SICE Annual Conference 2014
September 9-12, 2014, Hokkaido University, Sapporo, Japan.

Head Detection and Tracking for an Intelligent Room

Panca Mudjirahardjo, Joo Kooi Tan, Hyoungseop Kim and Seiji Ishikawa

Department of Control Engineering, Kyushu Institute of Technology, Kitakyushu, Japan
E-mail: {panca,etheltan,ishikawa}@ss10.cntl.kyutech.ac.jp, kimhs@cntl.kyutech.ac.jp
1-1 Sensuichou,Tobata-ku, Kitakyushu City, Fukuoka, 804-8550, Japan
Tel :093-884-3191

Abstract: We present a novel feature extraction method, which employs *a histogram of transition feature*, as an input to a SVM classifier. This feature relies on foreground extraction. We also evaluate some foreground extraction method. To evaluate the performance of this feature, we use it for head detection. Then, by applying a combination of the Harris corner detector and Lucas-Kanade tracker and motion pattern, we track the head position. The performance of the proposed method is experimentally shown.

Keywords: Foreground extraction, histogram of transition feature, head detection, head tracking.

1. INTRODUCTION

One area of computer vision research that has benefit by this advance and receiving considerable attention in the last few years is person tracking. Person tracking is a broad field encompassing the detection, tracking and recognition of bodies, heads, faces, expressions, gestures, actions, and gaze directions. Applications include surveillance, human computer interaction, teleconferencing, computer animation, virtual holography, and intelligent environments.

In intelligent environments, continuous monitoring of a room condition is required. This environment becomes significant with an aging population and changing demographics and household needs. In a fast-paced society, people find it difficult to manage their daily household tasks. Human housekeepers and helpers are commonly used while the regularity and reliability of such assistance are often requested. To overcome such problem, they use monitor cameras or homecare robotics to monitor a room condition.

An intelligent environment has been studied by some researchers. Mickelson [1] studied a system of head detection and tracking. He used multi-modal approach to the detection, using shape, motion, and size cues. At the core of the detector was an elliptical shape filter. Some failure occurred when the tracker locked on to an object whose elliptical shape is far better fit than a human head. Baranwal et al. [2] developed a home environment that was able to automatically monitor the motion of the occupants and quickly and accurately determine abnormal motions. Their vision-based approach used optical flow and an ellipse model for human body image. However, this research required the entire human body to be analyzed by a computer vision system. Nagayasu et al. [3] improved the three key functions for the operation of appliances in an intelligent room, i.e., detection of hand waving, skin color registration, and recognition of the number of fingers. Huang et al. [4] developed a real-time tracking of people in intelligent environments to maintain an awareness of all the dynamic events and activities taking place in them.

In addition, researches in the field of head detection have been carried out. Huang et al. [5] proposed a method to detect human heads in crowds from stereo images. They detected a human head based on its shape, with an assumption that human heads look like isolated balls from an elevated vantage point. A detector which relies on an object shape often causes a mistake when there is another object with the same shape [1]. Aziz et al. [6] explored a property of the graph skeleton and labeled body parts from the silhouette to deal with occlusions among people in motion. They proposed a skeleton-based head detection approach which can count people. Zeng et al. [7] used the multilevel HOG-LBP feature to detect the head-shoulders of people for people counting. Their methods offered a robust head-shoulder detector, but their multilevel calculation caused large processing time.

Due to the fact that a head motion can represent the movement of a body which may be partially covered, this research focus its attention on the detection and tracking of a human head. In this paper, we propose a novel feature for head detection. It is called *histogram of transition feature*. Compared with HOG and LBP feature, our feature has two advantages over them. First, the feature dimension is less than them, and second, generating calculation is simpler than them. Due to these advantages, the proposed detector rapidly generates the transition feature with high recognition capability.

2. OVERVIEW OF THE PROPOSED METHOD

We describe the proposed method in the following. There are two main processes in the proposed method. The first process is the detection of a human head from detected motion. After we perform frame differencing as in [1,8], we do foreground refinement by hole filling [2]. The resultant image is fed into the calculation of transition feature. We modify the calculation of the transition feature in [9,10]. Finally we recognize a human head by using a SVM classifier.

The second process is the tracking of head motion. We extract tracking points on motion objects in every two successive frames by using Harris corner detector followed by the Lucas-Kanade tracker [11-13]. **Fig. 1** depicts the overview of the proposed system. “A” is a connection node.

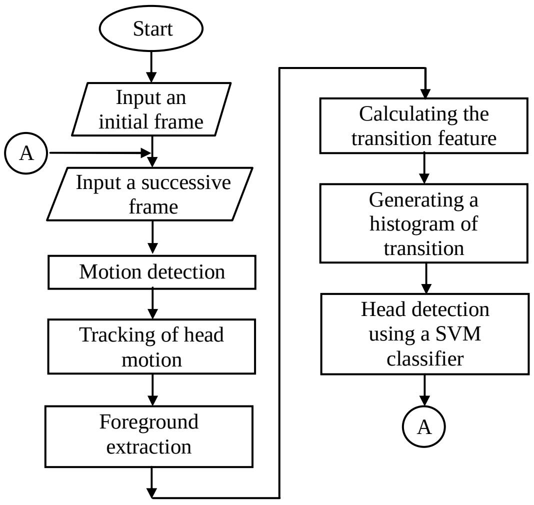


Fig. 1. Overview of the proposed head detection and tracking method

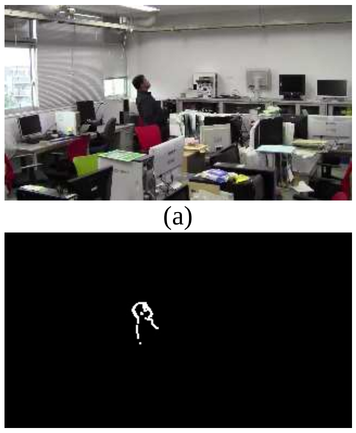
3. METHOD

3.1. Motion detection

Occasionally the most naive approach is found to yield adequate results. This is the case with motion detection. Simple frame differencing is used to find pixels corresponding to moving objects. If a pixel’s intensity, $I(x,y,t)$, changes significantly from one frame to the next, it is considered moving [1,8].

$$I_t(x,y,t) = \frac{\partial}{\partial t} I(x,y,t) = I(x,y,t) - I(x,y,t-1) \qquad (1)$$

This approach requires little computation and has minimal latency. First, we change the color image into gray image for both image frame $I(x,y,t\text{-}1)$ and $I(x,y,t)$. Then we apply equation (1). The result is still a gray image, then, with a gray level threshold, we change the gray image into a binary image. The results of this procedure can be seen in **Fig. 2**. We call the white pixels as a *motion pattern*.


(a)

(b)

Fig. 2. Motion detection using frame differencing: (a) A raw frame of video (b) the result of the motion detection.

3.2. Tracking of head motion

For tracking, we combine the result of Lucas-Kanade tracker and motion detection. Due to the fact that a head is always on top of a body, we assume a head to be on top of a motion object.

First, from the sequence image, we extract feature points in a given image by using Harris corner detector. Then we track the feature point of a head by using Lucas-Kanade tracker [11-13].

The feature points are tracked over some frames and their locational information is stored into a coordinate space. Suppose that a feature point n (n=0,1,2,…,N-1) is tracked through T image frames and its position on the frame t (t=0,1,2,…,T-1) is denoted by ($x_t^{(n)}$,$y_t^{(n)}$). We then define a sequence of T coordinates of the feature point by the following form;

$$\begin{aligned} X_0 &= \left[x_0^{(0)}, y_0^{(0)}, x_1^{(0)}, y_1^{(0)}, \ldots\ldots, x_{T-1}^{(0)}, y_{T-1}^{(0)}\right] \\ X_1 &= \left[x_0^{(1)}, y_0^{(1)}, x_1^{(1)}, y_1^{(1)}, \ldots\ldots, x_{T-1}^{(1)}, y_{T-1}^{(1)}\right] \qquad (2) \\ &\vdots \\ X_{N-1} &= \left[x_0^{(N-1)}, y_0^{(N-1)}, x_1^{(N-1)}, y_1^{(N-1)}, \ldots\ldots, x_{T-1}^{(N-1)}, y_{T-1}^{(N-1)}\right] \end{aligned}$$

We create a range of interest (ROI) of a head based on the coordinate of the feature point on top side as ROI$_A$. To avoid noise, we decide a ROI$_A$ has the number of coordinates above a threshold. If the number of coordinates in ROI$_A$ is below the threshold, then we move ROI$_A$ to the next coordinate. We check again the number of coordinates in ROI$_A$. The process is repeated until we find a ROI$_A$. If we can’t find a ROI$_A$, then we use the previous ROI$_A$.

Second, we create ROI$_B$ based on motion pattern on top side of a motion object. To avoid noise, we decide a ROI$_B$ of a head has the number of motion pattern above a threshold. Similarly as finding process of ROI$_A$, the process is repeated until we find ROI$_B$.

We define a ROI of a head, ROI$_H$, as

$$\mathrm{ROI_H} = \begin{cases} \mathrm{ROI_A} \cap \mathrm{ROI_B} & \text{if } \mathrm{ROI_A} \cap \mathrm{ROI_B} > th \\ \text{the previous } \mathrm{ROI_H} & \text{otherwise} \end{cases} \qquad (3)$$

After we decide a ROI$_H$, then we do the next process, i.e., foreground extraction for this ROI$_H$.

3.3. Foreground Extraction

Our transition feature relies on foreground extraction. Due to foreground extraction is not our main topic, we explain it not in detail.

The simple algorithm to extract foreground is that we determine some reference pixel coordinates as foreground. Then we compare another pixel’s intensity (I_x) to the reference pixel’s intensity (I_R).

$$I(x,y) = \begin{cases} \text{foreground} & \delta(I_x, I_R) < th \\ \text{background} & \text{otherwise} \end{cases} \qquad (4)$$

where $\delta(.)$ is a distance function.

Another algorithm to extract foreground is Linear Neighborhood Propagation (LNP) algorithm for image segmentation. This method is more complicated. The detail algorithm is explained in [17].

Table 1. Linear Neighborhood Propagation

Input : A set of partially labeled data X, the label y_i for each labeled data object x_i, the number of nearest neighbors k.
Ouput : The labels of the unlabeled data points.
1. Find the k nearest neighbors of each data in X.
2. Estimate the weights w_{ij} that best reconstruct x_i from its neighbors by minimizing the cost with the constraints.
3. Predict the labels of all the unlabeled data in X by solving the quadratic optimization problem with linear constraints.

The third algorithm to extract foreground is processing after motion detection by Eq.(1). Once the foreground pixels are obtained, the foreground is further refined by expanding the blob in three directions and then taking their intersection. This helps in filling the voids and empty spaces in blobs [2] (See **Fig. 3**).

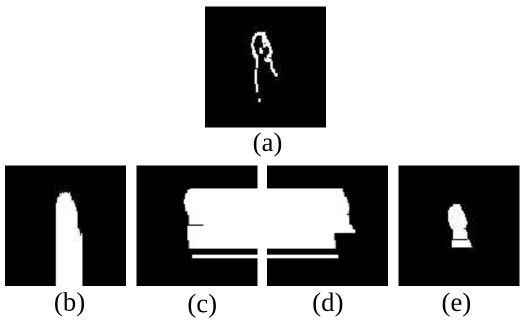


Fig. 3. Foreground refinement by hole filling: (a) An original image, (b) top to bottom, (c) left to right, (d) right to left, (e) refined foreground

3.4. Transition Feature

Our feature refers to [9,10]. Transition feature has been used successfully in a handwritten recognition, but it hasn't been used in a head detection yet. Due to a simple calculation to create a feature vector, we apply it's idea for a head detection. We do some modifications on it to be able to be used for head detection.

The idea is to compute the location and number of transitions from background to foreground along horizontal and vertical lines. This transition calculation is performed from right to left, left to right, top to bottom, and bottom to top. Since a constant dimension feature is required as input to the SVM classifier, an encoding scheme was developed.

In the first stage of feature extraction, the transition in each direction is calculated. Each transition is represented as a fraction of the distance across the image in the direction under consideration. These fractions are computed in the increasing order, differed from [10] in decreasing order. For example, when calculating the location of transitions from left-to-right, a transition close to the left edge would have a low value and a transition far from the left edge would have a high value as illustrated in **Fig. 4**.

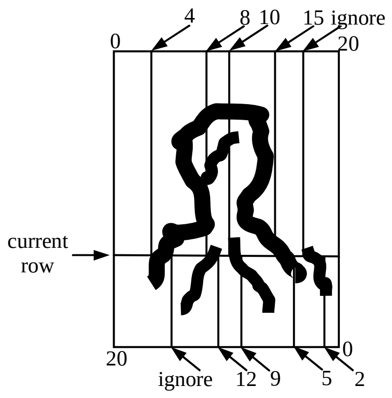


Fig. 4. The first stage of transition feature extraction shown for transitions from the left and from the right on one row of the image

The maximum number of transitions, M, are counted on each line. If there are more than M transitions in a line, then only the first M are counted, the rest are ignored. M is set to 4. If there are less than M transitions on a line, then the "nonexistent" transitions are assigned a value of 0.

More precisely, by a line we mean a row or a column of the head image. Let h be the height of the image and w be the width of the image. We assign exactly M values to each line, say $t_1, t_2, \ldots, t_M$. Assume that there are n transitions on a line located at (x_i, y_i) for $i = 1,2, \ldots n$. The algorithm for calculating the transition feature can be represented as follows: It doesn't require normalization as in [10]:

```
For $i$ = 1 to min($n$,$M$)
    If the line is row then
        $t_i = y_i$;
    Else
        $t_i = x_i$;
    end if;
End for;
If $n < M$ then
    For $i$ = $n$+1 to $M$
        $t_i = 0$;
    end for;
end if;
```

The transitions are resampled to a 4-point sequence for each direction and assembled into a feature vector. The four transitions for each row (column) are represented as two-dimensional (2-D) array, $t = [t_{ij}]$ for $i = 1,\ldots,h(w)$ and $j = 1,\ldots,4$.

The second stage is generating a histogram of transition. It is different from [10] where they calculated local averaging on the columns of t. Histogram of transition shows how often the location of the transition occurs at each transition. An example for generating a histogram of transition for transition left-to-right is shown in **Fig. 5**.

This histogram of transition creates a feature vector to be fed into the input of a SVM classifier [14].

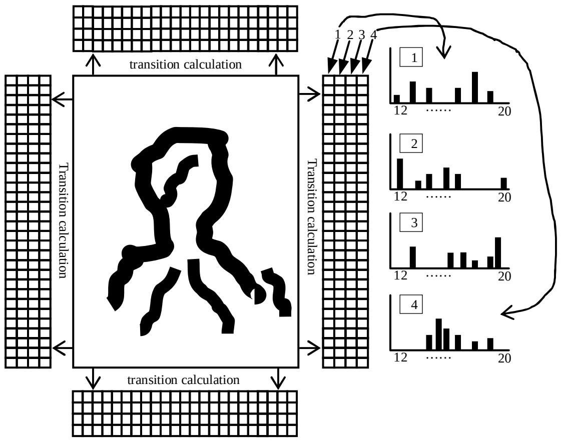


Fig. 5. The second stage of transition feature calculation consisting of generating the histogram of transition.

HOG feature contains gradient information of a pixel among its neighbor. Thus they give a high magnitude at the edge. On the other hand, LBP feature gives a binary pattern with a pixel among its neighbor. Histogram of transition feature looks like the HOG feature: It gives the edge position from right, left, top and bottom side. In contrast to HOG feature, the calculation of the histogram of transition feature is simpler.

4. EXPERIMENTAL RESULTS

The experimental environment is as follows: Operating system is Windows 7 ultimate; the processor is Intel® core™ i7 CPU 870 @2.93GHz and the used software is Microsoft Visual Studio 2010.

4.1. Head Detection

To test the proposed head detection method, we do comparison with HOG [15] and LBP feature [16]. For robust detection, we use backgrounds and negative samples at outdoor scenery. We use INRIA data for training and testing images. The image size (ROI_H) is 20 × 30 pixels. For training, positive sample of 2,000 images, negative sample of 4,500 images are used. For testing, positive sample of 100 images, negative sample of 300 images are employed.

For the image preprocessing to extract transition feature, we extract the foreground using a difference function (4). First, we determine five reference pixel coordinates as foreground. These coordinates are fixed for all the training and the test data. The coordinates should represent position of head and shoulder, that are (10,8), (10,15), (10,22), (5,22) and (15,22), see **Fig. 6 (a)**. Then, we check all pixels' intensity to the five reference pixel's intensity with Euclidean distance, by Eq. (4). If a pixel's intensity has distance to the one or more of five reference pixel's intensity less than a threshold, then the pixel should be a foreground, otherwise as background.

In addition, we compare the method of foreground extraction. Another method is image segmentation by applying Linear Neighborhood Propagation (LNP) [17]. As in LNP, we initialize to label some pixels manually as foreground and background and letting some unlabeled pixels to be labeled automatically. For this requirement, we label 20 and 8 pixels as foreground and background, respectively. These labeled pixels are shown in **Fig. 6(b)**. These label coordinates are fixed for all the training and the test data. The result of foreground extraction is shown in **Fig. 7**.

The result of head detection is summarized in **Table 2** and **Table 3**.

Table 2. Evaluation of feature extraction method

Feature	The number of array	Detection rate (%)		Execution time (ms)
		Positive	Negative	
HOG	648	84	98.3	0.353
LBP	1020	93	80	0.261
Histogram of transition	400	91	99.7	0.077

Table 3. Evaluation of foreground extraction method

Foreground extraction method	Head detection rate (%)		Execution time (ms)
	Pos.	Neg.	
LNP	52	93	32.931
Distance function (eq. 2)	89	99.7	0.077
Frame differencing then dilation	92	99.7	0.077
Frame differencing with refined foreground	80	93	0.077

Image segmentation by using LNP method needs label accurately. This method requires label pixel manually, then proceeds to segmentation automatically. This method forced unlabeled pixels to be labeled pixels. In case of an image without foreground, the resulting image should have foreground and background. Some result of LNP method are shown in **Fig. 7(c)**.

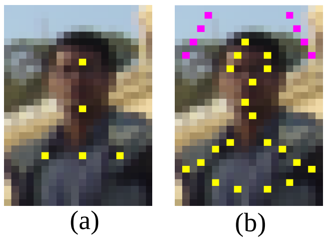


Fig. 6. Reference pixel coordinates: (a) Distance function method, (b) LNP method: Yellow are foreground, magenta are background.

In **Table 3**, head detection rate which use frame differencing then dilation is better than with refined foreground, because the head-shoulder shape more clear than the refined one, as shown in **Fig. 8**.

Based on **Table 2**, transition feature is a reliable method for head detection, and **Table 3** shows that, for motion scene, frame differencing method can be applied for foreground extraction.

4.2. Head Tracking

Based on the two above tables, we combine distance function of color image and frame differencing as foreground extraction and histogram of transition feature for a feature vector. A distance function method gives a pattern of a head for a standstill condition. Thus the system can detect a head when there is no motion or little motion. The outline of the proposed feature is shown in **Fig. 9**. The "∪" sign is a union operation.

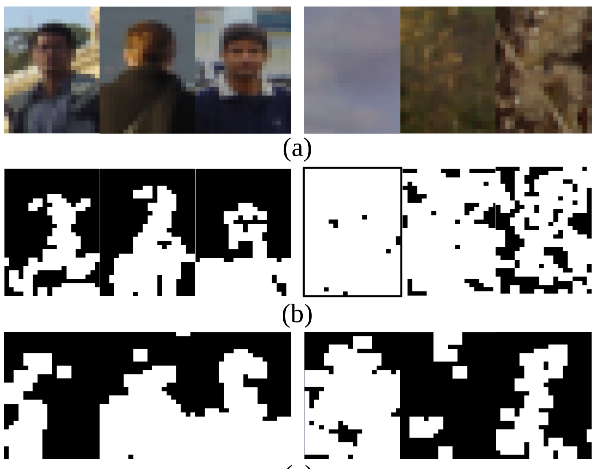


Fig. 7. The result of foreground extraction: (a) An original image, positive and negative, (b) the result of distance function method, (c) the result of LNP method.

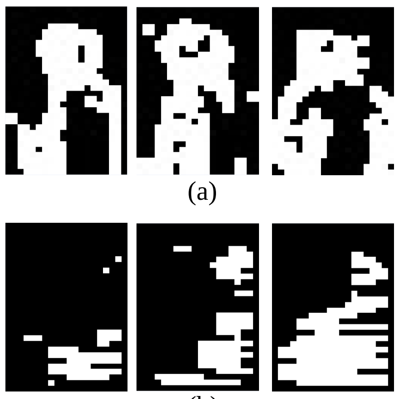


Fig. 8. The result of foreground extraction by using frame differencing (a) with dilation operation and (b) with refined foreground.

The experimental results are shown in **Fig. 10**. For evaluation of the results, let us define *recall*, *precision* and *FPR* by

$$recall = \frac{TP}{TP+FN} \times 100\% \quad (6)$$

$$precision = \frac{TP}{TP+FP} \times 100\% \quad (7)$$

$$FPR = \frac{FP}{FP+TN} \times 100\% \quad (8)$$

Here
TP : head is detected as head.
FN : head is detected as non-head.
FP : non-head is detected as head.
TN : non-head is detected as non-head.

Then the evaluation of performance is shown in **Table 4** and **Table 5**.

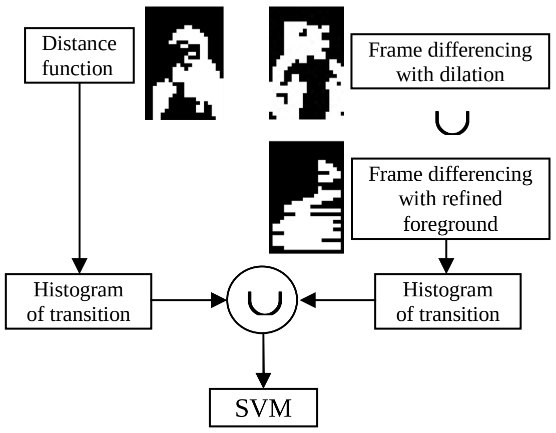

Distance function
Frame differencing with dilation
∪
Frame differencing with refined foreground
Histogram of transition
∪
Histogram of transition
SVM

Fig. 9. The outline of the proposed feature. It has 800 dimensions.

5. CONCLUSION

In this paper, we proposed a head detection and tracking method for an intelligent room. We use a histogram of transition feature as a novel method for feature extraction and we use frame differencing as a foreground extraction.

As future work, we are going to conduct experiments to improve foreground extraction and motion recognition.

REFERENCES

[1] Mickelson, J.S., "Design and application of a head detection and tracking system", *Master Thesis*. Massachusetts Institute of Technology, (June 2000).
[2] Baranwal, M., Khan, M.T., Silva, C.W.D., "Abnormal motion detection in real time using video surveillance and body sensors". *International Journal of Information Acquisition.* Vol. 8, No. 2, pp. 103-116, (July 2011).
[3] Nagayasu, T., Asano, H., Terabayashi, K., Umeda, K., "Improvement of an intelligent room that detects hand waving motion for operation of home appliances". *Proceeding of SICE Annual Conference 2011*, pp. 821-826 (Sept. 2011).
[4] Huang, K.S., Trivedi, M.M., "Video arrays for real-time tracking of person, head, and face in an intelligent room", *Journal Machine Vision and Applications*, Vol. 14, pp. 103-111 (2003)
[5] Huang, X., Li, L., Sim, T., "Stereo-based human head detection from crowd scenes", *Proceeding of International Conference on Image Processing*, Vol. 2, pp. 1353-1356 (Oct., 2004).
[6] Aziz, K.E., Merad, D., Fertil, B., Thome, N., "Pedestrian head detection and tracking using skeleton graph for people counting in crowded environments", *Proceeding of MVA2011 IAPR Conference on Machine Vision Applications*, pp. 516-519 (June 2011).
[7] Zeng, C., Ma, H., "Robust head-shoulder detection by PCA-based multilevel HOG-LBP detector for people counting", *Proceeding of International Conference on Pattern Recognition*, pp. 2069-2072

(2010).

[8] Viola, P., Jones, M., Snow, D., "Detecting pedestrians using patterns of motion and appearance". *Technical Report on Mitsubishi Electric Research Laboratories*, pp. 1-11 (Aug. 2003).

[9] Mudjirahardjo, P., "Penerapan jaringan perambatan-balik untuk pengenalan kode pos tulisan tangan" (The implementation of back-propagation network for recognition of handwriting post code), *Master Thesis*, Universitas Gadjah Mada, (Nov. 2001).

[10] Gader, P.D., Mohamed, M., Chiang, J.H., "Handwritten word recognition with character and inter-character neural networks", *IEEE Trans. On Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 27, No. 1. Pp. 158-164 (Feb. 1997)

[11] Park, M., Tan, J. K., Nakashima, Y., Kim, H., Ishikawa, S., "Detecting human flows on a road different from main flows". *Proceeding of The Sixteenth International Symposium on Artificial Life and Robotics 2011* (AROB 16th 2011), (2011).

[12] Mudjirahardjo, P., Tan, J.K., Kim, H., Ishikawa, S., "Abnormal motion detection in an occlusive environment", *Proceeding of SICE Annual Conference 2013*, pp. 1398-1402 (Sept. 2013).

[13] Mudjirahardjo, P., Tan, J.K., Kim, H., Ishikawa, S., "Fast motion detection in a dynamic background", *Proceeding of The Nineteenth International Symposium on Artificial Life and Robotics 2014* (AROB 19th 2014), pp. 896-900 (Jan. 2014).

[14] Hsu, C.W., Chang, C.C., Lin, C.J., "A practical guide to support vector classification", *Proceeding of IEEE Intelligent Vehicles Symposium*, Vol. 1, pp. 1-6 (April 2010).

[15] Dalal, N., Triggs, B., "Histograms of oriented gradients for human detection", *Proceeding of Computer Vision and Pattern Recognition*, Vol. 1, pp. 886-893 (June 2005).

[16] Heusch, G., Rodriguez, Y., Marcel, S., "Local binary patterns as an image preprocessing for face authentication", *Proceeding of Automatic face and Gesture Recognition* (FGR 2006), pp. 9-14 (April 2006)

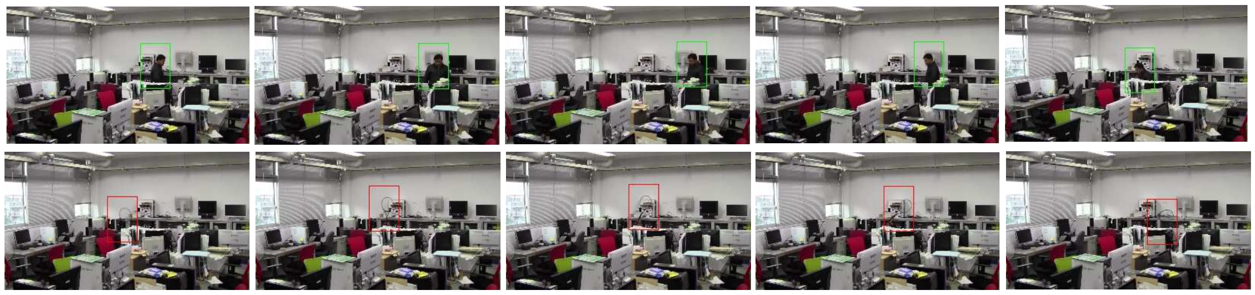
[17] Wang, F., Wang, J., Zhang, C., Shen, H.C., "Semi-supervised classification using linear neighborhood propagation", *Proceeding of Computer Vision and Pattern Recognition* (CVPR2006), Vol. 1, pp. 17-22 (June 2006).

Table 4. Evaluation of the performance

scene	Recall (%)			Precision (%)			FPR (%)		
	HOG	LBP	Histogram of Transition	HOG	LBP	Histogram of Transition	HOG	LBP	Histogram of Transition
1	87.2	85.0	**89.8**	88.0	85.2	**90.2**	8.1	8.8	**7.5**
2	77.1	**84.4**	83.7	80.5	78.1	**82.5**	10.4	11.5	**10.2**

Table 5. Execution time

Method of feature	Execution time (ms)
Histogram of transition (800 dimensions)	168.87
LBP (1020 dimensions)	678.29
HOG (648 dimensions)	1041.78


(a)

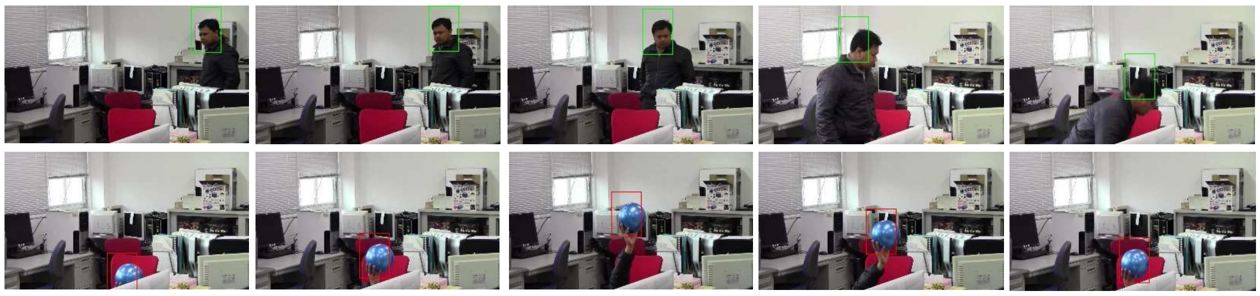

(b)

Fig. 10. Performance of head detection and tracking. The green box shows that the proposed system detects a motion and a head, the red box shows that the system detects a motion but not a head . (a) Scene 1: The system can distinguish a head and a racket. (b) scene 2: The system can distinguish a head and a ball