Detecting Humans and Visualizing Human Motions for People Image Analysis

Hironobu Fujiyoshi

Department of Computer Science, Chubu University
1200 Matsumoto-cho, Kasugai, Aichi, 487-8501 Japan
hf@cs.chubu.ac.jp
http://www.vision.cs.chubu.ac.jp/

Abstract. An invisible environmental robot is a system that recognizes ever-changing human states based on input provided by a group of sensors such as cameras situated in an environment. An important role of such systems is to help create a comfortable space for human users. To implement invisible robots, especially the functions related to people image analysis, it is essential to make them capable of detecting people in video images, detecting faces, tracking their movements and recognizing human activities. As an approach to people image analysis, this paper introduces methods for detecting humans and visualizing human motions.

1 Introduction

A robot is defined as a system with various capabilities that can be categorized as sensory (acquiring data from the outside world), cognitive (understanding the significance of this data), decision-making (deciding what to do), and behavioral (the resulting actions it performs for humans in the outside world). If we consider a situation where numerous robots and humans are in an environment where they share the same space and information, then the environment itself provides functions of memory, communication, sensors and effectors. This sort of system is also a type of robot and is called an environment robot or invisible robot (in the sense that it has no visible shape or form) [1].

For example, an automatic surveillance system that uses large numbers of outdoor cameras is also a type of robot, and can be regarded as one application of invisible robots to outdoor situations. One of the origins of research into this sort of intelligent video surveillance for the detection and tracking of outdoor objects was the VSAM (Video Surveillance and Monitoring) project at DARPA (Defense Advanced Research Projects Agency) in 1997, which studied automatic video surveillance systems [2]. Further progress has been made since then, and today a lot of effort is being put into the implementation of products based on these technologies.

Expectations are also growing with regard to the development of technologies that can understand the intentions of people in spaces such as offices, homes and public buildings, and can assist them in their actions. In the Aware Home

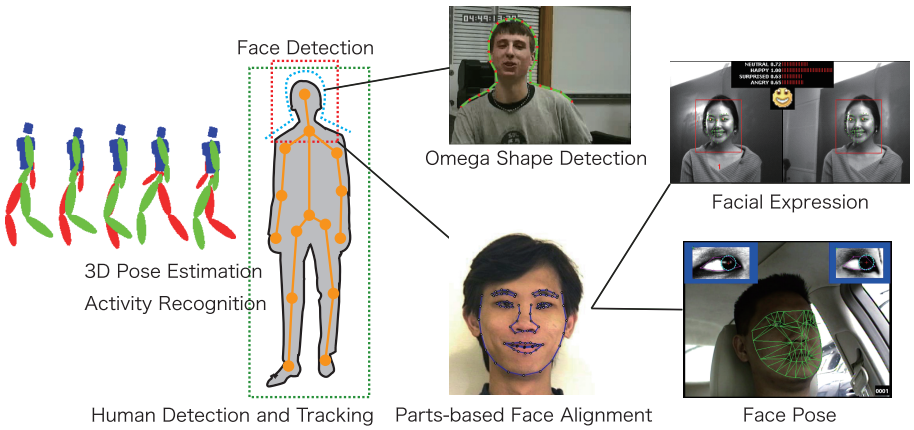


Fig. 1. Examples of human sensing by means of PIA technology.

project at Georgia Institute of Technology [3], researchers are looking into how large numbers of cameras and other sensors can be incorporated into a domestic living space and used to sense the movements of people in this space around the clock. Studies are also being made regarding the provision of comfortable spaces for human users based on information obtained by sensing, such as the Easy Living project at Microsoft Research [4], which is targeted at living rooms. These active living systems depends on the use of sensing technology, in which people image analysis(PIA) is essential, to recognize the ever-changing situations of people in real time. In particular, the key technologies for PIA are: detecting humans in moving images, tracking their movements, detecting faces, tracking the location of face parts, and understanding motions. Of these, detecting humans is regarded as a difficult problem due to changes in the shapes of humans. However, with recent advances in computer processing speeds, a detection method has been proposed where the whole image is raster scanned using detection windows and humans are detected based on image local features and statistical learning methods. In the techniques of tracking movements and understanding motions, more feature points on non-rigid objects such as humans are required and needed to be consecutively tracked for a long time.

In this paper, we describe human detection and human motion visualization methods which are recent approaches to people image analysis technology.

2 People Image Analysis (PIA)

The quality of user actions can be enhanced by sensing the ever-changing situations of humans (including their movements, trajectry, posture, line of sight and facial expressions) and supporting the recognition of human behavior intentions. One issue with this sort of sensing that the PIA project[5] aims to address is

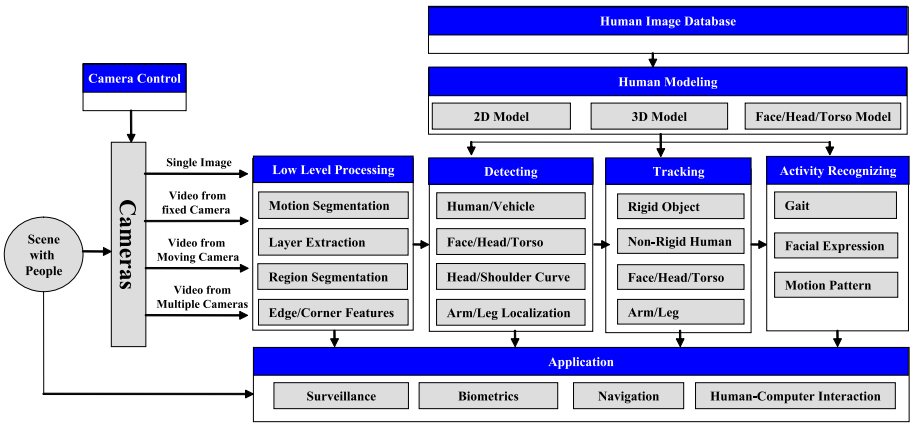


Fig. 2. Architecture of PIA technology.

that of using information from static and moving images to recognize actions by detecting and tracking humans, faces and parts of the body. Figure 1 shows an example where PIA is used to sense various parts of a human body. We are also engaged in cross-cutting efforts to develop computer vision techniques such as detecting humans, detecting faces, tracking their movements and recognizing the activities of humans, based on the development of low-level vision algorithms such as feature point extraction/tracking, layer extraction and motion segmentation adapted to diverse camera configurations ranging from a single fixed camera to multiple moving cameras (see figure 2). By combining these techniques according to requirements, it is possible to implement applications suitable for various purposes such as video surveillance, biometrics, activity navigation and human-computer interaction.

Unfortunately there is not enough space here for a discussion of each technique of PIA, so this paper describes the techniques used for the detection of humans and the visualization of human motions used to connect this sensing information with activity recognition.

3 Human Detection by Joint Features

Many recent object detection methods use a combination of statistical learning and image local features. Image local features that can be used for human detection include edge orientation histograms (EOH) which are based on the ratio of cumulative edge intensities between regions [6], histograms of oriented gradients (HOG) which are based on histograms of gradient orientations in a local region [7], and edgelets which are based on use of short lines and curves connecting edge segments [8]. These edge-based features can express the shape of an object within a localized region.

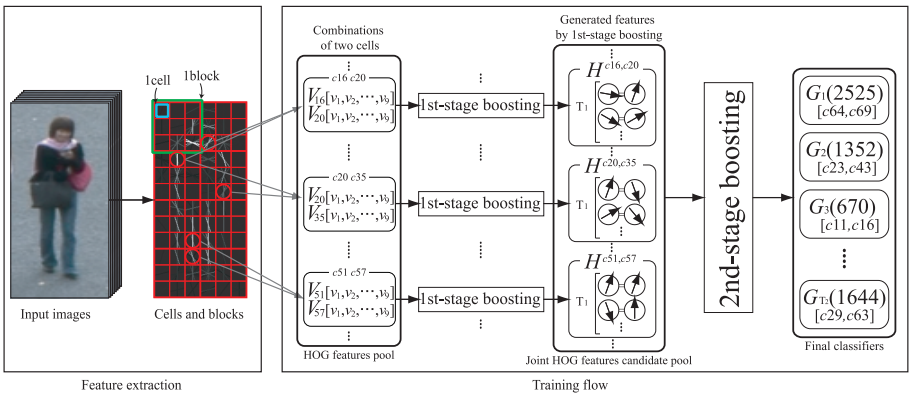


Fig. 3. Two-stage Real AdaBoost based on Joint features.

The shapes of humans can be broadly divided into the following two characteristics:

(1) The Ω-shaped head and shoulder region and its continuation through the upper and lower body.

(2) The left-right symmetry of the head, shoulders, torso, legs and so on.

Characteristics of type (1) can be detected with shapelet features that express local information by using AdaBoost to combine the edge characteristics in four directions within a local region [9]. For characteristics of type (2), joint Haar-like features have been proposed to represent co-occurrence by simultaneously observing multiple features with AdaBoost weak classifiers [10]. In both methods it is possible to grasp relations between features by using boosting to combine multiple low-level features, resulting in a high detection accuracy. Thus in recent years the way in which boosting is used to obtaine relations of multiple features has become an important issue.

We have proposed an object detection method [11] based on two-stage Real AdaBoost[12] and joint features with which it is possible to automatically capture the symmetry and continuity of object shapes. These joint features are made by using two-stage boosting to combine the HOG features of two different regions.

3.1 Joint Features and Two-Stage Boosting

Figure 3 illustrates the creation of joint features and the final classifier structure. The creation and learning of joint features is performed using a two-stage Real AdaBoost algorithm. Here, we separately describe the first-stage Real AdaBoost that creates the joint features, and the second-stage Real AdaBoost that trains the final classifier.

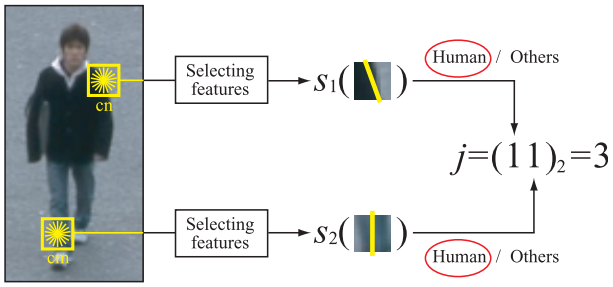


Fig. 4. Co-occurrence of features.

Creation of Joint Features by First-Stage Real AdaBoost To create joint features, individual features are extracted each from two different regions. HOG features are calculated in two cells c_m and c_n at different locations, and the feature co-occurrence values as shown in figure 4 are used to generate joint features in the first-stage Real AdaBoost. This captures the relations of cells as well as the symmetry of object shape and edge continuity. From the features that represent co-occurrence for cells at two different locations, c_m, c_n, Real AdaBoost selects those that are effective in discrimination. The joint feature $H^{c_m,c_n}(x)$, which is the strong classifier of the first-stage Real AdaBoost, is constructed with the following equation:

$$H^{c_m,c_n}(x) = \sum_{t=1}^{T} h_t^{c_m,c_n}(x). \tag{1}$$

The above process is applied to all the cell combinations, resulting in the creation of a joint feature candidate for each combination. For example, if the input image measures 30×60 pixels and the cell size is 5×5 pixels, then the image is divided into a total of 72 cells with ${}_{72}C_2 = 2{,}556$ possible combinations, so 2,556 joint feature candidates $Hc^{m,c_n}(x)$ are created. All these joint feature candidates are pooled together and input to the second-stage Real AdaBoost described below.

Construction of Final Classifier by Second-Stage Real AdaBoost In the second-stage Real AdaBoost, the final classifier is constructed based on the input from the joint feature candidate pool created by the first-stage Real AdaBoost. In this way, it is possible to automatically select joint features that are useful for classification.

Evaluation of Detection Performance To confirm the effectiveness of the joint features, we compared the proposed joint features with HOG features [7] and shapelet features [9] of conventional methods. The test results are shown in figure 5(a). We found that the detection performance was better than that

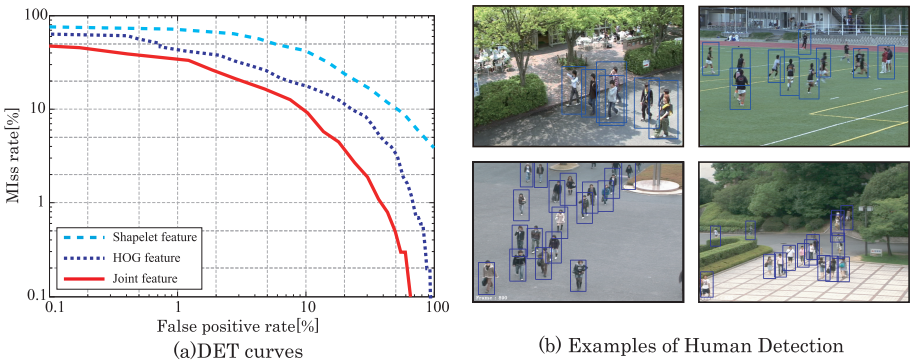


Fig. 5. DET curves.

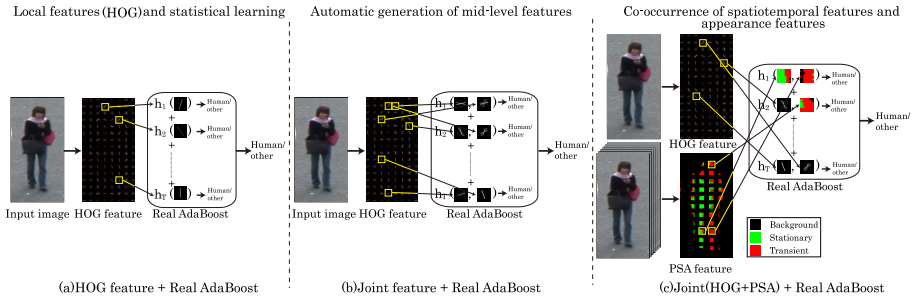


Fig. 6. Extraction of features in human detection.

of the conventional method. This is because even in patterns that are difficult to classify with edge features alone (e.g. HOG features), by combining HOG features in cells at two different locations it was possible to grasp patterns that are difficult to classify. Figure 5(b) shows examples of the detection of humans using joint features. As this figure shows, robust detection was possible even with target objects that were occluded or of differing scales.

Effectiveness of Joint Features Figure 6(b) shows how joint features are grasped using Real AdaBoost. With HOG features and Real AdaBoost, a single weak classifier performed classification using a single HOG feature, but with joint features a single weak classifier performs classification using multiple HOG features that are included in two regions at different locations. This makes it possible to automatically grasp symmetries and continuous edges in object shapes that cannot be grasped by conventional techniques that use only single HOG features, and thus allows humans to be detected with greater accuracy.

Figure 7(a) shows the results of visualizing HOG features selected by the first-stage Real AdaBoost, and figure 7(b) shows the results of visualizing HOG

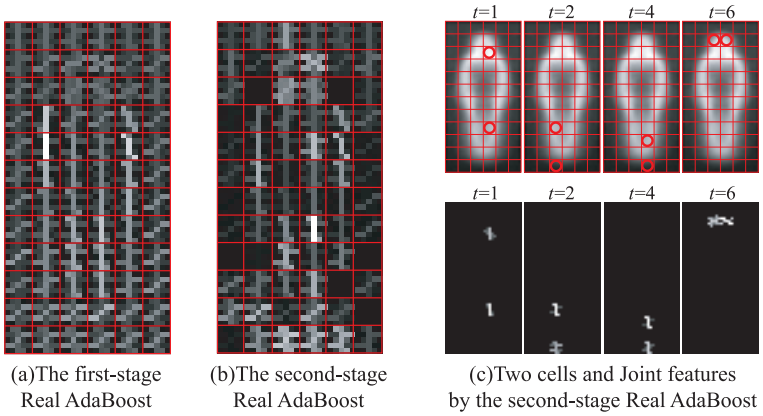


Fig. 7. Visualization of selected Joint features.

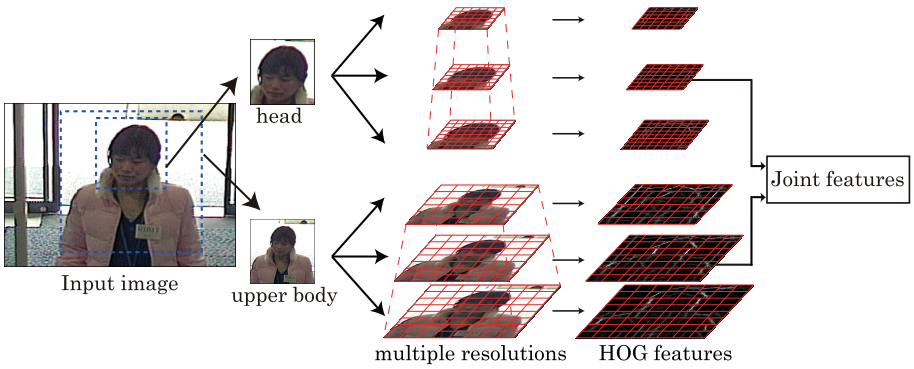


Fig. 8. Co-occurrence of HOG features at multiple resolutions.

features selected by the second-stage Real AdaBoost. Also, figure 7(c) shows the two cells and joint features selected by the second-stage Real AdaBoost at each round. The HOG feature gradient directions are represented by 9 directions, and a higher luminance represents a higher score of weak classifier in Real AdaBoost, making the feature more useful for classification.

The likelihood of selecting HOG features away from the outline of the human in figure 7(b) became smaller, even for the HOG features selected in figure 7(a). This is because these features are not judged to be useful for classification in the feature selection of the second-stage Real AdaBoost. Next we will focus on figure 7(c). As this figure shows, cells along the outline of the human figure are selected for joint features selected by the second-stage Real AdaBoost.

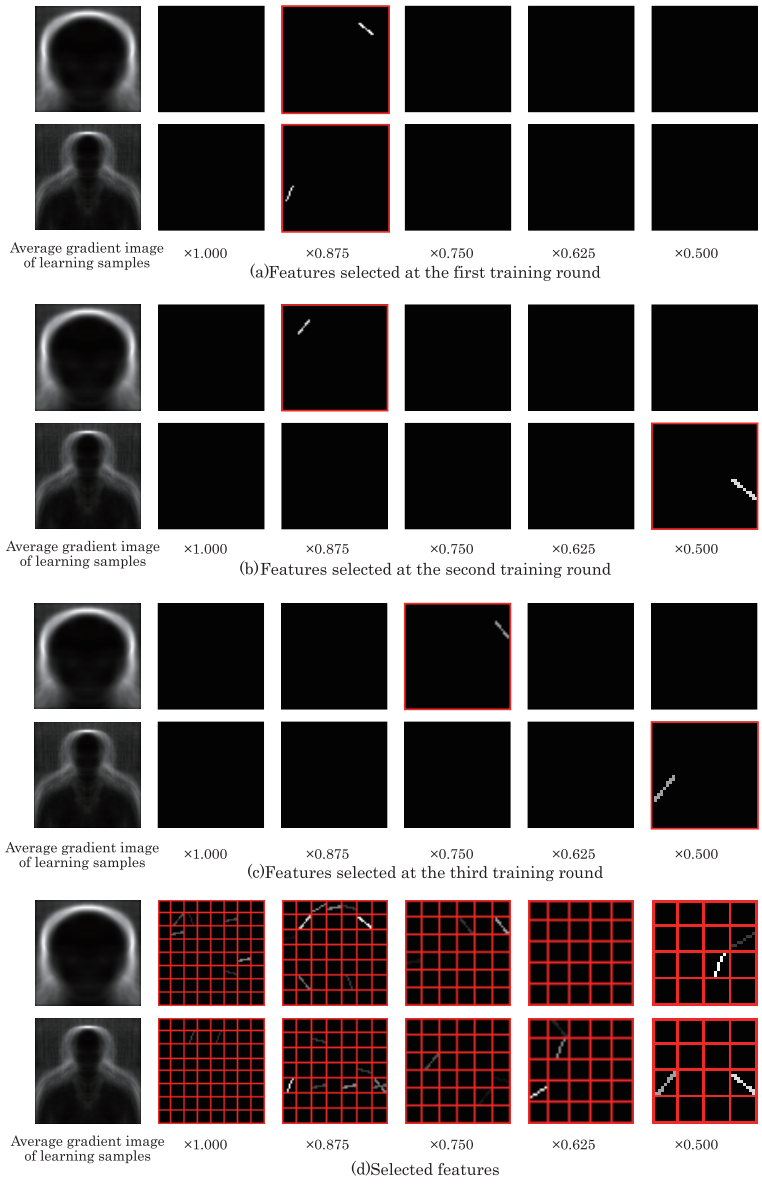


Fig. 9. Visualization of selected HOG features.

3.2 Co-occurrence between Multiple Resolutions

We have managed to increase the detection accuracy by using joint features to grasp shapes that are useful for the detection of humans. However, the features of shapes such as human heads and shoulders are not necessarily best repre-

sented at the same resolution. By extracting HOG features while varying the size of the input image, we can represent co-occurrence between different resolutions. Changing the resolutions allows us to select HOG features at the optimal resolution for each location such as human heads and shoulders. Figure 8 shows the flow of co-occurrence representation between multiple resolutions. First, the input image is separated into a head part and an upper body part, and these are downsampled to produce multiple resolution images from which the HOG features are extracted. The HOG features extracted from each cell are then used to calculate joint features.

Figures 9(a), (b) and (c) show visualizations of the HOG features selected in each round of Real AdaBoost, and figure 9(d) shows visualizations of the accumulation of HOG features selected at each learning round. At the start of learning where the feature selection trends are likely to appear, it can be seen that high-resolution HOG features are selected for the head part while low-resolution HOG features are selected for the upper body. This is because, as can be seen from the average gradient images of the learning samples, the gradients around the head have low variation and high-resolution HOG features are selected, while the gradients are more diffuse for the upper body and low-resolution HOG features are selected. The HOG features from the low-resolution images are based on histograms obtained over broad regions, so they are better able to accommodate this scattering. Also, with the weak classifiers of the second round it is possible to grasp the lines at the left side of the head and along the right shoulder, and with the weak classifiers of the third round it is possible to grasp the lines at the right side of the head and along the left shoulder. By grasping the symmetry of these features at multiple resolutions, it is possible to select features without being affected by partial occlusion.

3.3 Co-occurrence of Spatiotemporal Features

In the framework of joint features, it is possible to add other features to the HOG features that represent the appearance of humans (figure 10(b)). As a feature based on spatiotemporal features that have been used for the detection of moving objects, we added the results of pixel state analysis (PSA)[14] shown in figure 10(c), and thereby realized human detection with even higher accuracy [13].

Spatiotemporal Features Based on Pixel State Analysis Pixel state analysis is a technique that involves modeling temporal changes in pixel states to discriminate each pixel into one of three states - background, stationary or transient. This discrimination is performed by using the images from 5 frames ahead and behind to calculate abrupt changes in intensity (motion trigger) and stability (stability measure). In this way it is possible to represent the motion of objects. If we consider frame T in figure 11, with a single image it is not possible to tell if the person is standing still or walking, but if we also consider the images of frames before and after then it is possible to see that the left leg is moving

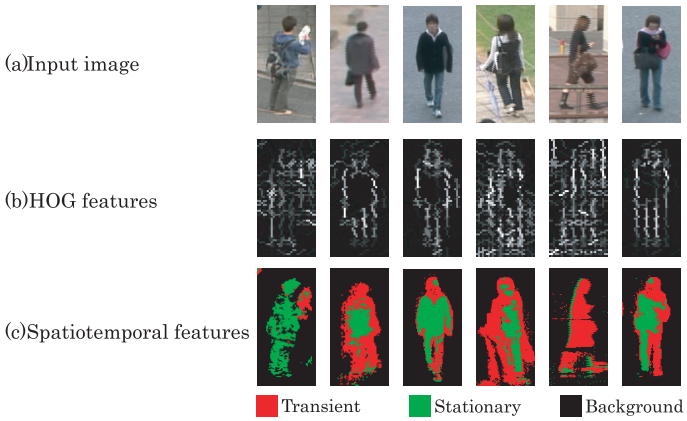


Fig. 10. Examples of HOG features and pixel state analysis.

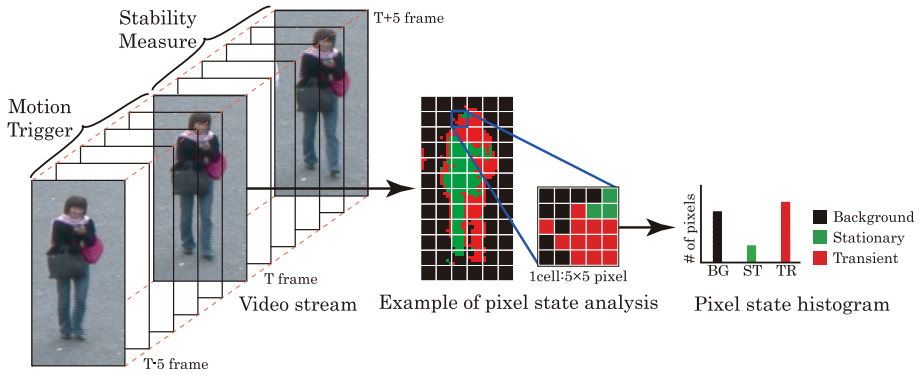
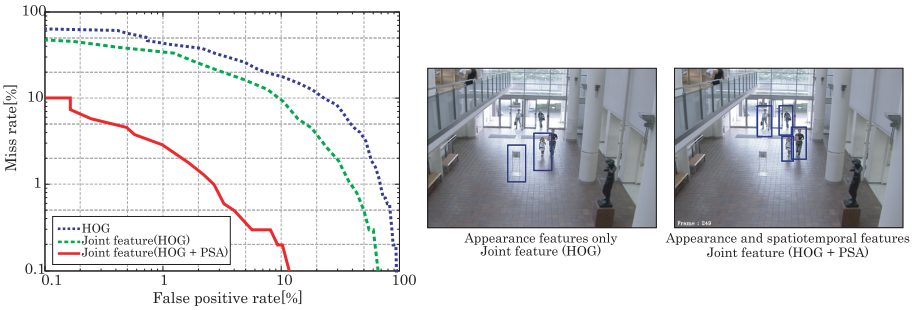


Fig. 11. Pixel state analysis and state histogram.

forwards and pivoting around the right leg. By using these sequential images to perform pixel state analysis, it is possible to represent movements such as walking in terms of transient and stationary states.

Figure 11 shows an example of pixel state analysis and the state histogram computation method. From the results of pixel state analysis, the state histogram is created by adding the number of pixels in each state that are included in one cell.

The Effects of Co-occurrence with Spatiotemporal Features Figure 12(a) shows the DET curves of human detection test results obtained using HOG and PSA joint features. Compared with conventional HOG features, the joint features(HOG+PSA) were able to achieve a detection rate of at least 99% at a false detection rate of 5.0%. Figure 12(b) show an example of human detec-



(a) DET curves (b) Effects of using spatiotemporal features

Fig. 12. DET curves and examples of human detection.

tion using this technique. Objects resembling humans resulted in false detections when using only appearance features, but with the addition of spatiotemporal features it was possible to reduce the false detection rate.

Figure 13 shows the proportion of HOG features and PSA features selected at each round of learning, together with some examples of the features selected during this process. PSA features are mostly selected in the initial round, but a greater proportion of HOG features are selected as the number of learning rounds increases. This is thought to be because the ability of PSA features to express the movement of objects allows an initial discrimination to be roughly made between people and non-people, after which a more precise classification boundary is formed by using HOG features which also include appearance information.

3.4 Co-occurrence of Depth Information

When detecting humans from images captured with a visible-light camera, it is sometimes difficult to acquire useful appearance information for human detection due to the complexity of the background texture. We have therefore proposed human detection using a time of flight (TOF) camera that can acquire depth information about the distance from the camera to the subject. A TOF camera is a camera that illuminates the subject with infra-red light from multiple LEDs positioned on the front of the camera and measures the time taken for this light to be reflected back from the subject, thereby acquiring information about the depth of the subject from the camera. By handling depth information obtained from a TOF camera simultaneously with the appearance features, it is possible to detect humans with greater accuracy. With the addition of depth information, it becomes possible for Real AdaBoost weak classifiers to grasp the distance relationship between object and the background as well as the appearance of these objects. This makes it possible to suppress the effects of occlusion and complex backgrounds.

As shown in figure 14, features obtained from depth information are processed by dividing the distance image into cells and using the Bhattacharyya distance

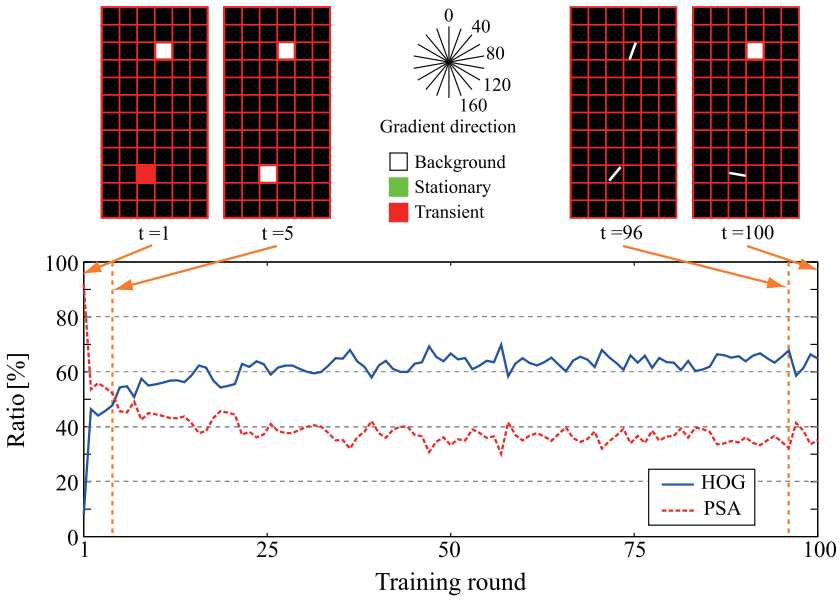


Fig. 13. Proportion of selected features.

to calculate the similarity between the distance histogram of the cell in question and the distance histograms of its eight neighboring cells, thereby obtaining the distance for each cell as a feature quantity. This feature expressed the relative distance relationship between the cell in question and its neighboring cells. Figure 15 shows the flow of human detection by co-occurrence of features obtained from distance histograms and HOG features which are appearance information. In this way, as shown in figure 16, it is possible to detect humans with high precision even when they are overlapping.

In this section we discussed object detection methods based on joint features using co-occurrence between multiple features as shown in figure 17. For patterns that are difficult to classify with simple HOG features, joint features combine the HOG features of two cells in different locations and are thereby able to correctly classify these difficult patterns. Evaluation tests using images of humans showed that joint features are capable of highly accurate human detection. To achieve even higher detection accuracy, we discussed the effects of co-occurrence of PSA (spatiotemporal) features, co-occurrence between HOG features obtained from images of multiple resolution, and co-occurrence with distance histogram features.

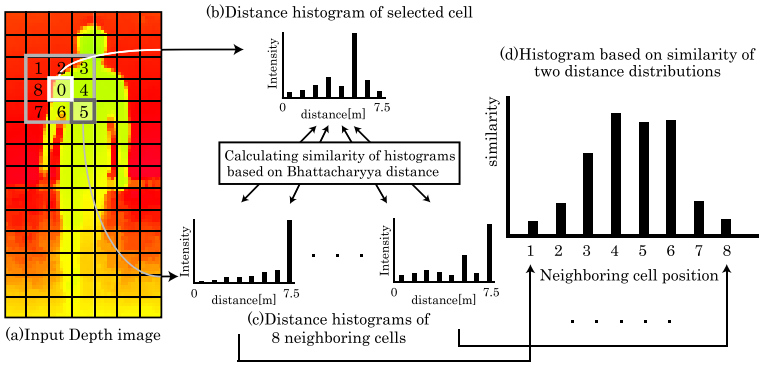


Fig. 14. Features obtained from distance histograms.

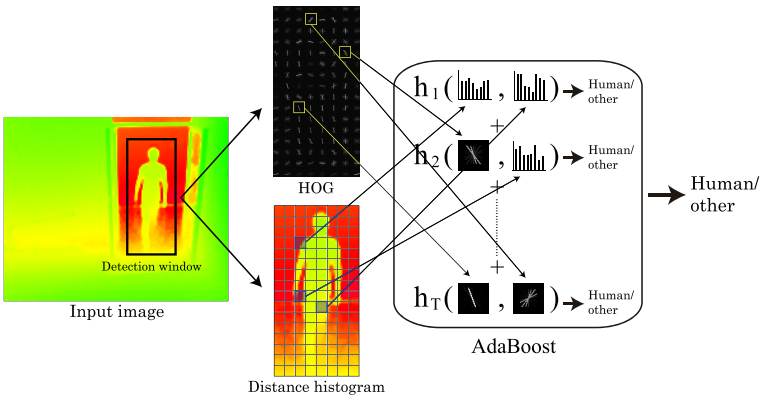


Fig. 15. Co-occurrence of HOG features and distance histogram features.

4 Visualization of Human Motions Using SIFT Features Point Tracking

Human motion analysis and its visualization are important for recognizing human activities. This section presents a method for visualizing a pedestrian traffic flow using results of feature point tracking. Using a scale invariant feature transform (SIFT) is a method for detecting keypoints and describing the characteristic features of these keypoints, which are invariant to changes caused by rotation, scaling, and illumination [15]. We have developed a new approach to keypoint tracking using the SIFT technique. In our approach, we use mean-shift searching to track a keypoint based on the information obtained from the SIFT technique. The mean-shift algorithm [16, 17] locates the nearest mode of a point sample distribution [18, 19]. Collins [20] proposed a method of scale change mean-shift based on color features, and She et al. [21] proposed a method that uses edge

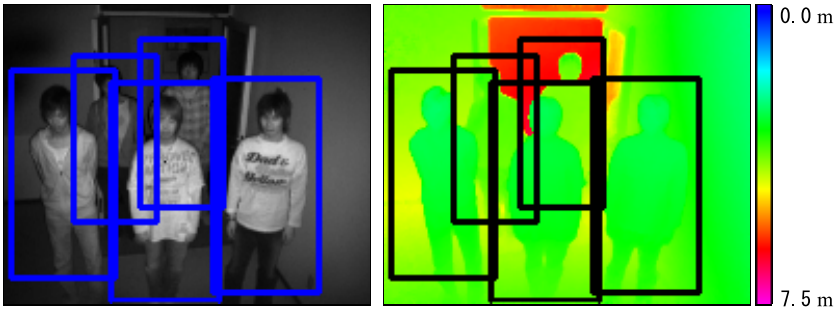


Fig. 16. Examples of human detection with a TOF camera.

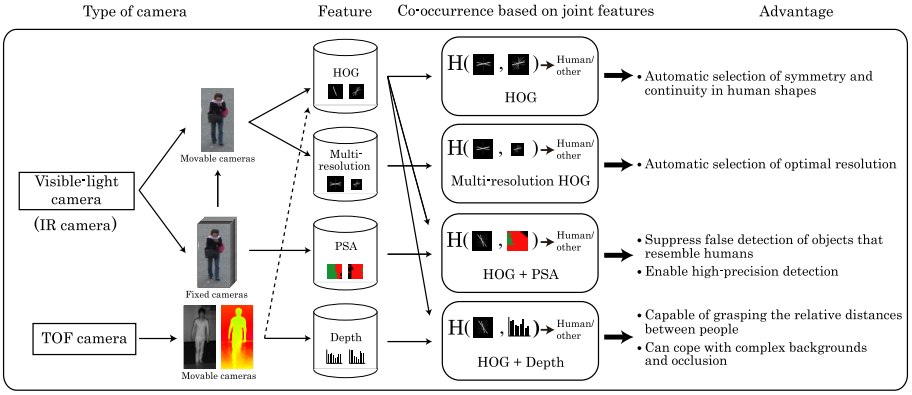


Fig. 17. Using joint features to represent co-occurrence features.

features. These features are used to form a weight-map of the mean-shift and are suitable for tracking the regions of a non-rigid object, but not suitable for the tracking of keypoints.

This section presents a mean-shift tracker to search the mode in image and scale spaces using a weight-map obtained by the SIFT technique. Our approach uses two interleaved mean-shift procedures to track the spatial location and to estimate the scale parameter of keypoints in an image. Since the SIFT feature is invariant to changes caused by rotation, scaling, and illumination, we obtain better tracking performance than that of conventional approaches such as the widely-used Kande-Lucas-Tomasi (KLT) feature tracker algorithm [22, 23]. Using the trajectory of the points tracked by the proposed method, we also show that it is possible to visualize a pedestrian traffic flow.

4.1 SIFT Feature Point Tracking

Since the SIFT descriptor computes invariant features from a local image patch, SIFT features around the keypoint tend to have high similarity in neighboring pixels. Our algorithm uses mean-shift searching based on a weight-map computed using the SIFT technique around the tracked keypoints. The weight-map is used to search a mode in image and scale spaces by using two interleaved mean-shift procedures. These two procedures are described below. Figure 18 shows a process of keypoint tracking using an image sequence.

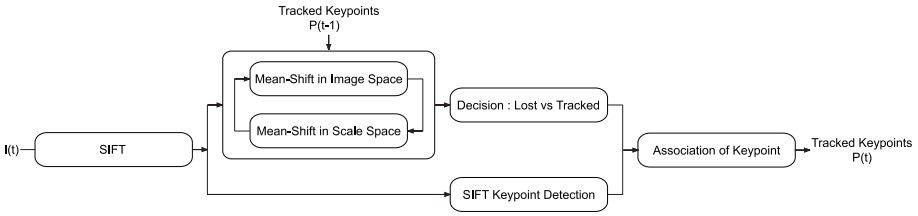


Fig. 18. Process of keypoint tracking using an image sequence.

Initial Tracking Point Detection Initial keypoints are detected by the SIFT keypoint detector and represented as a local feature by the SIFT descriptor, therefore, each detected keypoint has a 128-dimensional vector $\mathbf{v} = (v_0, \cdots v_{127})$ and a scale parameter s.

Mean-Shift Searching The mean-shift algorithm is a simple nonparametric method for locating the nearest mode of a sample distribution. It has been adopted as an efficient tracking technique. When the mean-shift method is applied to keypoint tracking, the gradient density is formed by the weight $\omega(\mathbf{x}_i, s)$ at each image pixel $\mathbf{x}_i$. The core of the mean-shift tracking algorithm is the computation of a keypoint motion vector from a location $\mathbf{x}$ to a new location $\mathbf{x}'$.

Generally, a weight map is determined using a color-based appearance model. In the work done by Comaniciu et al. [17], the weights were obtained by comparing a histogram q_u, where u is the histogram bin index, with a histogram of colors $q_u(\mathbf{x}_0)$ observed within a mean-shift window at the current location $\mathbf{x}_0$. In this paper, weight-maps are determined using the similarity between SIFT features at the location $\mathbf{x}_0$ of the previous frame $t-1$ and the current frame t. We augment the mean-shift tracker by using two interleaved mean-shift procedures to track the mode in image and scale spaces, which represents the spatial location and the scale parameter of the keypoint, repectively.

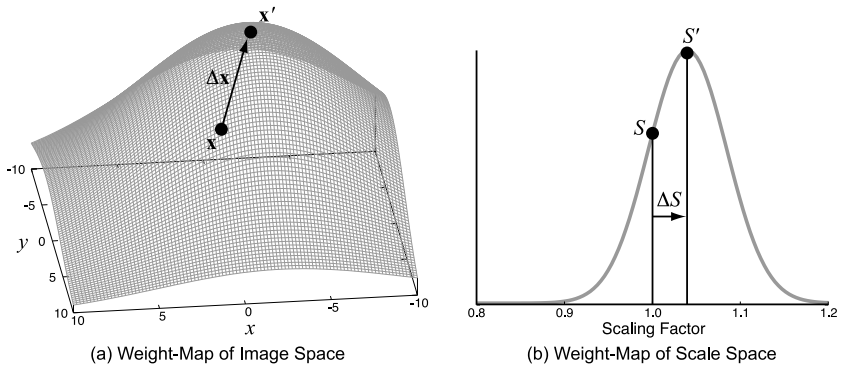


Fig. 19. Weight-map examples.

step1 Mean-Shift in Image Space

Given the scale s in the current frame, the SIFT features $\mathbf{v}_i$ are computed using following equation:

$$\mathrm{SIFT}(\mathbf{x}, s) = \mathbf{v}_{\mathbf{x}s} = (v_{\mathbf{x}s,0}, \cdots, v_{\mathbf{x}s,127}). \tag{2}$$

Then, we compute a location weight map $\omega(\mathbf{x}_i, s)$ from the distance between reference SIFT feature $\mathbf{v}_0$ and the SIFT feature $\mathrm{SIFT}(\mathbf{x}_i, s)$ at the location $\mathbf{x}_i$ with the scale s using the following equation:

$$\omega(\mathbf{x}_i, s) = \exp\left(\frac{-d(\mathbf{x}_i, s)^2}{2\sigma_d^2}\right), \tag{3}$$

where $d(\mathbf{x}_1, s)$ is the Euclidean distance given by

$$d(\mathbf{x}, s) = ||\mathrm{SIFT}(\mathbf{x}, s) - \mathbf{v}_0|| = \sqrt{\sum_{k=0}^{127}(v_{\mathbf{x}s,k} - v_{0,k})^2}. \tag{4}$$

Then the spatial mean-shift vector is obtained as

$$\Delta\mathbf{x} = \frac{\sum_{i=0}^{N} K_{loc}(\mathbf{x}_i - \mathbf{x}, \sigma_{xy})\omega(\mathbf{x}_i, s)(\mathbf{x}_i - \mathbf{x}_0)}{\sum_{i=0}^{N} K_{loc}(\mathbf{x}_i - \mathbf{x}, \sigma_{xy})\omega(\mathbf{x}_i, s)}, \tag{5}$$

where K_{loc} is a spatial kernel function given by

$$K_{loc}(\mathbf{x}, \sigma_{xy}) = \exp\left(\frac{-(x^2 + y^2)}{2\sigma_{xy}^2}\right). \tag{6}$$

Finally, we can get the new location $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$ from the mean-shift vector as shown in figure 19(a).

step2 Mean-Shift in Scale Space
Our approach uses a mean-shift procedure to estimate the scale parameter of the keypoint at the location obtained in step1. We create a scale weight-map $\omega(\mathbf{x}_i, s)$, which is a 1D array, using the following equation:

$$\omega(\mathbf{x}', sS_j) = \exp\left(\frac{-d(\mathbf{x}', sS_j)^2}{2\sigma_d^2}\right). \tag{7}$$

This mean-shift in scale space is performed on the 1D array of results to locate the mode, as shown in figure 19(b). The scale mean-shift vector is then obtained using this equation:

$$\Delta S = \frac{\sum_{j=0}^{M} K_{scale}(S_j - 1, \sigma_s)\omega(\mathbf{x}', sS_j)S_j}{\sum_{j=0}^{M} K_{scale}(S_j - 1, \sigma_s)\omega(\mathbf{x}', sS_j)}, \tag{8}$$

where S is the current scale, and K_{scale} is a kernel function for scale space given by

$$K_{scale}(S, \sigma_s) = \exp\left(\frac{-S^2}{2\sigma_s^2}\right). \tag{9}$$

Here, $S_j (j = 0, \cdots, M)$ is a numeric sequence that increases at equal intervals, and its median value is 1.0 (For example, $S_j = \cdots, 0.9, 1.0, 1.1, \cdots$). S_j is not a value on the scale parameter of the keypoint. S_j means a scaling factor of the scale parameter s for reference. If the value of S_j is 1.0, it means that there are no scale changes in the current frame. In the equation (8), we use $S - 1$ so that the response of the kernel function K_{scale} will be a maximum value where there is no scale change. The scale is updated by $s' = s\Delta S$ using the mean-shift vector ΔS in scale space.

step3 Iteration

Iterate by interleaving steps 1 and 2 until both $|\Delta\mathbf{x}| < \epsilon_{xy}$ and $|\Delta S - 1| < \epsilon_S$.

Rejection of Tracking Failure Point Our keypoint tracker sometimes loses features when they became occluded or leave an image. To make a decision whether a feature is lost or not, we compute the Euclidean distance of the SIFT features at the new location $\mathbf{x}'$, and previous location $\mathbf{x}$ using equation (4). If the distance is above a given threshold, the keypoint at the new location $\mathbf{x}'$ is deemed a lost feature point and rejected.

Association of Keypoints As shown in figure 18, we use the SIFT keypoint detector in parallel with a mean-shift procedure for keypoint tracking in order to add new keypoints that belong to any new objects appearing in the image. Finally, we obtain trajectories of these keypoints by associating tracked keypoints and newly detected keypoints.

Tracking Example of Non-rigid Object Figure 20 shows examples of keypoint tracking using our proposed method. In this video, pedestrians are walking in different directions. Each tracked point expresses the trajectory of the last 50 frames. We can see that our proposed method can obtain a greater number of long trajectories of keypoints than that obtained by KLT.

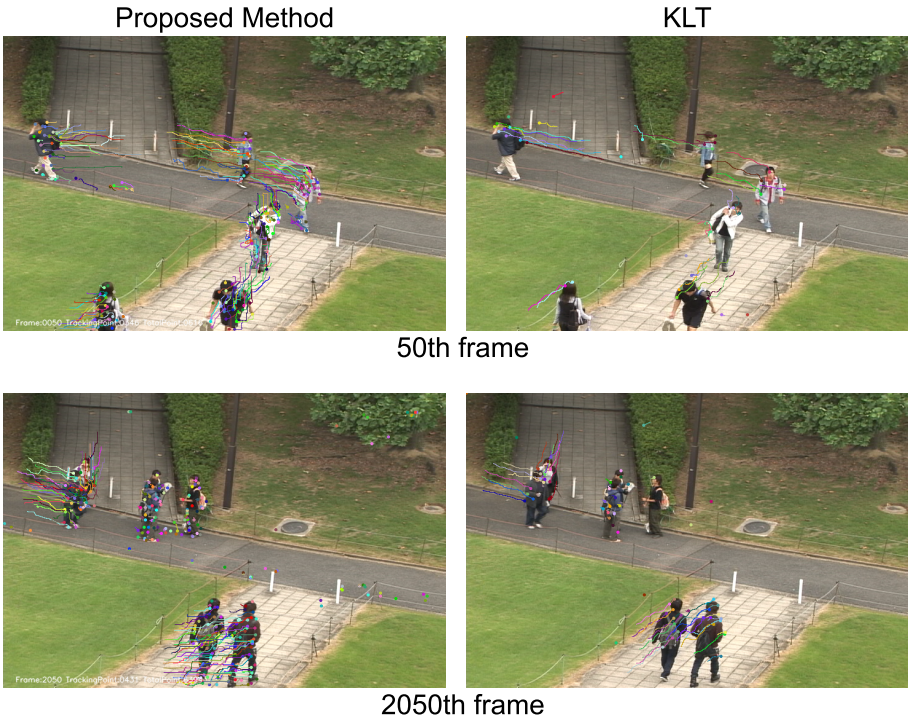


Fig. 20. Examples of feature point tracking for images of pedestrians.

4.2 Visualization of Pedestrian Traffic Flow

This section describes a technique used to visualize a pedestrian traffic flow. The technique uses the result of feature point tracking by the proposed method. The visualization procedure consists of two processes: a consistency check and flow representation.

Consistency Check In the visualization of pedestrian flow, it is important to be able to observe the direction and frequency of movement. To visualize pedestrian flow, we first check the consistency of a keypoint moving in a given

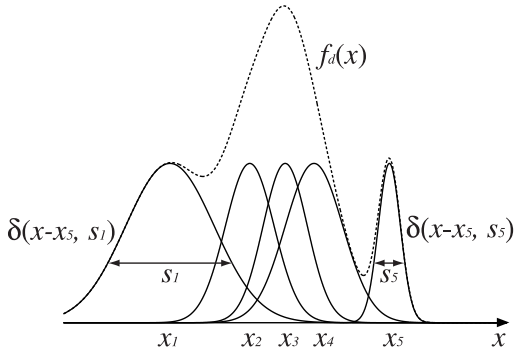


Fig. 21. Color intensity by density of points.

Fig. 22. Visualization by using scale information.

direction using the following equations:

$$\cos\theta = \frac{\mathbf{v}_t \cdot \mathbf{v}_{t-1}}{|\mathbf{v}_t||\mathbf{v}_{t-1}|} > th, \tag{10}$$

$$\mathbf{v}_t = (\mathbf{x}_t, \mathbf{x}_{t-1}), \mathbf{v}_{t-1} = (\mathbf{x}_{t-1}, \mathbf{x}_{t-2}). \tag{11}$$

If the value of $\cos\theta$ is close to 1, there are no great fluctuations in the direction of the movement. If the value of $\cos\theta$ is less than 0.9, we reject the keypoint as an outlier that is not good for using to visualize flow.

Flow Representation To express the movement by color information, a color is selected from a hue corresponding to the direction of the movement. The intensity of dense $f_d(\mathbf{x})$ in direction d at the location $\mathbf{x}$ is expressed by the following equation:

$$f_d(\mathbf{x}) = \sum_{t=1}^{T}\sum_{i=1}^{N} \delta(\mathbf{x} - \mathbf{x}_i^t, s_i), \tag{12}$$

$$\delta(\mathbf{x}, s) = \exp\left(\frac{-(x^2+y^2)}{2s^2}\right), \tag{13}$$

where T is total frames, N is number of tracking points, $\mathbf{x}_i^t$ is a location of the tracking point of the number i in frame t, and δ is a Parzen window function,

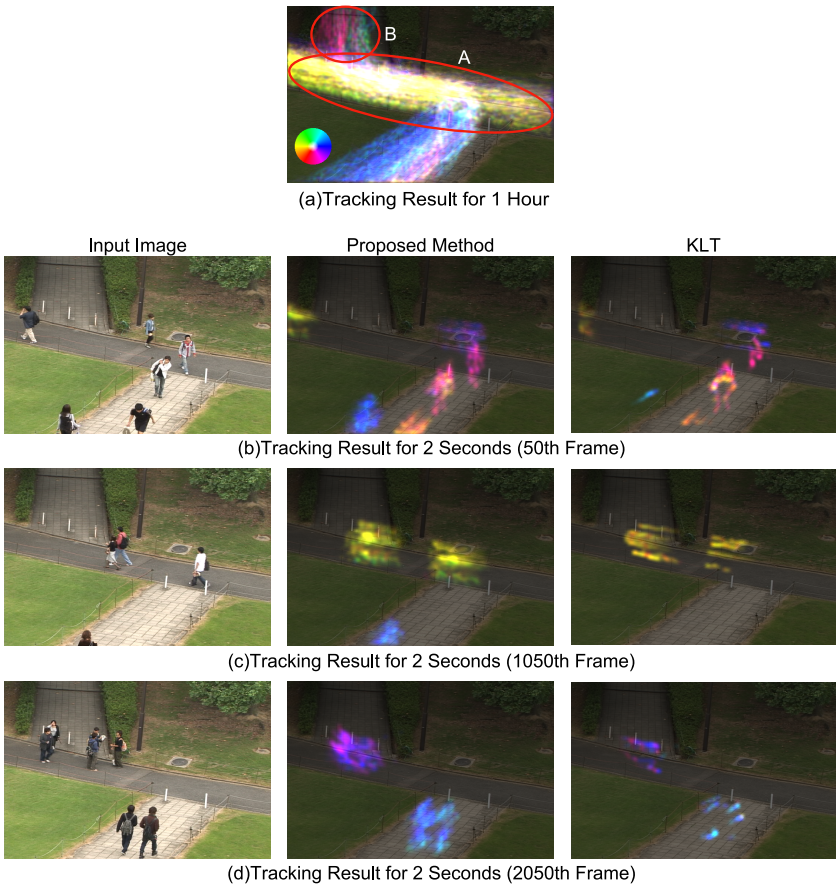


Fig. 23. Examples of visualization of pedestrian traffic flow.

which is based on Gaussian distribution. At this time, scale s_i of the tracking point is used as a standard deviation of Gaussian distribution, as shown in figure 21. The color intensity corresponding to the direction of the movement will be strongly expressed where the distribution density of a keypoint is high.
Figure 22 shows the value of s for a visualization example of pedestrian. Using the location and scale parameter of keypoints, we can obtain a rough silhouette of pedestrian, as shown in figure 22(c).

Visualization Example Figure 23(a) shows visualization examples of pedestrian flow accumulating tracked points over 1 hour(100,000 frames). The circle in the lower left shows a color map of the direction of the movement. From the visualization, we can see that there are a lot of people who were crossing to the left in area A. In area B, we also see that there are two movements in opposite directions. Figure 23(b) shows visualization examples of pedestrian flow for

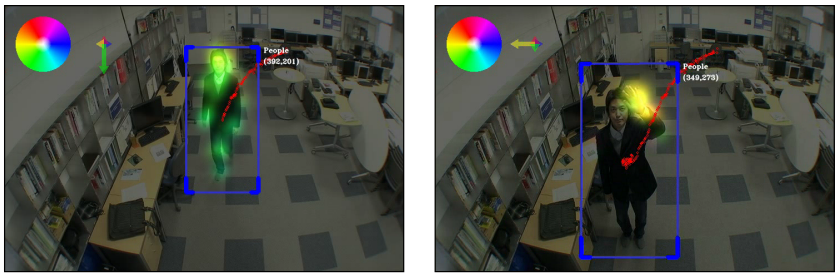


Fig. 24. Examples of people image analysis.

every 2 seconds (60 frames). Since the SIFT feature has a scale parameter, the proposed method can obtain better human shapes than that of the KLT.

5 Conclusion

In this paper we have introduced PIA technology that supports safe and secure everyday living, which is starting to being regarded as increasingly important. We also described human detection and human motion visualization techniques which can provide invisible robots with visual functions that allow them to observe humans. As shown in figure 24, for an invisible robot to be capable of recognizing the state of humans in environments that are always changing, it must be able to figure out where they are (human detection) and what sort of movements they are making (motion analysis). Visual recognition techniques such as these will be essential for the implementation of more advanced invisible robot sensing technology that can understand the behavior intentions of human users.

In the future, to enable invisible robots to provide services that are more stable and finely-tuned, it will become important to analyze what people are trying to do by using the relationships between people and other objects and their surroundings based on recognizing ordinary objects and understanding scenes. Also, by combining recognition techniques with special cameras such as TOF cameras that have been used in applications such as instrumentation, it is expected that these systems will become capable of more precise recognition by using information that cannot be obtained with ordinary visible-light cameras, thereby increasing their practical utility.

References

1. T. Kanade: “Invisible Robot” (in Japanese), pp. 449-465, (2004)
2. R. Collins, A. Lipton, H. Fujiyoshi and T. Kanade: “Algorithms for cooperative multi-sensor surveillance ”, Proceedings of the IEEE, Vol. 89, No. 10, pp. 1456-1477, (2001)
3. Aware House, http://www.gatech.edu/innovations/futurehome/

4. Easy Living, http://research.microsoft.com/easyliving/
5. People Image Analysis, http://www.consortium.ri.cmu.edu/
6. K. Levi and Y. Weiss: "Learning Object Detection from a Small Number of Examples: the Importance of Good Features.", IEEE Computer Vision and Pattern Recognition, vol. 2, pp. 53-60, (2004)
7. N. Dalal and B. Triggs: "Histograms of Oriented Gradients for Human Detection", IEEE Computer Vision and Pattern Recognition, pp. 886-893, (2005)
8. B. Wu and R. Nevatia: "Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors", IEEE International Conference on Computer Vision, vol. 1, pp. 90-97, (2005)
9. P. Sabzmeydani and G. Mori: "Detecting Pedestrians by Learning Shapelet Features", IEEE Computer Vision and Pattern Recognition , pp1-8, (2007)
10. T. Mita, T. Kaneko, B. Stenger and O. Hori: "Discriminative Feature Co-occurrence Selection for Object Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 7, pp. 1257-1269, (2008)
11. T. Mitsui, H. Fujiyoshi, "Object Detection by Joint Features based on Two-Stage Boosting", IEEE International Workshop on Visual Surveillance (in conjunction with ICCV '09), (2009)
12. R. E. Schapire and Y. Singer: "Improved Boosting Algorithms Using Confidence-rated Predictions", Machine Learning, No.37, pp.297-336, (1999)
13. Y. Yamauchi and H. Fujiyoshi, "People Detection Based on Co-occurrence of Appearance and Spatiotemporal Features", International Conference on Pattern Recognition, (2008)
14. H. Fujiyoshi and T. Kanade: "Layered detection for multiple overlapping objects", IEICE Transactions on Information and systems, pp. 2821-2827, (2004)
15. D. G. Lowe: "Distinctive image features from scale-invariant keypoints" International Journal of Computer Vision, Vol. 60, No. 3, pp. 91-110, (2004)
16. D. Comaniciu, P. Meer: "Mean shift analysis and applications" IEEE International Conference Computer Vision, pp. 1197-1203, (1999)
17. D. Comaniciu, P. Meer: "Real-time tracking of non-rigid objects using mean shift" IEEE Conference on Computer Vision and Pattern Recognition, pp. 142-149, (2000)
18. D. Comaniciu, P. Meer: "Kernel-based object tracking" IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 564-577, (2003)
19. K. Fukunaga and L. Hostetler : "The estimation of the gradient of a density function", with applications in pattern recognition IEEE Transactions on Information Theory, pp. 32-40, (1975)
20. R. Collins: "Mean-shift blob tracking through scale space" IEEE Conference on Computer Vision and Pattern Recognition, pp. 234-240, (2003)
21. E. G. Miller and K. Tieu: "Color eigenflows: Statistical modeling of joint color changes" ,IEEE International Conference on Computer Vision, pp. 607-614, (2001)
22. C. Tomasi and T. Kanade: "Detection and tracking of point features" Technical report, CMU-CS-91-132, (1991)
23. J. Shi and C. Tomasi: "Good features to track" IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, (1994)