

A Method for Visualizing Pedestrian Traffic Flow Using SIFT Feature Point Tracking

Yuji Tsuduki¹ and Hironobu Fujiyoshi²

¹ Dai Nippon Printing, Japan

² Dept. of Computer Science, Chubu University, Japan

hf@cs.chubu.ac.jp

<http://www.vision.cs.chubu.ac.jp/>

Abstract. This paper presents a method for visualizing a pedestrian traffic flow using results of feature point tracking. The Kanade-Lucas-Tomasi feature tracker algorithm for point feature tracking is widely used because it is fast; however, it is sometimes fails to accurately track non-rigid objects such as pedestrians. We have developed a method of point feature tracking using a scale invariant feature transform (SIFT). Our approach uses mean-shift searching to track a point based on the information obtained by a SIFT. 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, respectively. Since a SIFT feature is invariant to changes caused by rotation, scaling, and illumination, we can obtain a better tracking performance than that of a conventional approach. Using the trajectory of the points obtained by our method, it is possible to visualize pedestrian traffic flow using the location and scale obtained by SIFT feature point tracking.

1 Introduction

Visualizing a pedestrian traffic flow and its analysis are important for visual surveillance, marketing, and etc.. This paper 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 [1]. Mikolajczyk and Schmid [2] recently evaluated a variety of approaches and identified the SIFT algorithm as being the most resistant to common image deformations. Therefore, SIFT is commonly used in a number of real-world applications, such as image registration [3] and object recognition [4]. Keypoint matching using the Euclidean distance between SIFT features is a simple and very efficient way to track keypoints through an image sequence if the keypoints belong to rigid objects such as vehicles, as shown in Figure 1(a). However, keypoint matching sometimes fails to accurately track the keypoints on non-rigid objects such as pedestrians, as shown in Figure 1(b). This is because SIFTs are very sensitive to shape change in the image.

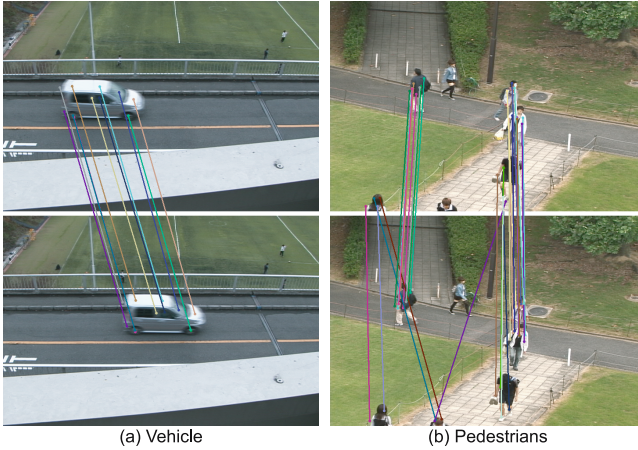


Fig. 1. Examples of SIFT keypoint matching. (a) Video of vehicle passing from left to right; (b) Video of pedestrians walking in different directions.

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 [5,6] locates the nearest mode of a point sample distribution [7,8]. Collins [9] proposed using a method of scale change mean-shift based on color features, and She et al. [10] proposed a method that uses edge 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.

In this paper, we propose 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 [11,12]. Using the trajectory of the points obtained by the proposed method, we also show that it is possible to visualize a pedestrian traffic flow.

The rest of this paper is organized as follows; Section 2 describes SIFT feature point tracking using two interleaved mean-shift procedures; Section 3 shows the experimental results; Section 4 shows visualization examples of pedestrian traffic flow by the proposed method; Section 5 summarizes and describes consideration and future work.

2 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.

2.1 Algorithm

Figure 2 shows a process of keypoint tracking using an image sequence following our method.

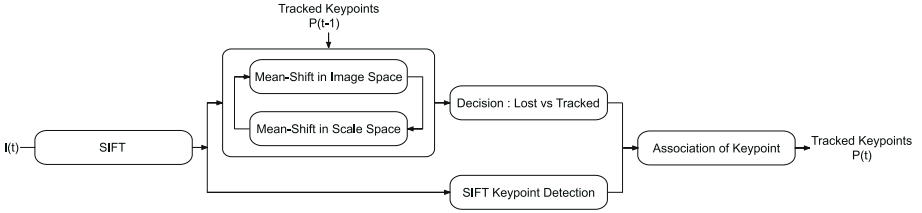


Fig. 2. Process of keypoint tracking using an image sequence

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 $\mathbf{v} = (v_0, \dots, v_{127})$ 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 recently 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. [6], 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, respectively.

Step1 Mean-Shift in Image Space

Given the scale s in the current frame, the SIFT features \mathbf{v}_i are computed using equation (3). 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 $\text{SIFT}(\mathbf{x}_i, s)$ at the location \mathbf{x}_i with the scale s using the following equation:

$$\omega(\mathbf{x}_i, s) = \exp\left(\frac{-d(\mathbf{x}_i, s)^2}{2\sigma_d^2}\right), \quad (1)$$

$$\begin{aligned} d(\mathbf{x}, s) &= \|\text{SIFT}(\mathbf{x}, s) - \mathbf{v}_0\|, \\ &= \sqrt{\sum_{k=0}^{127} (v_{\mathbf{x}s,k} - v_{0,k})^2}, \end{aligned} \quad (2)$$

$$\text{SIFT}(\mathbf{x}, s) = \mathbf{v}_{\mathbf{x}s} = (v_{\mathbf{x}s,0}, \dots, v_{\mathbf{x}s,127}). \quad (3)$$

Then the spatial mean-shift vector is obtained as

$$\Delta\mathbf{x} = \frac{\sum_{i=0}^N K_{loc}(\mathbf{x}_i - \mathbf{x}, \sigma_{xy})\omega(\mathbf{x}_i, s)(\mathbf{x}_i - \mathbf{x}_0)}{\sum_{i=0}^N K_{loc}(\mathbf{x}_i - \mathbf{x}, \sigma_{xy})\omega(\mathbf{x}_i, s)}, \quad (4)$$

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

$$K_{loc}(\mathbf{x}, \sigma_{xy}) = \exp\left(\frac{-(x^2 + y^2)}{2\sigma_{xy}^2}\right). \quad (5)$$

Finally, we can get the new location $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$ from the mean-shift vector as shown in Figure 3(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:

$$\omega(\mathbf{x}', sS_j) = \exp\left(\frac{-d(\mathbf{x}', sS_j)^2}{2\sigma_d^2}\right). \quad (6)$$

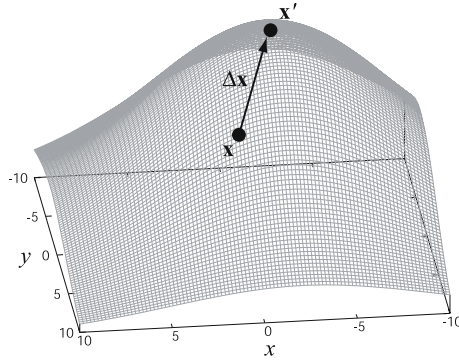
This mean-shift in scale space is performed on the 1D array of results to locate the mode, as shown in Figure 3(b). The scale mean-shift vector is then obtained using this equation:

$$\Delta S = \frac{\sum_{j=0}^M K_{scale}(S_j - 1, \sigma_s)\omega(\mathbf{x}', sS_j)S_j}{\sum_{j=0}^M K_{scale}(S_j - 1, \sigma_s)\omega(\mathbf{x}', sS_j)}, \quad (7)$$

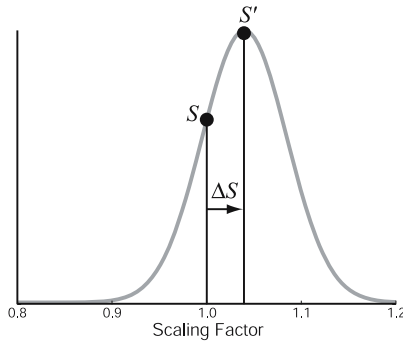
where S is the current scale, and K_{scale} is a kernel function for scale space given by

$$K_{scale}(S, \sigma_s) = \exp\left(\frac{-S^2}{2\sigma_s^2}\right). \quad (8)$$

Here, $S_j (j = 0, \dots, M)$ is a numeric sequence that increases at equal intervals, and its median value is 1.0 (For example, $S_j = \dots, 0.9, 1.0, 1.1, \dots$). S_j is not a value on the scale parameter of the keypoint. S_j means a scaling factor of the



(a) Weight-Map of Image Space



(b) Weight-Map of Scale Space

Fig. 3. Weight-Map Examples

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 (7), 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 (2). 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 2, 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.

2.2 Example of Scale Searching

The value of scale s corresponds to a local region centered on the keypoint for describing SIFT features. Figure 4 shows a tracking example of the location and scale when the image is magnified. White circles in Figure 4(a) show the location of the tracked keypoint, and blue circles show the size of the scale estimated by our proposed method. We can see that the same range of keypoints has been selected automatically, even though the size of the image has changed. Figure 4(b) shows the scaling rate of the image and the rate of the scaling rate estimated using our method. From the graph, we see that the ratio of scale estimated by our proposed method is almost the same as the ratio of image magnification. We used the least-square method to fit the plots, and we obtained a gradient of 0.91, which indicates a high correlation. Our proposed method can calculate the scale and the location of the feature point at the same time because it iterates the mean-shift search in image and in scale space.

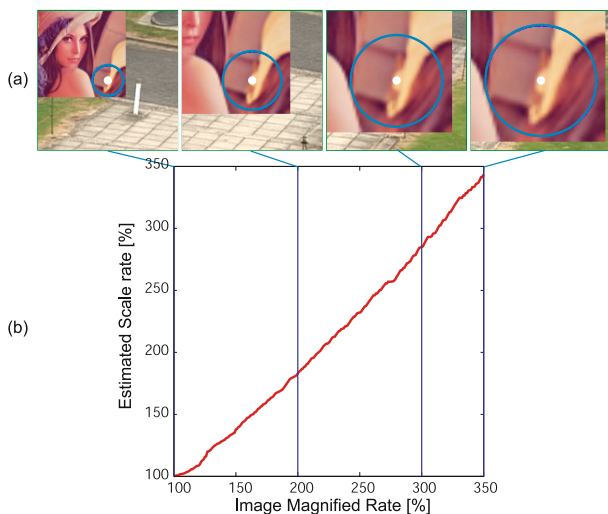


Fig. 4. Tracking Example of Location and Scale

3 Experimental Results

First, we outline our experimental setup and discuss the issue of generating ground-truth data. Then, this section contains our experimental results obtained using a synthesized image sequence and shows a pedestrian sequence as a tracking example.

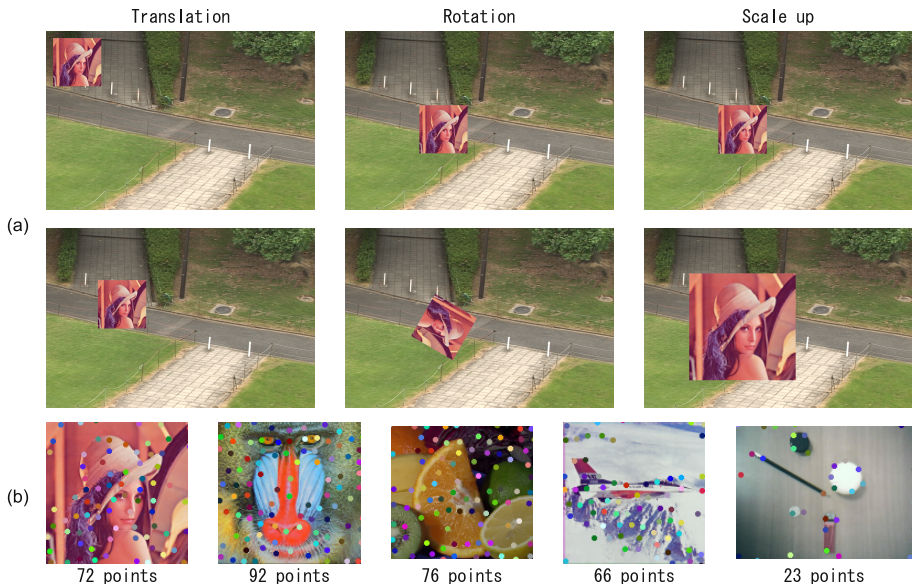


Fig. 5. Example of Experimental Image

3.1 Experimental Setup

We used synthetic images to quantify our method. We collected a dataset of images and applied the following transformations to each image: (1) translation; (2) rotation; and (3) scaling. To generate an image sequence, we overlapped the transformed image and the background image, as shown in Figure 5(a). For each image, we generated an image sequence of 180 frames per transformation. We investigated the difference in tracking performance between the KLT tracker and our method. To make the difference clear, the same initial keypoints were used in this experiment by both methods for translation and rotation sequences. Figure 5(b) shows examples of initial keypoints for each tracked image.

3.2 Ground-Truth Data

The transform (expressed as an affine motion) between two frames in a row is given. Therefore, ground-truth for each frame was made and used for the evaluation. We consider the match to be valid if the keypoint and ground truth are sufficiently close in location. We calculated the Euclidean distance between each tracked keypoint and ground-truth. If the distance was below the threshold, the tracked keypoint was determined to be a successfully tracked point. We then computed a tracking success rate from the total number of successfully tracked points.

3.3 Results

Figure 6 shows the tracking success rate calculated from all the frames (180 per frames for each sequence) used in 5 sequences. The horizontal axis represents

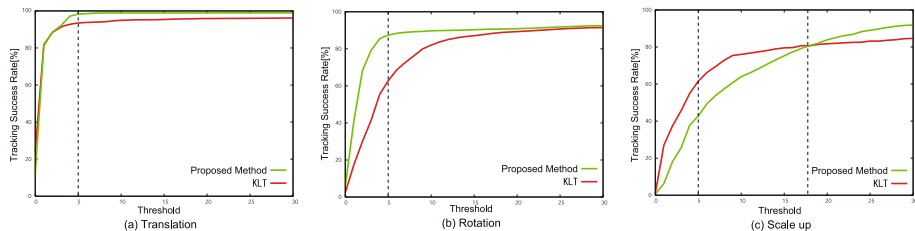


Fig. 6. Experiment Result

Table 1. Tracking Success Rate [%] in Threshold 5

	translation	rotation	scale-up	avg
proposed method	98.3	87.3	46.7	77.3
KLT	93.4	62.7	61.5	72.5

the threshold, and the vertical axis represents the tracking success rate. Table 1 shows the tracking success rate when the threshold is set to within 5 pixels. Our proposed method can obtain a higher tracking success rate than that of the KLT in the translation and the rotation. Because the SIFT features are invariant to rotation, high tracking accuracy is achieved. In the scale-up, the tracking accuracy of the KLT method is better than that of our proposed method when tracking threshold is below 18. Since initial keypoints selected by KLT are corner points, the KLT works well when tracking keypoints even if the scale changes. However, our proposed method becomes better than the KLT when the value of threshold is above 18. Because our method can estimate the scale of the keypoint adaptively, the use of mean-shift searching with a weight-map makes it attract to the mode in local areas.

3.4 Tracking Example of Non-rigid Object

Figure 7 shows examples of non-rigid object 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.

4 Visualization of Tracked Points

This section describes a technique used to visualize a pedestrian 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.



Fig. 7. Examples of Feature Point Tracking for Images of Pedestrians

4.1 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 direction using the following equations:

$$\cos \theta = \frac{\mathbf{v}_t \cdot \mathbf{v}_{t-1}}{|\mathbf{v}_t| |\mathbf{v}_{t-1}|} > th, \quad (9)$$

$$\mathbf{v}_t = (\mathbf{x}_t, \mathbf{x}_{t-1}), \mathbf{v}_{t-1} = (\mathbf{x}_{t-1}, \mathbf{x}_{t-2}). \quad (10)$$

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.

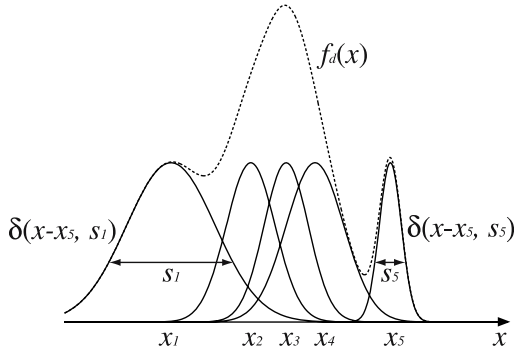


Fig. 8. Color Strength by Density of Points

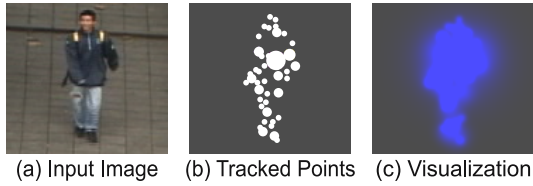


Fig. 9. Visualization by using Scale Information

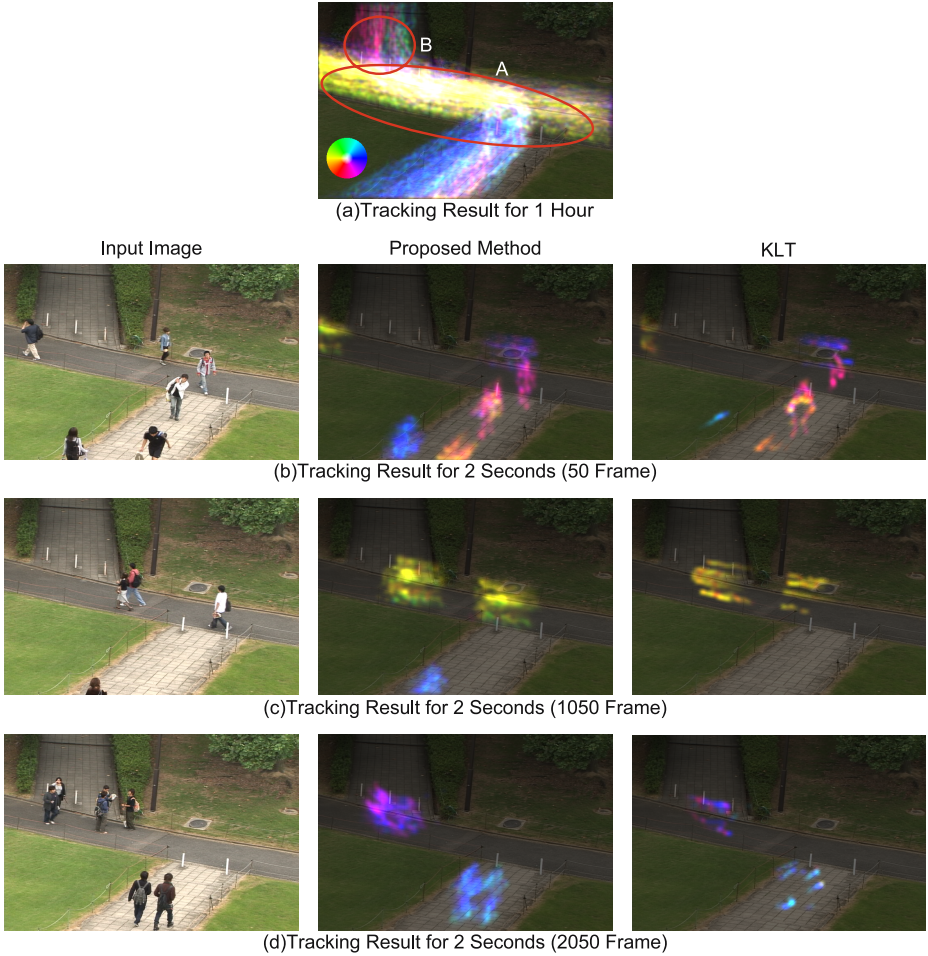


Fig. 10. Visualization Result

4.2 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:

$$f_d(\mathbf{x}) = \sum_{t=1}^T \sum_{i=1}^N \delta(\mathbf{x} - \mathbf{x}_i^t, s_i), \quad (11)$$

$$\delta(\mathbf{x}, s) = \exp\left(\frac{-(x^2 + y^2)}{2s^2}\right), \quad (12)$$

where T is total frames, N is number of tracking points, \mathbf{x}_i^t is a location of the chase point of the number i in frame t , and δ is a Parzen window function, 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 8. The color intensity corresponding to the direction of the movement will be strongly expressed where the distribution density of a keypoint is high. Figure 9 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 people, as shown in Figure 9(c).

4.3 Visualization Example

Figure 10(a) shows visualization examples of pedestrian flow accumulating tracked points over 1 hour(100,000 frames). The circle in the left 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 10(b) shows visualization examples of pedestrian flow for 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

We developed a feature point tracking method that used the mean-shift that of SIFT features. We demonstrated that high accuracy of keypoint tracking was archived for translation and rotation according to the SIFT features. Even if the tracking object was scaled up, it was still possible to track it by updating the scale of the SIFTs adaptively. Moreover, the visualization method of the feature point tracking result was shown as an example of the tracking of a pedestrian. In the future, we intend to develop a method to automatically detect movement in different directions from a regular flow in order to detect unusual events.

References

1. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision* (2004)
2. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: *Proceedings of Computer Vision and Pattern Recognition* (2003)
3. Brown, M., Lowe, D.: Recognising panoramas. In: *Ninth IEEE International Conference on Computer Vision* (2003)

4. Lowe, D.G.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision, Corfu, Greece (1999)
5. Comaniciu, D., Meer, P.: Mean shift analysis and applications. In: IEEE Int. Conf. Computer Vision, Kerkyra, Greece (1999)
6. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition (2000)
7. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence (2003)
8. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Transactions on Information Theory (1975)
9. Collins., R.: Mean-shift blob tracking through scale space. In: IEEE Conference on Computer Vision and Pattern Recognition (2003)
10. Miller, E.G., Tieu, K.: Color eigenflows: Statistical modeling of joint color changes. In: IEEE International Conference on Computer Vision (2001)
11. Tomasi, C., Kanade, T.: Detection and tracking of point features, Technical report, CMU-CS-91-132 (1991)
12. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition (1994)