Keypoint of Interest Based on Spatio-temporal Feature Considering Mutual Dependency and Camera Motion

Takahiro Suzuki and Takeshi Ikenaga
Graduate School of Fundamental Science and Engineering
Waseda university
Tokyo, Japan
Email: takahir0@toki.waseda.jp, ikenaga@waseda.jp

***Abstract*—Recently, cloud systems start to be utilized for services to analyze user's data in the region of computer vision. In these services, keypoints are extracted from images or videos and the data is identified by machine learning with large database of cloud. Conventional keypoint extraction algorithms utilize only spatial information and many unnecessary keypoints for recognition are detected. Thus, the systems have to communicate large data and require processing time of descriptor calculations. This paper proposes a spatio-temporal keypoint extraction algorithm that detects only Keypoints of Interest (KOI) based on spatio-temporal feature considering mutual dependency and camera motion. The proposed method includes an approximated Kanade-Lucas-Tomasi (KLT) tracker to calculate the positions of keypoints and optical flow. This algorithm calculates the weight at each keypoint using two kinds of features: intensity gradient and optical flow. It reduces noise of extraction by comparing with states of surrounding keypoints. The camera motion estimation is added and it calculates camera-motion invariant optical flow. Evaluation results show that the proposed algorithm achieves 95% reduction of keypoint data and 53% reduction of computational complexity comparing a conventional keypoint extraction. KOI are extracted in the region whose motion and gradient are large.**

***Keywords*-Keypoint extraction; SIFT; Temporal analysis.**

I. Introduction

Recently, Scale-Invariant Feature Transform (SIFT) [1] has attracted attention in computer vision because of its robustness in keypoint detection. Since SIFT can describe scale, rotation and illumination invariant features from images, matching between distinct images is executed accurately. By fully utilizing this characteristics, wide range of application is being considered. For example, it is used for object recognition [2], human or other object tracking [3], [4], recognizing panorama [5] and 3-D reconstruction [6]. In object recognition field, Bag-of-Features (BoF) was proposed by using combinations of SIFT descriptor. It generates one histogram from many keypoints which are extracted from one image. These are breakthroughes to recognize general objects. In addition, Support Vector Machine (SVM) was proposed as a machine learning algorithm. It utilizes non-linear kernel and classifies obtained keypoints with high accuracy. It needs to analyze a lot of keypoints to learn. It is also applied to recognition systems [7], [8] and shows high accuracy rate of recognition.

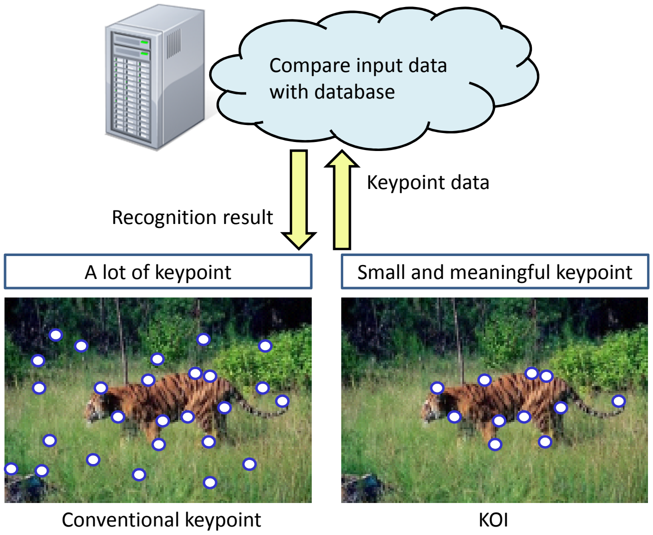

Figure 1: Conventional keypoints and KOI.

Recently, applications whose learned data is stored in cloud systems start to be released in relation to image recognition. A lot of keypoints are extracted from input images. All obtained keypoints are communicated with database. In this case, the amount of data makes it difficult to communicate data with high speed and stably. Recognition system needs only keypoints in object parts of interest. We call them Keypoints of Interest (KOI). If only KOI are extracted from input images, it generates two merits:

- Reduction of descriptor data communicated with cloud systems,
- Reduction of computational complexity of descriptor calculations.

Figure 1 shows the concept of this work. PCA-SIFT [9] which reduces the dimension of SIFT descriptor is also proposed. However, conventional keypoint extractions use only spatial data and extracts a lot of unnecessary keypoints. Other extended methods, SURF [10], GLOH [11], CSIFT [12] and ASIFT [13] , also utilize only spatial data.

This paper proposes a spatio-temporal keypoint extraction

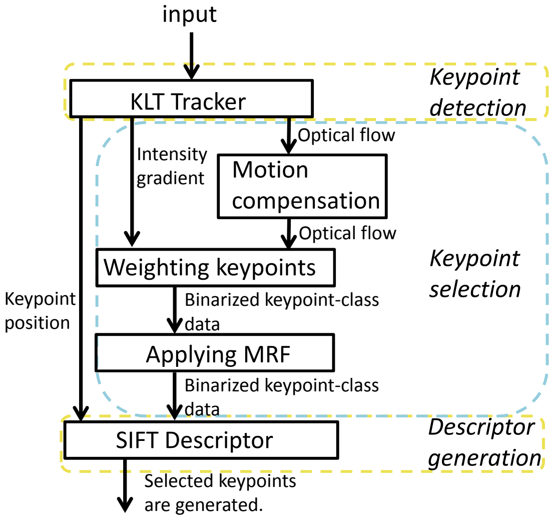

Figure 2: The flow of entire processing.

algorithm that detects only KOI based on spatio-temporal feature considering mutual dependency and camera motion. Candidate KOI are detected by Kanade-Lucas-Tomasi (KLT) tracker. KOI are selected by two kinds of features: intensity gradient and optical flow. However, important regions do not necessarily include information of large motion and gradient. Thus, we propose noise reduction by using Markov Random Field (MRF) that connects adjacent keypoints and determine the keypoint class. This algorithm extracts KOI that have many features including motion and intensity gradient. To deal with moving cameras, the compensation of motions is added. It can extract camera-motion invariant optical flow. Reduction of number of keypoints and computational complexity are evaluated in surveillance scenes.

Next section shows the proposed algorithm. Section III shows evaluation results. Finally, section IV concludes.

II. Keypoint extraction

Keypoint extraction is utilized for recognition and finding corresponding point between two images. The algorithm is divided into following two key parts.

- Keypoint detection
- Descriptor generation

The keypoint detection is the process which decides keypoint's position near characterized region. The SIFT descriptor generation calculates the histograms with information about neighboring region. These are common processes in all keypoint extraction methods. This paper utilizes low complexity keypoint extraction based on corner detection and plural images in database [14] as a conventional method. This algorithm also contains two key parts. It performs high-speed keypoint extraction maintaining almost same accuracy with SIFT.

III. KOI extraction based on spatio-temporal feature considering mutual dependency and camera motion

In this section, we show the method that extracts KOI from an input movie. This paper proposes the following four methods.

- Calculation of optical flow by approximated KLT tracker
- Weighting keypoints by two elements: intensity gradient and optical flow
- Applying MRF to keypoint class
- Calculation of camera-motion invariant optical flow by camera motion estimation

The entire flow containing these methods is shown in Fig. 2. We choose the KLT tracker [15], [16] as a keypoint detection method because it simultaneously calculates positions of keypoints and optical flow which is utilized in keypoint selection part. This algorithm contains the keypoint selection part between the keypoint detection part and the descriptor generation part. In the keypoint selection part, first, this algorithm weights keypoint by two elements and calculates values which describe likelihood of KOI at each keypoint. Then, these values are arranged and keypoint class is determined by threshold. However, the results include a number of noise because important regions do not necessarily include motion and gradient. Thus, keypoints are connected by MRF and a graph cut algorithm is used to reduce noise from the output keypoints. In addition, to deal with moving cameras, the motion compensation is executed by camera motion estimation and camera-motion invariant optical flows are extracted. SIFT descriptor is calculated at only selected keypoints. This section shows each algorithm in more detail.

A. Calculation of optical flow by approximated KLT tracker

KLT tracker is one of the algorithms which detect keypoints and calculate optical flow. It uses filters which computes second-order difference of adjacent pixels. It needs to refer many adjacent pixels during detection from general images with noise. According to the number of referred pixels, the process time becomes very long. Thus, an integral image and box filters are utilized for speeding up.

First, keypoint detection by box filter is shown. We use Hessian matrix, $\mathbf{H}$, which is composed of second-order difference of adjacent pixels:

$$\mathbf{H} = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}. \tag{1}$$

In general, the elements are weighted by Gaussian function. However, it is not suitable for an integral image because weighting has to be determined at each pixel. Thus, this filter is approximated and it becomes easy to compute by integral image. An approximated filter is shown in Fig. 3. This approximation is also used by SURF. L_{xx}, L_{yy}, L_{xy} are obtained by filter process of integral image. After that, they are used to compute the function which decides corners. When the position is a corner, it satisfies the equation,

$$V = det(\mathbf{H}) - \omega tra(\mathbf{H}) > T, \tag{2}$$

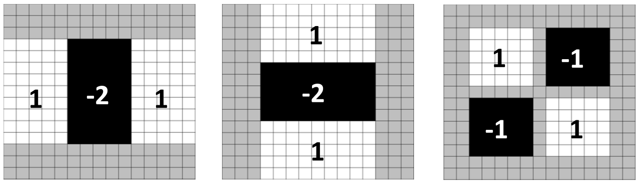

Figure 3: Approximated Filter (L_{xx}, L_{yy}, L_{xy}).

where ω is a parameter and T is a threshold. If the threshold becomes larger, corners decrease. It is adjusted to keep the number of keypoints optimal.

After the keypoint detection, The KLT tracker calculates optical flows. Optical flows are also calculated by second-order difference of adjacent pixels. Thus, the Hessian matrix is reused. The optical flow, $[u, v]$, is calculated by the equation:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}^{-1} \begin{bmatrix} L_{xt} \\ L_{yt} \end{bmatrix}. \quad (3)$$

L_{xt} and L_{yt} are calculated by the frame differences and filtering of x and y directions. In addition to gradient information, it utilizes the frame difference. This calculation also uses the integral image.

B. Weighting keypoints by two elements: intensity gradient and optical flow

In this paper, we choose two elements for weighting keypoints. The elements are intensity gradient and optical flow. With respect to intensity gradient, there is a high possibility that objects with many intensity gradients are the recognition targets. For example, book covers, posters and traffic signs are pointed out. With respect to optical flow, there is a high possibility that objects with motion are the recognition targets. For example, human, animals and vehicles are pointed out. Conventional keypoint extraction algorithms generally utilize only gradient information. Thus, it is expected to extract important keypoints including motion information if we use the temporal information. The weights of two elements are calculated at each keypoint which is obtained by the KLT tracker. The two different weights are normalized and summed up. This flow is described in Fig. 4.

Ways to obtain these values are shown next. The weight of intensity gradient is calculated by the Hessian detector [17]. The value, V, has already calculated in (1) and (2). V of (2) describes the strength of intensity gradient. It is obtained by the corner detection part of keypoint extraction. In other hand, the weight of optical flow is calculated by norm of optical flow. The value is obtained by (3). This calculation is a low complexity because the values have been already calculated. These two values are calculated at each keypoint and summed up after normalization. The weight is quantized data: $\mathbf{X} = \{x_1, x_2, \cdots, x_N\}$ where $x_i \in \{0, 1, \cdots, 255\}$. This weight data is binarized by threshold. Threshold is arranged by the number of KOI which the applications require. This process generates keypoint class $Y = \{y_1, y_2, \cdots, y_N\}$ at each keypoint where $y_i \in \{0, 1\}$. If the value of y_i is 1, the keypoint i is KOI.

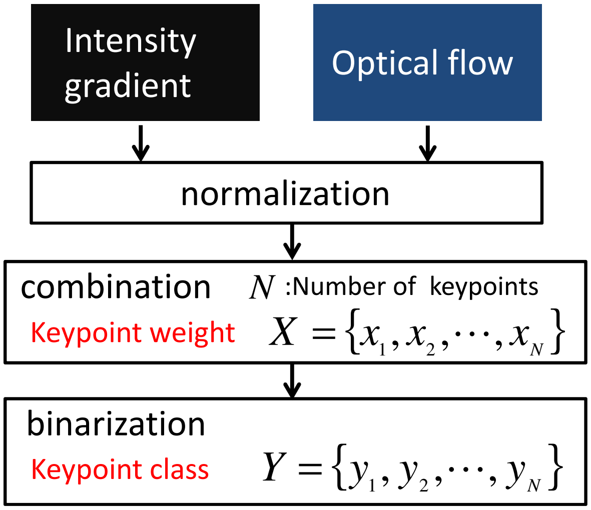

Figure 4: The flow of weight on keypoints.

However, important regions do not necessarily include motion and gradient. For example, gradient of human body is not large. It depends on their clothes. Several KOI can be extracted because the gradient of contour contains large values by proposed method in this section. Next, MRF is applied to reduce noise data and smoothing the keypoint class using the result of this section and adjacent keypoint data. The keypoint class is integrated on the each region.

C. Applying MRF to keypoint class

To solve the problem that keypoints in important region do not necessarily include large motion and gradient values, this paper applies MRF [18], [19] to keypoint class. MRF is usually used to reduce the noise of image in the region of image processing. MRF is the graph structure which represents the dependence between nodes. In this case, the nodes are keypoints and the dependency is defined in this section. Keypoints are connected by the weight of the distance from each other because the candidate keypoint whose adjacent keypoints are KOI tends to be KOI. The example of connections is shown in Fig. 5. In the circle, the keypoints are connected and they are easy to become same class. We utilize graph cut to reduce noise and determine keypoint classes. The graph cut algorithm changes keypoint class z_i and minimizes the energy equation:

$$E(Z) = \sum_i g_i(z_i) + \sum_{i,j} h_{ij}(z_j, z_i). \quad (4)$$

In this case, global solution is calculated because the keypoint class is binary. To solve this minimization problem, Min-Cut/Max-flow algorithm is used. Each function is defined as

$$g_i(z_i) = \lambda |y_i - z_i|, \quad (5)$$

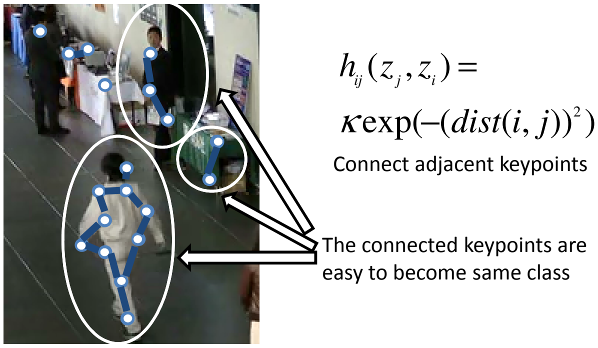
$h_{ij}(z_j, z_i) =$
$\kappa \exp(-(dist(i,j))^2)$
Connect adjacent keypoints
The connected keypoints are easy to become same class

Figure 5: The connection of keypoints.

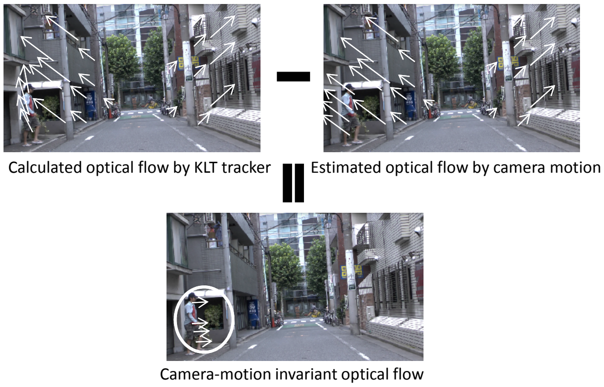
Calculated optical flow by KLT tracker
Estimated optical flow by camera motion
Camera-motion invariant optical flow

Figure 6: Calculation of camera-motion invariant optical flow.

$$h_{ij}(z_j, z_i) = \kappa \exp(-(dist(i,j))^2). \tag{6}$$

Equation (5) is data term. The outputted z_i is changed to approximate inputted y_i. Equation (6) is smoothing term. The strength of connection depends on distances between keypoints. We assume it is gaussian distribution. The nearer the keypoint distance, the stronger connection this function generates. $dist(i,j)$ represents the distance between keypoint i and keypoint j. λ and κ are parameters which are determined experimentally. They determine the strength of data term and smoothing term. If λ is larger than κ, the result approximates to the inputted data. If κ is larger than λ, the result approximates to the majority class of inputted keypoints. The calculated $Z = \{z_1, z_2, \cdots, z_N\}$ where $z_i \in \{0, 1\}$ is the output keypoint class. If the value of z_i is 1, the keypoint i is KOI. This calculation is faster than noise reduction of image which each node is a pixel because there are fewer nodes of the proposed method.

D. Calculation of camera-motion invariant optical flow by camera motion estimation

In the practical scene, cameras move like motion of pan or zoom. There are a number of scenes of zoom and pan in surveillance or in-vehicle camera. To apply this algorithm to moving cameras, this paper proposes a calculation of camera-motion invariant optical flow by camera motion estimation not to obtain large weight from the parts which do not move in fact. The overall flow is shown in Fig. 6 including zoom scenes. Optical flows are obtained by the KLT tracker at each keypoint. However, they include the influence of camera motion. For example, a number of optical flows which contain radical directions are generated like Fig. 6 from the parts which do not move in fact. Thus, we calculate the camera motion from these optical flows and the optical flows which are influenced by only camera motion is estimated. These are subtracted and the optical flow without influence of camera motion is obtained. Next, the method that estimates a camera motion is shown.

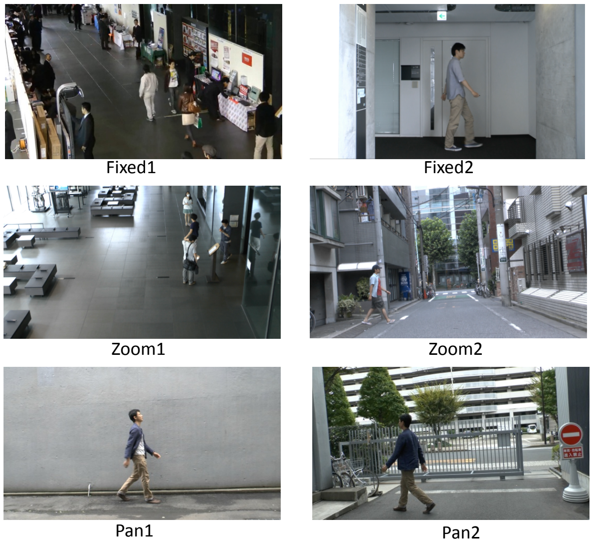

Figure 7: Test sequences.

A camera motion is estimated by all obtained optical flow by the KLT tracker. The motion vector of camera is defined as $\mathbf{T} = [t_x, t_y, t_z]^T$ where t_z represents the motion of a depth. The coordinate of the keypoint i is defined as $\mathbf{x}_i = [x_i, y_i, z_i]^T$. The optical flow, $\mathbf{v}_i = [u_i, v_i]^T$, is calculated by

$$\begin{aligned} u_i &= \frac{x_i t_z}{z_i} - \frac{f t_x}{z_i}, \\ v_i &= \frac{y_i t_z}{z_i} - \frac{f t_x}{z_i}. \end{aligned} \tag{7}$$

$\mathbf{T}$ is estimated by minimizing the function J:

$$J = \sum_{i=1}^{N} (\hat{u}_i - u_i)^T (\hat{u}_i - u_i), \tag{8}$$

where u_i is the calculated optical flow by (7) and $\hat{u}_i$ is the calculated optical flow by the KLT tracker. $\mathbf{T}$ is changed to minimize J. The result is substituted for (7) again. Estimated optical flows and obtained optical flows from inputted video are subtracted. The result is the camera-motion invariant optical flows.

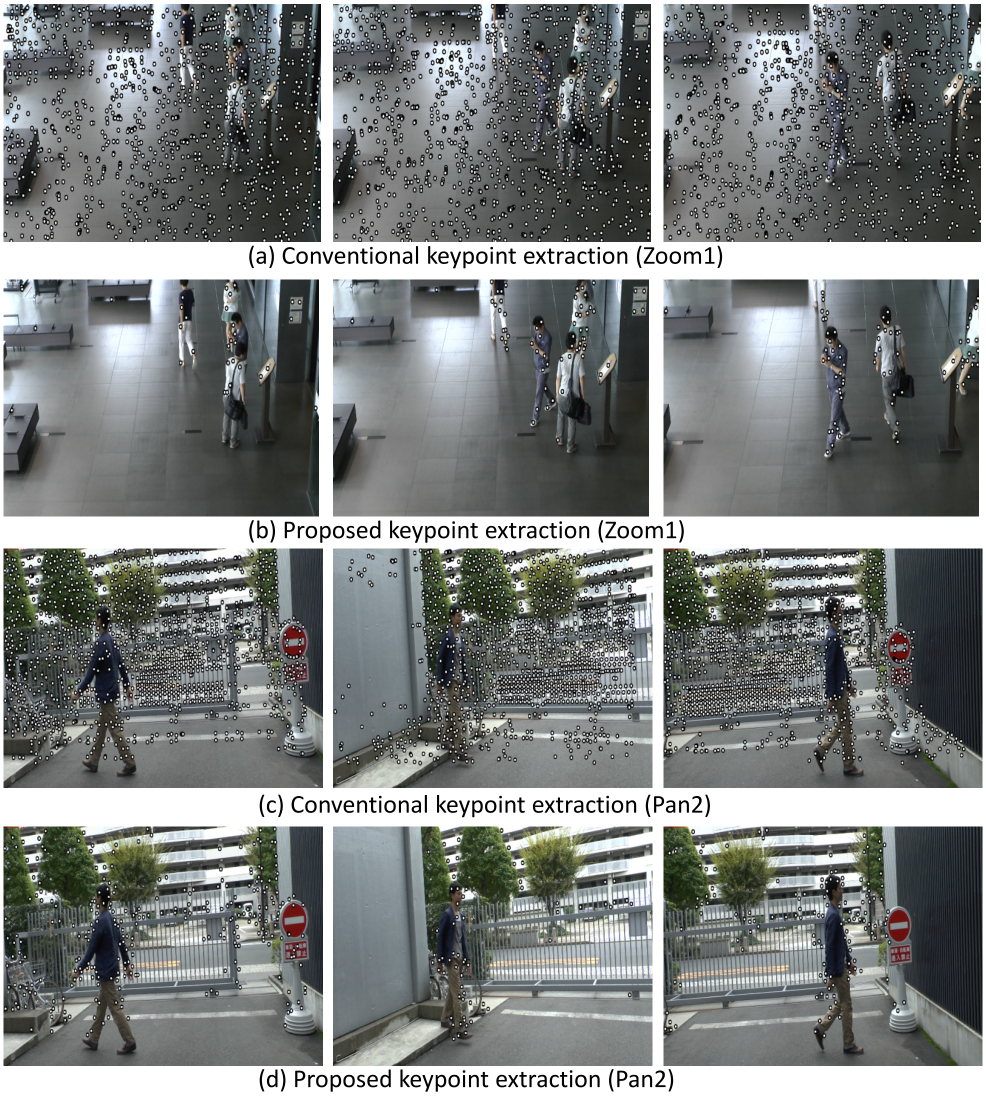
(a) Conventional keypoint extraction (Zoom1)
(b) Proposed keypoint extraction (Zoom1)
(c) Conventional keypoint extraction (Pan2)
(d) Proposed keypoint extraction (Pan2)

Figure 8: The comparison between (a) conventional keypoint extraction and (b) proposed algorithm in movie1, (c) conventional keypoint extraction and (d) proposed algorithm in movie2.

IV. Evaluation results

This section shows evaluation results that compare the proposed method with the general keypoint extraction which utilizes the corner detector and SIFT descriptor introduced in section 2. The development environment on software is Visual Studio C++ 2008. CPU is Intel Core i7-2600 CPU 3.40GHz. The resolution of the video we used is Full-HD (1920×1080), 60 fps. In this paper, we evaluated six test sequences. We assume the surveillance cameras and each movie includes scenes that people walk on the path. In addition, they are taken by three camera motions including fixed cameras, zoom and pan to confirm effectiveness of camera motion estimation. The test sequences are shown in Fig. 7.

First, the number of keypoints which are detected by both methods are compared in Tab. I. It shows the average among all frames of the movie. The proposed algorithm achieves the 94%, 96%, 96%, 95%, 94% and 90% reduction of keypoints in all movies. Almost same results are obtained from Fixed1-2, Zopm1-2 and Pan1. However, Pan2 is low reduction comparing with others. It is considered that movie of Pan2 includes complex texture that has large illuminance gradients in background. In Fixed1, several keypoints in poster or other display items whose gradient is large are extracted. From these parts, many KOI are obtained in Pan2. In all video, the reduction of keypoints is confirmed. In addition, processing time is compared. The proposed algorithm reduces about 75%, 78%, 52%, 58%, 53% and 50% computational complexity than the conventional keypoint extraction in all movies. In the processing of Fixed1-2, the motion estimation parts are excepted. Thus, the complexity reduction is higher than others.

TABLE I: The number of keypoint and processing time between conventional method and proposed method.

		Conventional method	Proposed method
Fixed1	Number of keypoints	1192	66
	Processing time	754	190
Fixed2	Number of keypoints	1188	53
	Processing time	821	179
Zoom1	Number of keypoints	1202	47
	Processing time	759	367
Zoom2	Number of keypoints	1205	63
	Processing time	851	354
Pan1	Number of keypoints	1121	64
	Processing time	765	358
Pan2	Number of keypoints	1143	108
	Processing time	771	380

In other movies, motion estimation is calculated and almost same complexity reductions are obtained. In all video, the reduction of computational complexity is confirmed.

Figure 8 shows the video result of the conventional method and the proposed algorithm. The white circles are the keypoint obtained by each algorithm. It shows the proposal detects keypoints from only human which moves largely and outstanding texture whose gradient is large. In other video, the proposed algorithm extracts a number of keypoints from human body and the part including outstanding texture. By using only these keypoints, it is expected to analyze human or other outstanding object behaviors in surveillance and in-vehicle cameras combining motion features. In this algorithm, the parameters during weighting gradient and motion can be adjusted. Thus, if we want to obtain keypoints from only human, the parameters are adjusted to obtain the intended keypoints. The parameters can be determined by machine learning algorithm which learns correct KOI in advance. The correct KOI is determined by applications which this algorithm is applied. In this paper, application is not specified. Thus, the weight of gradient and motion is same in this evaluation.

V. Conclusion

Reduction of data amount of keypoints and reduction of computational complexity are required for cloud application. Conventional keypoint extractions utilize only spatial information and extract a lot of unnecessary keypoints. This paper proposes a keypoint selection algorithm from many keypoints including unnecessary ones based on spatio-temporal feature considering mutual dependency and camera motion. The proposed method includes an approximated KLT tracker to calculate the positions of keypoints and optical flow. It calculates the weight at each keypoint using two kinds of features: intensity gradient and optical flow. It reduces noise by comparing with states of surrounding keypoints. Optical flows are compensated by camera motion estimation and it calculates camera-motion invariant optical flows. Evaluation results show that the proposed algorithm achieves about 95% reduction of keypoints and 53% reduction of computational complexity. KOI are extracted in human bodies that move widely and the objects whose gradient is large. This algorithm is expected to be applied to surveillance cameras and in-vehicle cameras when they start to utilize cloud system to recognize motion of humans or other outstanding objects.

Acknowledgment

This work was supported by KAKENHI (23300018).

References

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int.Journal of Computer Vision, 60, pp. 91-110, 2004.

[2] D. G. Lowe, “Object recognition from local scale-invariant features,” In International Conference on Computer Vision, Corfu, Greece, pp. 1150-1157, 1999.

[3] Yuji Tsuzuki, Hironobu Fujiyoshi, Takeo Kanade, “ Mean Shift-based Point Feature Tracking using SIFT,” Journal of Information Processing Society, Vol. 49, No. SIG 6, pp. 35-45, 2008.

[4] Huiyu Zhou , Yuan Yuan , Chunmei Shi, “ Object tracking using SIFT features and mean shift,” Computer Vision and Image Understanding, v.113 n.3, pp. 345-352, 2009.

[5] Matthew Brown and David G. Lowe, “ Recognising panoramas,” International Conference on Computer Vision, pp. 1218-25, 2003.

[6] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, R. Szeliski, “ Building rome in a day,” In ICCV, 2009.

[7] M. Pontil and A. Verri, “Support Vector Machines for 3D Objec Recognition, ” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.20, no.6, pp.637-646, 1998.

[8] G. Guo, S.Z. Li and K. Chan, “ Face Recognition by Support Vector Machines, ” Proceedings of the 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition, pp.195-201, 2000.

[9] Y. Ke, R. Sukthankar, “ A More Distinctive Representation for Local Image Descriptors,” Proceedings of Computer Vision and Pattern Recognition, pp. 506-513, 2004.

[10] „H.Bay, T.Tuytelaars and L. V. Gool, “SURF: speeded up robust features, ” In ECCV, pp. 404-417, 2006.

[11] Krystian Mikolajczyk, Cordelia Schmid, ”A performance evaluation of local descriptors,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp. 1615–1630, 2005.

[12] A.E. Abdel-Hakim, A.A. Farag,”Csift: a sift descriptor with color invariant characteristics,” Proceedings of the Computer Vision and Pattern Recognition, pp. 1978–1983, 2006.

[13] J. M. Morel and G. Yu, “Asift: A new framework for fully affine invariant image comparison, ” SIAM Journal on Imaging Sciences, 2, 2, pp. 438-469, 2009.

[14] Takahiro Suzuki, Takeshi Ikenaga, “ Low Complexity Keypoint Extraction Based on SIFT Descriptor and Its Hardware Implementation for Full-HD 60 fps Video,” IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E96-A, No. 6, pp. 1376-1383, 2013.

[15] C. Tomasi and T. Kanade, “Detection and Tracking of Point Features,” Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[16] Bruce D. Lucas and Takeo Kanade, “ An Iterative Image Registration Technique with an Application to Stereo Vision,” International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.

[17] Beaudet, P. R., “ Rotational invariant image operators,” In Proceedings of the 4th International Joint Conference on Pattern Recognition (ICPR), pp. 579-583, 1978.

[18] K. Tanaka, “ Statistical-mechanical approach to image processing,” J. Phys. A: Mathematical and General, vol. 35, no. 37, pp.R81-R150, 2002.

[19] A. Willsky, “ Multiresolution Markov models for signal and image processing,” Proceedings of the IEEE, vol. 90, no. 8, pp. 1396-1458, 2002.