Paper

Automatic Self-Improvement Scheme in Optical Flow-Based Motion Estimation for Sequential Fisheye Images

Arief Suryadi Satyawan[†1, †4], Junichi Hara[†2, †3] and Hiroshi Watanabe[†1, †2] (member)

Abstract   This paper aims to present an innovative design of motion estimation for sequential fisheye images. This design is an extended version of the original Lucas and Kanade's (LK) concept that used to design for calculating optical flow from general perspective images. The extended design consists of the LK concept and an additional self-improvement mechanism that automatically finds the maximum performance of the estimated motion. This extended scheme works much better than the original LK's idea or some block-based motion estimations. Moreover, to some extent, this proposed method is working extremely well to overcome some critical characteristics of the sequential fisheye images. These characteristics include distortion error on the fisheye image area, inconsistent brightness level, fluctuating number of object motion, changing the shape of object motion, or poor camera stability.

Keywords: motion estimation, sequential fisheye images, optical flow, Lucas and Kanade, self-improvement mechanism.

1. Introduction

For about three decades, the motion estimation technique has been developed and mainly deployed for many applications. These applications can be found in the object tracking, estimating 3D structure, video compression, video frame-rate conversion, and video surveillance system. The motion estimation technique forecast the apparent motion in time-varying images that obtained from the projection process of the real-world one [1]. This motion occurs because of exact object motion, camera motion, an illumination change, and noise. The objective of the motion estimation technique is then to define a motion vector for each pixel or group of pixels in an image by using a specific calculation scheme. This calculation scheme involves the observed image and at least one of its neighbors.

To the extent our observation, at least there are two types of motion estimation scheme that have been

developed massively in recent years [1]. The first is optical flow-based motion estimation that calculates the apparent motion vectors directly from two successive images. The calculation performs to solve the brightness constancy assumption between pixels in an image and each counterpart pixel in the following or previous image. The optical flow is very well known in computer vision applications. Some of them are mentioned above. The second one is block-based motion estimation. This concept directly calculates the similarity of each block of pixels in an image with each counterpart block of pixels in one of the following or previous images. Therefore, a motion vector can be obtained for each block. The block-based motion estimation is very valuable in the video compression application.

Besides the two major motion estimation concepts, other schemes have been developed in recent years [1]. Among these, there is the global motion estimation which deals with a video sequence that contains camera motion. This motion then is modeled by parametric transformations. How to estimate the parametric transformations is known as the global motion estimation. This motion estimation is essential for the sports video content analysis, including zoom in and out applications. Another method is the motion estimation in the transform domain. This method utilizes correspondence between translations in the spatial domain with a change of phase of the Fourier coefficients. Therefore, the motion is estimated from a

phase correlation between blocks of pixels. In one case, the transform domain approach uses a discrete transform rather than the Fourier transform. That is applicable since the discrete cosine transform (DCT) has been early applicable in the video coding application. One more is the pel-recursive motion estimation. The concept calculates the motion vectors recursively to get the best result. In other words, to attain the least error of motion vectors, the calculation of motion vectors can be done many times.

So far, the above motion estimation concepts have successfully worked for serving general perspective images. However, applying those techniques for taking advantage of the fisheye images, particularly for the above applications, still become the most significant challenge to be faced. Despite the fact that the fisheye images provide more extensive visual information than the general perspective images, these images suffer from some distortion (radial and tangential) due to the projection characteristic of the fisheye lens[2)~4)]. Therefore, the motion of an object shown in the sequential fisheye images tends to move in a nonlinear fashion, and at the same time, that object's shape deforms irregularly.

Hard efforts have been conducted to develop a motion estimation for serving omnidirectional images. They firstly found a way of transforming the fisheye images to appropriate general perspective images. Therefore, the advanced motion estimation schemes with slight modification can be applied to further applications. I. Bauermann et al.[5)] came up with an idea of converting omnidirectional images (360^0) to panoramic images or general perspective images before applying the existing motion estimation scheme in MPEG-4 AVC/H.264 video coding standard. It seems like the conversion process is started from pre-warping the omnidirectional images to panoramic images. After that, a resampling step is applied to scale the image in horizontal and vertical direction. By doing this, the panoramic images can be feed to the motion estimation process in the encoder.

Z. Jiali et al.[6)] proposed the affine motion compensated frame for estimating motion from a panoramic video sequence. In their concept, a general perspective camera captures the wide-field-of-view along a given axis on a cylindrical or spherical surface. Furthermore, image stitching and registration method are applied to obtain a continuity of series of images. Since there is not only translation motion inside a frame, but also non-translation motion between two frames, the motion in

this video sequence is assumed as geometrical affine motion. Therefore, the well-known Inter-Frame predictive scheme adopted by the International Telecommunication Union, Telecommunication standardization sector (ITU-T), H.26x, or Moving Picture Experts Group (MPEG), MPEG-X standard, can be modified to serve this kind of video sequence by using the affine motion compensated frame.

A. Djamal et al.[7)] introduced the adapted block-matching method for optical flow estimation in catadioptric images. In their idea, the modified version of the classical block-based motion estimation is applied for estimating motion directly from either synthetic catadioptric images or the real one. The modification is done by changing the shape and size of both the block of pixels and the window search area.

This research focuses on developing a motion estimation scheme that can be used directly for the fisheye images. For some application, such as the video frame-rate conversion, slow-motion scene, scene flow, and fisheye compression, shifting the correction process to the final processing is a useful strategy.

Furthermore, the basic concept of motion estimation applied to this research is the optical flow. By assuming that this concept will perform very well to overcome the deformation of objects on the fisheye images, the research firstly applies the basic idea of optical flow that previously proposed by Lucas and Kanade (LK), Horn and Schunck (HS), and an equidistant motion model to obtain preliminary results. From then on, a proposed concept is employed to achieve a better result. The proposed concept extends the LK's scheme with a supplementary self-improvement mechanism that autonomously looks for the highest performance of the motion estimation.

This method is similar to the pel-recursive motion estimation, particularly in making use of updating mechanism of motion vector value according to the previous prediction error. Nevertheless, either the way of calculating motion vectors or updating the result is very different. The conventional pel-motion estimation recursively estimates a motion vector of a pixel directly from neighboring pixels in the same frame (in scanned order) or to the corresponding pixel location in the next image frame. The refinement process is then performed by finding the least error between the current and the previous estimation[1)]. In the proposed scheme, on the other hand, the calculation of motion vectors is carried out between two successive frames by using LK's

scheme, while the further refinement process relies on the quality of a reconstructed image produced by transforming the first image by using the obtained motion vectors as transformation parameters.

The advantage of the proposed scheme is that the obtained motion vectors are more reliable, although, the motion speed of an object in the fisheye images area varies. Moreover, the proposed scheme can be used for overcoming motion in the fisheye images that caused by a change of intensity, a change of object shape, fluctuating number of object motion, or poor camera stability. If such fisheye image characteristics are assumed as integrated parts of higher order Tylor series that is ignored when the optical flow constraint problem is defined, the proposed method can be a practical solution for the absence of this parameter.

2. Motion Estimation Design Methodology

The fundamental process of estimating motion directly from sequential fisheye images cannot be separated from the performance evaluation mechanism. This because it is critical to confirm that the predicted motion occurs accurately. Therefore, it can be used for further applications. For that reason, in this research, the organic method of motion estimation is designed to follow a flowchart shown in Fig. 1.

The images input is two successive fisheye images. These images are taken from a sequential fisheye image. Every pair of fisheye images is then inputted into an optical flow calculation process. The heart of estimating motion between the two consecutive fisheye images is in this process. The optical flow calculation process will produce motion vectors from the first fisheye image to the second one (forward motion vectors). These vectors
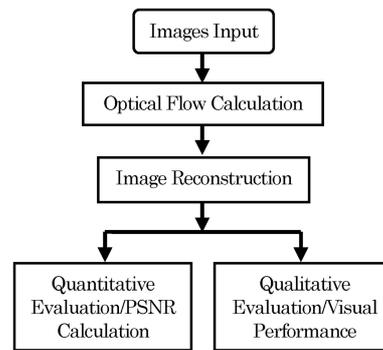

Fig. 1 The motion estimation design methodology.

also denote as estimated motion vectors. In the next step, to be able to evaluate the performance of the estimated motion vectors, an imitation of the second fisheye image firstly has to be reconstructed. This step transforms the first fisheye image by making use of the motion vectors. Therefore, a reconstructed fisheye image can be generated. This reconstructed fisheye image finally can be evaluated quantitatively or qualitatively at the final process by comparing it with a proper fisheye image reference (e.g., the second fisheye image).

3. The Sequential Fisheye Images Input

There are twelve sequential fisheye images trained for evaluating the performance of each motion estimation concept. The characteristics of each sequential fisheye image, such as the number of fisheye image (frames), types of information, and information characteristic, are described explicitly in Table 1. The information in all sequential fisheye images, except the cube and flowers, is captured by a Ricoh Theta S fisheye camera[8]. In this experiment, the data is only obtained through one side lens. The cube sequence is developed by using Blender software, while the flowers sequence is collected from

Table 1 The characteristics of sequential fisheye images.

The Name of Sequential Fisheye Images	The Number of Frames	Frame Resolution	Types of Information	The Characteristic of Information
Cube (CQ)	209	640x640	Synthetic	A cube moves from left to right side with constant speed.
Flowers (FQ)	118	1088x1088	Synthetic	Flowers blossom freely.
Hand (HQ)	151	640x640	Real object	A hand moves from left to right side with constant speed.
Man (MQ)	93	640x640	Real object	A man sits in front of the fisheye camera and moves his hands in opposite direction rapidly.
Truck I (TQ-I)	13	960x960	Real object	A truck moves fast toward to the fisheye camera.
Truck II (TQ-II)	40	960x960	Real object	A truck moves away out of the fisheye camera.
Car I (CQ-I)	14	960x960	Real object	A small white car moves toward to the fisheye camera with constant speed.
Car II (CQ-II)	19	960x960	Real object	A small red car moves toward to the fisheye camera with constant speed.
Train (TQ)	84	960x960	Real object	A small train moves from right to left side with constant speed.
Traffic I (TRQ-I)	117	960x960	Real object	A small number of cars move slowly on the road.
Traffic II (TRQ-II)	99	960x960	Real object	A large number of cars move slowly on the road.
People (PQ)	158	960x960	Real object	Many people walk with constant speed.

the website of the multimedia communications and signal processing (LMS), Friedrich - Alexander University Erlangen - Nurnberg (FAU), Germany[9]. The camera position is in a static position when it is capturing the information contained in some sequential fisheye images, except the traffic I, traffic II, and people. Therefore, the last three fisheye image sequences also consist of small camera motion.

4. Optical Flow Calculation

The optical flow previously is used for obtaining motion vectors by calculating the spatiotemporal intensity derivative of two consecutive perspective images[1)10)].

Now it can be utilized in the same manner for serving two consecutive fisheye images, and it is applicable for all fisheye images in a fisheye image sequence.

The basic concept of the optical flow comes from an assumption of brightness similarity of pixels between two successive fisheye images (FN1 and FN2), as shown in Fig. 2. In this figure, FN denotes frame number (the fisheye image number). The assumption also means that intensity of a pixel projected on FN1 at t ($I_1(x_1, y_1)$) is similar to its emergence on FN2 at $t+1$ ($I_2(x_2, y_2)$), even though they are not in the same position due to the motion. Therefore, the assumption can be expressed as follows:

$$I_1(x_1, y_1, t) = I_2(x_1 + u, y_1 + v, t + 1) \qquad (1)$$

where the motion vectors for this case are denoted by u and v (horizontal and vertical, respectively). Furthermore, a construction of the lower-order Taylor series expansion of Eq. (1), concerning such location of the pixel on FN1 (x_1, y_1), can simplify Eq. (1) to become a general optical flow equation as follows:

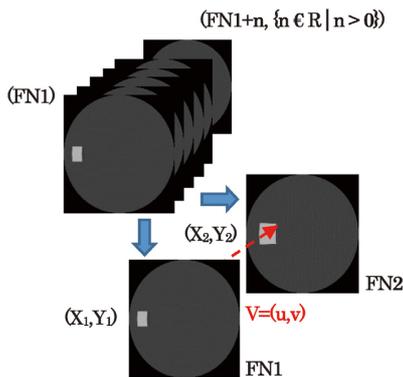$$u I_x + v I_y = -I_t \qquad (2)$$

Fig. 2 The optical flow calculation from two successive fisheye images.

where $I_x = \partial I / \partial x$ and $I_y = \partial I / \partial y$ are the first derivatives of the fisheye image regarding horizontal and vertical successively, while $\partial I / \partial t$ is the difference between the first derivative of I_2 and I_1. Nevertheless, since there are two unknowns in Eq. (2), this equation now becomes an optical flow problem (aperture problem) that have to be solved in a specific way.

4.1. Lucas and Kanade's Method

Lucas and Kanade (LK) propose a solution for Eq. (2) by incorporating $n \times n$ neighboring pixels[11)12)]. They assume that pixels surrounding a pixel being observed behave almost the same as the observed pixel. Therefore, Eq. (2) can be expressed as follows:

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{U} = - \underbrace{\begin{pmatrix} \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{B} \qquad (1)$$

From Eq. (3), the motion vectors (u and v) can be obtained. This equation is also well known as a solution for image registration technique.

4.2 Horn and Schunk's Method

Horn and Schunk (HS) introduce a solution for Eq. (2) by considering a global constraint of smoothness[13)14)]. Different from the LK's scheme that considers the local variation of a pixel, HS' scheme considers the global variation of an image. This condition means that motion vectors of a pixel always depends on the value of its neighbors. Therefore, the iterative calculation should be performed to update the value of the motion vectors. The iterative calculation of the motion vector for each pixel can be stated as follows:

$$u^{k+1} = u^{-k} - \frac{I_x(I_x u^{-k} + I_y v^{-k} + I_t)}{\alpha^2 + I_x^2 + I_y^2} \qquad (4)$$

and

$$v^{k+1} = v^{-k} - \frac{I_y(I_x u^{-k} + I_y v^{-k} + I_t)}{\alpha^2 + I_x^2 + I_y^2} \qquad (5)$$

The notations of u, v, I_x, and I_y are the same as the ones mentioned in the LK's method, except α is a parameter used for scaling the global smoothness period (typically around 1), and k is the iteration step ($k \in R / k > 0$).

4.3 Equidistance Projection Model

The equidistance projection (EP) model assumes that pixels on a fisheye image are composed by using an equidistance projection scheme[4)15)]. In this scheme, motion on the fisheye image can be defined as follows:

$$u = a(x - x_0)^2 + a(y - y_0)^2 + c, and$$
$$v = b(x - x_0)^2 + b(y - y_0)^2 + d \qquad (6)$$

Therefore, Eq. (2) can be expressed as follows:

$$
\underbrace{\begin{pmatrix} \sum n^2 I_x^2 k^2 & \sum n^2 I_x I_y k^2 & \sum n^2 I_x k & \sum n^2 I_x I_y k \\ \sum n^2 I_x I_y^2 k^2 & \sum n^2 I_y^2 k^2 & \sum n^2 I_x I_y k & \sum n^2 I_y^2 k \\ \sum n^2 I_x k & \sum n^2 I_x I_y k & \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y k & \sum n^2 I_x k & \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}_{E}
$$
$$
= -\underbrace{\begin{pmatrix} \sum n^2 I_x I_t k \\ \sum n^2 I_y I_t k \\ \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{F} \qquad (7)
$$

After calculating Eq. (7), E parameters (a, b, c, and d) can be divined. Therefore, the motion vectors denoted by Eq. (6) now can be obtained.

4.4 Lucas and Kanade's Method with Automatic Self-Improvement (Proposed Approach)

This approach applies original LK's concept as a fundamental concept of calculating optical flow. However, the calculation of the optical flow from two successive fisheye images is extended by incorporating an evaluation performance calculation (PSNR calculation). In image or video processing technology, PSNR (peak-to-peak signal-to-noise ratio) generally measures a level of likeness between two images. The PSNR can be calculated as follows[1]:

$$PSNR = 20 log_{10}\left(\frac{2^n}{MSE^{1/2}}\right) \qquad (8)$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[A(i,j) - \tilde{A}(i,j)\right]^2 \qquad (9)$$

where a fisheye image (A) and its replication ($\hat{A}$) have a size of $N \times M$ pixels. Each pixel location is indicated by pixel indices $1 \le i \le M$ and $1 \le j \le N$. The MSE is also the abbreviation for the mean square error.

The reconstructed fisheye image ($\hat{A}$) is developed from simple linear geometry transformation of (A) using the motion vectors of u and v[14)16)]. In other words, each pixel (x, y) in A is mapped to each correspondence pixel (x^*, y^*) in $\hat{A}$, and their relationship can be expressed as $x^* = ux$ and $y^* = vy$. Since the destination point during the transformation process could lie in between two pixels or there would be more than one vector go in the same

Fig. 3 The LKI Scheme.

direction, both x^* and y^* would be difficult to be defined. To overcome this condition simple bilinear interpolation can be applied at the end of fisheye image reconstruction process[16)17)].

In this paper, this proposed method will then be named Lucas and Kanade's concept with an improvement (LKI). Fig. 3 shows a flowchart for this approach. By setting the PSNR to zero for the first time (IPSNR = 0), the calculation begins with a way of finding motion vectors between two successive fisheye images (FN1 and FN2) by using Eq. (3). From then on, the motion vectors (u and v) and the first fisheye image (FN1) are used for developing a reconstructed fisheye image (RI) as if it is the same as the second fisheye image (FN2). The RI is then compared with the FN2 to get a performance value. The performance value is indicated as a new PSNR. Until this stage, one cycle calculation has been performed. If the new PSNR is higher than the previous one, there will be the next cycle with the previous FN1 is updated by the new RI. Otherwise, the calculation process will be stopped, and the motion vectors will be indicated as an aggregate of all motion vectors obtained in each cycle (u_{total} and v_{total}).

In the final stage, a final reconstructed fisheye image (FRI) can be generated by transforming the original FN1 with transformation vectors u_{total} and v_{total}. As a result, the final performance evaluation can be obtained from the final PSNR, FR1, and the final inter-different fisheye image (FIDI). The final IDI shows the difference between the FN2 and the final FRI.

As far as our observation, there are at least three image quality metrics used for comparing an image with its replication. These are the mean square error (MSE), the peak-signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM)[1]. The MSE calculates the average of the squares of the errors between an estimated image and the original one, while the PSNR enhances the MSE calculation by scaling it according to the dynamic range of pixel intensity (the bit depth of pixel). The SSIM measures a similarity index between two images by combining three significant factors of an image, such as loss of correlation, luminance distortion, and contrast distortion.

The SSIM works with the highest computational complexity that caused by utilizing those three elements, although it provides the best accuracy among those image quality metrics. On the other hand, the MSE runs fastest since it only considers the differences between the two images, but of the three image quality metrics, the MSE offers the lowest accuracy. Selecting one of those image quality metrics for a specific purpose can be conducted based on many different aspects. One of them is to consider a trade-off between the computational complexity and the accuracy. Therefore, in this research, the PSNR is selected since it gives better accuracy with modest complexity. Additionally, the PSNR calculation can perform very well and more efficient, even though it works repeatedly.

5. Results and Discussion

This section will discuss the experimental processes comprehensively, including experimental setup, additional comparison schemes, estimated motion vectors, qualitative evaluation, quantitative evaluation, and processing time.

Because of the page limitation, there are only a few pictures that can be shown in this paper for the qualitative evaluation. However, in the quantitative measurement, the performance of each motion estimation scheme for all sequential fisheye images can be presented entirely in this paper.

5.1 Experimental Setup

To be able to implement all motion estimation schemes for serving different types of sequential fisheye images, a set of program code related to appropriate simulation environment is developed by using Matlab R2016.

The motion is estimated from two successive fisheye images, and it is applied to all fisheye images within each sequential fisheye image listed in Table 1. The objective of the experiment is to pay attention to how far the proposed scheme (LKI) can increase the performance of motion vectors, in comparison with other schemes. Two indicators of performance can be judged from qualitative (visual) and quantitative (PSNR) measurement. While the PSNR denotes a value of peak to peak error between the second image (FN2) and the reconstructed image (RI), visual performance can be seen directly from the quality of the inter-different fisheye image (IDI) and the RI. The finest IDI, the best RI (the RI is similar to the FN2). The highest PSNR, the lowest error of motion vectors.
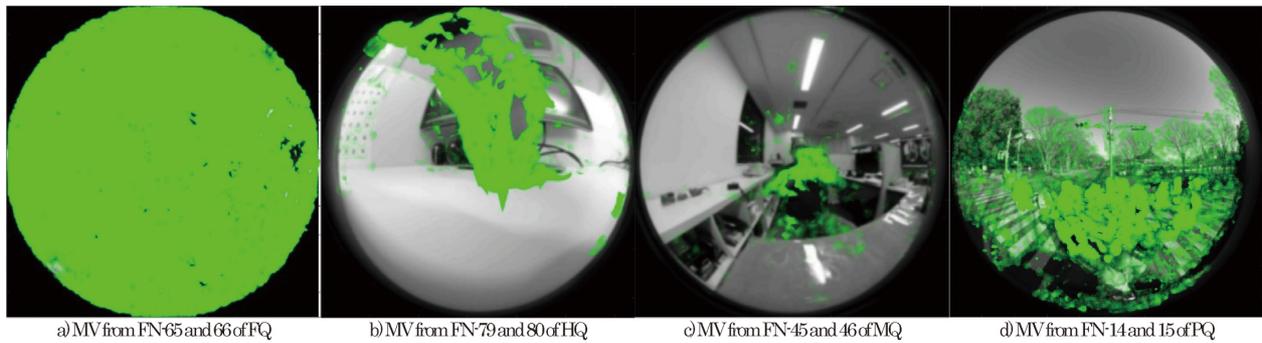
5.2 Additional Comparison Schemes

In this research, the performance of LKI scheme is benchmarked against the one obtained from each block-based motion estimation (BMA) schemes. These BMA schemes are exhaustive search (ES), three-step search (TSS), new three-step search (NTSS), simple and efficient TSS (SETSS), four-step search (FSS), diamond search (DS), and adaptive rood pattern search (ARPS). These algorithms are fundamental concepts of the BMA that are used for video coding standards, such as MPEG series or H.26x family.
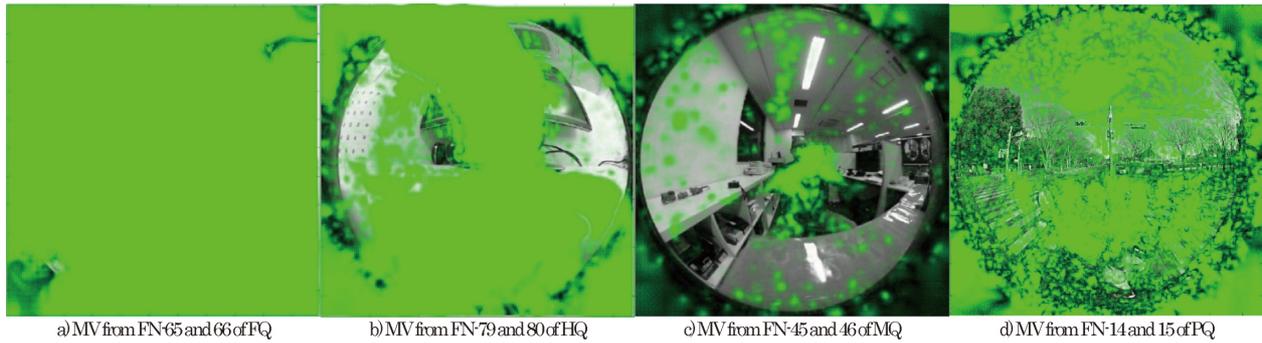
A complete report of these concepts can be found in the reference[18]. The size of the macro-block used in this experiment is set to 8×8 pixels.

5.3 The Estimated Vectors

Motion vectors (MV) or optical flow (OF) can be obtained directly from each sequential fisheye image listed in Table 1 by using every OF scheme mentioned in section 3. Some examples of the MV related to the pair of fisheye images input taken from the flower (FQ), man (MQ), hand (HQ), or people (PQ) can be seen in Fig. 4 and Fig. 5. Explicitly, the examples of the MV obtained from the LK's scheme are shown in Fig. 4, while the other examples of the MV produced by the LKI process are depicted in Fig. 5. In compare to the LK's scheme, the LKI process can produce MV more densely. The MV occurs wherever the motion of an object is located in, or if there is a motion caused by a change of illumination or

a) MV from FN-65 and 66 of FQ   b) MV from FN-79 and 80 of HQ   c) MV from FN-45 and 46 of MQ   d) MV from FN-14 and 15 of PQ

Fig. 4 The MV after LK's process.

a) MV from FN-65 and 66 of FQ   b) MV from FN-79 and 80 of HQ   c) MV from FN-45 and 46 of MQ   d) MV from FN-14 and 15 of PQ

Fig. 5 The MV results after LKI process.

noise in the fisheye image area.

5.4 The Qualitative Evaluation

Example of fisheye images taken from each sequential fisheye image listed in Table 1 can be seen in Fig. 6 (a - d), Fig. 7 (a - d), and Fig. 8 (a - d). Those figures are also the second fisheye image input (FN2).

Some examples of RI produced by the LK process can be seen in Fig. 6 (e - h), 7 (e - h), and 8 (e - h), while some other examples of IDI associating with its RI can be seen in Fig. 6 (m - p), 7 (m - p), and 8 (m - p). Those RI figures show that each RI is not the same as each associated FN2. This condition is also reflected from each associated IDI. The white area in those IDI results occurs significantly. This condition also means that the error of the MV is very high. Nevertheless, the results are much better when the LKI scheme is applied, as shown in Fig. 6 (i - l), Fig. 7 (i - l), and Fig. 8 (i - l) for the RI results, and Fig. 6 (q - t), Fig. 7 (q - t), and Fig. 8 (q - t) for the IDI results. Each RI is almost the same as each associated FN2. Moreover, those IDI results are almost totally grey. This condition means that the error of the MV is very low.

5.5 The Quantitative Evaluation

In the quantitative evaluation, the performance of the LKI is proven to be higher than the other optical flow methods or even the seven types of block-based scheme. The PSNR stays at a high level for each sequential fisheye image trained, as shown in Fig. 9 to Fig 20.

Regarding some sequential fisheye images that the motion is scattered equally in the fisheye area at the same time, such as the flowers, traffic I, traffic II and people, the associated PSNR for the LKI process tends to be constant in the higher level, by comparison with other schemes.

The PSNR for the flowers (FQ), traffic I (TRQ-I), traffic II (TRQ-II), and people (PQ) are about 40 dB, 40 dB, 38 dB, and 35 dB, respectively. Those PSNR results can be seen in Fig. 10, Fig. 18, Fig. 19, and Fig. 20, consecutively. Additionally, in which few sequential fisheye images consist of small camera motion, such as the TRQ-I, TRQ-II, and PQ, the PSNR obtained from the LKI scheme for each of them stays at the highest level. In other words, small camera motion can be detected by the LKI process very well.

Concerning the man sequence (MQ), the PSNR obtained from all schemes, except the LKI, seems to be fluctuated dramatically from about 35 dB to 55 dB. This condition happens since the object motion is dominated by two hands of a man that moves in opposite direction around the center area of the fisheye image. Nevertheless, the PSNR derived from the LKI scheme only varies slightly from about 45 dB to 55 dB. These results can be seen in Fig. 12. Moreover, this condition also means that a rapid motion around the center area

(a) FN-106 from CQ   (b) FN-66 from FQ   (c) FN-80 from HQ   (d) FN-46 from MQ

(e) RI for (a)   (f) RI for (b)   (g) RI for (c)   (h) RI for (d)

(i) RI for (a)   (j) RI for (b)   (k) RI for (c)   (l) RI for (d)

(m) IDI between (a) and (e)   (n) IDI between (b) and (f)   (o) IDI between (c) and (g)   (p) IDI between (d) and (h)

(q) IDI between (a) and (i)   (r) IDI between (b) and (j)   (s) IDI between (c) and (k)   (t) IDI between (d) and (l)

Fig. 6 The FN2 inputs (a - d), RI results after LK's process (e - h), RI results after LKI process (i - l), IDI results after LK's process (m - p), and IDI results after LKI process (q - t).

(a) FN-15 from TQ-I   (b) FN-21 from TQ-II   (c) FN-16 from CQ-I   (d) FN-21 from CQ-II

(e) RI for (a)   (f) RI for (b)   (g) RI for (c)   (h) RI for (d)

(i) RI for (a)   (j) RI for (b)   (k) RI for (c)   (l) RI for (d)

(m) IDI between (a) and (e)   (n) IDI between (b) and (f)   (o) IDI between (c) and (g)   (p) IDI between (d) and (h)

(q) IDI between (a) and (i)   (r) IDI between (b) and (j)   (s) IDI between (c) and (k)   (t) IDI between (d) and (l)

Fig. 7 The FN2 inputs (a - d), RI results after LK's process (e - h), RI results after LKI process (i - l), IDI results after LK's process (m - p), and IDI results after LKI process (q - t).
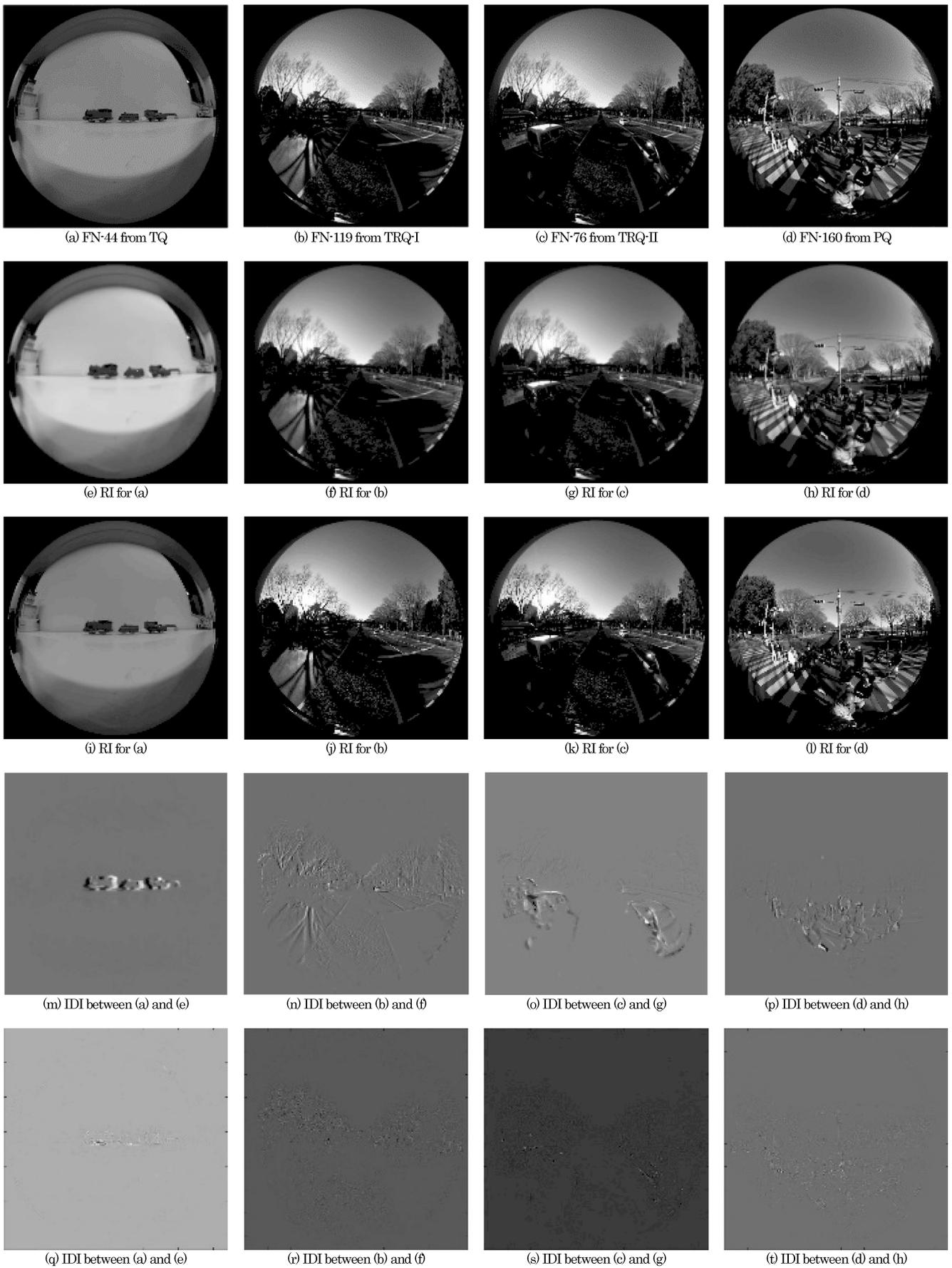
Fig. 8 The FN2 inputs (a - d), RI results after LK's process (e - h), RI results after LKI process (i - l), IDI results after LK's process (m - p), and IDI after LKI process (q - t).

of the fisheye image can be estimated very well by the LKI scheme.

Some sequential fisheye images that present single object with constant motion, such as the cube (CQ), hand (HQ), and train (TQ), present an exciting result, as shown in Fig. 9, Fig. 11, and Fig. 17, respectively. When the LK, HS, EP, and the other seven types of block-based scheme are applied, and the position of the object is located at the peripheral area of the fisheye image, the PSNR for each scheme is very high (at a maximum value). However, when the object is moving toward to the center area of the fisheye image, the PSNR for each scheme is going down gradually. It touches a minimum value when the position of the object is located in the center area of the fisheye image. The PSNR is then going up again when the object is leaving the center area of the fisheye image. It reaches the maximum value again when the position of the object is located at another peripheral area of the fisheye image. The gap between the maximum and minimum PSNR is about 30 dB, 24 dB, and 20 dB for the cube, hand, and train, consecutively. As a result, those schemes are not good enough to produce MV, mainly when the motion of the object mostly occurs around the center area of the fisheye image. Nevertheless, the condition is much better when the LKI scheme is applied. The PSNR for the LKI scheme obtained when the object is nearby the peripheral area of the fisheye image is as high as the PSNR for other schemes. From then on, the PSNR is decreasing when the object is moving to the center area of the fisheye image, but the decrease is slight. The PSNR then goes down to a minimum when the object is located at the center area of the fisheye image, but the PSNR is still much higher than the minimum PSNR obtained from the other schemes. The distance between the maximum and minimum PSNR is now only about 9 dB, 6 dB, and 3 dB for the cube, hand, and train, respectively.

Some other sequential fisheye images, such as the truck I (TQ-I), truck II (TQ-II), car I (CQ-I) and car II (CQ-II), show other intriguing results, as shown in Fig. 13, Fig. 14, Fig. 15, and Fig. 16, respectively. When the LK, HS, EP and the seven types of block-based scheme are employed, and the position of the object is in front of the fisheye lens, but far away from the fisheye lens, the PSNR for each scheme is very high (at a maximum value). However, when the position of the object is closing to the fisheye lens, the PSNR for each scheme is going down gradually. It touches a minimum value when

the object is located at the closest distance to the fisheye lens. The gap between the maximum and minimum PSNR obtained from those motion estimation schemes for the truck I, truck II, car I, and car II is around 26 dB, 30 dB, 35 dB, and 20 dB, respectively. However, the PSNR disparity obtained from the LKI scheme for the same sequence order can be reduced to about 6 dB, 12 dB, 12 dB, and 12 dB, respectively. Of the eleven motion estimation schemes, the LKI scheme generally achieves the highest score of PSNR for those sequences. Once again, the LKI process is beneficial to increase the performance of MV, especially to overcome the distortion problem around the center area of fisheye images.

The MV result obtained from the TRQ-I sequence almost remains constant at a high PSNR, although there is a slight fluctuation. However, in some frames, the PSNR attained by the LKI scheme is lower than that of the LK or EP method, as presented in Fig. 18. This condition occurs since, in this experiment, the LK, EP, or LKI scheme runs independently with a different parameter of calculation. Either the LK or EP scheme uses a constant window size (15×15 neighboring pixels) to obtain the MV using Eq. (3). This window size is generally acceptable for every sequential fisheye images used in this experiment, even though it does not give the best result for all fisheye images in every sequence, especially for handling a large motion. Nevertheless, in the case of some fisheye images in the TRQ-I and few fisheye images in the TRQ-II, the PSNR obtained from the LK or EP remains at a high level.

This condition happens because those fisheye images consist of simple object motions located outside the center area of the fisheye images. On the other hand, the LKI scheme in this experiment uses 10×10 neighboring pixels for every cycle of the PSNR calculation. This window size is very stable to find an initial MV. This situation means that the possibility of obtaining outlier vectors is very low. Therefore, the rest cycles will not propagate a lot of false vectors. However, the first PSNR calculation in the LKI scheme may be lower than that of the LK or EP method. Another advantage of using the LKI scheme is that the PSNR value for these kinds of sequence tends to be fluctuating in a tiny margin.

In the LKI scheme, the process of finding a maximum PSNR may resemble the mechanism of the hill climbing. The hill climbing behaves to look for a maximum state of a function or to obtain a local maximum of an objective function using an iterative calculation[1]. Furthermore, the LKI scheme seeks a PSNR maximum through a
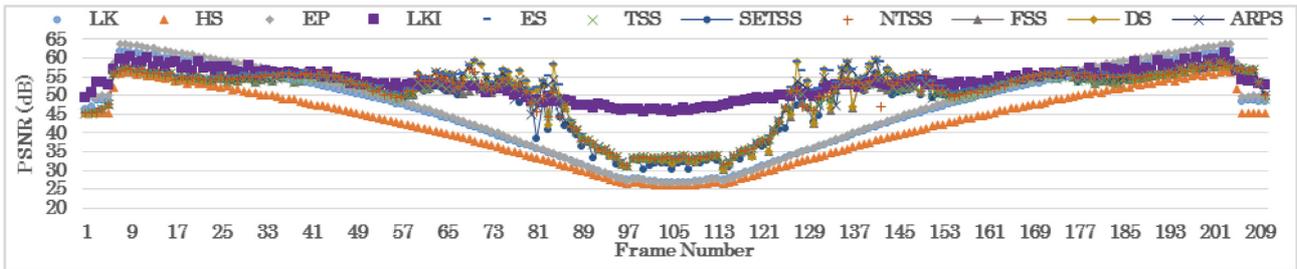
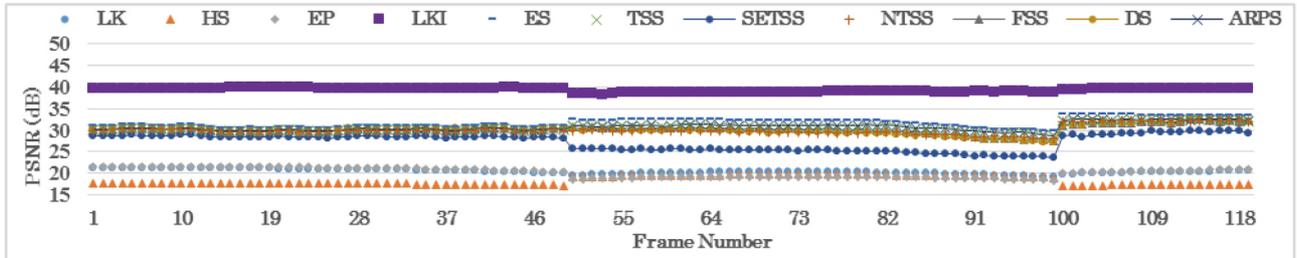**Fig. 9** PSNR versus frame number for the cube sequence (CQ).

Fig. 10 PSNR versus frame number for the flowers sequence (FQ).
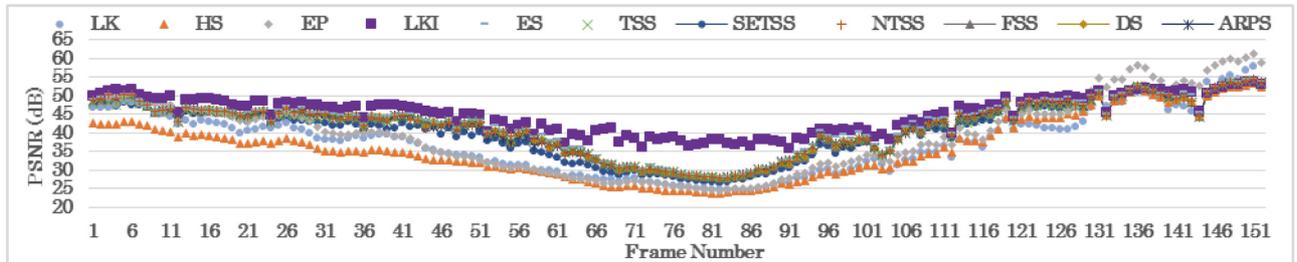

Fig. 11 PSNR versus frame number for the hand sequence (HQ).
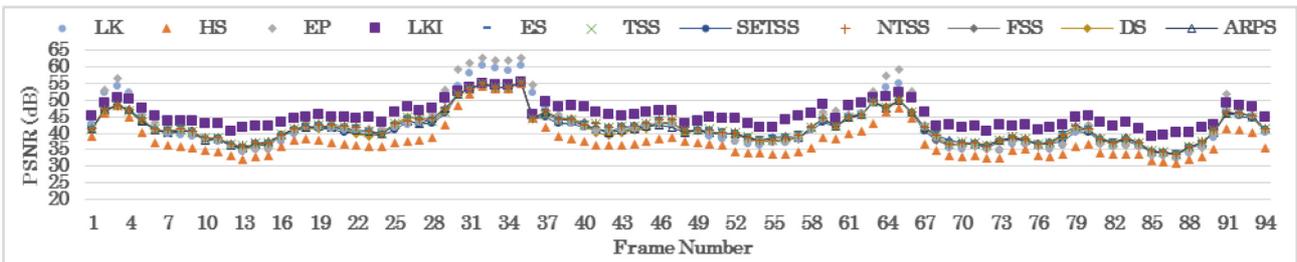

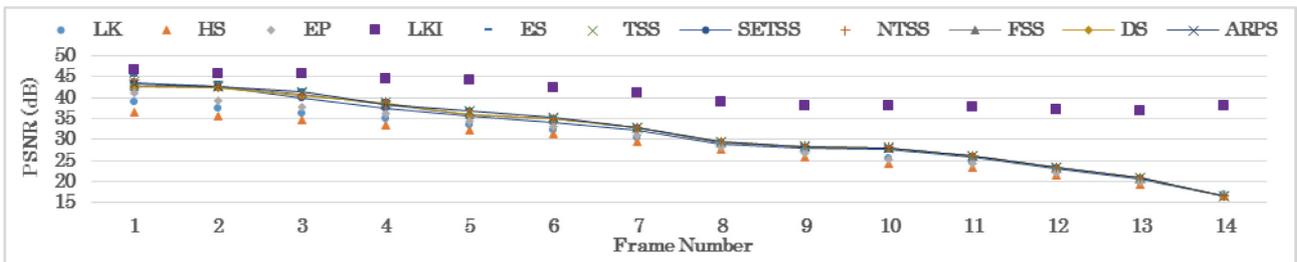**Fig. 12** PSNR versus frame number for the man sequence (MQ).

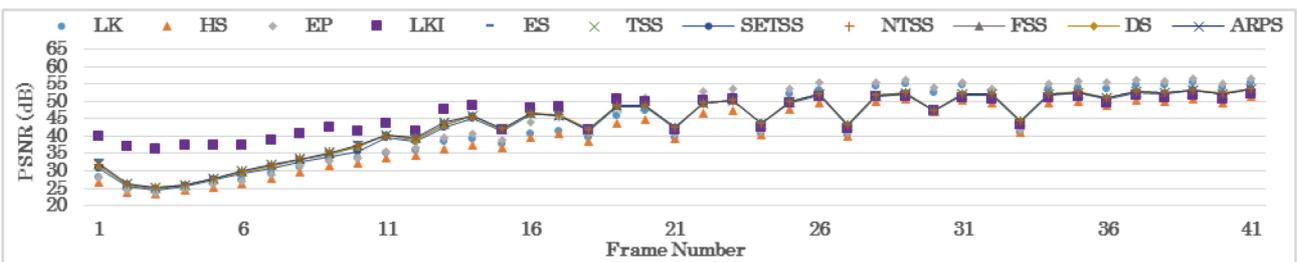**Fig. 13** PSNR versus frame number for the truck I sequence (TQ-I)

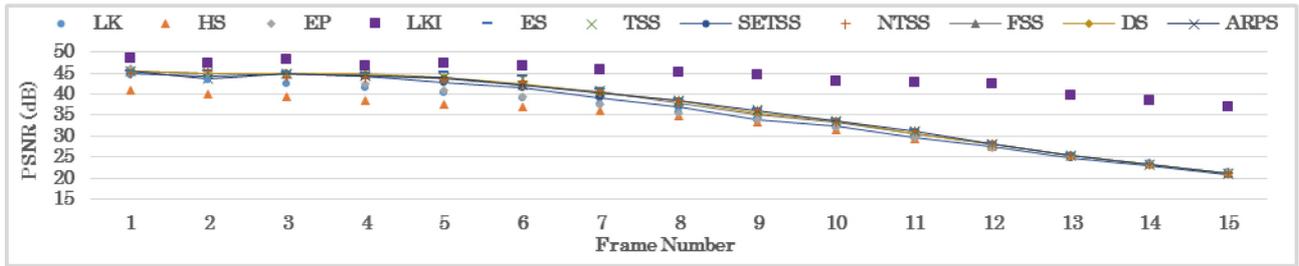**Fig. 14** PSNR versus frame number for the truck II sequence (TQ-II).
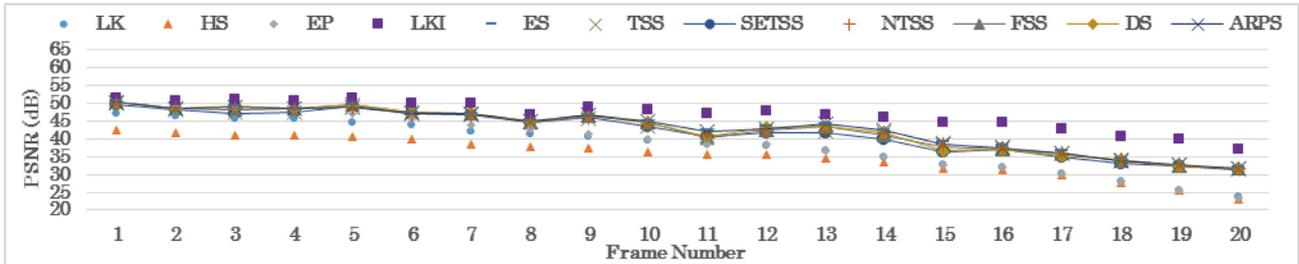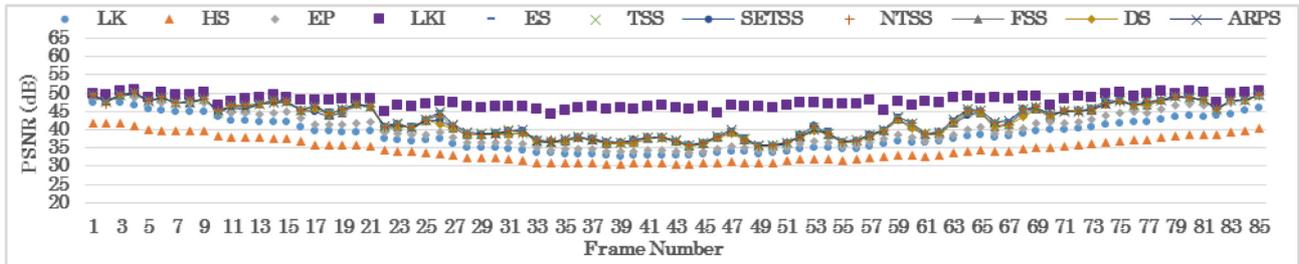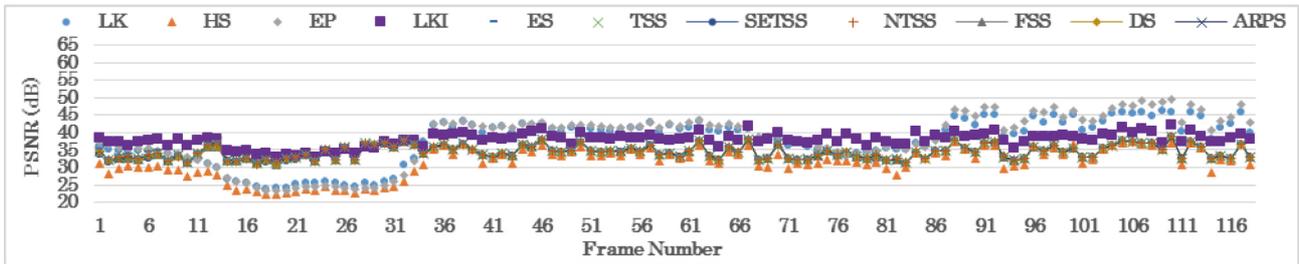
Fig. 15 PSNR versus frame number for the car I sequence (CQ-I)

Fig. 16 PSNR versus frame number for the car II sequence (CQ-II).

Fig. 17 PSNR versus frame number for the train sequence (TQ).

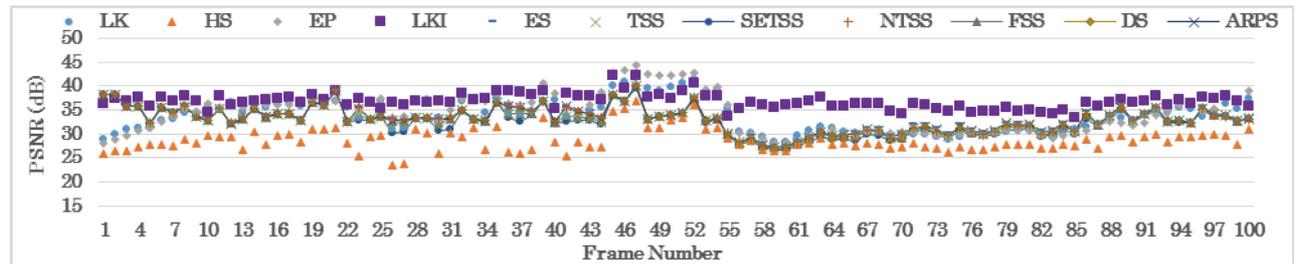**Fig. 18** PSNR versus frame number for the traffic I sequence (TRQ-I).

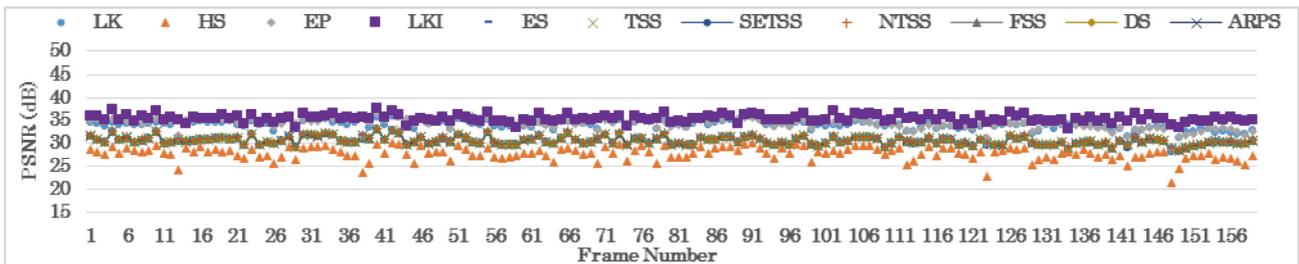**Fig. 19** PSNR versus frame number for the traffic II sequence (TRQ-II).

Fig. 20 PSNR versus frame number for the people sequence (PQ).

repetitive PSNR calculation. By setting the initial PSNR ($PSNR$ [0]) to zero, the succeeding PSNR calculations will perform to find a value that is larger than its neighbors ($PSNR$ [0] < $PSNR$ [i] > $PSNR$ [i+1], {$i \in R$ / i > 0}). The objective of finding the maximum PSNR in the LKI scheme, basically, is to obtain the best aggregate of motion vectors from all cycles of PSNR calculation. To be able to achieve that objective, there are two influential factors in each cycle of the PSNR calculation (i) that have to be concerned. The first is the quality of motion vectors produced by the optical flow calculation. The second is the quality of the reconstructed fisheye image built by transforming the origin of the fisheye image using variables denoted by the motion vectors. In practice, the process of finding a maximum PSNR when the motions of objects occur massively in the center area of sequential fisheye images is more difficult than that of appearance in the peripheral region. This condition happens since the motions of objects seem to be moving with higher speed, or the shape of objects tends to be deformed enormously in the center area of sequential fisheye images. The same situation even happens when the motions of objects occur too close to the camera, or the illumination changes dramatically. Moreover, from the quantitative perspective, the number of cycles of PSNR calculation also increases significantly due to those conditions.

The LKI scheme works well to estimate motion directly from sequential fisheye images because the recursive PSNR based calculation inside this scheme has a resilient effort to adapt to the characteristic of object motion inside the sequential fisheye images. For instance, the LKI scheme succeeds to obtain motion vectors not only from slower motions of objects located nearby the peripheral area but also from faster motions of objects discovered around the center area of the fisheye images. The LKI works with a small number of cycle of PSNR calculation when the movements of objects occur in the peripheral region, while it works with a higher number of cycle of PSNR calculation when the movements of objects appear around the center area of the fisheye images. For such condition, this method runs successfully to maintain the performance of the motion vectors at the higher level, in comparison with the seven types of block-based motion estimation. Those block-based motion estimations cannot keep the PSNR performance at the same level, especially when the movements of objects appear around the center area of the fisheye images. In a further experiment, the LKI method, to some extent, is also applicable for estimating motion from general perspective images. However, it has been tested with only a small number of consecutive general perspective images. From that test, the performance of PSNR or inter-different fisheye image is auspicious. The LKI mostly works with a constant number of cycle of PSNR calculation for this kind of image.

5.6 The Processing Time

The amount of time spent on obtaining MV from two successive fisheye images is defined as processing time. Table 2 presents some examples of the processing time needed by each motion estimation scheme to obtain MV from two successive fisheye images associated with each sequential fisheye image (SFI). Generally, it can be seen that the LKI can obtain MV under one second for all samples of the pair of fisheye images, except the pairs taken from the truck I, car I, car II, traffic II, and people.

The LKI is very fast enough to find the best MV, especially when it has to obtain MV from synthetic sequences, such as the cube or flowers. However, the processing time for the flowers is longer than that of the

Table 2 The processing time (in second).

SFI\MV Schemes	LK	HS	EP	LKI	ES	TSS	SETSS	NTSS	FSS	DS	ARPS
Cube (CQ)	0.05	22.7	3.3	0.22	2.2	0.3	0.18	0.32	0.29	0.41	0.21
Flowers (FQ)	0.15	63.45	5.65	0.64	5.63	0.77	0.47	0.82	0.72	0.81	0.37
Hand (HQ)	0.04	21.6	3.18	0.53	2.23	0.29	0.22	0.31	0.25	0.28	0.15
Man (MQ)	0.05	21.26	3.16	0.6	2.22	0.27	0.21	0.2	0.2	0.17	0.1
Truck I (TQ-I)	0.12	49.48	4.97	7.4	3.4	0.42	0.29	0.51	0.45	0.53	0.31
Truck II (TQ-II)	0.11	50.44	4.9	0.3	3.42	0.43	0.34	0.38	0.36	0.35	0.24
Car I (CQ-I)	0.11	49.48	4.99	5.3	3.45	0.43	0.31	0.45	0.41	0.44	0.27
Car II (CQ-II)	0.11	50.1	4.86	3.89	3.45	0.43	0.3	0.43	0.4	0.41	0.27
Train (TQ)	0.05	21.27	3.18	0.8	2.28	0.29	0.21	0.27	0.25	0.25	0.17
Traffic I (TRQ-I)	0.11	49.77	4.9	0.32	3.44	0.43	0.33	0.32	0.32	0.27	0.17
Traffic II (TRQ-II)	0.11	48.77	5.01	2.42	3.47	0.42	0.32	0.34	0.34	0.3	0.15
People (PQ)	0.11	46.07	4.98	3.2	3.42	0.42	0.32	0.33	0.33	0.32	0.17

cube. This because the number of object motions in the flowers is more than that of the cube. However, they have the same characteristics, such as constant illumination, constant speed, and minor motions. On the other hand, the processing time for other sequences that have the same number of object motions with the cube, such as the hand, man, and train, is longer than that of the cube. This because they are real sequences that consist of inconstant illumination.

Regarding the truck I and truck II, the processing time for the truck II is lower than one second, while the truck I is higher than one second. This condition happens because the samples from the truck II consists of a small object, whereas the samples from the truck I consists of a massive object.

Besides, although the samples of the pair of fisheye images taken from the car I and car II is almost the same, the time processing for the car I is longer than that of the car II. This condition occurs since there is a significant difference in the brightness level of the object (the car) between the two samples. On the other hand, even though the samples taken from the traffic I are almost the same as the traffic II, the processing time for the samples taken from traffic II is slower than that of the traffic I. This condition happens due to the difference of the number of objects.

The processing time for each sample taken from the truck I, car I, car II, traffic II, and people, is higher than one. This condition happens because those samples consist of complex characteristics, such as many object motion (the traffic II and people), different illumination, and different brightness level of the object motion.

6. Conclusion

In this paper, the design and analysis of the optical flow-based motion estimation with the automatic self-improvement scheme (LKI) for sequential fisheye images is shown. The final result proves that this proposed scheme can increase the performance of the MV. In the visual observation, the MV obtained from the LKI scheme can produce the finest RI that is almost the same as the fisheye image reference. That result also causes increases in the PSNR number. The PSNR obtained from the LKI scheme remain at a higher level in comparison with the ones obtained by using the original LK, HS, or ES. This condition occurs for every kind of sequential fisheye images trained. Furthermore, the PSNR results obtained from the LKI scheme are also better than the ones derived from the seven types of block-based motion estimations, including ES, TSS, SETSS, NTSS, FSS, DS, and ARPS.

Besides, to some extent, the LKI scheme also works very well to obtain MV from a sequential fisheye image with complex characteristics, such as inconsistent brightness, fluctuating number of object motion, changing shape of object motion, or poor camera stability. To be able to obtain MV from such kinds of fisheye images, the LKI process needs only below one second. However, when the characteristic of the fisheye images becomes more complex, the processing time increases slightly, but its performance remains at the high level.

References

1) A.C. Bovik: "The Essential Guide to Video Processing sourcebook", second edition, Elsevier Inc (2009)

2) D. Scaramuzza: "Omnidirectional Camera. Computer Vision: A Reference Guide", ISBN: 978-0-387-30771-8 (Print) 978-0-387-31439-6 (Online), Springer, 10.5167/uzh-106115 (Apr. 2014)

3) A.S. Satyawan, J. Hara, H. Watanabe: Motion Estimation on Fish-Eye Images Using Modified Motion Model", FIT2017, H-023 (Sep. 2017)

4) C. Zuo, Y. Liu, Y. Li: "A Distortion Correction Algorithm for Fish-Eye Panoramic Image of Master-Slave Visual Surveillance System", Optical Review, Volume 20, Issue 5, pp.367-373 (2013)

5) I. Bauermann, M. Mielke, E. Steinbach: "H.264 Based Coding of Omnidirectional Video", In: K. Wojciechowski, B. Smolka, H. Palus, R. Kozera, W. Skarbek, L. Noakes (eds) Computer Vision and Graphics. Computational Imaging and Vision, Springer, Dordrecht, vol 32 (2006)

6) Z. Jiali, Z. Yongdong, S. Yanfei and N. Guangnan: "Panoramic Video Coding Using Affine Motion Compensated Prediction", Springer-Verlag Berlin Heidelberg, pp.112-121 (2007)

7) A. Djamal, K. Djemaa, "An Adapted Block-Matching Method for Optical Flow Estimation in Catadioptric Images", International Conference on Multimedia Computing and Systems (ICMCS 2014), DOI: 10.1109/ICMCS.2014.6911139, IEEE, pp 69-74 (2014)

8) Ricoh Theta S: https://theta360.com/en/about/theta/s.html

9) N. Asuni and A. Giachetti: "TESTIMAGES: A Large-Scale Archive for Testing Visual Devices and Basic Image Processing Algorithms", in Proc. Smart Tools and Apps for Graphics Conference (STAG), pp.63-70 (available online at http://testimages.tecnick.com or www.lms.lnt.de/fisheyedataset) (Sep. 2014)

10) D.A. Forsyth, J. Ponce, "Computer Vision: A modern approach", second edition, Pearson (2011)

11) B.D. Lucas, T. Kanade: "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of Imaging Understanding Workshop, pp.121-130 (1981)

12) S. Baker and I. Matthews: "Lucas-Kanade 20 Years On: A Unifying Framework", International Journal of Computer Vision, Volume 56, Number 3, pp.221-255 (Feb. - Mar. 2004)

13) B.K. P. Horn and B.G. Schunck: "Determining optical flow", Artificial Intelligence, Volume 17, pp.185-203 (1981)

14) D. Fortun, P. Bouthemy, C. Kervrann: "Optical Flow Modeling and Computation: A Survey", Computer Vision and Image Understanding, Volume 134, pp.1-21 (May 2015)

15) A. Radgui, C. Demonceaux, E. Mouaddib, D. Aboutajdine and M. Rziza: "An Adapted Lucas-Kanade's Method for Optical Flow Estimation in Catadioptric Images", The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras-OMNIVIS, inria-00325390 (2008)

16) D. Salomon: "Transformation and Projections in Computer Graphics", Springer-Verlag London Limited (2006)
17) Y. Altunbasak, R.M. Mersereau and A.J. Patti: "A Fast Parametric Motion Estimation Algorithm With Illumination and Lens Distortion Correction", IEEE Transactions on Image Processing, Volume 12, Number 4, pp.395-408 (Apr. 2003)
18) J. Casey: "An Investigation of Block Searching Algorithms for Video Frame Codecs," Master's Dissertation, Dublin Institute of Technology, Republic of Ireland (2008)

Acknowledgments

Arief Suryadi Satyawan received his master degree (M.T.) in information system and telecommunication engineering from Bandung Institute of Technology, Indonesia, in 2007. He has joined Indonesian Institute of Science since 1996. He is currently a Ph.D. student at Waseda University, Japan. His current research is the omnidirectional vision.

Junichi Hara received his B.E. and M.E. degree in control engineering from Kyushu Institute of Technology (1988) and Osaka University (1990), respectively. He received his Ph.D. degree in information science from Osaka University in 2010. He has joined Ricoh Co., Ltd., since 1990. He is engaged in research and development of image processing, coding, communication, etc. He has also participated in standardization work such as JPEG 2000, JPEG XR, JPEG System, and JPEG 360.

Hiroshi Watanabe received his B.E., M.E. and Ph.D. degrees from Hokkaido University, Japan, in 1980, 1982, and 1985, respectively. He joined Nippon Telegraph and Telephone Corporation (NTT) in 1985, and was engaged in the research and development of image and video coding system at NTT Human Interface Labs., and NTT Cyber Space Labs., until 2000. He has been involved with the development of JPEG, MPEG standards under JTC 1/SC 29. He is currently a professor at the Department of Communications and Computer Engineering, School of Fundamental Science and Engineering, Waseda University. His research interests include image and video processing, video coding, and multimedia distribution. He is a member of a steering committee of PCSJ. He is also a member of the IEEE, IEICE, IPSJ, ITE, and IIEEJ.