Motion Estimation for Sequential Fisheye Images by Extending Optical Flow Concept

Waseda University

Department of Computer Science and Communications Engineering

Research on Audiovisual Information Processing

Arief Suryadi Satyawan

February, 2019

Abstract

The emergence of many kinds of omnidirectional cameras recently has given a broader opportunity for enhancing many types of application with a wide-field-of-view capability. However, making use of images obtained from an omnidirectional camera is not as easy as that of a general perspective camera due to a distortion characteristic attached to those images. Therefore, overcoming the distortion while utilizing omnidirectional images is still a big challenge. In this research, we focus on estimating motion directly from sequential fisheye images. For about several years, the motion estimation (ME) technique has been developed and mainly implemented for many applications. These applications can be found in the object tracking, 3-D structure estimation, motion analysis, video frame-rate conversion, video compression, and video surveillance system.

There are two ME developments conducted in this research. The first is the 2-D ME, and the second is the 3-D ME. The 2-D ME estimates motion directly from sequential fisheye images produced by a fisheye camera. The 3-D ME forecasts the 3-D scene flow by utilizing both of sequential fisheye images and a sequential stereo fisheye image produced by a stereo fisheye camera. Both the 2-D ME and 3-D ME are design based on the optical flow concept, and the conventional Lucas and Kanade's (LK) approximation is developed in each ME concept.

In the 2-D ME development, we have succeeded to make use of the LK's optical flow scheme as a fundamental methodology of estimating motion directly from sequential fisheye images. In the fisheye image circumstance, it seems like this method works very efficient. However, the conventional LK's approach seems to be difficult to overcome a massive distortion in the center region of the fisheye image. Therefore, we propose an extension of the conventional LK's concept. The extended version of the conventional LK's approach is named "LKI", which is Lucas and Kanade's concept with an improvement. The LKI scheme exploits the conventional LK's method to work many times under the supervision of a parametric performance evaluation. As a result, the performance of the estimated motion vectors throughout the fisheye image area remains the same at a high level.

In the meantime, rational utilization of the fundamental LK's concept for supporting the 3-D scene flow estimation has been conducted successfully in another development.

The conventional LK's approach has been proven to be effective for calculating not only the optical flow but also the disparity between stereo fisheye images. Hence, our 3-D scene flow estimation framework that employs the optical flow and disparity calculations can produce a promising result. The proposed scheme can generate the 3-D scene flow directly from a stereo fisheye camera. This condition is applicable wherever the moving object occurs in the fisheye image area. In other words, our approach can reveal the 3-D motion captured from 180^0 FoV.

Declaration

This dissertation is composed of two main projects. The first is the two-dimensional optical flow based motion estimation for sequential fisheye images. The second is the three-dimensional scene flow estimation for sequential stereo fisheye images. The content mainly adopted author’s published journal paper and proceedings of conferences.

Acknowledgments

Firstly, I would like to greatly thank Professor Hiroshi WATANABE at Waseda University for overall supporting during my research and my doctoral study. He has been provided valuable ideas, and these are not only throughout this dissertation but also through lifelong education. He had a lot of influence on the ideas presented in this dissertation. He has been very supportive and helped a lot, mainly, at the level quality of presentation, and he never let me wait too long for any request.

I also would like to willing thank my sub-advisor Professor Wataru KAMEYAMA, at Waseda University for his insightful comments, constructive criticism at different stages on my research.

Next, I would like to greatly thank Professor Jiro KATTO, at Waseda University for his remarkable questions and comments on my dissertation.

I would like to thank the boards of the Riset-Pro scholarship program initiated by the government of Republic Indonesia, for giving an opportunity to attend this course leading to the doctoral degree course in Waseda University, Tokyo, and giving provision to finish my research.

Most importantly, none of this would have been possible without the love and patience of my family. I would like to express my heartfelt gratitude to my wife, son, parents, and sisters for providing me valuable support and encouragement throughout my doctoral study.

I extend many thanks to all members of Hiroshi Watanabe Laboratory at Waseda University, my colleagues and friends for their support, help, interest, and valuable hints for discussion about research to fulfill all my need in all time of my doctoral study.

Contents

List of Abbreviations

2-D	Two-dimensional
3-D	Three-dimensional
A	
AAE	Average angular error
AVC	Advanced video coding
ARPS	Adaptive rood pattern search
B	
BMA	Block matching algorithm
C	
CQ	Cube sequential fisheye images
CQ-I	Car I sequential fisheye images
CQ-II	Car II sequential fisheye images
D	
dB	Decibel
DCT	Discrete cosine transform
DS	Diamond search
E	
EP	Equidistance projection
ES	Exhaustive search
F	
FN	Frame number
FoV	Field-of-View
FQ	Flowers sequential fisheye images
FRI	Final reconstructed image
FSS	Four-step search
H	
HS	Horn and Schunck

HQ	Hand sequential fisheye images
I	
IDI	Inter-different fisheye image
L	
LC	Left camera
LK	Lucas and Kanade
LKI	Lucas and Kanade’s concept with an improvement
M	
ME	Motion Estimation
MPEG	The moving picture experts group
MQ	Man sequential fisheye images
MSE	Mean square of error
MV	Motion vectors
N	
NTSS	New three-step search
O	
OF	Optical flow
P	
PQ	People sequential fisheye images
PSNR	Peak-to-peak signal-to-noise ratio
R	
RC	Right camera
RI	Reconstructed fisheye image
S	
SAE	Standard deviation average angular error
SETSS	Simple and efficient TSS
SSIM	Structural similarity index measure
T	
TQ	Train sequential fisheye images

TQ-I	Truck I sequential fisheye images
TQ-II	Truck II sequential fisheye images
TRQ-I	Traffic I sequential fisheye images
TRQ-II	Traffic II sequential fisheye images
TSS	Three-step search

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

For about several years, the motion estimation (ME) technique has been developed and mainly implemented for a lot of applications. These applications can be found in the object tracking, 3-D structure estimation, motion analysis, video frame-rate conversion, video compression, and video surveillance system [1].

In the two-dimensional (2-D) domain, the ME scheme works by forecasting the apparent movement of an object in time-varying images that obtained from the projection process of the real-world one. This motion occurs because of exact object motion, an illumination change, camera motion, and noise. The objective of the 2-D ME technique is then to determine 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. There are so many kinds of the 2-D ME schemes that have been developed until now. These include the optical flow-based ME, block-based ME, pel-recursive-based ME, and transform domain ME [1] [2].

Besides, estimating the three dimensional (3-D) scene flow directly from multi-view general perspective images has successfully developed in recent years [3]. The optical flow that is interpreted as the 2-D motion field of objects or pixels on an image plane. Similarly, the scene flow is illustrated as the 3-D motion field of objects or points in the real-world. This term also means that an uncomplicated projection of the scene flow onto an image plane is the optical flow. To be able to estimate the 3-D scene flow, on our approach uses a combination of calculation of multi-view of the 3-D projection onto an image plane with information about geometrical of 3-D surface or depth. Forecasting the 3-D scene flow is very important. It can be used for reconstructing the 3-D object, analyzing 3-D motion, and developing 3-D videos.

Although both the 2-D and 3-D ME has been successfully developed, the application of these tools seems to be useful even for narrow-angle-view purposes. This condition is acceptable since those tools are designed to support images projected by conventional narrow-angle-view cameras as well. To be able to extend the capability of view, and at the same time, to support the 3-D applications, multi-narrow-angle cameras, therefore, field-

of-view (FoV) capability can be extended to about 180^0 (hemisphere FoV) or even 360^0 (omnidirectional FoV) [4]. In the meantime, wide-angle cameras have been emerging massively on the market. One of them is the fisheye camera that can capture scene phenomenon up to two times hemisphere FoV. However, applying the above ME tools directly to serve sequential images captured by the fisheye camera is not easy. This is because the sequential fisheye images suffer from massive distortion. Therefore, development of both the 2-D and 3-D ME to serve sequential fisheye images is still challenging.

1.2 Objective

The objective of this dissertation is to describe comprehensively the development of both the 2-D and 3-D ME directly from fisheye images. Both of them are developed mainly for increasing the capability of estimating a motion of object located in the hemisphere FoV (in front of the fisheye camera). The advantages of these developments are not only to increase the capability of view, but also to simplify the number of cameras used for capturing such large view. Instead of developing a method that firstly invests a hard effort to refine the fisheye images before implementing a conventional ME method to estimate the captured motion [5] [6] [7], this research aims to estimate motion directly from fisheye images domain. Therefore, the design of ME should adapt very well to overcome a massive distortion attached to such images.

There are two main developments conducted in this research. The first is the 2-D ME, and the second is the 3-D ME. The 2-D ME estimates motion directly from sequential fisheye images produced by a fisheye camera. The 3-D ME forecasts the 3-D scene flow by utilizing both of sequential fisheye images and a sequential stereo fisheye image produced by a stereo fisheye camera. Each of the developments will be discussed comprehensively in a chapter inside this dissertation.

1.3 Research Contribution

In this research, there are three main contributions. These contributions can be explained in the following lists.

1. This research introduces a solution for estimating both 2-D and 3-D motion directly from sequential fisheye images.

2. The performance of the 2-D motion vectors (MV) resulted from a conventional LK's calculation can be improved significantly by executing the calculation repetitively. The multiple computations work to find the best performance of the MV by minimizing an error metric in the fisheye image pixel domain. Specifically, this error matric minimizes the difference between a fisheye image reference determined as the destination points of the MV and its replication constructed from the estimated MV.
3. This research also reveals that the 3-D motion (scene flow) can be generated from sequential fisheye images taken from two fisheye cameras set up as a stereo configuration. In this case, those fisheye images from the configuration are made synthetically. By combining the optical flow obtained from every two consecutive fisheye images and the disparity between stereo fisheye images, the 3-D scene flow can be generated from a calculation framework. Once again, in this research, both optical flow and disparity calculation are conducted by using the conventional LK's scheme.

1.4 Dissertation Organization

This dissertation consists of four main chapters. Each of them will be described as follows.

The first chapter briefly introduces the background of this research, the objective of this research, and finally the organization of this dissertation. Following this chapter, there are two consecutive chapters that describe the two ME developments in this research.

The second chapter describes the design of the 2-D ME. In this research, we use the optical flow-based ME to estimate the 2-D motion directly from sequential fisheye images. The design firstly introduces the conventional Lucas and Kanade (LK), Horn and Schunck (HS), and equidistance projection method to obtain the motion vectors from every two successive fisheye images. A promising result is shown by the conventional LK's approach. This approach does not only work efficiently but also can adapt very well with the kinds of motion on the fisheye image. However, in the center region of the fisheye image, it cannot overcome the massive distortion. Therefore, we proposed the extension of the conventional LK's approach. The extended version of the conventional LK's approach is "LKI", which

is named Lucas and Kanade's concept with an improvement, and it is described clearly in this chapter.

The third chapter, on the other hand, presents the design of the 3-D ME, and it is also known as the 3-D scene flow. In this research, we implement a binocular fisheye camera system to produce a sequential stereo fisheye image. Furthermore, we combine the optical flow and the disparity calculation to support the 3-D scene flow calculation framework. The design is presented comprehensively in this chapter.

Finally, in chapter four, a summary of our works and further questions are inventoried clearly in this chapter.

Chapter 2

The 2-D Optical Flow-Based Motion Estimation for Sequential Fisheye Images

2.1 Introduction

2.1.1 Background and Motivation

As it has been mentioned in chapter 1, the two-dimensional (2-D) motion estimation (ME) technique has a great value to reveal the real-world motion from a video sequence. Specifically, the 2-D ME estimates the existence of motion in time-varying images that obtained from the projection process of the real-world one inside a camera. This motion occurs due to true object motion, an illumination change, camera motion, and noise. The objective of the 2-D ME 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 method involves the image being investigated and at least one of its neighbors. Finally, the motion vectors can be used for further applications, such as video analysis, motion tracking, frame-rate conversion, video compression, and video surveillance system [1].

Many 2-D ME techniques have been developed well mainly to serve a video sequence consisting of general perspective frames (images). These images have an insignificant geometric distortion, and it is available in a narrow-angle perspective fashion. On the other hand, recently there have been many kinds of omnidirectional cameras in the markets [4]. These cameras capture larger field-of-view (FoV) than the narrow-angle perspective cameras do, and one of the omnidirectional cameras is the fisheye camera. This camera primarily provides images that cover almost 180^0 FoV from an inexpensive single camera. Nevertheless, estimating motion directly from larger FoV images is still challenging to do since the fisheye images undergo massive geometric distortion. Therefore, this research focuses on the development of an appropriate ME tool that deals with sequential fisheye images.

2.1.2 Problem Formulation

There are three main problems in this research. These problems are the fisheye image's distortion, the appropriate ME tools for the fisheye images, and the suitable image

quality metric used to measure the performance of motion vectors or for supporting the ME tool. Each of them will be described briefly in the following sections.

2.1.2.1 The Fisheye Image's Distortion

Although fisheye images bring many advantages due to that they provide more extensive visual information; these images suffer from distortion. In the maximum FoV, 180^0, there are at least two types of distortion adhering to the fisheye images [4] [8]. The first is tangential distortion, and the second is radial distortion. The tangential distortion is insignificant, as well as invisible. Therefore, it can be ignored. However, the radial distortion is visible tremendously. Hence, to make use of the fisheye images for further needs, the radial distortion has to be managed appropriately. In Figure 2.1 and 2.2, the effect of the radial distortion upon the fisheye images can be seen clearly. While Figure 2.1 shows three samples of syntactical fisheye images [9], Figure 2.2 presents three examples of real fisheye images obtained from a Ricoh Theta S camera [10].

a) Frame 5th b) Frame 35th c) Frame 78th

Figure 2.1. Three samples of synthetic fisheye image from the flowers sequence [9].

a) Frame 19th Frame 70th Frame 90th

Figure 2.2. Three samples of real fisheye image from the hand sequence.

The radial distortion tended to change the shape of objects when these objects were captured in the different location of the fisheye image area.

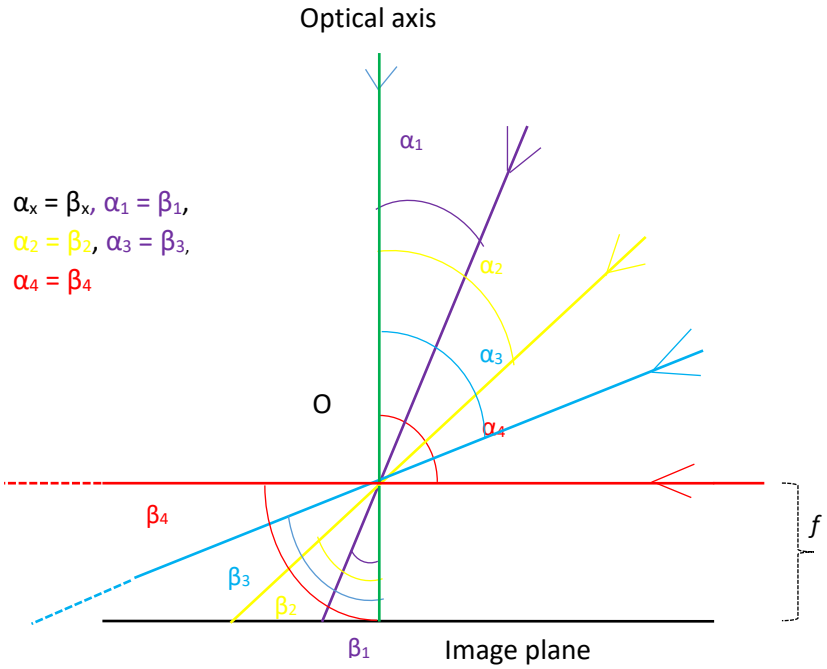
The distortion attached to a fisheye image is influenced by the uncommon projection characteristic of the fisheye camera itself. A fisheye lens is designed to capture a hemispherical field in front of the fisheye camera. The angle of the view also can reach 180^0. Therefore, unlike a general image (a narrow-angle perspective image), a fisheye image is not assumed to obey the conventional pinhole projection model, but it follows the fisheye projection families. The reason for this assumption is caused by the fact that the projected view will need an infinite image plane to cover that huge-of-view [10] [11]. In Figure 2.3, the difference between the basic geometric concept of the pinhole projection model and the fisheye projection model is presented clearly. It can be seen in Figure 2.3.a that the central perspective (pinhole) projection model assumes that the angle (α_x) between a ray coming from an object point and the optical axis is the same as the angle (β_x) between the ray after passing center projection (O) and the optical axis. The relationship between the two angles can be expressed as equation (2.1).

$$\alpha_x = \beta_x \quad \{x \in R | x > 0\} \qquad (2.1).$$

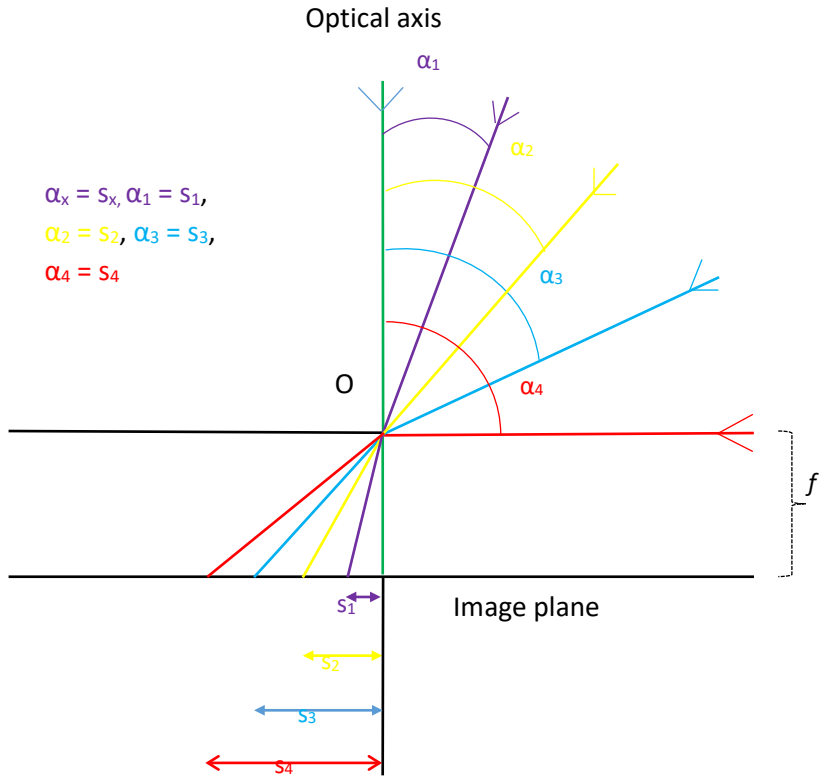
Furthermore, for 180^0 FoV, when the object point's ray is perpendicular to the optical axis, this ray will be projected onto the image plane at an infinite distance. This drawback can be avoided by using the fisheye projection model, as shown in Figure 2.3.b. The fisheye projection model, basically assumes that in an ideal situation, the angle (α_x) between an incident ray coming from the corresponding object point and the optical axis has a linear relationship with the distance (s_x) between that ray after reaching the image point and the optical axis. This relation can be expressed as follows:

$$\alpha_x = s_x \quad \{x \in R | x > 0\} \qquad (2.2).$$

This projection model tends to be refracted the incoming ray in line with the optical axis. In the maximum FoV, 180^0, an object point's ray will be projected at the periphery area of the image space (the fisheye image area). Hence, the hemispherical field-of-view can be projected in a finite image plane. Furthermore, to be able to project the hemispherical FoV in a finite image plane, one of the fisheye projection models in table 2.1 can be selected [12]. Of the four models, the equidistance projection is the most suitable model



(a) The perspective projection model.



(b) The fisheye image projection model.

Figure 2.3. The difference between the perspective projection and the fisheye projection model.

for the fisheye image. Despite the fact that the fisheye projection model projects the hemispherical FoV into the image space, the distribution of the projected view is not uniform due to the properties of the lens. The projection of the incoming rays occurs densely in the center area of the image, whereas it is indicated sparsely in the peripheral region of the image [11] [13] [14]. This phenomenon is then assumed as the radial distortion.

Table 2.1. The fisheye projection models.

No	Name of the fisheye projection	Formulation
1	Stereographic	$s = 2f \tan(\alpha/2)$
2	Equidistance	$s = f\alpha$
3	Equisolid	$s = 2f \sin(\alpha/2)$
4	Orthogonal	$s = f \sin(\alpha)$
s = the distance between image point and principal axis, *f = focal length, and* *α = the angle between principal axis and object point's incoming ray*		

Beside the fisheye projection model, the radial distortion can be assumed to meet polynomial fisheye transform [13] [15]. There are two types of polynomial fisheye transform. The first is the odd polynomial model. This model can be expressed as follows:

$$r_d = r_u + \sum_{n=1}^{\infty} k_n r_u^n \quad \{n \in R, n > 1\} \tag{2.3}$$

where r_d is the distorted radius, r_u is the undistorted radius, and k is the polynomial coefficient. The second is the division polynomial model. This model can be stated as follows:

$$r_d = r_u \left(1 + \sum_{n=1}^{\infty} k_n r_u^{2n} \right) \quad \{n \in R, n > 1\} \tag{2.4}$$

where the symbols are the same as the ones for equation 2.3.

To some extent, the above radial distortion concepts might be useful for practical solutions. However, the distortion in the fisheye images also consists of an illumination change and other noise caused by the construction of the fisheye lens.

2.1.2.2 The Motion Estimation Tool

To the degree our perception, there are several kinds of the 2-D-ME concept that has been developed recently, and all of them seem to come from a thought that motion is a displacement vector that wraps a reference image into the following destination image [1]. A constraint equation for this though can be expressed as follows:

$$A(x, y, t) = A(x - d_1, y - d_2, t - 1) \quad (2.5)$$

where $\boldsymbol{d} = (d_1, d_2)$ denotes the apparent displacement vector field, *A* is marked as an image, and *t* and *t-1* describes the arrangement of the images position in time.

The optical flow-based ME calculates the apparent motion vectors directly from two successive images. The calculation works to solve the brightness constancy suspicion between pixels in an image and each counterpart pixel in the following or previous image (*I(x,y,t)* and *I(x,y,t-1)* or *I(x,y,t)* and *I(x,y,t+1)*). The brightness constancy suspicion seems to be similar to the equation (2.5). By using partial differentiation, which is the same as the least-squares operation, the equation (2.5) can be stated also as the spatiotemporal constraint equation as follows (noted that the *A* now become *I*):

$$v_x \frac{\partial I}{\partial x} + v_y \frac{\partial I}{\partial y} + \frac{\partial I}{\partial t} = 0 \quad (2.6)$$

where v_x and v_y are motion vectors in respect to horizontal and vertical respectively and ∂ is the partial derivative operator. Moreover, by solving the equation (2.6), the motion vectors can be obtained. Since there will be two unknowns in the equation (2.6), the calculation has to avoid for only one pixel. At least a small region of pixels called the aperture has to be involved to solve that equation. The optical flow-based ME is very well known in computer vision applications.

The block-based ME 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 [16]. Therefore, a motion vector can be obtained for each block.

Different with the optical flow-based ME, the block-based ME use an error function such as the mean-square error (MSE) or the mean-absolute error (MAE) to judge the similarity between a block of pixels from the target frame and each block of pixels in a search area in the reference frame. This condition can be expressed as follows:

$$\varepsilon(d) \triangleq \sum_k (A(\boldsymbol{n} + k, t) - A(\boldsymbol{n} + k - d, t - 1))^2 \quad (2.7)$$

where $\boldsymbol{n} = (x,y)^T$ center position of a square block, $\boldsymbol{d} = (d_1,d_2)^T$ denotes a vector displacement. The rest of work is to seek the displacement vectors that minimizes this error as follows:

$$d_o \triangleq \min_d \varepsilon(d) \qquad (2.8)$$

The block-based motion estimation is very valuable in the video compression application.

Besides the two major motion estimation concepts, other methods have also been developed in the same years. Among these, there is the global ME 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. The global ME considers motion field (***d***) in the equation (2.5) to consist of affine motion model as follows [17]:

$$d_1(x,y) = (a_1 + a_2x + a_3y), d_2(x,y) = (a_4 + a_5x + a_6y) \qquad (2.9),$$

or quadratic motion model as follows:

$$d_1(x,y) = q_1 + q_2x + q_3y + q_7x^2 + q_8xy,$$
$$d_2(x,y) = q_4 + q_5x + q_6y + q_7xy + q_8y^2 \qquad (2.10).$$

Therefore, the least-square minimization can be applied to the equation (2.5) the same as the optical flow-based ME do earlier. This motion estimation is necessary for the sports video content analysis, including zoom in and out applications.

Another method is the pel-recursive ME. This method calculates a motion vector recursively for each pixel in a frame. The calculation is conducted to minimize the displaced frame difference (DFD). The DFD is basically similar to the equation (2.5), and it is defined as follows (noted that the *A* now become *f*):

$$DFD = f(n,m,k) - f(n - \Delta x, m - \Delta y, k - 1) \qquad (2.11),$$

where *f* denotes as a frame, *k* determines a temporal location, (*n*,*m*) is a spatial location, Δx and Δy are displacements. From then on, by using the steepest descent approach, minimizing the square of the DFD can produce an iterative calculation of the motion vector $D_{n,m}^{(i)}$. By considering the current pixel position (*n*,*m*) and the motion vector $D_{n,m}^{(i)}$ at iteration *i*, the recursion can be defined as follows:

$$D_{n,m}^{(i+1)} = D_{n,m}^{(i)} - \frac{\varepsilon}{2}\nabla_D DFD^2(n,m,\Delta x,\Delta y) \qquad (2.12),$$

where ∇_D is the gradient with respect to D, and $\varepsilon > 0$ denotes a constant gain.

Besides the above method, there is the motion estimation in the transform domain. This method exploits correspondence between translations in the spatial domain and a change of phase of the Fourier coefficients. Hence, the displacement is estimated from a phase correlation between blocks of pixels. In one case, the transform domain approach employs a discrete transform rather than the Fourier transform. That is applicable because the discrete cosine transform (DCT) has been previously applicable in the video coding application.

2.1.2.3 The Image Quality Metrics

As far as our observation, there are at least three image quality metrics used for comparing an image target with an image reference. These are the mean square error (MSE), the peak-signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) [1] [18].

The MSE calculates the average of the squares of the errors between the image target and the reference one, and it can be calculated as follows:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[A(i,j) - \hat{A}(i,j)\right]^2 \qquad (2.13),$$

where the image target (A) and the image reference ($\hat{A}$) have a size of N x M pixels. Each pixel location is indicated by pixel indices $1 \leq i \leq$ M and $1 \leq j \leq$ N. Both images should have the same size.

PSNR enhances the MSE calculation by scaling it according to the dynamic range of pixel intensity (the bit depth of pixel denoted by 2^n). Hence, the PSNR can be calculated as follows:

$$PSNR = 20log_{10}\left(\frac{2^n}{MSE^{1/2}}\right) \qquad (2.14).$$

The SSIM measures a similarity index between two images (f and g) by combining three significant factors of an image, such as loss of correlation $s_{(f,g)}$, luminance distortion ($l_{(f,g)}$), and contrast distortion ($c_{(f,g)}$). The SSIM's formulation can be expressed as follows [39]:

$$SSIM(f,g) = \left(l(f,g).c(f,g).s(f,g)\right) \qquad (2.15),$$

where

$$l(f,g) = \frac{(2\mu_f\mu_g + C_1)}{(\mu_f^2 + \mu_g^2 + C_1)}, c(f,g) = \frac{(2\sigma_{fg} + C_3)}{(\sigma_f\sigma_g + C_3)}, and$$

$$s(f,g) = \frac{(2\sigma_f\sigma_g + C_2)}{(\sigma_f^2 + \sigma_g^2 + C_2)}.$$

The notations of μ, σ, and C are a mean, a standard deviation, and a non-zero constant, respectively.

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 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.

2.1.2.4 Contribution and Constraint

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 analysis, video frame-rate conversion, slow-motion scene, scene flow, and fisheye compression, shifting the correction process to the final processing is a valuable strategy.

Moreover, the basic concept of motion estimation implemented in 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 initially 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 [1] [2]. After that, a proposed scheme is employed to achieve a better result. The proposed scheme extends the LK's scheme with a supplementary self-improvement mechanism that autonomously looks for the highest performance of the motion estimation.

This scheme is somewhat similar to the pel-recursive motion estimation, especially 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. 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.

Additionally, this research is constrained by some assumptions. The first assumption is that each sequential fisheye image consists of no camera rotation. The second is that the focal length of the sequential fisheye images is constant, therefore, those images are viewed in the maximum FoV. Another constraint is that the distance of the motion of each objects from the fisheye camera is not defined in the calculation. Finally, the camera motion is assumed to be small, or it occurs because the process of taking videos did not use a camera stabilization.

2.2 Related Works

Hard endeavors have been led to develop a motion estimation for serving omnidirectional images. They initially found a way of transforming the fisheye images to appropriate general perspective images. Hence, the advanced motion estimation schemes with slight modification can be used for further applications. I. Bauermann et al. started 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 [19]. It seems like the conversion process is began from

pre-warping the omnidirectional images to panoramic images. From then on, a resampling step is applied to scale the image in horizontal and vertical direction. Consequently, the panoramic images can be feed to the motion estimation process in the encoder.

Z. Jiali et al. proposed the affine motion compensated frame for estimating motion from a panoramic video sequence [20]. In their concept, a general perspective camera captures the wide-field-of-view along a given axis on spherical surface or a cylindrical. Moreover, image stitching and registration method are applied to obtain a continuity of series of images. Because 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. developed the adapted block-matching method for optical flow estimation in catadioptric images [21]. In their idea, the modified version of the classical block-based motion estimation is applied for estimating motion directly from synthetically or real catadioptric images. The modification is conducted by changing the shape and size of both the block of pixels and the window search area.

2.3 Research Methodology

2.3.1 Motion Estimation Design Methodology

In this research, the organic method of motion estimation process is designed to meet a flow diagram shown in Figure 2.4. It can be seen that 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. Hence, it can be used for further applications. 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.

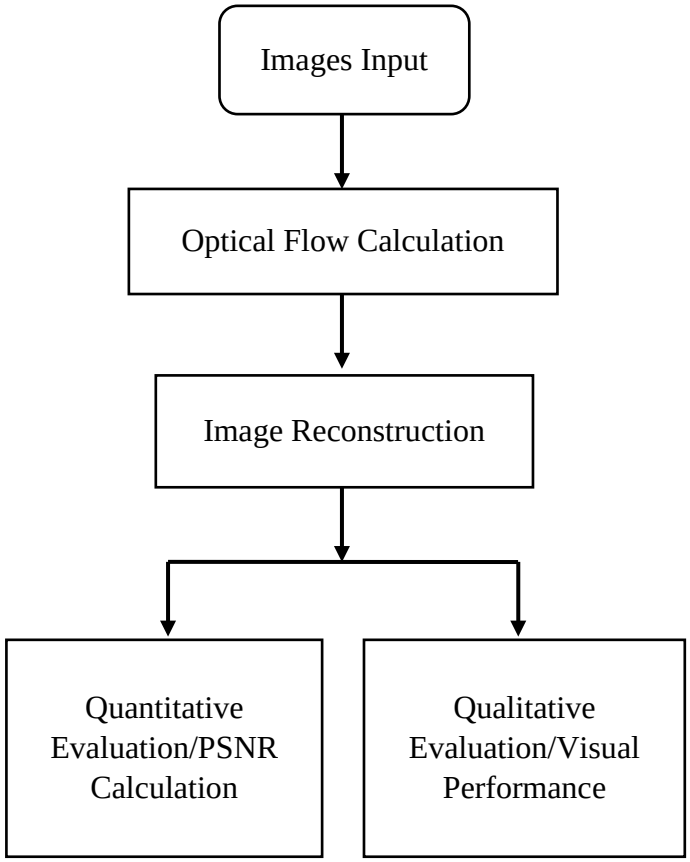


Figure 2.4. The flow diagram of the motion estimation design methodology.

The optical flow calculation process will produce motion vectors from the first fisheye image to the second one (forward motion vectors). These vectors 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).

2.3.2 The 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] [2] [22]. Now it can be utilized in the same manner for serving two consecutive fisheye images, and it is applicable to 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 Figure 2.5. In this figure, FN denotes frame number (the fisheye image number).

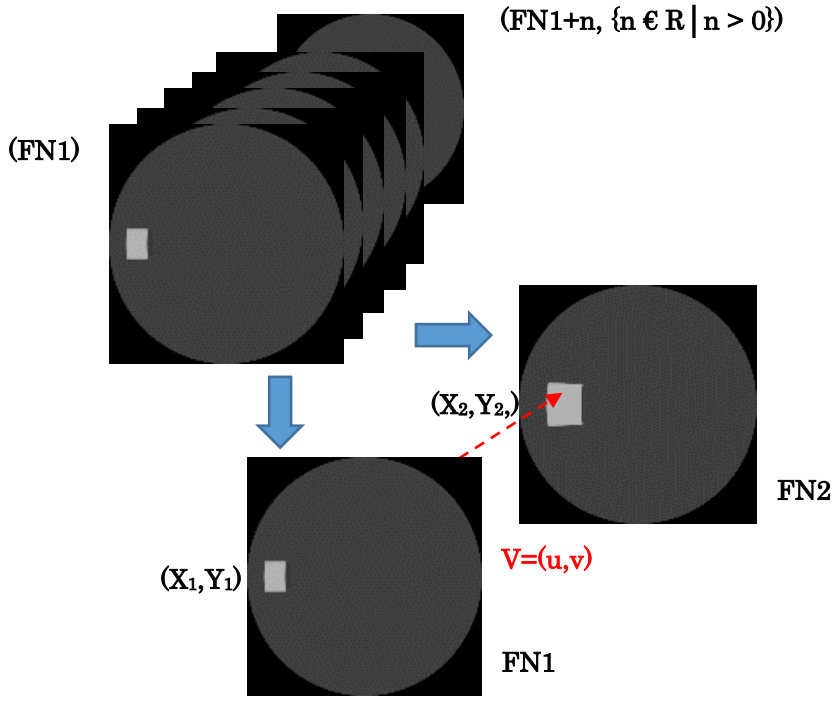


Figure 2.5. The optical flow calculation from two successive fisheye images.

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:

$$I_1(x_1, y_1, t) = I_2(x_1 + u, y_1 + v, t + 1) \quad (2.16)$$

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 equation (2.16), concerning such location of the pixel on FN1 (x_1,y_1), can simplify that equation to become a general optical flow equation as follows [23]:

$$uI_x + vI_y = -I_t \quad (2.17)$$

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 equation (2.17), this equation now becomes an optical flow problem (aperture problem) that have to be solved in a specific way.

2.3.2.1 Lucas and Kanade's Method

Lucas and Kanade (LK) propose a solution for the equation (2.17) by incorporating *n x n* neighboring pixels. They assume that pixels surrounding a pixel being observed behave almost the same as the observed pixel. Therefore, the equation (2.17) can be expressed as follows [24] [25]:

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{U} = \underbrace{-\begin{pmatrix} \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{B} \tag{2.18}$$

From the equation (2.18), the motion vectors (u and v) can be obtained. This equation is also well known as a solution for image registration technique.

2.3.2.2 Horn and Schunk's Method

Horn and Schunk (HS) introduce a solution for the equation (2.17) by considering a global constraint of smoothness [26]. 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:

$$u^{k+1} = u^{-k} - \frac{I_x\left(I_x u^{-k} + I_y v^{-k} + I_t\right)}{\alpha^2 + I_x^2 + I_y^2} \tag{2.19}$$

and

$$v^{k+1} = v^{-k} - \frac{I_y\left(I_x u^{-k} + I_y v^{-k} + I_t\right)}{\alpha^2 + I_x^2 + I_y^2} \tag{2.20}$$

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 ($k \in R \mid k > 0$).

2.3.2.3 Equidistance Projection Model

The equidistance projection (EP) model assumes that pixels on a fisheye image are composed by using an equidistance projection method [23] [27] [28]. In this scheme, motion on the fisheye image can be defined as follows:

$$u = a(x - x_0)^2 + a(y - y_0)^2 + c \quad (2.21)$$

$$v = b(x - x_0)^2 + b(y - y_0)^2 + d \quad (2.22).$$

Therefore, the equation (2.17) can be expressed as follows:

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 k^2 & \sum n^2 I_x I_y k^2 & \sum n^2 I_x k & \sum n^2 I_x I_y k \\ \sum n^2 I_x I_y^2 k^2 & \sum n^2 I_y^2 k^2 & \sum n^2 I_x I_y k & \sum n^2 I_y^2 k \\ \sum n^2 I_x k & \sum n^2 I_x I_y k & \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y k & \sum n^2 I_x k & \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}_{E} =$$

$$\underbrace{-\begin{pmatrix} \sum n^2 I_x I_t k \\ \sum n^2 I_y I_t k \\ \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{F} \quad (2.23).$$

After calculating the equation (2.23), *E* parameters (*a*, *b*, *c*, and *d*) can be divined. Therefore, the motion vectors denoted by the equation (2.21) and (2.20) now can be obtained.

2.3.2.4 Lucas and Kanade's Method with Automatic Self-Improvement

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 section 2.1.2.3, the discussion about the image quality metrics has been done briefly, and the PSNR will be used not only for the image quality assessment but also for judging the performance of the estimated motion vectors.

By considering the equation (2.13), the reconstructed fisheye image ($\hat{A}$) is developed from a simple linear geometry transformation of (A) using the motion vectors of *u* and *v*. 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 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 the fisheye image reconstruction process [29] [30].

In the following discussion, this approach will then be named Lucas and Kanade's concept with an improvement (LKI) [31]. Figure 2.6 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 the equation (2.18). 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.

2.3.3 Sequential Fisheye Images

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 2.2. The cube sequence is developed by using the blender software, while the flowers sequence is collected from the website of the multimedia communications and signal processing (LMS), Friedrich - Alexander University Erlangen - Nurnberg (FAU), Germany [9].

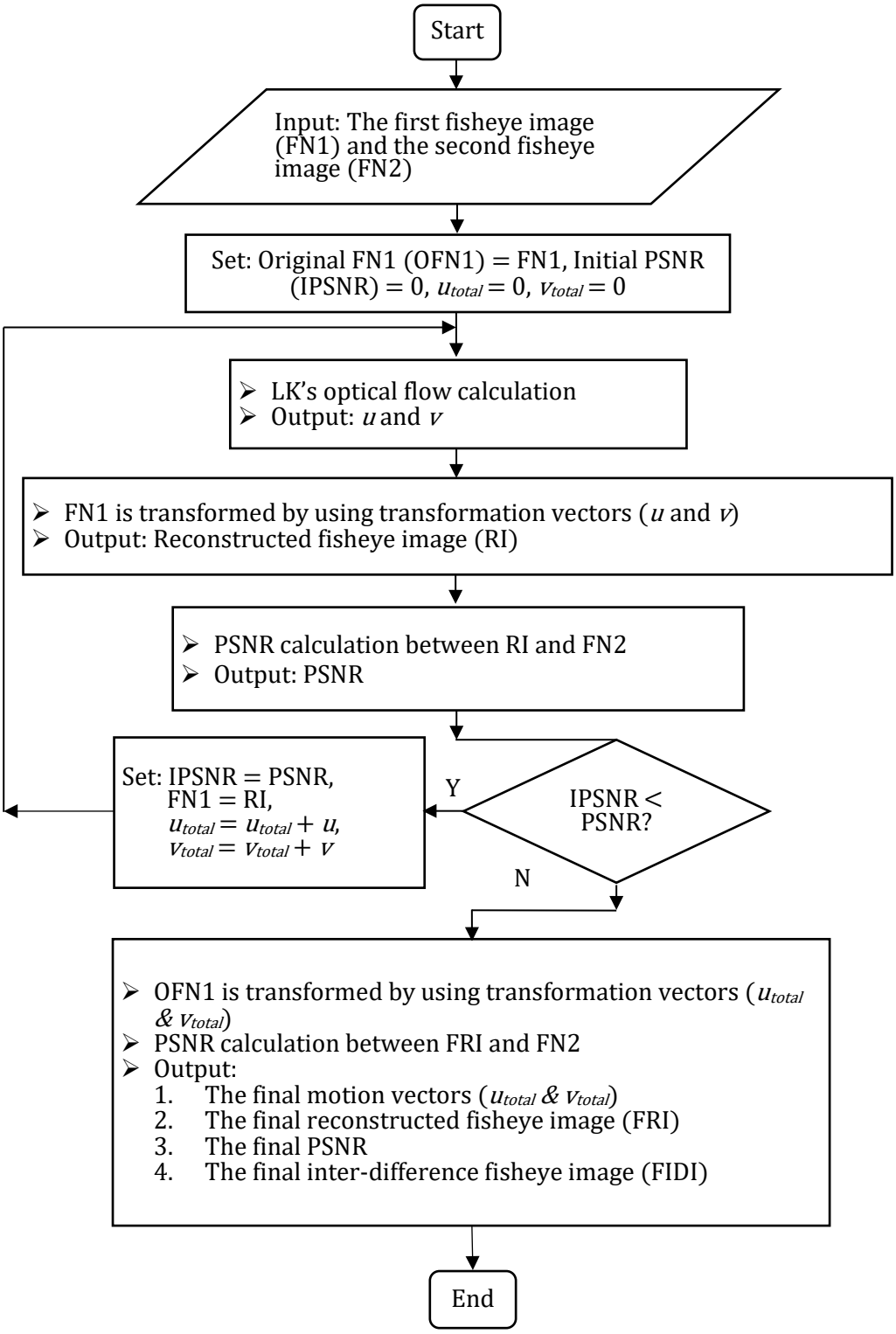


Figure 2.6. The flowchart of the LKI scheme.

Table 2.2. The characteristics of sequential fisheye images.

The Name of Sequential Fisheye Images	The Number	Frame Resolution	Types of Information	The Characteristic of Information

	of Frames			
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 information in all sequential fisheye images, except the cube and flowers, is captured by a Ricoh Theta S fisheye camera [32]. In this experiment, the data is only obtained through one side lens. 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.

2.4 Research Results

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

2.4.1 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 R2016a.

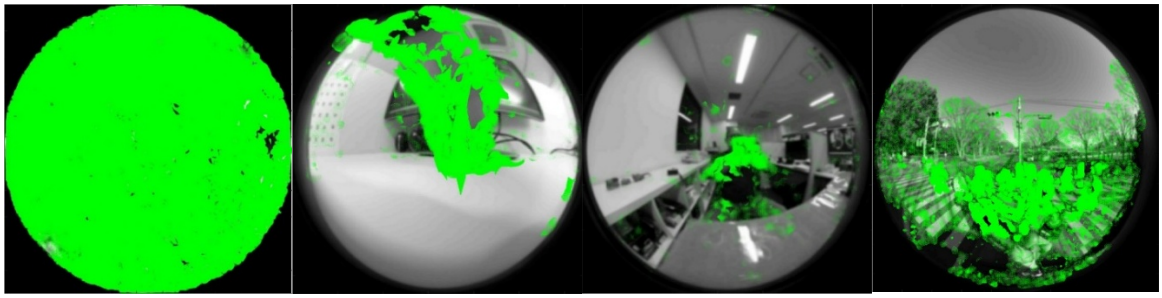
The motion is estimated from two successive fisheye images, and it is applied to all fisheye images within each sequential fisheye image listed in Table 2.2. The objective of the experiment is to pay attention to how far the proposed scheme (LKI) can increase the performance of motion vectors, in comparison 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.

2.4.2 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 [16]. The size of the macro-block used in this experiment is set to 8×8 pixels.

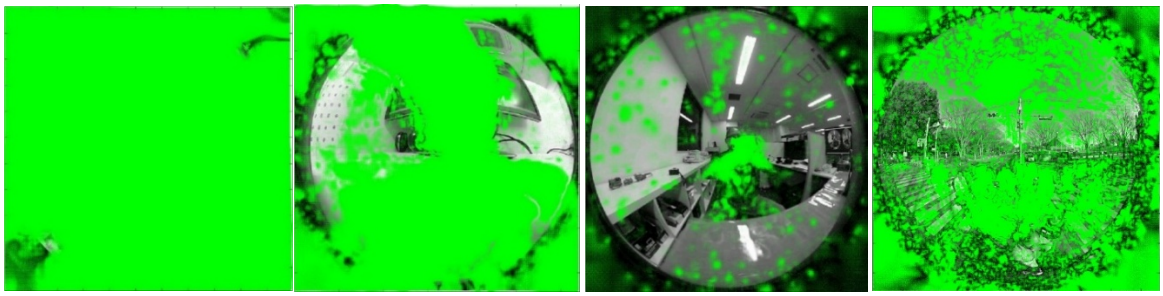
2.4.3 The Estimated Vectors

Motion vectors (MV) or optical flow (OF) can be obtained directly from each sequential fisheye image listed in Table 2.2 by using every OF scheme mentioned in section 2.3.2. 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 Figure 2.7 and Figure 2.8. Explicitly, the examples of the MV obtained from the LK's scheme are shown in Figure 2.7, while the other examples of the MV produced by the LKI process are depicted in Figure 2.8. In compare to the LK's scheme, the LKI process can produce MV more densely [33]. 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 noise in the fisheye image area.



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

Figure. 2.7. 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

Figure 2.8. The MV results after LKI process.

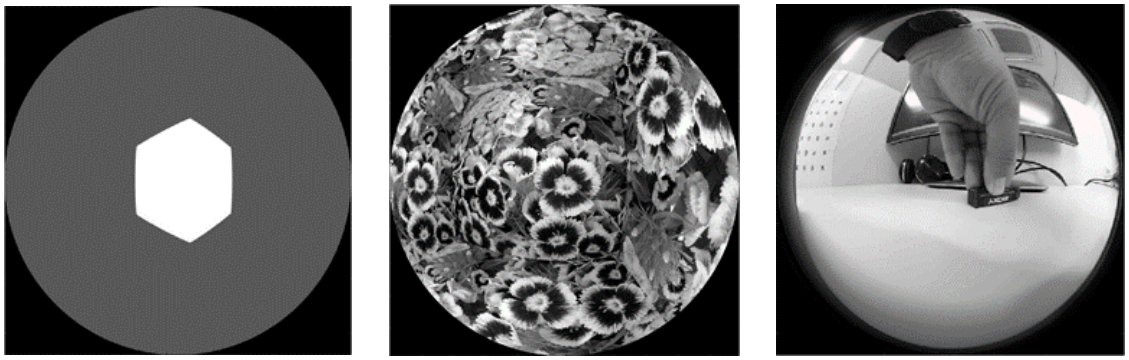
2.4.4 Qualitative Evaluation

Example of fisheye images taken from each sequential fisheye image listed in Table 2.2 can be seen in Figure 2.9 (a - l). Those figures are the second fisheye image input (FN2). Some examples of the RI produced by the LK process can be seen in Figure 2.10 (a - c), Figure 2.11 (a - c), Figure 2.12 (a - c), and Figure 2.13 (a - c). Some other examples of the IDI associating with its RI can be seen in Figure 2.10 (d - f), Figure 2.11 (d - f), Figure 2.12 (d - f), and Figure 2.13 (d - f). Those RI figures show that each RI is not the same as each associated FN2. This condition is also reflected in 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 Figure 2.10 (g - i), Figure 2.11 (g - i), Figure 2.12 (g – i), and Figure 2.13 (g - i) for the RI results, and Figure 2.10 (j - l), Figure 2.11 (j - l), Figure 2.12 (j - l), and Figure 2.13 (j - l) 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.

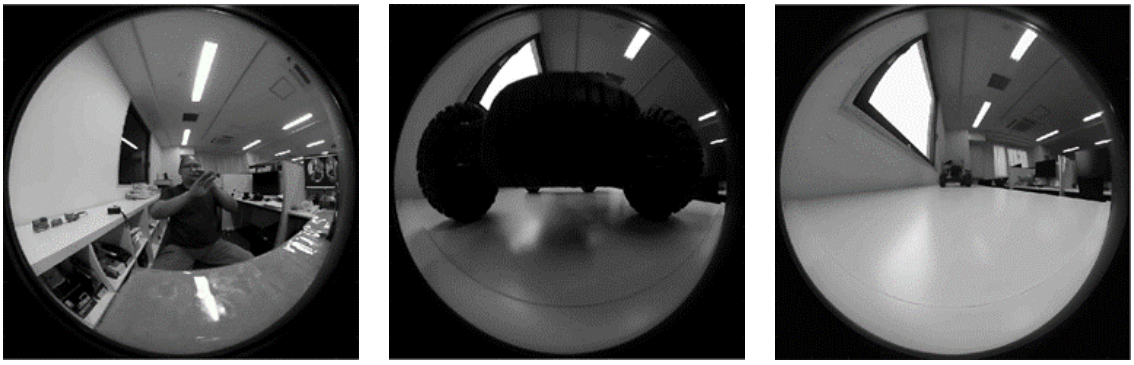
2.4.5 Quantitative Evaluation

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 Figure 2.14 to Figure 2.25.

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 Figure 2.15, Figure 2.23, Figure 2.24, and Figure 2.25, 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.



(a) The FN-106 of CQ (b) The FN-66 of FQ (c) The FN-80 of HQ

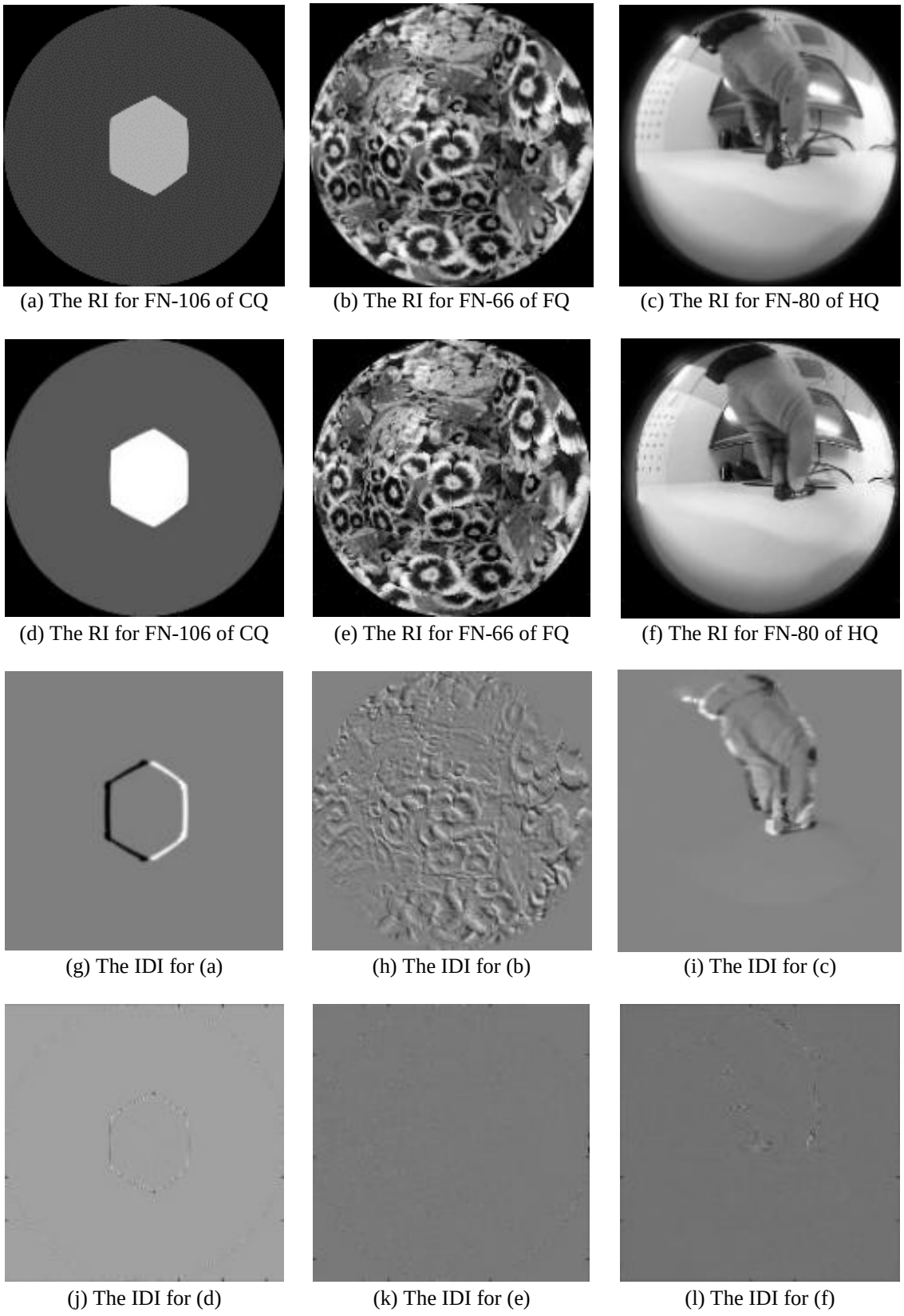


(d) The FN-46 of MQ (e) The FN-15 of TQ-I (f) The FN-21 of TQ-II

(g) The FN-16 of CQ-I (h) The FN-21 of CQ-II (i) The FN-44 of TQ

(j) The FN-119 of TRQ-I (k) The RI for FN-76 of TRQ-II (l) The RI for FN-160 of PQ

Figure 2.9. The examples of the fisheye image obtained each sequential fisheye images listed in Table 2.2



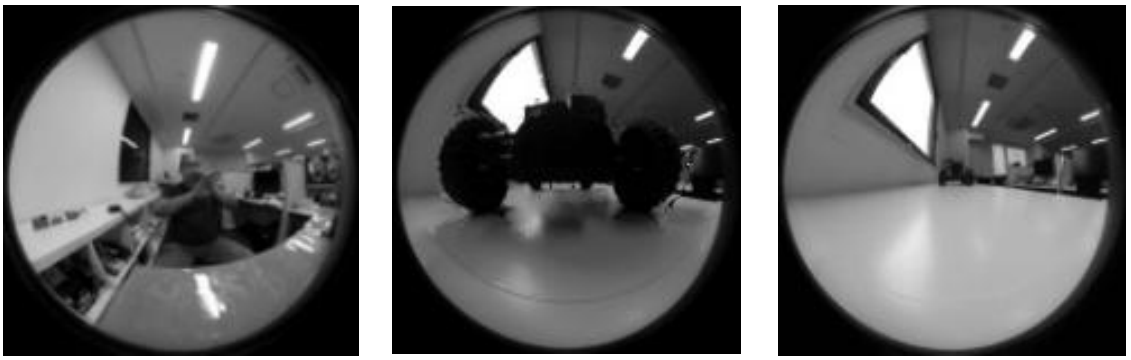
(a) The RI for FN-106 of CQ (b) The RI for FN-66 of FQ (c) The RI for FN-80 of HQ

(d) The RI for FN-106 of CQ (e) The RI for FN-66 of FQ (f) The RI for FN-80 of HQ

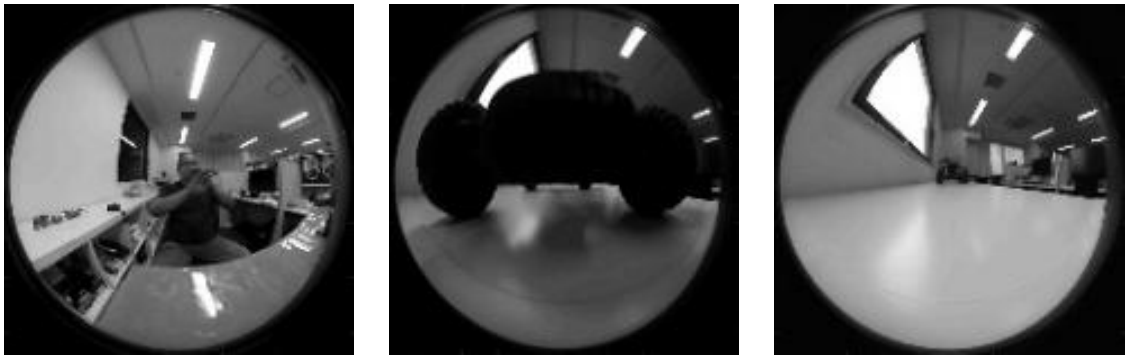
(g) The IDI for (a) (h) The IDI for (b) (i) The IDI for (c)

(j) The IDI for (d) (k) The IDI for (e) (l) The IDI for (f)

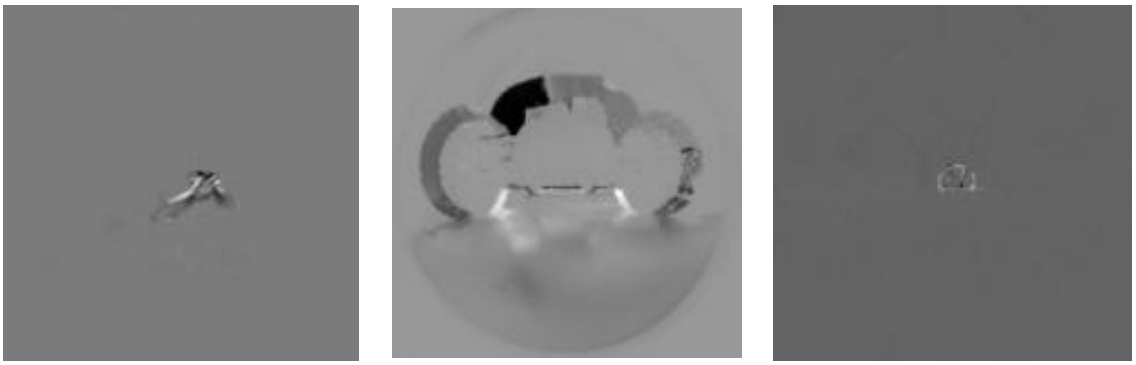
Figure 2.10. The RI results after LK's process (a - c), RI results after LKI process (d - f), IDI results after LK's process (g - i), and IDI results after LKI process (j - l).



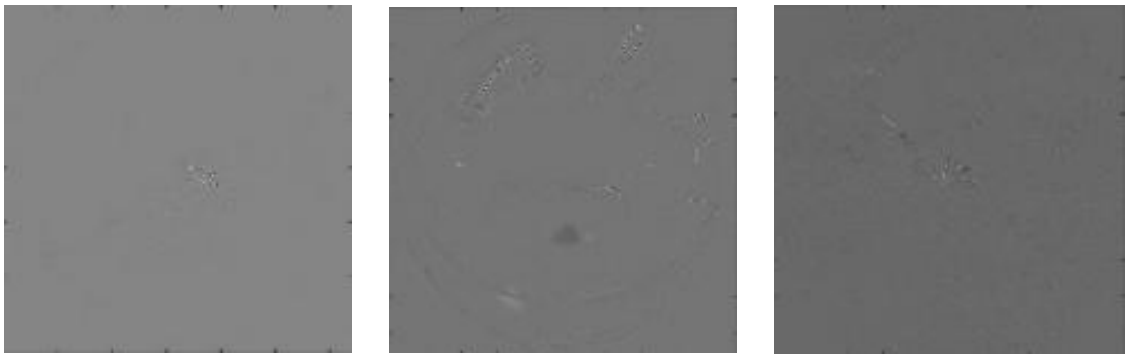
(a) The RI for FN-46 of MQ (b) The RI for FN-15 of TQ-I (c) The RI for FN-21 of TQ-II



(d) The RI for FN-46 of MQ (e) The RI for FN-15 of TQ-I (f) The RI for FN-21 of TQ-II



(g) The IDI for (a) (h) The IDI for (b) (i) The IDI for (c)



(j) The IDI for (d) (k) The IDI for (e) (l) The IDI for (f)

Figure 2.11. The RI results after LK's process (a - c), RI results after LKI process (d - f), IDI results after LK's process (g - i), and IDI results after LKI process (j - l).

(a) The RI for FN-16 of CQ-I

(b) The RI for FN-21 of CQ-II

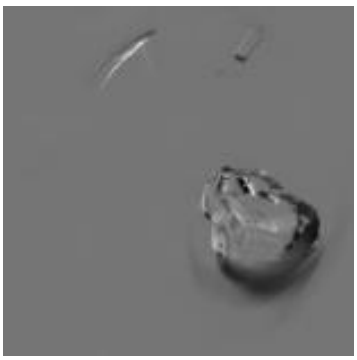
(c) The RI for FN-44 of TQ

(d) The RI for FN-16 of CQ-I

(e) The RI for FN-21 of CQ-II

(f) The RI for FN-44 of TQ

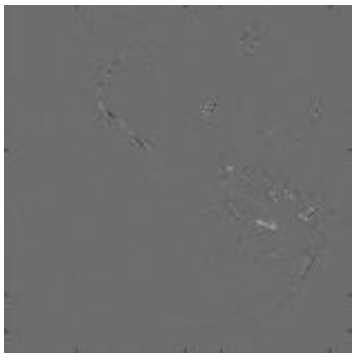
(g) The IDI for (a)



(h) The IDI for (b)

(i) The IDI for (c)

(j) The IDI for (d)



(k) The IDI for (e)

(l) The IDI for (f)

Figure 2.12. The RI results after LK's process (a - c), RI results after LKI process (d - f), IDI results after LK's process (g - i), and IDI results after LKI process (j - l).

(a) The RI for FN-119 of TRQ-I

(b) The RI for FN-76 of TRQ-II

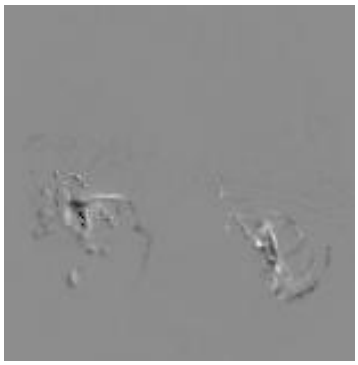
(c) The RI for FN-160 of PQ

(d) The RI for FN-119 of TRQ-I

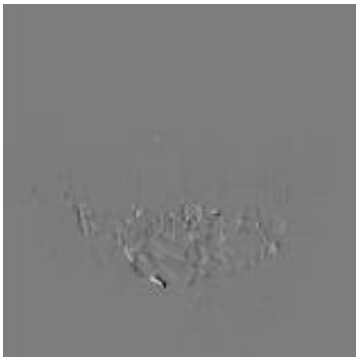
(e) The RI for FN-76 of TRQ-II

(f) The RI for FN-160 of PQ

(g) The IDI for (a)



(h) The IDI for (b)



(i) The IDI for (c)

(j) The IDI for (d)

(k) The IDI for (e)

(l) The IDI for (f)

Figure 2.13. The RI results after LK's process (a - c), RI results after LKI process (d - f), IDI results after LK's process (g - i), and IDI results after LKI process (j - l).

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 the 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 Figure 2.19. Moreover, this condition also means that a rapid motion around the center area 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 Figure 2.16, Figure 2.18, and Figure 2.24, 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.

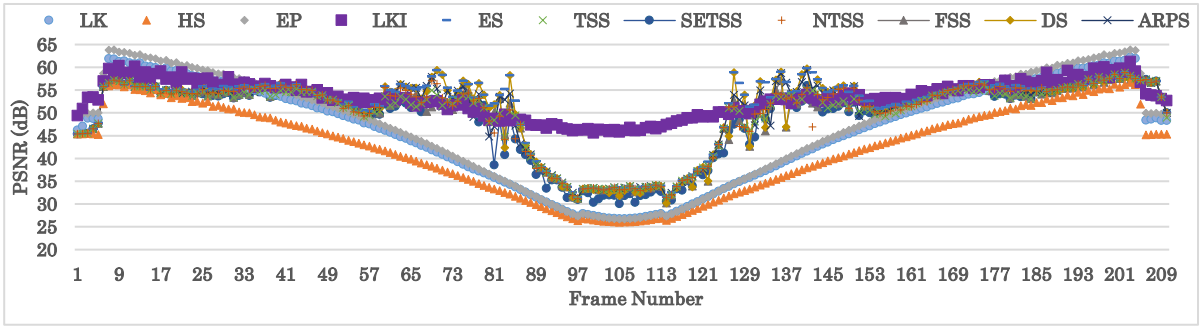


Figure 2.14. PSNR versus frame number for the cube sequence (CQ).

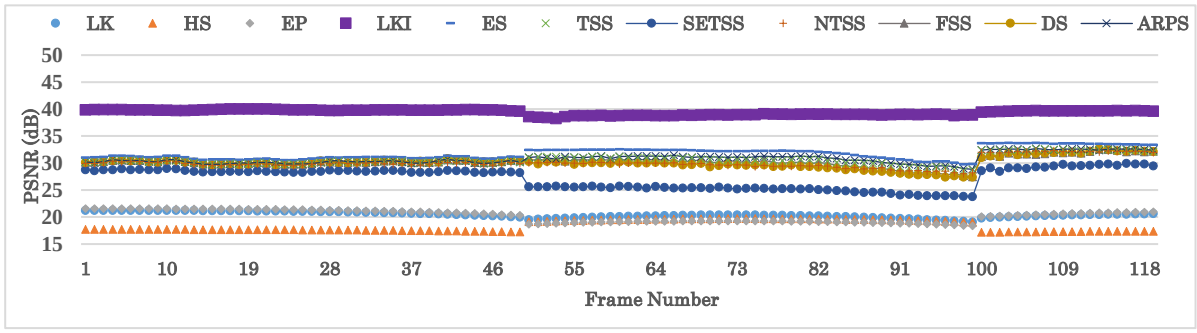


Figure 2.15 PSNR versus frame number for the flowers sequence (FQ).

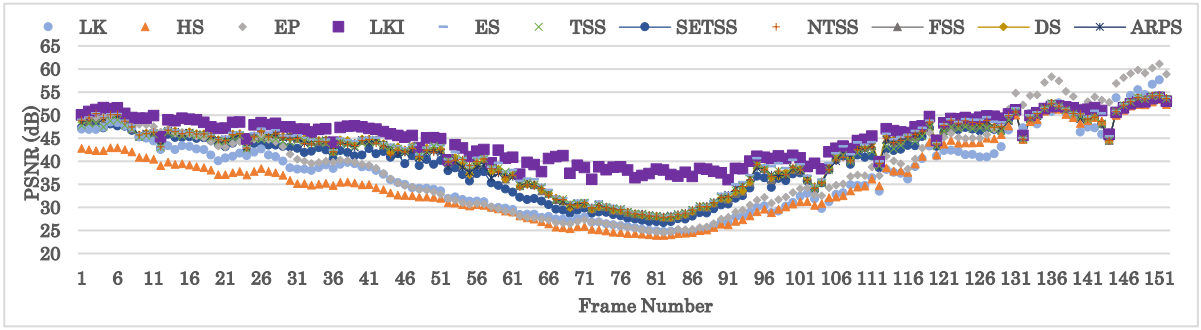


Figure 2.16 PSNR versus frame number for the hand sequence (HQ).

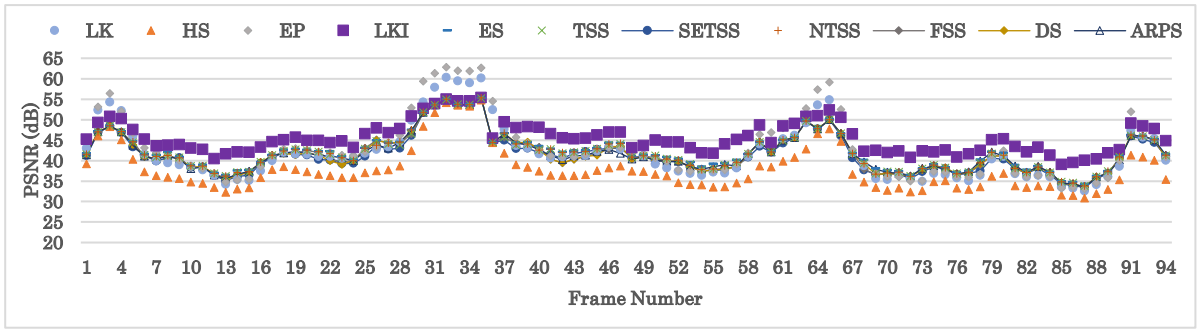


Figure 2.17. PSNR versus frame number for the man sequence (MQ).

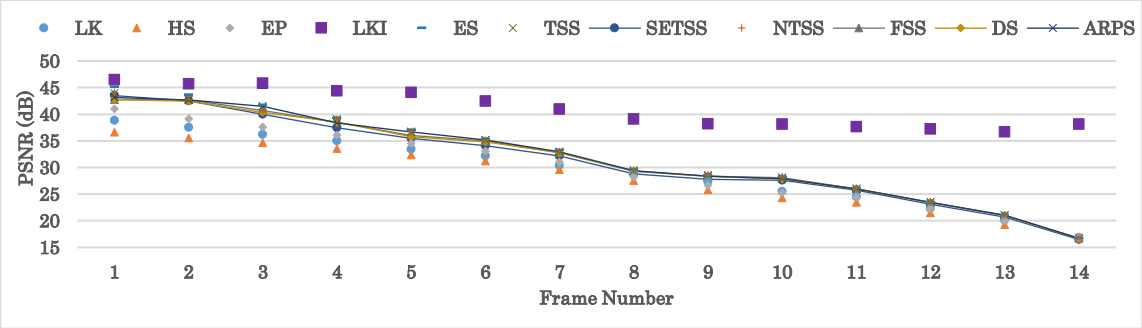


Figure 2.18 PSNR versus frame number for the truck I sequence (TQ-I).

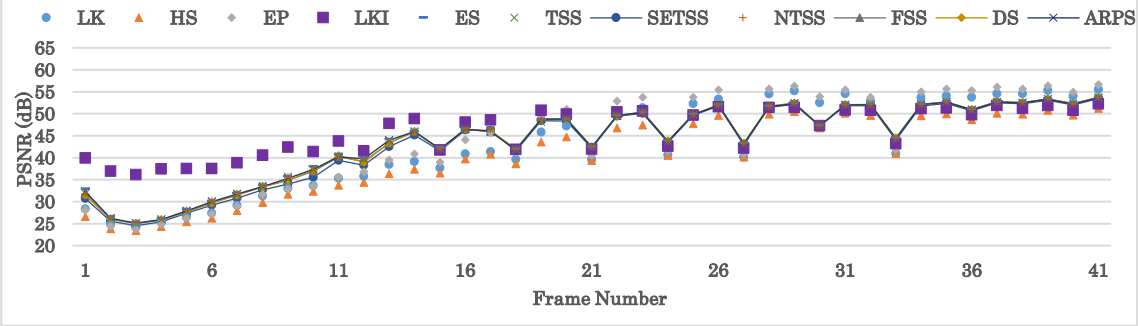


Figure 2.19 PSNR versus frame number for the truck II sequence (TQ-II).

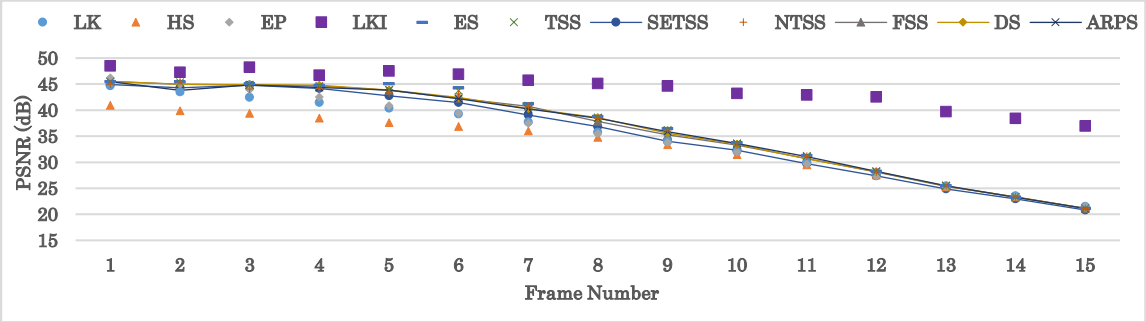


Figure 2.20. PSNR versus frame number for the car I sequence (CQ-I).

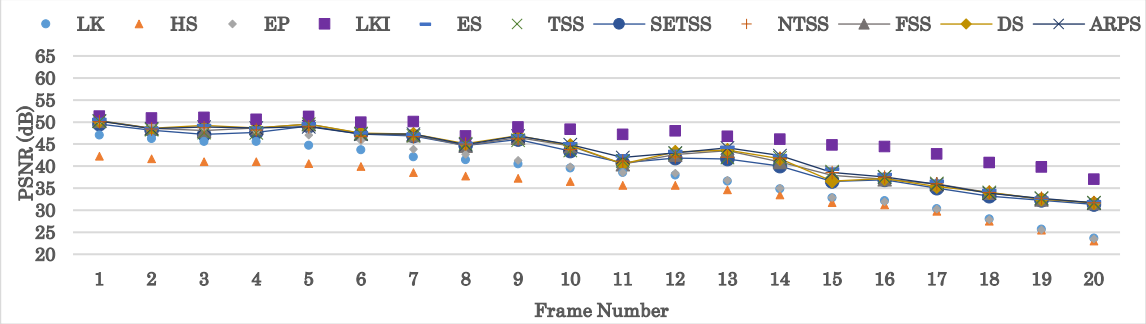


Figure 2.21 PSNR versus frame number for the car II sequence (CQ-II).

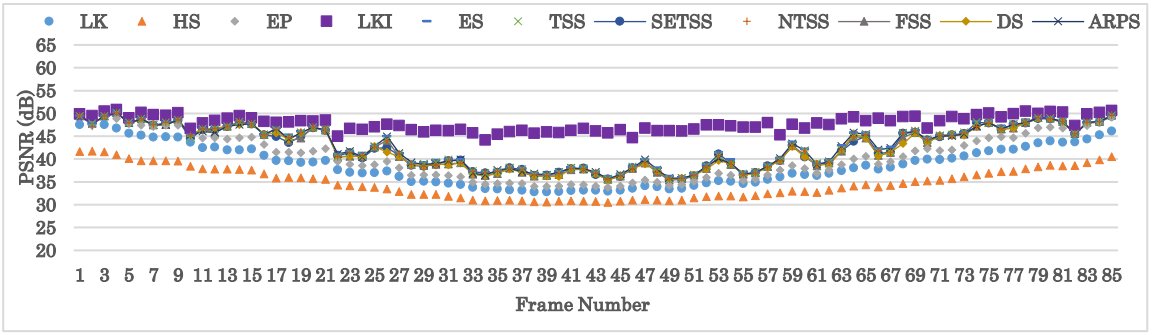


Figure 2.22 PSNR versus frame number for the train sequence (TQ).

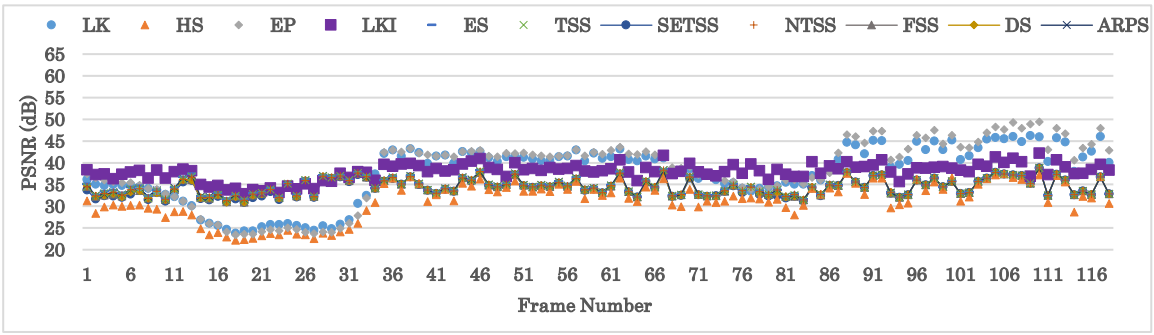


Figure 2.23. PSNR versus frame number for the traffic I sequence (TRQ-I).

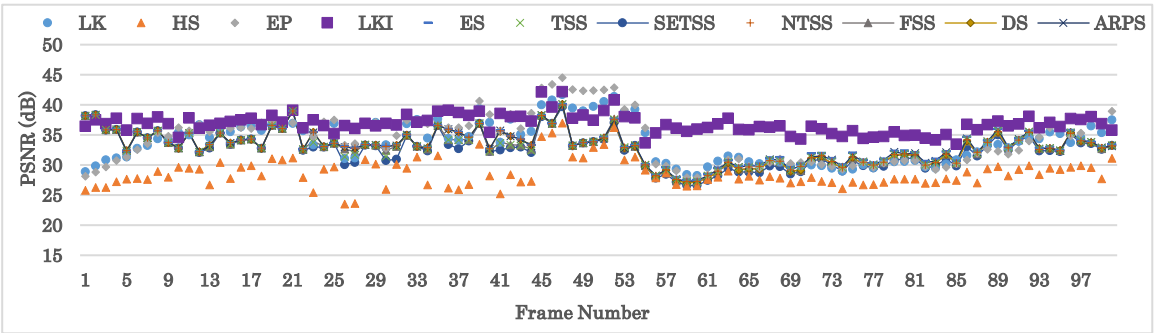


Figure 2.24 PSNR versus frame number for the traffic II sequence (TRQ-II).

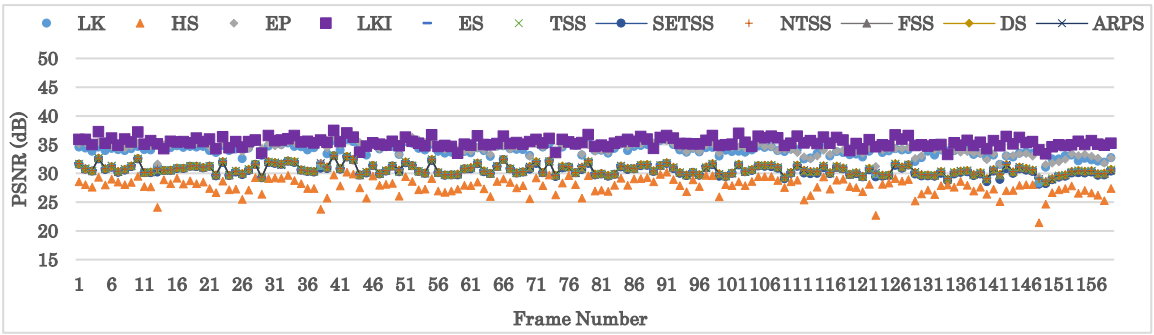


Figure 2.25 PSNR versus frame number for the people sequence (PQ).

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 Figure 2.20, Figure 2.21, Figure 2.22, and Figure 2.23, 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 Figure 2.23. 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 x 15* neighboring pixels) to obtain the MV using the equation (2.18). 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 [34]. 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 x 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 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 € 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.

2.4.6 The Processing Time

The amount of time spent on obtaining MV from two successive fisheye images is defined as processing time [22]. Table 2.3 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 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.

Table 2.3 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

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.

The above results are achievable by running codes on the Intel Pentium i7-3770 CPU, 8 GB RAM. We develop the codes by using MATLAB R2016a, which is installed on Window 7 operating system.

2.4.7 The Advanced Results

Although the LKI scheme has been succeeded to be used for estimating motion directly from sequential fisheye images, the capability of this method can still be evaluated in further measurement scenarios. One of the scenarios is to handle a sudden movement or a motion that is larger than a couple of pixels. To be able to implement the sudden movements of objects, the input pair of fisheye images can be obtained from one (frame1 and frame2), two (frame1 and frame3), three (frame1 and frame 4), four (frame1 and frame5); or five (frame1 and frame6) image difference inside sequential fisheye images [35]. There are four sequential fisheye images used in this experiment. These sequential fisheye images are the Hand, Man, Cube, and Truck I. Moreover, the same as the previous experiment, the performance of motion vectors is determined from the quality of the reconstructed fisheye image (RI).

In the quantitative evaluation, it can be seen from Figure 2.26 to Figure 2.29 that the performance of the LKI goes down gradually proportional to a reduction of image difference. When the image difference reaches 5, the PSNR of each sequence generally reduces until about 15 dB from the maximum value (when the image difference is 2). However, this condition is far better in comparison with the result obtained by using block-based (BB) motion estimation scheme. The gap can reach about 14 dB in the worst situation.

The results can also be seen in the qualitative evaluation. The quality of the RI for each sequence are presented in Figure 2.30. Each RI's inside that figure represents a sample from each sequence. It can be seen that the RI for the Man and Cube sequence is still good, while the RI for the Hand sequence is distorted slightly. In case of the Truck I, since the position of the object in the frame1 and frame3 is very different (it caused by the different speed of truck while it was captured at t=1 and t=3), the obtained motion vectors cannot be determined very well. Therefore, the RI becomes worst. This sequential fisheye images can be categorized as an unpredictable movement of the object, or it is worse than the sudden movements of the object.

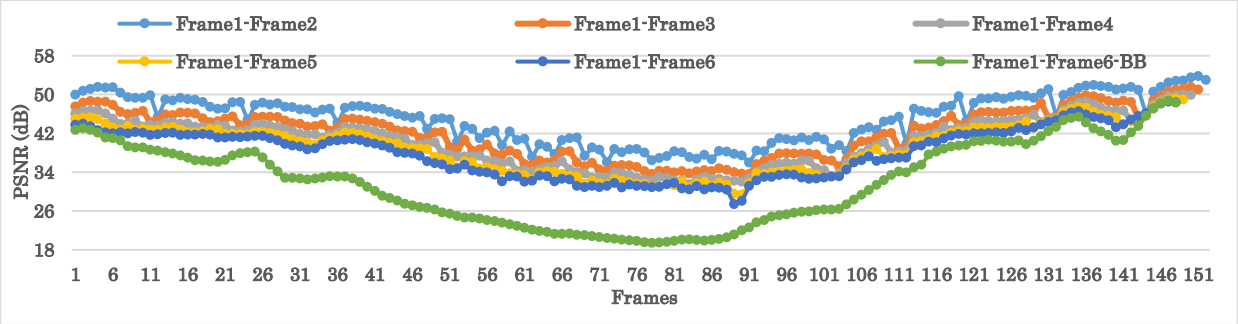


Figure 2.26. PSNR versus frame number for the hand sequence (HQ) (The advanced results).

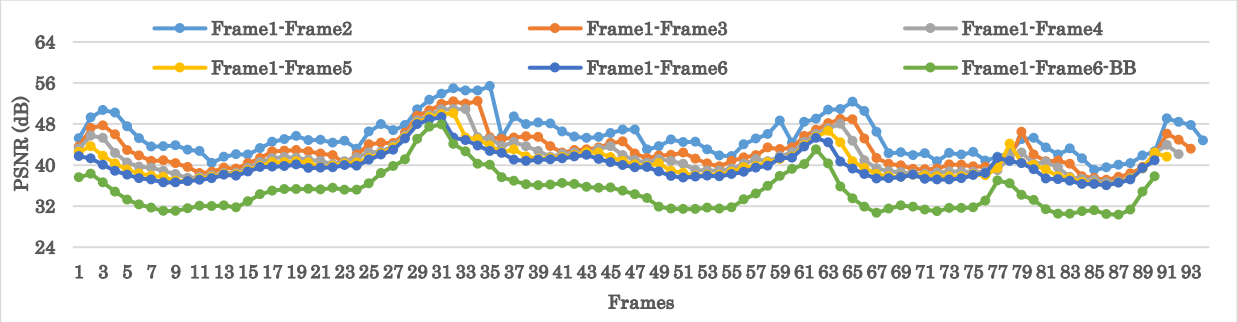


Figure 2.27 PSNR versus frame number for the man sequence (MQ) (The advanced results).

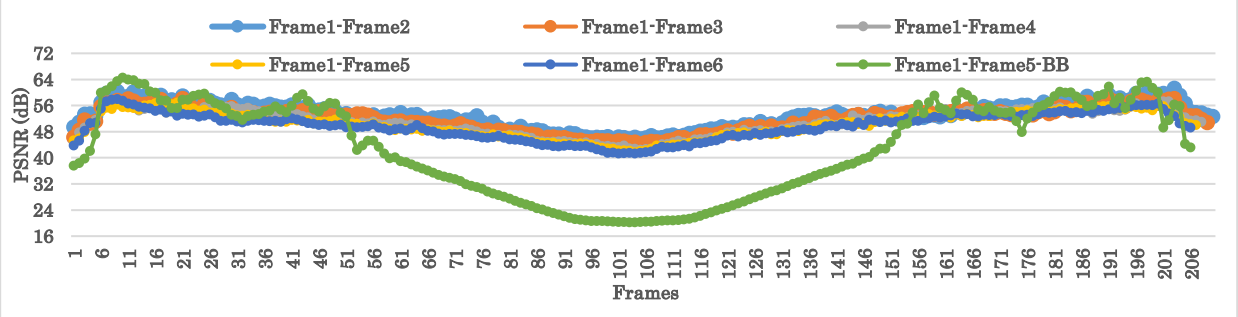


Figure 2.28. PSNR versus frame number for the cube sequence (CQ) (The advanced results).

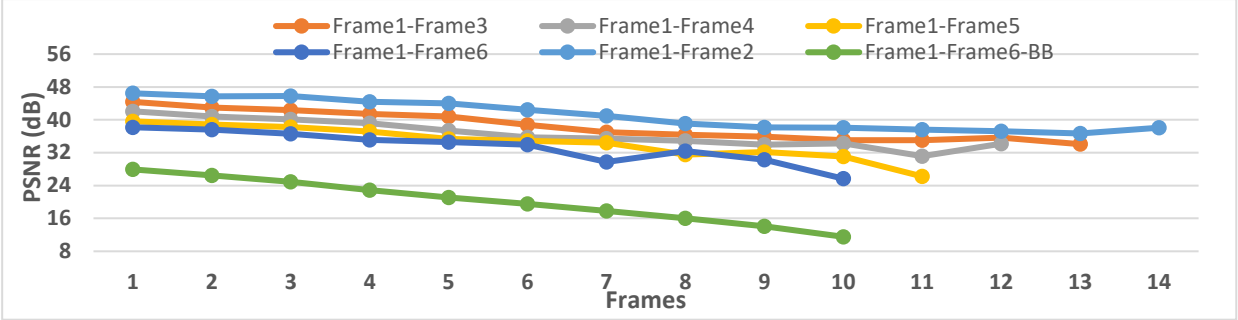
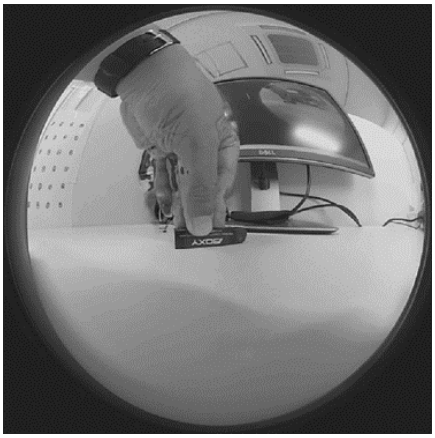
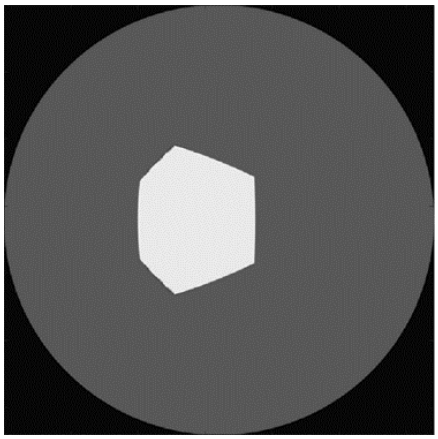


Figure 2.29 PSNR versus frame number for the truck I sequence (TQ-I) (The advanced results).

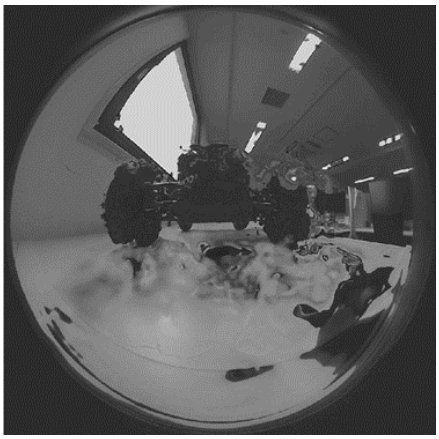


(a) The RI for hand sequence (The Advanced Results)

(b) The RI for man sequence (The Advanced Results)



(c) The RI for cube sequence (The Advanced Results)



(d) The RI for truck I sequence (The Advanced Results)

Figure 2.30. The qualitative performance of the estimated motion shown by the reconstructed images (RI)

2.5 Chapter Conclusion

In this research, the design and analysis of the optical-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 a high level.

The evaluation of the LKI scheme for handling four sequential fisheye images with a sudden movement of the object has been conducted. The LKI scheme works well to produce motion vectors from at least three frames difference. This means that to some extent, the LKI can estimate motion from sequential fisheye images with sudden movements of objects.

Chapter 3

3-D Optical Flow-Based Motion Estimation (Scene Flow) for Sequential Stereo Fisheye Images

3.1 Introduction

3.1.1 Background and Motivation

Estimating the three dimensional (3-D) scene flow directly from multi- view general perspective images has successfully developed in recent years. This technique has become a valuable tool to reveal real-world motion [3] [36]. Therefore, this method can be used for reconstructing the 3-D object, analyzing 3-D motion, developing 3-D videos, or autonomous vehicle [37]. Generally, to be able to generate the 3-D scene flow, many developed algorithms use multiple sequential images that are obtained from multiple general perspective cameras as well. Indeed, some recent developments have been succeeded to generate the 3-D motion by using only two general perspective cameras. This camera system is then mainly known as the stereo camera system [38]. Besides being more efficient, to some extent, this camera system is extremely good to reveal the motion of non-complex objects completely.

Nevertheless, to some other extent, either a stereo camera system or a multiple cameras system is not too efficient anymore, particularly if they use general perspective cameras. This condition is caused by the camera system maybe need more space to accommodate many cameras. Generally, to cover larger field-of-view (FoV), the camera system needs more than one general perspective camera. This because the general perspective camera has narrow FoV. Furthermore, in the stereo general perspective camera system, the FoV is obviously very limited. It might be needed more effort to maximize the capability of the FoV.

To be able to simplify a stereo camera system, while at the same time the stereo camera system has maximum FoV, one of the possible solutions is to employ two fisheye cameras. Recently, many kinds of fisheye cameras are available in the market. One of them is very small, and it has large FoV (2×180^0) [32]. By employing two fisheye camera, the capability of view of the stereo camera system can be expanded. As a result, it can be used for estimating the 3-D motion located surrounding the camera.

Even though the stereo fisheye camera has large FoV, the fisheye camera itself suffers from distortion. This distortion actually has been discussed in section 2. Hence, to be able to develop the 3-D motion estimation directly from sequential stereo fisheye images, we need to decide a suitable method carefully. In this research, we propose to make use of the optical flow and disparity calculation to obtain the 3-D scene flow.

3.1.2 Problem Formulation

In similar form to the optical flow, which is interpreted as the 2-D motion of objects or pixels on an image plane, scene flow is illustrated as the 3-D motion field of objects or points in the real-world [3] [39] [40]. This term also means that an uncomplicated projection of the 3-D scene flow onto an image plane is the 2-D optical flow. The relation between the two kinds of motion is very tight. Firstly, by considering a 3-D point of **A** *(X,Y,Z)* positioned in the real-world, the projection of this point onto an image coordinate $\mathbf{a}_n$ (x_n,y_n) of camera *n* is define by the equation (3.1) and equation (3.2), in which $[P_i]_j$ is j^{th} row of the projection matrix P_i. Those equations can be expressed as follows:

$$x_n = \frac{[P_i]_1(X,Y,T)^T}{[P_i]_3(X,Y,T)^T} \tag{3.1}$$

and

$$y_n = \frac{[P_i]_2(X,Y,T)^T}{[P_i]_3(X,Y,T)^T} \tag{3.2}.$$

From then on, by assuming the cameras are static, the relationship between the optical flow ($\vec{\mathbf{v}} = d\mathbf{a}_n/dt$) and the scene flow ($\vec{\mathbb{V}} = d\mathbf{A}/dt$) can be stated as follows:

$$\frac{d\mathbf{a}_n}{dt} = \frac{\partial \mathbf{a}_n}{\partial \mathbf{A}} \frac{d\mathbf{A}}{dt} \tag{3.3}.$$

According to [39], the scene flow then can be estimated by at least two cameras. Therefore, it has to be composed as a linear equation *VB = U*. This linear equation can be expressed as follows:

$$B = \begin{bmatrix} \frac{\partial x_1}{\partial X} & \frac{\partial x_1}{\partial Y} & \frac{\partial x_1}{\partial Z} \\ \frac{\partial y_1}{\partial X} & \frac{\partial y_1}{\partial Y} & \frac{\partial y_1}{\partial XZ} \\ \frac{\partial x_2}{\partial X} & \frac{\partial x_2}{\partial Y} & \frac{\partial x_2}{\partial Z} \\ \frac{\partial y_1}{\partial X} & \frac{\partial y_1}{\partial Y} & \frac{\partial y_1}{\partial Z} \end{bmatrix}, U = \begin{bmatrix} \frac{\partial x_1}{\partial t} \\ \frac{\partial y_1}{\partial t} \\ \frac{\partial x_2}{\partial t} \\ \frac{\partial y_2}{\partial t} \end{bmatrix}, VB = U \tag{3.4}.$$

The 3-D coordinate **A** in a stereo setup has an interesting association with disparity *(d)* and corresponding image coordinates (x_{Right} and y_{Right} are the right views, while x_{Left} and y_{Left} are the left views). Their relationship can be approximated as follows:

$$X = \frac{T(x_{Right}+x_{Left})}{2d}, Y = \frac{T(y_{Right}+y_{Left})}{2d}, Z = \frac{fT}{d} \qquad (3.5),$$

where *T* denotes as the baseline (distance between the two cameras), while *f* defines as the focal length of the cameras (assume that the two fisheye cameras have similar focal length, *f = 1*).

To be able to get *V* (*VB = U*) mentioned by the equation (3.4), we need to carefully obtain each partial derivative element inside the *B* and *U*. In this research, those elements can be obtained from the optical flow and the disparity between two fisheye images.

3.1.3 Contribution and Constrain

This research aims to estimate the 3-D scene flow directly from sequential stereo fisheye images. The main advantage of developing this research is to extend the FoV of the stereoscopic camera system by using low-cost fisheye cameras. Therefore, it can be used for further applications, particularly for the 3-D motion analysis in a fisheye domain, the 3-D video system, and the 3-D reconstruction system.

In this research, the 3-D scene flow estimation is developed by employing the 2-D motion estimation and the disparity between a stereo fisheye images. We start from the idea proposed by S. Vidula et al. to build understanding about utilizing the optical flow for generating the 3-D motion [39]. However, we avoid using more than two cameras. Hence, we need to consider the idea proposed by R. Li, and S. Sclaroff [40]. They use a stereo general perspective camera to get the optical flow and the disparity for calculating the 3-D motion. Although our proposed idea seems to be the same as them, we use a different way of calculating the optical flow and the disparity. Firstly, we make use of a stereo camera rather than a stereo general perspective camera. Secondly, since we found a promising result after making use of the Lucas and Kanade's optical flow approximation for calculating the optical flow directly from sequential fisheye images, we focus on extending this approximation to be used for the disparity calculation as well.

Furthermore, this research is constrained by some assumptions. Firstly, the stereo fisheye camera and its sequential fisheye images are made by using the blender software [41]. The use of this synthetic sequential fisheye images makes the quantitative

measurement of the result becomes easy. The qualitative evaluation can also be evaluated easily. Secondly, this research assumes that the sequential stereo fisheye images have the same sampling time. Lastly, the stereo fisheye camera has no rotation, and it works in the maximum FoV.

In term of calculating the quantitative performance of the 3-D scene flow, this research uses average angular error (AAE) and standard deviation angular error (SAE) [42]. The angular error (AE) can be computed as follows:

$AE = cos^{-1}(a.\bar{a})$ (3.6),

where a is the correct vector, while $\bar{a}$ is the estimated vector.

3.2 Related Works

In our observation, the concept of generating the 3-D scene flow is started by S. Vidula et al. [39]. The approach provides an enormous opportunity for obtaining the 3-D scene flow with or without taking geometry information continuously from the real 3-D surface. One of the options will make generating the 3-D scene flow more efficient since we can depend on 2-D information greatly, although in the same time we will face greater noise that makes the 3-D scene flow seems to be unreliable. This condition means that the 3-D scene flow can be generated by using the 2-D optical flow information, but it has to come from multiple cameras.

Another researcher, R. Li, and S. Sclaroff propose a different idea [40]. They combine the optical flow and the disparity between images into calculation framework of 3-D scene flow. They firstly construct a binocular camera system to produced sequential stereo images, therefore, in the following process, this information can be used for obtaining the optical flow and the disparity. Finally, the 3-D scene flow can be generated.

Another way of generating the 3-D scene flow is also developed by J. Quiroga et al. [43]. Instead of using multiple cameras, they construct a single camera together with RGBD sensor to obtain 2-D motion and depth before doing 3-D motion estimation in the final step. The result seems to be useful to be used for generating 3-D scene flow of a very close phenomenon.

A. Wedel et al. build the 3-D motion by using a stereoscopic scene flow estimation framework [36]. This framework is mainly constructed from the latest development of

optical flow and stereo matching estimation. It seems like combining the 2-D motion estimation and depth information to obtain the 3-D scene flow.

All of the above works successfully develop the 3-D scene flow estimation directly from general perspective images. Generating it directly from sequential fisheye images seems to be challenging now.

3.3 Proposed Approach

This section discussed the method of obtaining 3-D scene flow in this research comprehensively. The method is developed to follow a conceptual diagram, as shown in Figure 3.1 [44] [45]. At the beginning step, two identical fisheye cameras are configured to capture the same scene phenomenon, as it can be seen on the left side of the figure. The configuration of the two fisheye camera follows the general stereo configuration system that previously has been designed for two general perspective cameras [2] [3]. Having captured the scene phenomenon, each camera ideally will produce the same sequential fisheye images but has a different point of view.

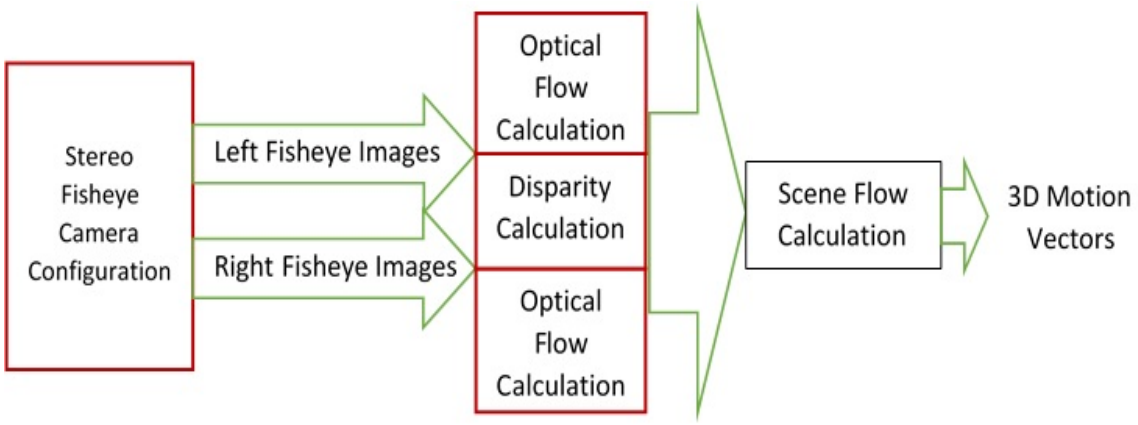


Figure 3.1. The conceptual diagram of obtaining 3-D scene flow [44] [45].

One fisheye camera provides the left point of view, while the other one presents the right one. Each of them is then fed to the second step, in which the calculation of the optical flow and disparity between two fisheye images perform. For each side (right and left), every two successive fisheye images from the time-varying fisheye images can be used for obtaining the optical flow, while every two adjacent fisheye images from both time-varying fisheye images can be used for obtaining the disparity between two fisheye images. The outputs of

this step will then be ready for feeding to the 3-D scene flow calculation process denoted by the equation (3.4). As a result, at the end of the process, the 3-D scene flow can be generated. This is also well known as the estimated motion in the 3-D domain.

The process of calculating both the optical flow and disparity between two fisheye images in the second step can be described as Figure 3.2 [44] [45]. In this figure, the fisheye camera 1 corresponds to the left-side camera, while the fisheye camera 2 correlates to the right-side camera. Either the optical flow or disparity calculation performs as a problem of finding an interrelated point between two images. Since one of the more straightforward ways of finding those points is to calculate the brightness similarity, we prefer to use the Lucas and Kanade's (LK) optical flow concept for calculating both the optical flow and disparity between two fisheye images. This concept is the same as the one used for estimating the 2-D motion in section 2.3.2.1. To be able to describe clearly about the use of the LK's approximation for those calculations, the equation (2.18) is rewritten, as follows:

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{U} = \underbrace{-\begin{pmatrix} \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{B} \qquad (3.7).$$

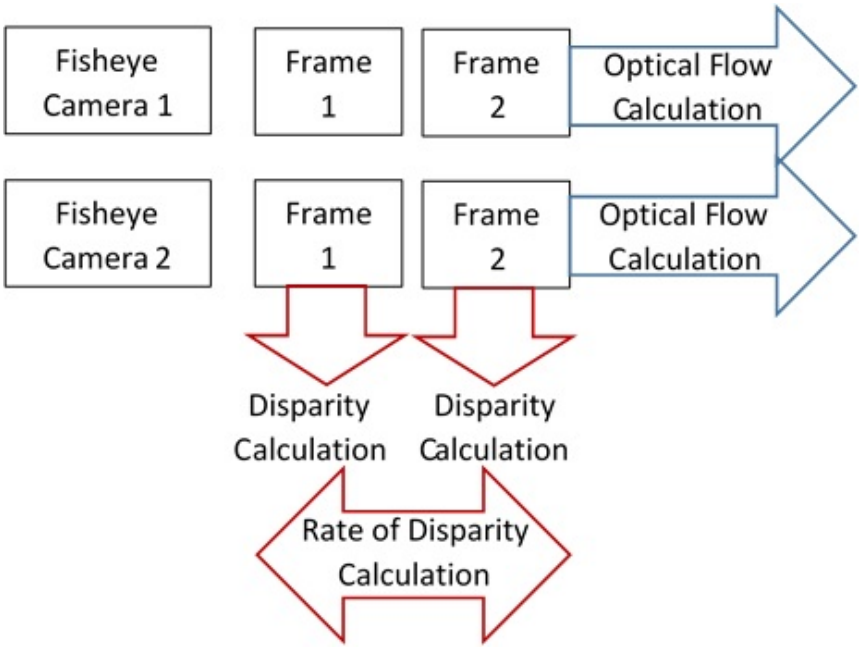


Figure 3.2. The process of calculating the optical flow and disparity between two fisheye images [44].

Note that I_x and I_y are the horizontal and vertical gradient of the fisheye image, while I_t is the image difference between two fisheye images. The number of neighboring pixels used in the calculation is denoted by *n.* In term of the optical flow calculation, the two fisheye images term means the two successive fisheye images taken from a sequential fisheye image. This calculation performs for either the left-side of sequential fisheye images or the right-side one. On the other hand, in case of the computation of disparity between two fisheye images, the two images are a pair of the fisheye images obtained from the two-side of sequential fisheye images. Additionally, for the two-side sequential fisheye images, the calculation of either the optical flow or disparity between two fisheye images performs every two successive fisheye images or two adjacent fisheye images respectively.

3.4 Experimental Setting

In this experiment, all sequential stereo fisheye images are made synthetically by using blander software [41]. By making use of this software, the two fisheye camera can be configured in an ideal stereo condition to produce some preferred synthetic fisheye images. Therefore, both qualitative and quantitative evaluation according to the generated 3-D scene flow output can be observed easily. The configuration of the two fisheye cameras with an object used for this experiment is described in Figure 3.3 [44] [45]. In that figure, two fisheye cameras are composed in a stereo configuration. The distance between the two cameras is 4 or 6 cm. The stereo fisheye camera is set up to capture an object that moves monotonically. The object is a cube with an edge of 10 cm, and it is placed in front of the stereo fisheye camera. The distance between the stereo fisheye camera and the cube is 15 cm. This cube then is moved forward 1, 2, 3, or 4 cm per step, started from a left-side position, and finished to a right-side position. The stereo fisheye camera then captures the movement of the cube (e.g., the first fisheye image shows the cube at the original position, the second fisheye image shows the cube after moving one cm, the third fisheye image displays the cube after moving two cm, etc.). From this scenario, there will be 190 fisheye images from each sequential fisheye images. Finally, the sequential fisheye images are applied to the scene flow calculation framework written in Table 3.1.

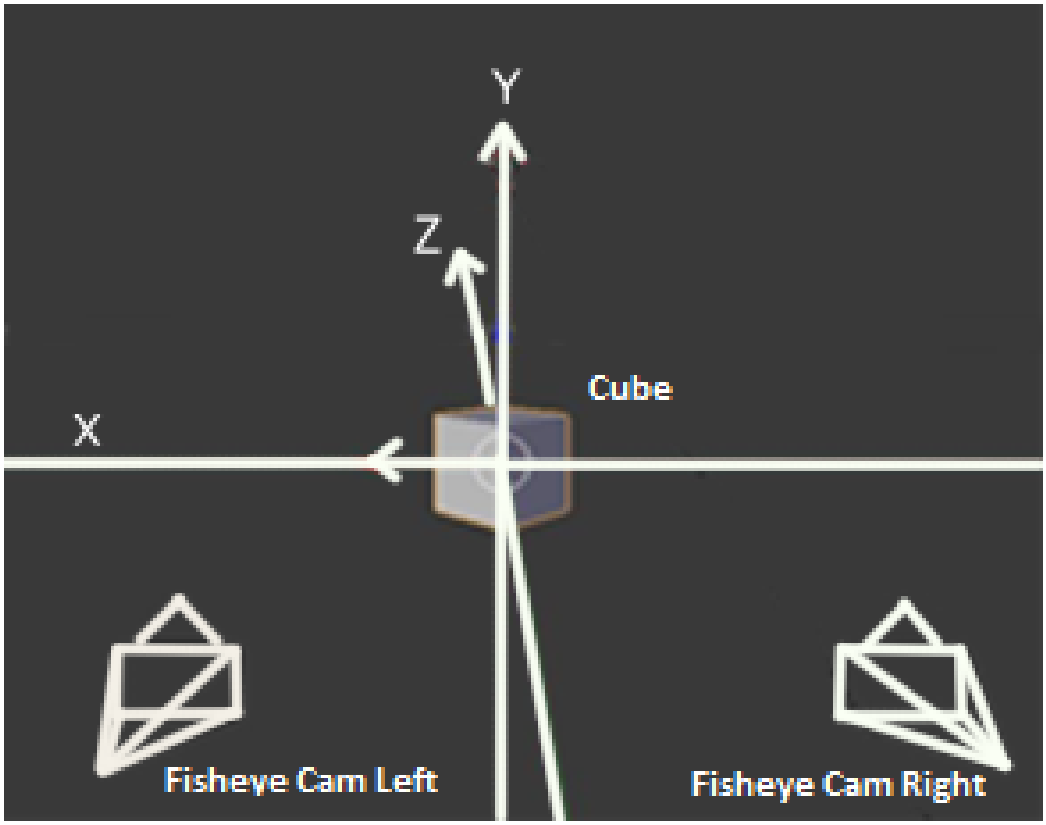


Figure 3.3. The stereo fisheye camera configuration with an object moving [44].

3.5 Research Results

In this section, the qualitative and quantitative performance of the 3-D scene flow estimation is discussed predominantly. In the qualitative assessment, there are three kinds of the position of the cube to be investigated, and they are mainly displayed in this chapter. Each of them aims to reveal the 3-D motion of cube according to the point of view of the stereo fisheye camera. The first position exploits the cube moving forward from the left side. This is also indicated that the cube is located in front of the stereo fisheye camera but on the left side. The second position reveals the cube that moves forward in the middle area. This condition means that the cube is exactly moving in front of the stereo fisheye camera. The last observe the cube moving forward to the left side, which means the cube is placed in front of the camera but on the right side.

On the other hand, the quantitative measurement presents qualitative numbers that indicate the error of the generated 3-D scene flow. Each number represents for all position based on the size of the step of cube's movement starting from the left side and finishing to

the right side. On the other hand, the quantitative measurement presents qualitative numbers that indicate the error of the generated 3-D scene flow. Each number represents for all position based on the size of the step of cube's movement starting from the left side and finishing to the right side.

Table 3.1. The 3-D scene flow algorithm.

I. ***Input****: Two successive fisheye images from either left or right the fisheye camera*
The optical flow calculation:
1: ***for*** *each two successive images,* ***do***
2: $im1_{st}$ *&* $im2_{nd}$ *←converting the two images to grayscale*
3: $im1_{st}$ *&* $im2_{nd}$ *← smoothing the two grayscale images by using the Gaussian smooth filter*
4: I_x*,* I_y *&* I_t *← obtaining spatial and temporal gradient of the images*
5: *u & v ← calculating vector flows by using matrix equation 3.7.*
6: ***end for***
7: ***Return***
II. ***Input****: The two stereo fish-eye images from either t=1 or t=2*
The disparity calculation:
1: ***for*** *each two successive images,* ***do***
2: im_{Left} *&* im_{Right} *← converting the two images to grayscale*
3: im_{Left} *&* im_{Right} *← smoothing the two grayscale images by using the Gaussian smooth filter*
4: I_x*,* I_y *&* I_t *(*im_{Left} *&* im_{Right}*) ← obtaining spatial and stereo gradient of the images*
5: *d ← calculating disparity by using equation 3.7 and then calculating rate of disparity*
6: ***end for***
7: ***Return***
III. ***Input****: The optical flow and disparity*
1: Calculating equation 3.4
2: Displaying 3D scene flow
3: Calculating AAE and SAE using equation 3.6
4: ***Return***

3.5.1 The Qualitative Performance

As we have discussed in the previous section, the first step of the 3-D scene flow estimation process is to calculate the optical flow directly from two successive fisheye images obtained from each of fisheye cameras in the stereo composition. The intake of the two successive fisheye images is also applicable for each sequential fisheye images. In Figure 3.4, those fisheye images input is presented clearly. Explicitly, in the first two rows of that figure, either (a) and (b) comprises two successive fisheye images. Each of them is obtained from left and right fisheye camera (LR and RC) respectively. These fisheye images tell the cube's position at the left side of the fisheye image area. Both (c) and (d) also come from the same sources (LR and RC), but these fisheye images present the cube's position in the middle of the fisheye image area. The rest of parts of that figure, (e) and (f) are also taken from the similar fisheye cameras; however, they show the cube's position at the right side of the fisheye image area.

From there on, by using the equation (2.18), the optical flow can be obtained, and the results can be seen in the last two rows of Figure 3.4. In this figure, (g), (h), (i), (j), (k), and (l) are the optical flow output for the consecutive images mentioned by (a), (b), (c), (d), (e), and (f) respectively. The optical flow output seems to occur densely only in the border area of the cube since there is a strong difference in intensity level between the cube and the background area. However, since the intensity around the edge of the cube (in the middle area of the cube) is almost constant, the optical flow cannot be obtained very well in this area. This phenomenon is visible clearly, especially when the cube is located in front of the stereo fisheye camera. Interestingly, when the cube is located at the left or right side position, the optical flow occurs almost the same, although it appears strongly only around the left and right edge of the cube. This condition means that there is a specific threshold of the difference in the intensity between the two areas, where the optical flow can still be generated by the system calculation. Besides, the results presented by Figure 3.4 are produced when the distance between the two fisheye cameras (the baseline) is 6 cm.

The second step is the computation of disparity between two fisheye images provided by the LC and RC. In that figure, (a) and (b) behave as a stereo fisheye image showing the cube's position at the left side of the fisheye image area, while (c) and (d) are a stereo fisheye image telling the cube in the middle position of the fisheye image area.

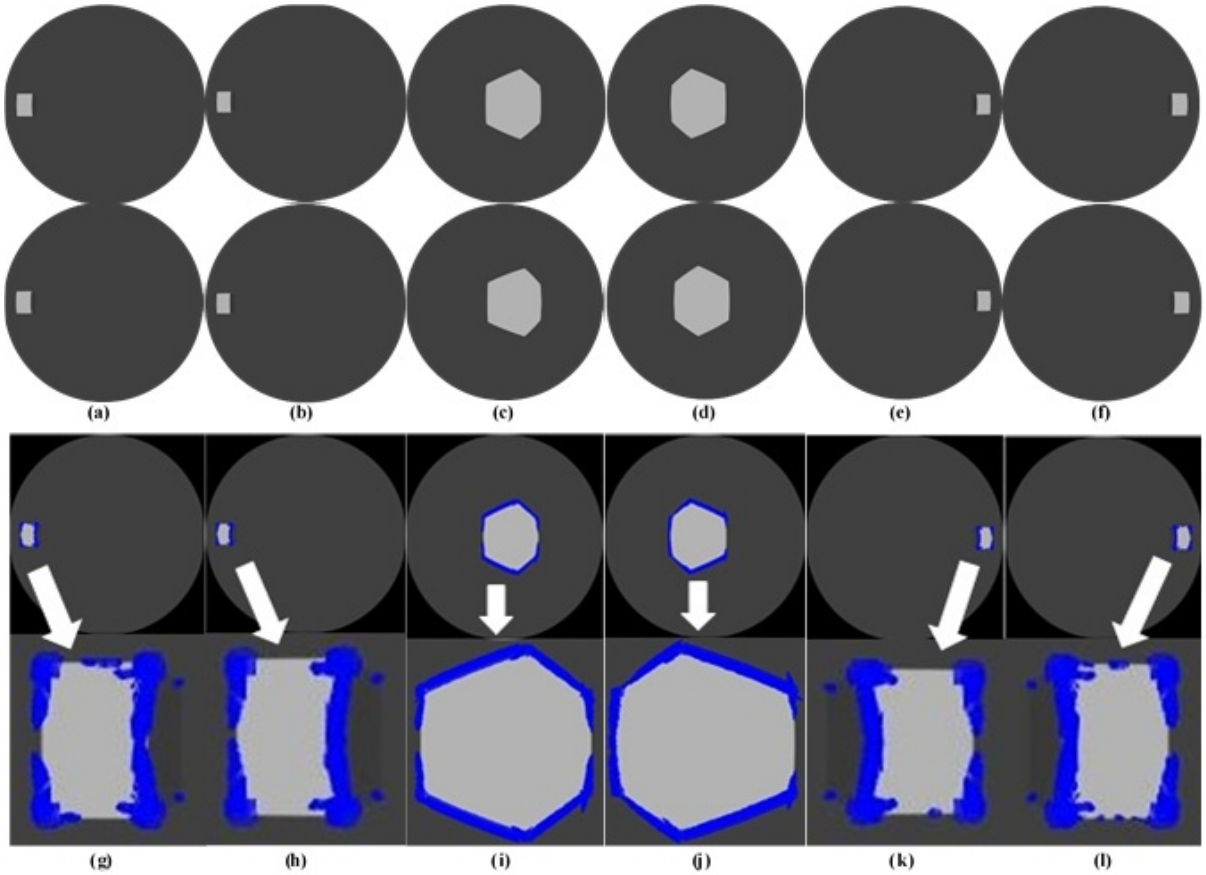


Figure 3.4. The example of two consecutive fisheye images produced by each fisheye camera. The distance between the two cameras is 6 cm. In the first two rows, (a), (c), and (e) are two consecutive fisheye images taken by the LC, while (b), (d), and (f) are two consecutive fisheye images captured by the RC. In the last two rows, (g), (h), (i), (j), (k), and (l) show the optical flow for the above consecutive images of (a), (b), (c), (d), (e), and (f) respectively.

The rest, (e) and (f) are also a stereo fisheye image presenting the cube's position at the right side of the fisheye image region. Furthermore, the disparity of each stereo fisheye image can be seen from the last two rows of Figure 3.5. In this figure, the computation of the disparity between each stereo fisheye image is composed when the distance between the two fisheye cameras is 6 cm.

The disparity of a stereo fisheye image occurs more often around the border area between the edge of the cube and the background. A consistent result can be obtained when the cube is placed in the middle side, although it is not too visible if the edge area of the cube tends to have uniform intensity. On the contrary, the disparity output obtained when the cube's position at the left side, is different with the disparity output obtained when the cube's position at the right side. It seems like there are so many false vectors appears when the cube's position at the right side.

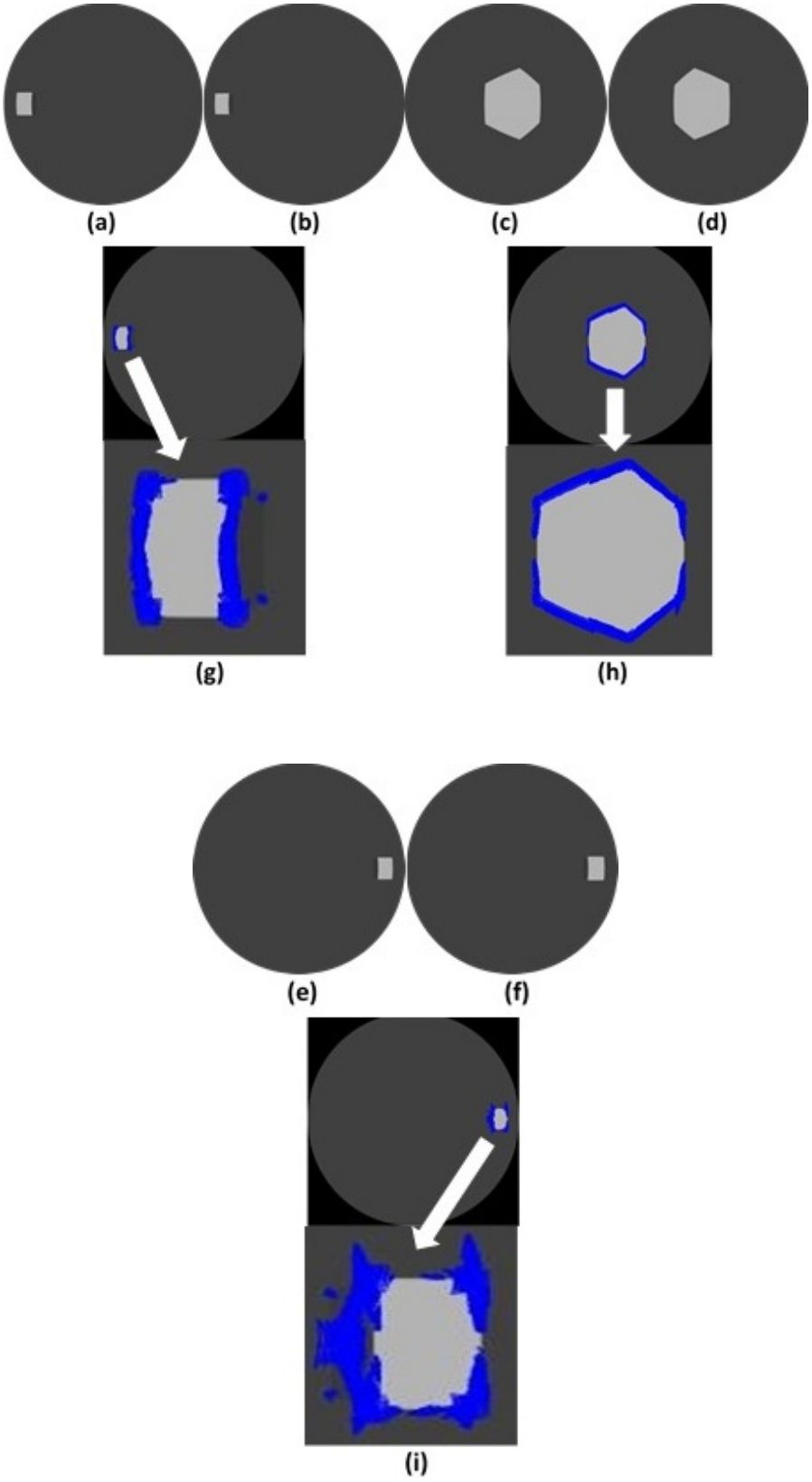


Figure 3.5. The example of the pair of fisheye images produced by the stereo camera. The distance between the two cameras is 6 cm. In the first row, (a) and (b), (c) and (d), and (e) and (f) are stereo fisheye images showing the position of the cube at the left, middle, and right side respectively. In the last two rows, (g), (h), and (i) show the disparity between the stereo fisheye images.

However, they will be ignored by the opposite one during the calculation of scene flow.

The 3-D scene flow calculation finally performs in the last step, and the results can be seen in Figure 3.6. In this figure, (a), (b), and (c) show the scene flow obtained when the cube is located at the left, center and right side position consecutively. The distance between the two fisheye cameras is 6 cm. From that figure, it can be seen that the scene flow can be generated although the cube's position in the stereo fisheye image is placed at the left, middle, or right side. However, when the stereo fisheye image shows the cube at the left or right side position, they occur more sparsely. It seems like the 3-D scene flow estimation is significantly affected by the results of optical flow and disparity calculation.

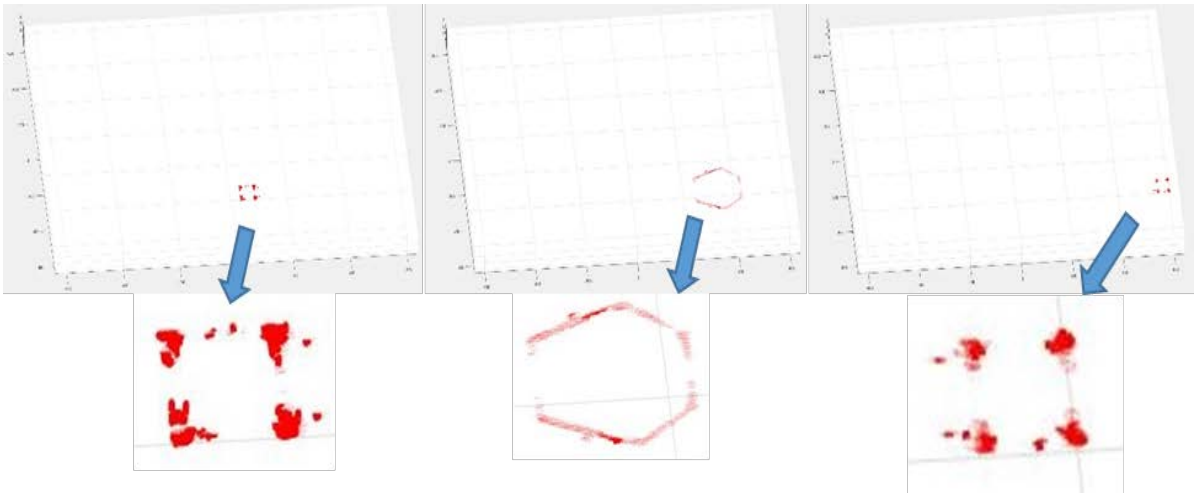


Figure 3.6. The visual performance of the 3-D scene flow according to different object location. The distance between the two cameras is 6 cm. The cube is moving in front of the stereo camera. (a), (b), and (c) show the 3-D scene flow obtained when the cube is located at the left, center, and right side respectively.

3.5.2 The Quantitative Performance

In the quantitative performance, the average angular error (AAE) and the standard deviation average angular error (SAE) of the estimated 3-D scene flow are presented in Figure 3.7, Figure 3.8, Figure 3.9, and Figure 3.10. While Figure 3.7 and Figure 3.8 show the AAE and SAE of the estimated 3-D scene flow obtained when the distance between two fisheye cameras is 4 cm, Figure 3.9 and Figure 3.10 illustrate the same quantitative evaluations, but they are obtained when the distance between the two fisheye cameras is 6 cm. In those figures, the frame number denoted by the horizontal axis also tells the cube's position in each fisheye image. It starts from 1 (left side position), and it finishes at 190

(right side position). The evaluation is applied to different types of cube's movements (1, 2, 3, and 4 cm), and different types of legend colors denote each movement.

Based on Figure 3.7 and Figure 3.8, it can be seen that the AAE and SAE are lower than around 0.1^{o} and 1.5^{o} respectively. Those conditions happen especially when the movement of the cube is 1 cm. However, the AAE and SAE increase slightly to around 0.2^{o} and 2.5^{o} consecutively, and those situations occur when the movement of the cube rises from 1 cm until 4 cm. These results are achieved when the distance between the two fisheye cameras is set up to either 4 or 6 cm, and the cube' position is located at the right or left side of the stereo fisheye images. Nevertheless, either AAE or SAE stays at higher levels, especially when the cube' position is located in the middle side of the stereo fisheye images. Specifically, when the cube's movement is only 1 cm, both AAE and SAE remains low at 0.22^{o} and 2.7^{o} respectively. This situation occurs when the distance between the two fisheye cameras is set up to either 4 or 6 cm. Nevertheless, the results become worst when the cube's movement is increased to 4 cm. The AAE and SAE rise to around 0.7^{o} and 5^{o} consecutively, particularly when the distance between the stereo fisheye cameras is set up to 4 cm. However, for the same cube's position (in the middle side), and the distance between the two fisheye cameras is set up to 6 cm, either AAE and SAE only increases to around 0.5^{o} and 4.5^{o} consecutively.

3.6 Chapter Conclusion

In this research, estimating 3-D scene flow directly from sequential stereo fisheye images can be done by incorporating the optical flow and disparity calculation simultaneously. Both of them can be calculated by using Lucas and Kanade's approximation with different composition of the fisheye images input. Based on the visual investigation, the 3-D scene flow can be estimated very well particularly in the area where the intensity difference occurs significantly, otherwise it is difficult to generate the 3-D vectors. The performance of the 3-D scene flow also depends on the motion of the object, the location of the object and the distance between the two fisheye cameras. Making further investigation regarding those aspects is essential. Therefore, the system calculation can adapt to these three aspects very well. To some extent, the results have provided a promising output.

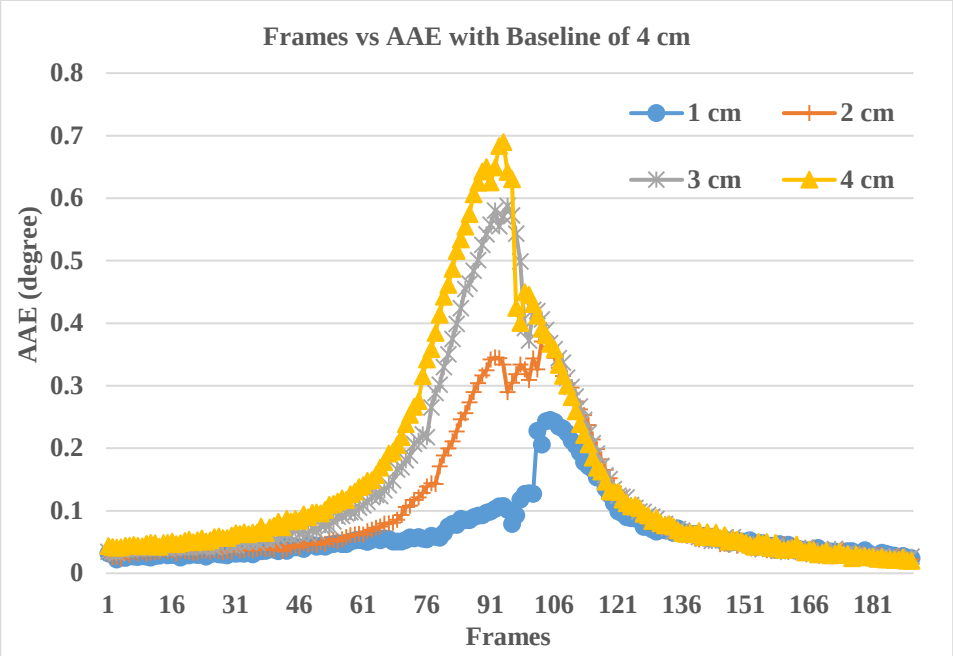


Figure 3.7. The AAE performance of the 3-D scene flow estimation. The baseline is 4 cm.

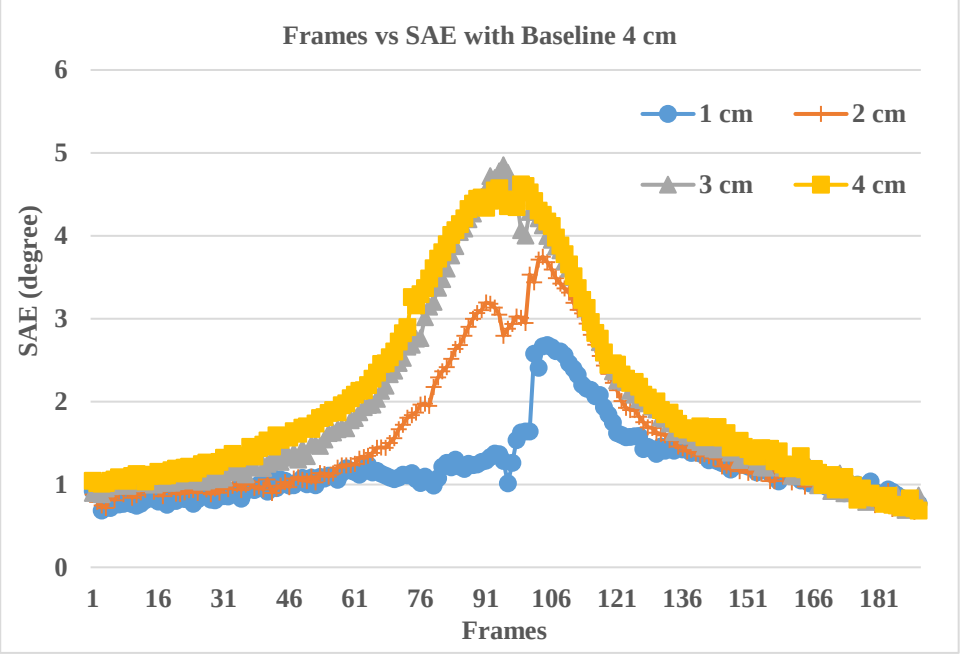


Figure 3.8. The SAE performance of the 3-D scene flow estimation. The baseline is 4 cm.

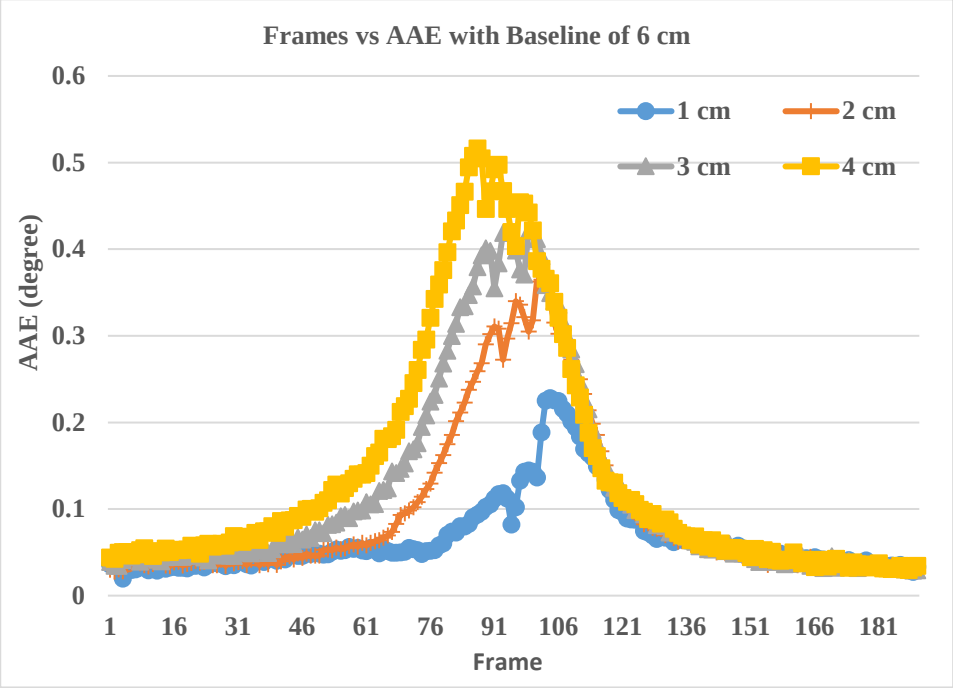


Figure 3.9. The AAE performance of the 3-D scene flow estimation. The baseline is 6 cm.

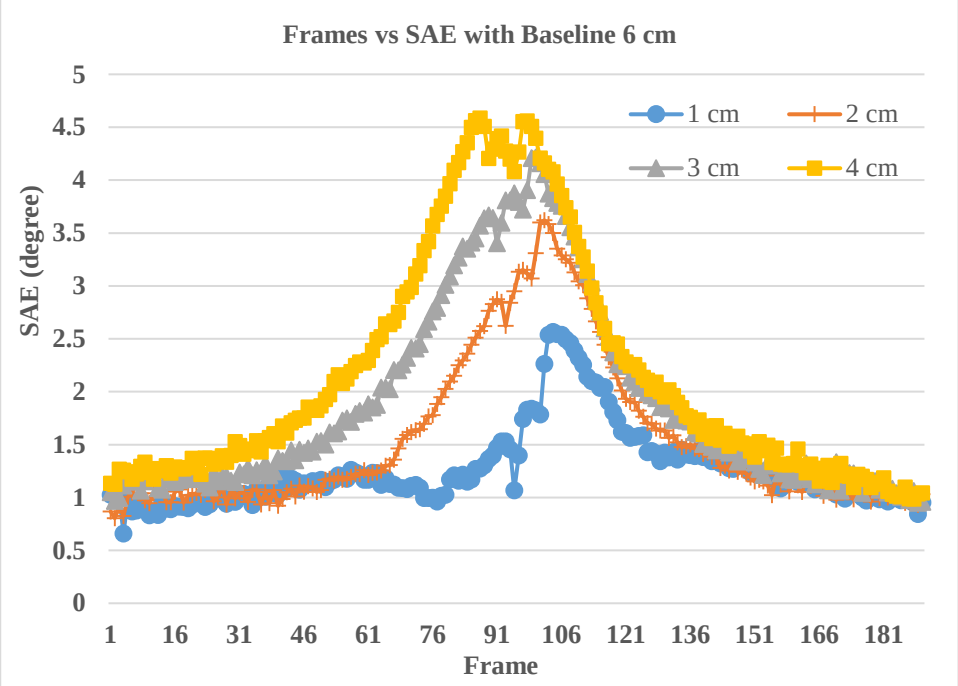


Figure 3.10. The SAE performance of the 3-D scene flow estimation. The baseline is 6 cm.

Chapter 4

Conclusion and Further Research

4.1 Conclusion

In this dissertation, innovative developments of motion estimation (ME) for sequential fisheye images has been discussed comprehensively. The development consists of the 2-D ME and 3-D ME. Both of them are developed by extending the optical flow concept. The 2-D ME is designed for estimating the 2-D motion directly from a sequential fisheye image. The 3-D ME is built for estimating the 3-D scene flow directly from sequential stereo fisheye images. By conducting this development, we will have a great possibility of implementing advanced visual applications with larger field-of-view (FoV) than the conventional visual application do.

In the 2-D ME development, we have succeeded to make use of the Lucas and Kanade's (LK) optical flow scheme as a fundamental methodology to estimate motion directly from sequential fisheye images. In the fisheye image circumstance, it turns out that this method works very efficient. This is because the basic approximation is very simple, and it can be exploited further. Although the initial approximation tends to be difficult to overcome a complex motion around the center area of the fisheye image, it can be exploited to work many times under the supervision of a parametric performance evaluation. This repetitive calculation scheme guides the estimated vector headed toward the right direction. The extended version of the LK's approach named "LKI", which is Lucas and Kanade's concept with an improvement has been proven working properly so far to overcome a massive distortion around the center area of the fisheye image. Therefore, the performance of the estimated motion vectors throughout the fisheye image area remains the same at a high level.

Another ME development written in this dissertation is the 3-D ME development. Sensible utilization of the fundamental LK's concept for supporting the 3-D scene flow estimation has been conducted successfully in this development. The conventional LK's approach has been proven to be an effective method for calculating not only the optical flow but also the disparity between stereo fisheye images. Hence, our 3-D scene flow estimation framework that employs the optical flow and disparity calculations can produce a promising result. The proposed scheme can generate the 3-D scene flow directly from a

stereo fisheye camera. This condition is applicable wherever the moving object occurs in the fisheye image area. In other words, our approach can reveal the 3-D motion captured from 180^0 FoV.

Nevertheless, the above achievements can be realized in certain constraints. These constraints firstly assume that there is only a small camera motion. Secondly, there is no camera rotation. The third one is that the processing circumstance is in a maximum FoV (180^0). The last is that the stereo FoV is captured in the same period by each fisheye camera.

4.2 Further Research

There are some aspects that can be exploited further from the current results. In the research of the 2-D optical flow-based ME, at least there are two aspects that still become questions to be answered. Firstly, when the system calculation is used to estimate a sudden movement or a huge movement, and particularly, the movement of the object occurs on the center area of the fisheye image area, it cannot predict that motion very well. This condition might be equal to the problem to determine the motion vectors directly from almost five fisheye image differences in a sequential fisheye image. Solving this problem will be useful for making a video frame rate conversion or a motion analysis application. Secondly, although the optical flow can be used in every preferred FoV (or every different focal length), it is essential to investigate the motion in transition between two different FoV. In this condition, the shape of the object seems to change dramatically. By revealing the 2-D motion in this transition, it will give a smooth effect to the vision.

In the 3-D development, on the other hand, it seems like the main problem is to obtain true 3-D motion vectors from an area, in which a feature cannot be defined very well. Extending the LK's approach might be an interesting challenge to be exploited. Therefore, the 3-D scene flow can be obtained for the entire moving object. Another question to be answered is to obtain the 3-D scene flow from different FoV of the stereo fisheye camera. This condition will lead the same question regarding motion in the transition FoV, but now it is related to the view effect in the generated 3-D domain.

Appendix A

Bibliography

[1] A. C. Bovik, The Essential Guide to Video Processing Sourcebook, London: Elsevier Inc., Jul. 2009.

[2] D. A. Forsyth and J. Ponce, Computer Vision: A Modern Approach, 2nd edition, California: Pearson, May. 2012.

[3] Z. Yan and X. Xiang, "Scene flow estimation: A Survey," CoRR, abs/1612.02590, New York, Dec. 2016.

[4] D. Scaramuzza, "Omnidirectional Camera," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed., New York, Springer, pp. 552-560, Feb. 2016.

[5] C. Zuo, Y. Liu, Y. Li, W. Xu and M. Zhang, "A distortion correction algorithm for fish-eye panoramic image of master-slave visual surveillance system," *Optical Review*, vol. 20, no. 5, pp. 367-373, Sep. 2013.

[6] C. Hughes, M. Glavin, E. Jones and P. Denny, "Review of geometric distortion compensation in fish-eye cameras," in *IET Irish Signals and Systems Conference (ISSC 2008)*, Galway, Jun. 2008.

[7] S. Abraham and W. Forstner, "Fish-eye-stereo calibration and epipolar rectification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 5, pp. 278-288, Aug. 2005.

[8] J. Courbon, Y. Mezouar, L. Eckt and P. Martinet, "A generic fisheye camera model for robotic applications," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, Oct. 2007.

[9] N. Asuni and A. Giachetti, "TESTIMAGES: A Large-Scale Archive for Testing Visual Devices and Basic Image Processing Algorithms," 22-23 Sep. 2014. [Online]. Available: http://testimages.tecnick.com or www.lms.lnt.de/fisheyedataset. [Accessed 14 Feb. 2018].

[10] E. Schwalbe, "Geometric modelling and calibration of fisheye lens camera systems," in *Proceedings 2nd Panoramic Photogrammetry Workshop, Int. Archives of Photogrammetry and Remote Sensing*, Berlin, Feb. 2005.

[11] J. Kannala and S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335-1340, Aug. 2006.

[12] X. Ying and Z. Hu, "Can We consider central catadioptric cameras and fisheye cameras within a unified imaging model," in *Computer Vision - ECCV 2004*, Prague, May. 2004.

[13] C. Hughes, P. Denny, E. Jones and M. Glavin, "Accuracy of fish-eye lens models," *Applied Optics,* vol. 49, no. 17, pp. 3338-3347, Jun. 2010.

[14] F. Devernay and O. Faugeras, "Straight lines have to be straight: automatic calibration and removal of distortion from scenes of structured enviroments," *Machine Vision and Applications,* vol. 13, no. 1, pp. 14-24, Jun. 2001.

[15] A. Basu and S. Licardie, "Alternative models for fish-eye lenses," *Pattern Recognition Letters,* vol. 16, no. 4, pp. 433-441, Apr. 1995.

[16] J. Casey, "An investigation of block searching algorithms for video frame codecs; Master's dissertation," Dublin Institute of Technology, Dublin, Jun. 2008.

[17] A. Furnari, G. M. Farinella, A. R. Bruna and S. Battiato, "Affine covariant features for fisheye distortion local modeling," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 696 - 710, Feb. 2017.

[18] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *2010 International Conference on Pattern Recognition*, Istanbul, Sep. 2010.

[19] I. Bauermann, M. Mielke and E. Steinbach, "H.264 based coding of omnidirectional video," in *Computer Vision and Graphics*, Warsaw, Springer, pp. 209-215, Sep. 2006.

[20] Z. Jiali, Z. Yongdong, S. Yanfei and N. Guangnan, "Panoramic video coding using affine motion compensated prediction," in *Multimedia Content Analysis and Mining*, Berlin, Springer, pp. 112-121, July. 2007.

[21] A. Djamal, A. Zohra and K. Djemaa, "An adapted block-matching method for optical flow estimation in catadioptric images," in *International Conference on Multimedia Computing and Systems (ICMCS 2014)*, Marrakesh, Aug. 2014.

[22] D. Fortun, P. Bouthemy and C. Kervrann, "Optical flow modeling and computation: A survey," *Computer Vision and Image Understanding*, vol. 134, pp. 1-21, May. 2015.

[23] A. S. Satyawan, J. Hara and H. Watanabe, "Motion estimation on fish-eye images using modified motion model," in *FIT 2017*, Tokyo, Sep. 2017.

[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI '81*, Vancouver, Apr. 1981.

[25] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221-255, Feb. 2004.

[26] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-2, pp. 81-87, Apr. 1981.

[27] A. Radgui, C. Demonceaux, E. Mouaddib, D. Aboutajdine and M. Rziza, "An adapted Lucas-Kanade's method for optical flow estimation in catadioptric images," in *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS*, Marseille, Oct. 2008.

[28] D. Alouache, Z. Ameur and D. Kachi, "An adapted block-matching method for optical flow estimation in catadioptric images," in *ICMCS 2014*, Marrakesh, Apr. 2014.

[29] D. Salomon, Transformation and Projections in Computer Graphics, Springer-Verlag: London, Mar. 2006.

[30] 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*, vol. 12, no. 4, pp. 395-408, Apr. 2003.

[31] A. S. Satyawan, J. Hara and H. Watanabe, "Automatic self-improvement scheme in optical flow-based motion estimation for sequential fisheye images," *ITE Transactions on Media Technology and Applications*, vol. 7, pp. 20-35, Jan. 2019.

[32] "RICOH THETA S," Ricoh Company, Ltd, December 2017. [Online]. Available: https://theta360.com/en/about/theta/s.html. [Accessed 9 September 2018].

[33] A. S. Satyawan, J. Hara and H. Watanabe, "An improvement of motion estimation on sequential fisheye images," in *IEICE General Conference*, Tokyo, Mar. 2018.

[34] A. S. Satyawan, J. Hara and H. Watanabe, "Motion estimation on fish-eye images," in *IEICE General Conference*, Nagoya, Mar. 2017.

[35] A. S. Satyawan, J. Hara and H. Watanabe, "The performance of self-enhancement method in optical flow-fased motion estimation for sequential fisheye images with sudden movement of the objects," in *IEICE Society*, Kanazawa, Sep. 2018.

[36] A. Wedel and D. Cremers, Stereo Scene Flow for 3D Motion Analysis, London: Springer, Jan. 2011.

[37] M. Manze and A. Geiger, "Object scene flow for autonomous vehicles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Jun. 2015.

[38] L. Zhang, B. Curless and S. M. Seitz, "Spacetime stereo: shape recovery for dynamic scenes," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Madison, Jun. 2003.

[39] S. Vedula, P. Rander, R. Collins and T. Kanade, "Three-dimensional scene flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 475-480, Mar. 2005.

[40] R. Li and S. Sclaroff, "Multi-scale 3D scene flow from binocular stereo sequences," *Computer Vision and Image Understanding*, vol. 110, no. 1, pp. 75-90, Apr. 2008.

[41] "blender," Open source 3D creation, [Online]. Available: https://www.blender.org/about/. [Accessed 7 January 2017].

[42] N. Sharmin and R. Brad, "Optimal filter estimation for Lucas-Kanade optical flow," *Sensor*, vol. 12, no. 9, pp. 12694-12709, Sep. 2012.

[43] J. Quiroga, T. Brox, F. Devernay and J. Crowley, "Dense semi-rigid scene flow estimation from RGBD images," in *Computer Vision – ECCV 2014*, Zurich, Sep. 2014.

[44] A. S. Satyawan, J. Hara and H. Watanabe, "Scene flow estimation from sequential stereo fisheye images: Preliminary study," in *International Conference on Signals and Systems (ICSigSys) 2018*, Bali, May. 2018.

[45] A. S. Satyawan, J. Hara and H. Watanabe, "Scene flow from stereo fisheye images," in *International Workshop on Advanced Image Technology (IWAIT) 2018*, Chiang Mai, Jan. 2018.

Appendix B

List of Publication

A. International Journal

[1] A. S. Satyawan, J. Hara, and H. Watanabe, “Automatic self-improvement scheme in optical flow-based motion estimation for sequential fisheye Images,” ITE Transactions on Media Technology and Applications, Vol. 7, No. 1, pp. 20-35, Jan. 2019.

B. International Proceeding

[1] A. S. Satyawan, J. Hara and H. Watanabe, "Scene flow estimation from sequential stereo fisheye images: Preliminary study," in *International Conference on Signals and Systems* (*ICSigSys*) *2018*, Bali, No. 20, pp. 109-114, May. 2018.

[2] A. S. Satyawan, J. Hara and H. Watanabe, "Scene flow from stereo fisheye images," in *International Workshop on Advanced Image Technology* (*IWAIT*) *2018*, Chiang Mai, No. 89, pp. 1-4, Jan. 2018.

C. Domestic Proceeding

[1] A. S. Satyawan, J. Hara and H. Watanabe, "The performance of self-enhancement method in optical flow-fased motion estimation for sequential fisheye images with sudden movement of the objects," in *IEICE Society Conference*, BS-7-8, Kanazawa, Sep. 2018.

[2] A. S. Satyawan, J. Hara and H. Watanabe, "An improvement of motion estimation on sequential fisheye images," in *IEICE General Conference*, BS-2-12, Tokyo, Mar. 2018.

[3] A. S. Satyawan, J. Hara and H. Watanabe, "Motion estimation on fish-eye images," in *IEICE General Conference*, BS-1-9, Nagoya, Mar. 2017.

[4] A. S. Satyawan, J. Hara and H. Watanabe, "Motion estimation on fish-eye images using modified motion model," in *FIT 2017Conference*, H-001, Tokyo, Sep. 2017.