International Journal on Advanced Science Engineering Information Technology

Vol.12 (2022) No. 1
ISSN: 2088-5334

An Elastic Frame Rate Up-Conversion for Sequential Omnidirectional Images

Arief Suryadi Satyawan [a,c,*], Salita Ulitia Prini [a], Syed Abdul Rahman Abu-Bakar [b], Yusuf Nur Wijayanto [a]

[a] *Research Center for Electronics and Telecommunication, Indonesian Institute of Sciences, Bandung 40135, Indonesia*
[b] *School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor, Malaysia*
[c] *Department of Electrical Engineering, Nurtanio University, Bandung 40174, Indonesia*
Corresponding author: [*]arie008@lipi.go.id; arief.suryadi@akane.waseda.jp

***Abstract*— A distinct innovation in the development of frame rate up-conversion for an omnidirectional video is presented in this paper. The innovation allows an omnidirectional video to decide the number of omnidirectional frames that the main algorithm can create based on the characteristic of the motion of objects in the omnidirectional video itself. This elastic omnidirectional frame production is achievable since the main algorithm works based on an optical flow method with a self-improvement mechanism. The optical flow concept has proven to be adaptable greatly with the character of such a video. The experiment results, in which ten omnidirectional videos were involved, show that the algorithm's exertion and adaptability named the elastic frame rate up-conversion (E-FRUC) are incredible. This achievement is confirmed by both the excellent visual quality and the high score of the PSNR (33 to 59 dB) of the reconstructed omnidirectional frame (RF). The E-FRUC can generate at least one RF from two consecutive omnidirectional images, although these images are synthetically designed with an ideal motion. The more complex the motion objects condition inside the omnidirectional video is, the more RFs the E-FRUC can create. By using the E-FRUC, the continuity of motion of moving objects can be preserved. Therefore it can be useful to maintain the frame rate of incomplete omnidirectional video frames when such video is played back. Such a condition usually happens when the omnidirectional video is transmitted through error-prone telecommunication networks.**

***Keywords*— Frame rate up-conversion; elasticity; omnidirectional video; optical flow.**

Manuscript received 18 Aug. 2020; revised 12 Dec. 2020; accepted 19 Apr. 2021. Date of publication 28 Feb. 2022.

I. INTRODUCTION

An incredible escalation of the utilization of visual applications has boosted a dramatic increase in the volume of video and image data. This situation is affected not only by the rapid rise in the development of visual capture devices but also by the impressive improvement in the visual quality of display equipment. Besides, either the high-resolution camera facility or high-performance display unit is now available in a piece of portable telecommunication equipment. It facilitates people to collect or exchange visual information more often through a telecommunication network anytime and anywhere [1]. This condition probably works well for the time being. However, those vast data will soon overload the bandwidth capacity of our telecommunication infrastructure, and at the same time, it will reduce the show sensation dramatically [2]. Therefore, some engineering methods are expected to anticipate or even solve those problems from some different points of view.

One undesirable effect from the above scenario is the possibility of playing back a video that consists of several incomplete video frames. This impact will occur if an application cannot retrieve a complete set of video frames from a source due to the busy network routers. The design of a state-of-the-art video codec compresses video information by eliminating temporal redundancy between two consecutive video frames; therefore, the volume of the video data decreases forcefully [3], [4].

In essence, this process builds predictor frames sent together with their reference by the encoder. The arrangement of coded video frames named as a group of coded frames usually consists of the reference frame and some predictor frames. The predictor frames can either follow or proceed with the reference one, and this composition depends on the direction of motion estimation. There would be several groups of the coded frame from complete video data. After the

encoder has sent the group of the coded frame, it has to arrive at the decoder side successfully so that the original video frames can be regenerated flawlessly. The reference frame is more robust than the predictor frame because it is decodable independently without the help of any predictor frames.

Contrarily, the predictor frame needs a decoded reference frame closely related to it to regenerate a video frame that resembles the original one. When a network resource is unoccupied, a video encoder can release the maximum number of predictor frames, and the network will allow them to go through. In contrast, when it becomes too busy, either the encoder or the network will sacrifice the predictor frames to the reference frame pass through. A coded frame-group seems to be very reliable because it can adapt to the network behavior. Nevertheless, if such a fully-occupied network happens more frequently, there will be a period in which only several decoded reference frames exist on the decoder side [4]. If this consecutive decoded reference frame has to be playback at the same rate as the original video, the motion displayed will not be as smooth as the original video due to a lack of reconstructed frames between the two decoded reference frames. Therefore, we need to rigorously maintain the number of frames per second at the decoder side to playback in typical visual reception or even more. This effort is useful for a decoder with the above scenario and helpful for a decoder that retrieves an encoded video from different types of digital media storage.

One potential solution to overcome the above problem is to develop a frame rate up-conversion (FRUC) [5], [6]. This mechanism will endeavor to rebuild an incomplete video as if it looks like the original form or increase the smoothness of motion on a video display. The FRUC is not new and in fact has been in used in the past. To the best of our knowledge, at least there are two types of FRUC in place [7], [8]. The first type makes use of the non-motion-compensated (NMC) scheme, while the second one utilizes the motion-compensated (MC) concept. The NMC is more straightforward than the MC; however, the performance of the MC is more satisfactory than the NMC. The NMC only tries to produce a repetitive frame, while the MC relies on the motion estimation (ME) technique to present accurate motion so that the MC offers good quality reconstructed frame and strong immunity against artifacts. Frame repetition, central weighted median, and linear averaging are examples of the NMC concept [5]–[9], while MC fetching, MC shifting, and MC linear averaging are three types of the MC method [10]–[14].

Although the state-of-the-art FRUC concepts provided a better temporal resolution, these methods have been proven successfully only to serve traditional video, a kind of video obtained from general perspective video cameras. The performance of these methods has yet to be tested in the case of videos taken from omnidirectional cameras. In comparison to the conventional video, image frames inside this omnidirectional video suffer a lot of distortion [15]–[21]. The most significant distortion comes from the radial distortion, and it is discernible clearly by human vision. As depicted in Fig. 1 [21], the shape of a moving object, a cube in the first row (from *a* to *c*) and a train in the second row (from *d* to *f*), looks to be different while the omnidirectional camera captured and projected them on three different locations of the omnidirectional image (at the left side, at the center, and the right side). Even though the cube images are synthetic omnidirectional images, while the train images are the real ones, they perform identical behavior. Moving objects captured at the peripheral location (*a*, *c*, *d,* and *f*) look smaller than that of the central position (*b* and *e*).

Moreover, in the central area, the shape of the object deformed massively. This is due to the construction of the omnidirectional lens that causes such inconsistent deformation [15], [17], and [21]. For example, Fig. 1, shows a hemisphere phenomenon captured by one lens contained in the omnidirectional camera.

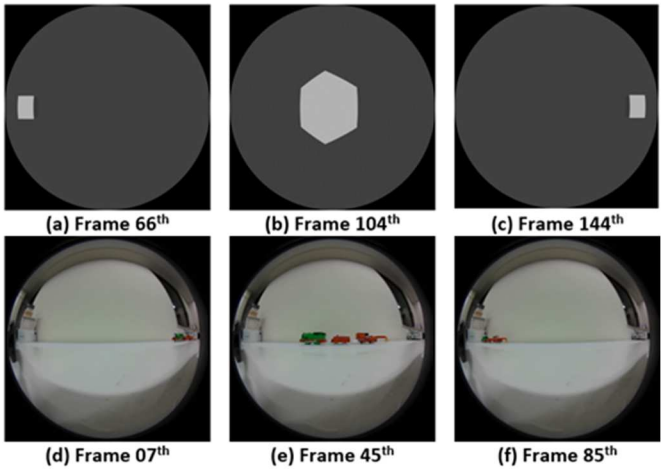


Fig. 1 Two sets of a sample of omnidirectional images [15].

The rays coming from the hemisphere phenomenon are distributed massively at the center field of the image but not in the peripheral area. This condition causes irregular motion as well. Any object that appears nearby the peripheral area of an omnidirectional image tends to move slower than when it is closer to the center area.

Based on the above anomaly, this research focuses on developing a FRUC method for omnidirectional video, using the MC scheme. Since the FRUC has to adapt very well to the characteristic motion occurring in such a video, it is still challenging to preserve block-based ME. Therefore, we propose a design that utilizes an optical flow-based ME. We assume that pixel-wise calculation is more adaptable to the typical motion in the omnidirectional video than computation among group of pixels. The results we obtained show the validity of our arguments.

We found that the automatic self-improvement mechanism in the optical flow-based ME algorithm (ASI-OME) is very suitable for estimating motion directly from sequential fisheye images [21]. Furthermore, the benefit of the mechanism can be used principally for the FRUC. Hence, in this research, the central concept of the FRUC for the omnidirectional video comes from that scheme. Interestingly, such an automatic method is very flexible to produce some intermediate image frames from two successive omnidirectional images that have unique characteristics, such as inconsistent motion, unstable intensity, or another additional noisy texture. As a result, the FRUC for the omnidirectional video can be generated elastically depending on the unique characteristics of the video.

II. Material and Method

This section will describe our innovative method proposed to generate synthetic images elastically from omnidirectional images.

A. The Elastic Frame Rate Up-Conversion (E-FRUC)

The elastic frame rate up-conversion employs the method proposed in [21]. However, we do not apply it only to find the best motion vectors (MV) for estimating motion from omnidirectional video. Our purpose is to use the automatic ME calculation to generate several omnidirectional images from two successive omnidirectional images, and they were played back in between these two images. As a result, the frame rate of the original omnidirectional video will increase proportionally to the number of omnidirectional images that have been generated.

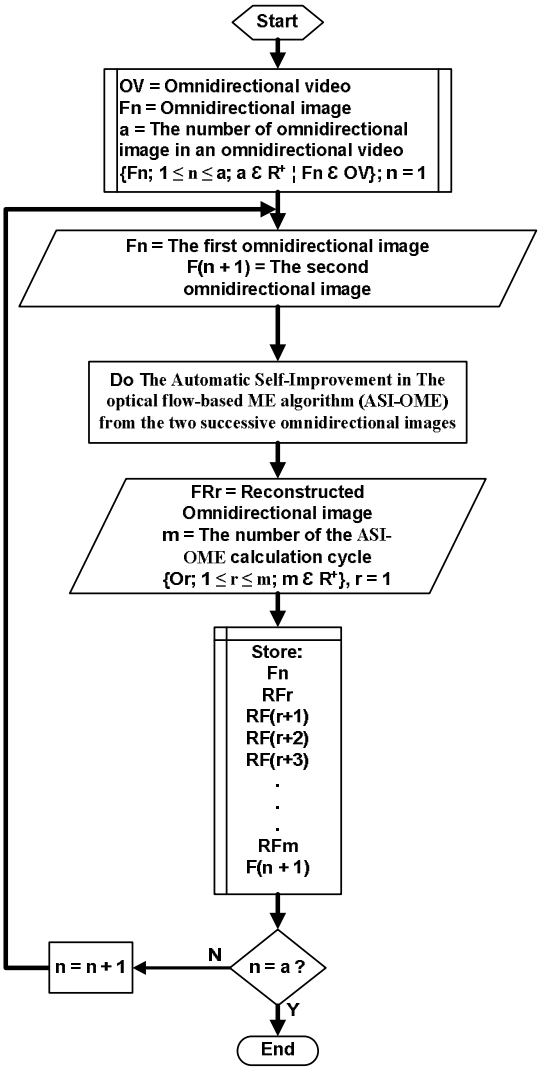


Fig. 2 The flow of the E-FRUC.

The primary process of the proposed FRUC follows the flow chart given in Fig. 2. At the top of the process, the process begins with upgrading the frame rate of the omnidirectional video (OV). This video is decomposed into several omnidirectional images arranged consecutively $\{F_n \mid F_1, F_2, F_3 \ldots F_a \mid a \in R^+\}$. Subsequently, every pair of successive omnidirectional images (F_1 with F_2, F_2 with F_3, F_3 with F_4…, F_{a-1} with F_a) is applied to the ASI-OME algorithm so that a number of the reconstructed omnidirectional images $\{RF_r \mid RF_1, RF_2, RF_3 \ldots RF_m\}$ can be obtained. These reconstructed omnidirectional images will stay between the two omnidirectional images input. The same process was available for the next two consecutive omnidirectional images until the last frame in the omnidirectional video is reached. Since the ASI-OME algorithm has performed well in getting MV from two successive omnidirectional images by computing an automatic motion vector trajectory mechanism, the reconstructed omnidirectional images will also exhibit almost actual motion.

The elastic term of the E-FRUC means that the number of RF generated between two consecutive omnidirectional images (F_n's) depends on the number of calculation cycles (m). Since m is undefined in a specific amount by the algorithm, there were different numbers of m for each consecutive Fn's input, although these omnidirectional images are members of the same omnidirectional video input.

B. Review of The Automatic Self-Improvement in the Optical flow-based Motion Estimation (ASI-OME)

The ASI-OME was developed by using the optical flow calculation scheme to produce high-quality MV from two consecutive omnidirectional images. One of these images acts as a reference image, while the other behaves as a departure image that assigns the MV to move toward the reference. The calculation scheme employs the Lucas-Kanade (LK) concept, but there is an additional improvement mechanism. The improvement scheme is to ensure that the quality of an aggregate of MV is well preserved by conducting repetitively the LK's calculation in stages. Each stage performs an LK's computation to produce MV, and it has to be evaluated by using a quantitative error measurement such as the peak-to-peak-signal-to-noise-ratio (PSNR).

In this evaluation process, the MV is transformed firstly into a reconstructed omnidirectional image before comparing it with the reference omnidirectional image. Consequently, a quantitative error can be calculated. If the number of PSNR is higher than the previous or initial one, the reconstructed omnidirectional image at the current stage can be used together with the reference image for the LK's computation in the following stage to obtain a new MV. What follows then, the same procedure starting from finding the new MV using the LK's calculation, followed by reconstructing the new omnidirectional making use of the new MV, and evaluating it using the PSNR mechanism, was applied to the new stage.

Once again, if the number of the new PSNR is higher than the previous one, there was another stage created. Otherwise, the ASI-OME algorithm will stop, and the aggregate of MV can be measured by accumulating the MV from each stage. As we mention in the proposed method, we utilize the ASI-OME process to obtain consecutive reconstructed omnidirectional images between the two successive omnidirectional images fed into the ASI-OME algorithm.

C. Configuration of the Lucas-Kanade (LK) Concept

LK's concept has been used to calculate MV from two successive general perspective images with an overwhelming performance [22]. The idea comes from the assumption that the intensity of a particular pixel in the first image remains

constant in the following one, although they are in a different location because of motion. The MV can be obtained using a linear approximation expressing the direct connection between those pixels. The simple equation for describing the linear approximation then can be obtained using the least square approach, which can be expressed as follows:

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

where I_x and I_y are the horizontal and vertical spatial derivatives of the pixels being observed in the first frame, respectively; I_t is derivative across the two image frames; U (u and v) is the desired MV, and n is a window determining the number of neighboring pixels involved to obtain a solution for the U.

The key to the success of the LK's scheme comes from maintaining the matrix given by the first equation to be solvable in a particular condition. That is to say, the solution for U occurs if matrix A has a proper invertible (i.e. A^TA is invertible). Moreover, the eigenvalues (λ_1 and λ_2) of the A^TA is not too small enough. To achieve that, we need to apply appropriate window size and at the same time, provide a minimum threshold for the eigenvalues that still maintain high-quality MV. Based on the modesty of the above expression, the LK's approximation also provides flexibility in expansion or modification. In [21], the original LK can be enhanced to serve omnidirectional images with high performance.

III. Results and Discussion

This section discusses experimental results that started from procedures applied for taking experimentation until the evaluation of results performance by using either the quantitative or qualitative judgments.

A. Experimental Procedure

The process flow of the experiment is described in Fig. 3. The objective of this experiment is to generate multiple reconstructed omnidirectional images (RF_1 to RF_m) between two consecutive omnidirectional images (F_1 to F_n) using E-FRUC. This process is applicable for whole successive omnidirectional images within an input of the omnidirectional video (OV). Hence, the frame rate of the video can be increased appropriately. To realize the proposed method (E-FRUC) described in Fig. 2, MATLAB 2016 programming language is adopted [23]. Several simulation processes were carried out for evaluating the qualitative and quantitative performances of the proposed method.

B. The Input of Omnidirectional Video

We used ten omnidirectional video sequences in this experiment. The characteristic of each video is described in Table 1. Two omnidirectional videos, Cube Sequence (CU) and Flowers Sequence (FU), were made synthetically. The Cube sequence presents a single object (cube) which was created using Blender [24], while the Flowers sequence, which shows multiple moving objects (flowers), was downloaded from [25]. The rest, eight omnidirectional videos, were obtained using Ricoh Theta S camera [26]. Five real omnidirectional videos; Hand Sequence (HS), Man Sequence (MS), Car-I Sequence (CS-I), Car-II Sequence (CS-II), and Train Sequence (TS), show a single object moving around the area of the video or image, while the remaining three videos; Traffic-I Sequence (TRS-I), Traffic-II Sequence (TRS-II), and People Sequence (PS), consist of multiple moving objects.

C. Evaluation Performance

We employed two performance metrics in this work. The first is peak-to-peak-signal-to-noise-ratio (PSNR). This is to provide a quantitative evaluation performance that compares the similarity between the image being observed (A) and the reference image (Â). Eqs. (2) and (3) are used for this purpose.

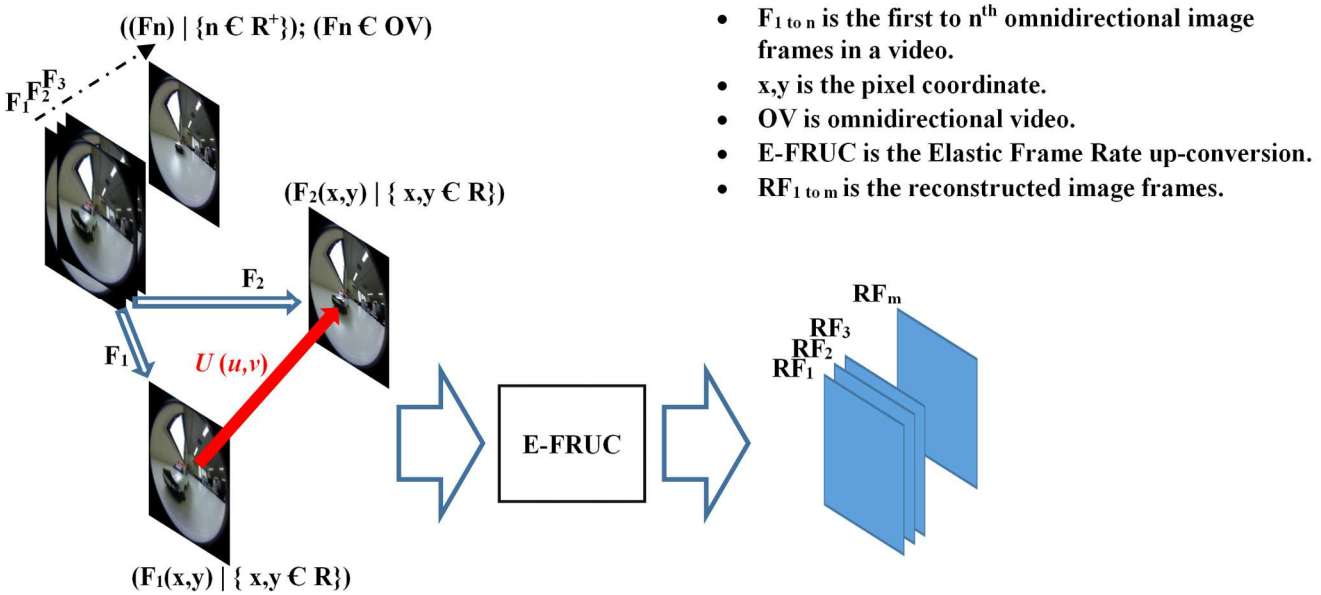


Fig. 3 The experimental process for an omnidirectional video

TABLE I
OMNIDIRECTIONAL VIDEO SEQUENCES [21]

The Name of Sequential Omnidirectional Image Frames	Types of Videos	The Number of Omnidirectional Image Frames	Omnidirectional Video Resolution	The Characteristic of Information
Cube Sequence (CS)	Synthetic	211	640x640	A cube moves from left to right side with constant speed.
Flowers Sequence (FS)	Synthetic	120	1088x1088	Flowers blossom freely.
Hand Sequence (HS)	Real object	153	640x640	A hand moves from left to right side with a constant speed.
Man Sequence (MS)	Real object	94	640x640	A man sits in front of the fisheye camera and moves his hands in the opposite direction slowly.
Car-I Sequence (CS-I)	Real object	16	960x960	A white small car moves toward to the fisheye camera with a constant speed.
Car-II Sequence (CS-II)	Real object	21	960x960	A red car moves toward to the fisheye camera with a constant speed.
Train Sequence (TS)	Real object	86	960x960	A small train moves from right to left side at a constant speed.
Traffic-I Sequence (TRS-I)	Real object	119	960x960	A small number of cars move slowly on a road.
Traffic-II Sequence (TRS-II)	Real object	101	960x960	A large number of cars move slowly on a road.
People Sequence (PS)	Real object	160	960x960	A group of people walking slowly.

$$\text{PSNR} = 20\,log_{10}\left(\frac{2^n}{MSE^2}\right) \quad (2)$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[A(i,j) - \hat{A}(i,j)\right]^2 \quad (3)$$

where *n* is the bit depth. Either A or Â should have the same resolution (M x N), and each pixel position is indicated by pixel indices $1 \le i \le M$ and $1 \le j \le N$. For every two successive omnidirectional image inputs, the PSNR was used for evaluating the performance of each reconstructed omnidirectional image (RFm | m Є R+) with the reference omnidirectional image (the second omnidirectional image input).

As for the second performance measurement, we employed direct visual quality evaluation on every reconstructed omnidirectional image (RF_m). We found that such visualization of the image, many a time, is capable of describing more on the real condition of every pixel on an image.

D. The Elasticity in the Number of the Generated Omnidirectional Image

This experimental phase aims at revealing the elasticity of the E-FRUC in terms of generating the omnidirectional images from every two successive omnidirectional image inputs. The results for this phase are shown in Table 2 and Fig. 4 (a and b). Each sequential omnidirectional video presents unique characteristics that can be explained as follows.

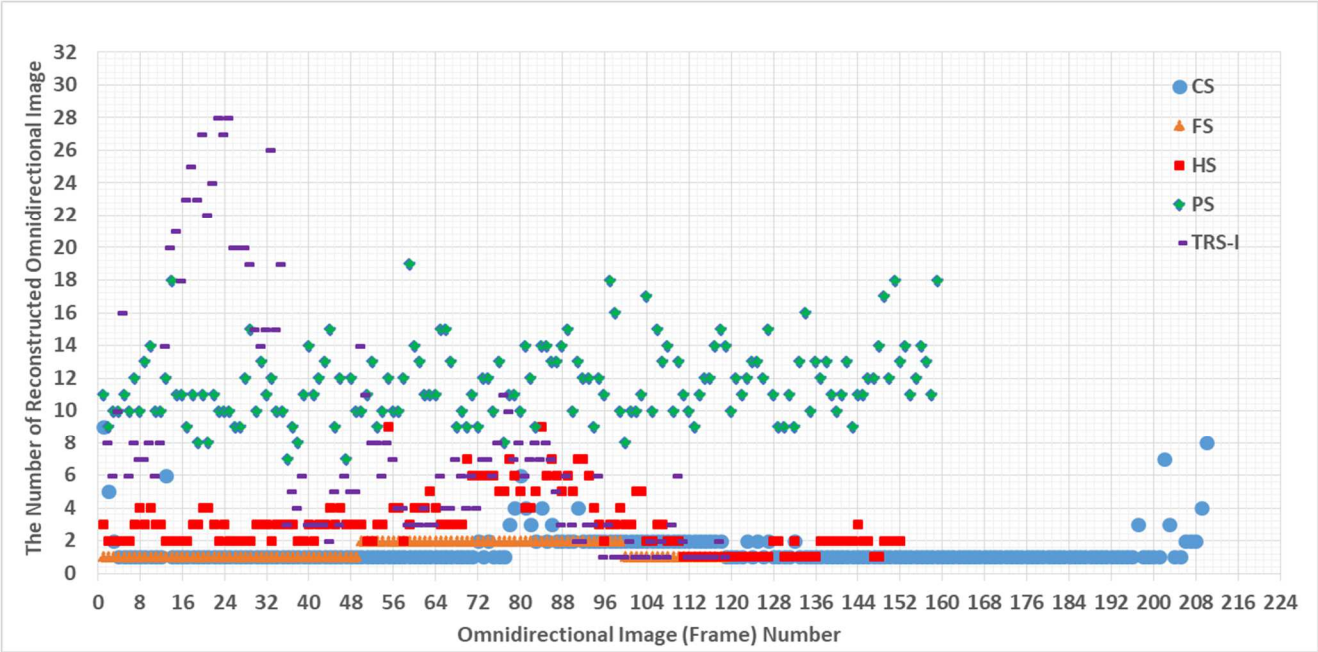
Even though both the Cube Sequence (CS) and the Flowers Sequence (FS) are synthetically created, the motion of the object in each video is different. As a result, the number of reconstructed omnidirectional image (RF) produced by the E-FRUC for each video is also different. The maximum number of RF for the CS is 9, while that of the FS is 2, as can be seen in Table 2. The RF can be generated in large numbers when the motion of the object in the CS happens at the peripheral area of the omnidirectional images input (F_1 with F_2 and F_{209} with F_{210}), as can be seen in Fig. 4 (a). In contrast, the RF for the FS tends to be consistent at a maximum number when the motion occurs around all areas of the omnidirectional images input (every two consecutive omnidirectional images started from F_{50} and finished to F_{99}). As can be observed in Table 2, 42.02% of the total number of omnidirectional images in the FS produces multiple RF, which is 14.42% higher than that of the CS. It seems like the more the motion occurs throughout the omnidirectional image area, the more opportunity to obtain multiple RF from the E-FRUC process.

The Hand Sequence (HS) and Train Sequence (TS), both display a similar motion model; although, the motion comes from different objects with different sizes. The hand that appears in the HS is bigger than the train in the TS, except that the train is longer than the hand. According to Table 2, the maximum number of the RF produced by the E-FRUC in TS is slightly more than that compared to that in the HS. There are 10 RFs for the TS, while the HS obtains 9 RFs. Using the E-FRUC, repetitive RF exists from almost 83% and 85% of the total number of omnidirectional images in the TS and HS, respectively. It is interesting to note that the distributions of the RF for both HS and TS spread out almost throughout the images in each sequence, as described in Fig. 4 (a) for HS and Fig. 4 (b) for TS.

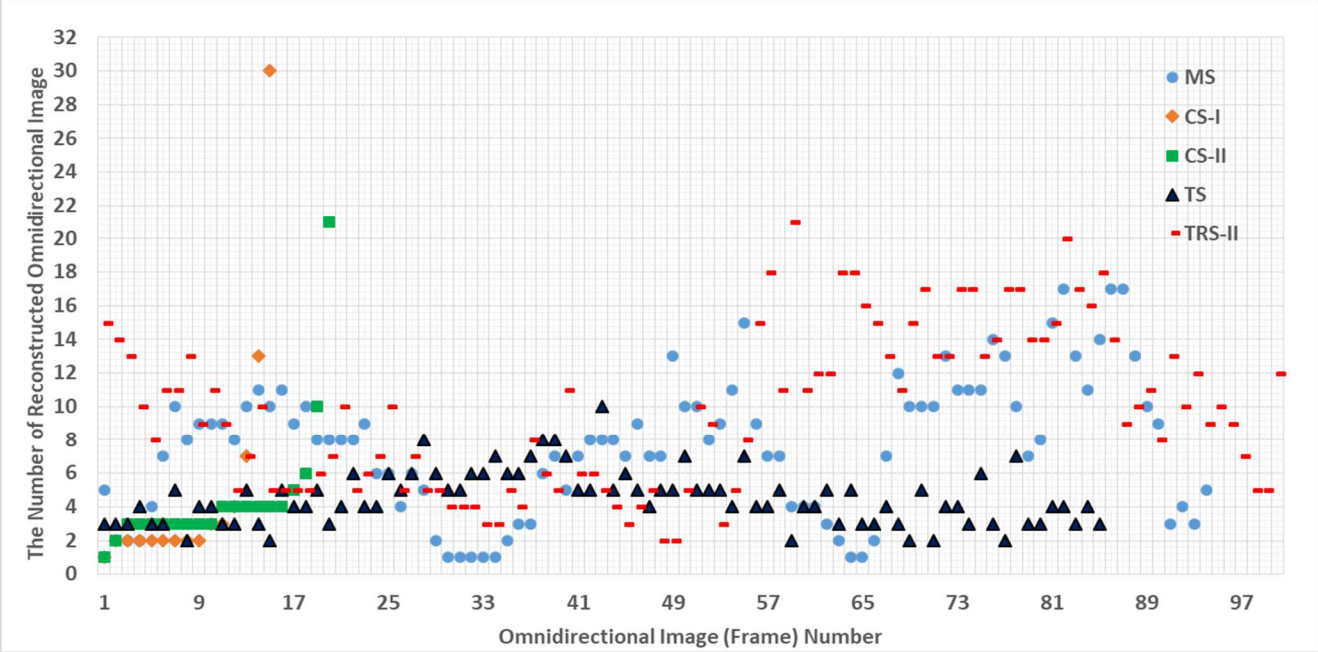
With respect to the motion model, Man Sequence (MS) is similar to the FS except that the MS is a real video. Based on Table 2, 92% of omnidirectional images in the MS produces RF more than one. This is twice as many as the number of omnidirectional images in the FS, which the E-FRUC can process to develop the RF. The distribution of multiple RF can be seen in Fig. 4 (b). Moreover, the maximum number of the RF obtained from two successive omnidirectional images within the MS can reach up to 16, whereas the FS can manage up to 9.

TABLE II
THE CHARACTERISTIC OF THE OMNIDIRECTIONAL IMAGES PRODUCED BY THE E-FRUC.

The Number of The Reconstructed Omnidirectional Image (RFm)	Cube (CS)	Flowers (FS)	Moving Hand (HS)	Man (MS)	Car I (CS-I)	Car II (CS-II)	Train (TS)	Traffic I (TRS-I)	Traffic II (TRS-II)	People (PS)
Maximum	9	2	9	16	30	21	10	28	21	19
Minimum	1	1	1	1	1	1	2	1	2	8
Average	5	1.5	5	8.5	15.5	11	6	14.5	11.5	13.5
More than the minimum	58 (27. 6%)	50 (42. 02%)	127 (83. 6%)	87 (92%)	14 (93. 3%)	19 (95%)	85 (100%)	102 (86. 4%)	100 (100%)	100 (100%)



(a) The number of RF for every two successive omnidirectional images from the CS, FS, HS, PS, and TRS-I.



(b) The number of RF for every two successive omnidirectional images from the MS, CS-I, CSII, TS, and TRS-II.

Fig. 4 The number of RF created by the E-FRUC for every two successive omnidirectional images input

Car-I Sequence (CS-I) and Car-II Sequence (CS-II) in general have similar motion models unless the color of one car is in contrast with the other. In CS-I, the main object is the white car, while in CS-II, it is the red car. The maximum numbers of RF that the E-FRUC can generate from both video sequences are considerably high, 30 for CS-I and 21 for CS-II (See Table 2). As for the distribution, 93.3% of the omnidirectional images in CS-I produce multiple RF, which is 1.7% lower than the generated RF number for CS-II. Those RFs are obtained from F2 to F14 and F2 to F20 for CS-I and CS-II, respectively, as illustrated in Fig. 4 (b).

There is a similarity between Traffic-I Sequence (TRS-I) and Traffic-II Sequence (TRS-II), in which both have several moving objects. i.e., vehicles. The difference is the number of vehicles captured between these videos. TRS-II has a more cars compared to TRS-I, and because of this, the number of RF produced by the E-FRUC process is at least 2 for all omnidirectional images within the TRS-II, as can be noticed in Table 2 and shown in Fig. 4 (b). Meanwhile, for TRS-I, only 86.4% of the sequential omnidirectional images can generate the RF for more than one time. However, a pair of omnidirectional images in the TRS-I can obtain a maximum of 28 RFs higher than the E-FRUC can generate in the TRS-II.

Finally, for every two consecutive omnidirectional images in People Sequence (PS), E-FRUC can produce RF with a minimum of eight frames and a maximum of 19 frames. In this video, many people walk across the street, and sometimes they are found to be close to each other. This phenomenon causes the movement seems to occur inside the omnidirectional image area at the same time.

E. The Quantitative Evaluation of The Performance of The Reconstructed Omnidirectional Images

This phase measures every reconstructed omnidirectional image (RF) performance by comparing each RF with a reference omnidirectional image. The image reference, in this case, is the second omnidirectional image, as mentioned by the first equation in section III-C. The comparison between an RF and an omnidirectional image is computed using PSNR measurement, mentioned by the 2nd and 3rd equations for the performance indicator. The results show an increasing trend of the PSNR for every RF increase from the first construction until the end. Fig. 5 shows the results of the PSNR for ten omnidirectional videos. For better visibility, each omnidirectional video result is separately shown. In this figure, each result comes from two consecutive omnidirectional images that the E-FRUC produces the most RF, as listed in Table 2. It can be seen that the PSNR increment for each sample of omnidirectional video is non-linear. The first RF stands at the lowest PSNR point followed by the second one with a significant increment. Then as can be observed, the PSNR of some following RFs increases gradually. Another observation is that the last RF usually has the highest amount of PSNR. Such behavior can be seen in all samples of the omnidirectional video.

F. The Qualitative Evaluation of the Performance of the Reconstructed Omnidirectional Images

In this phase, we try to reveal the visual quality of the reconstructed omnidirectional images (RF_m) qualitatively. Due to a lack of information given by the PSNR values regarding the position of an error bit in the pixel domain on the omnidirectional images, a simple visual observation directly to the image is a valuable assessment to carry out. Fig. 6 and Fig. 7 show some of these results. In those figures, a set of RFs expresses each omnidirectional video input. This set of RFs comes from the two consecutive omnidirectional images, which the cycle of the E-FRUC runs the most frequently, as illustrated in Fig. 5. Since the number of RF is many, and the availability of space is limited, the set of RF is presented only by three images, and the following discussion explains these results.

In Fig. 6 (a) to 6 (e), the visual condition of a pair of omnidirectional images input and three samples of RF from the Cube Sequence (CS) is presented.

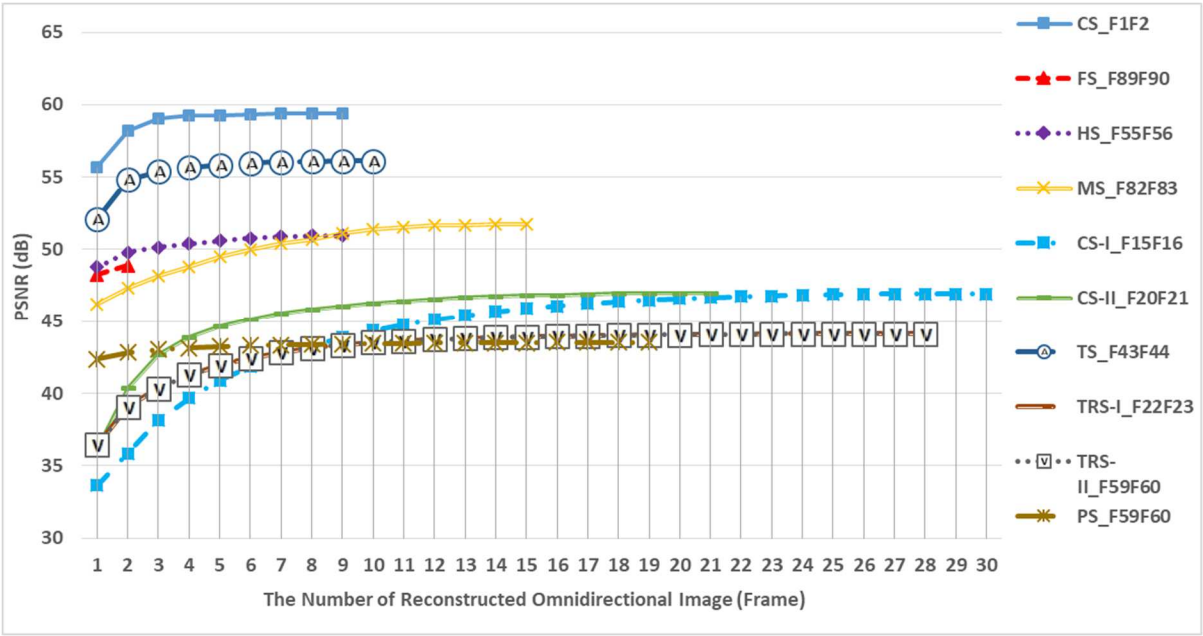
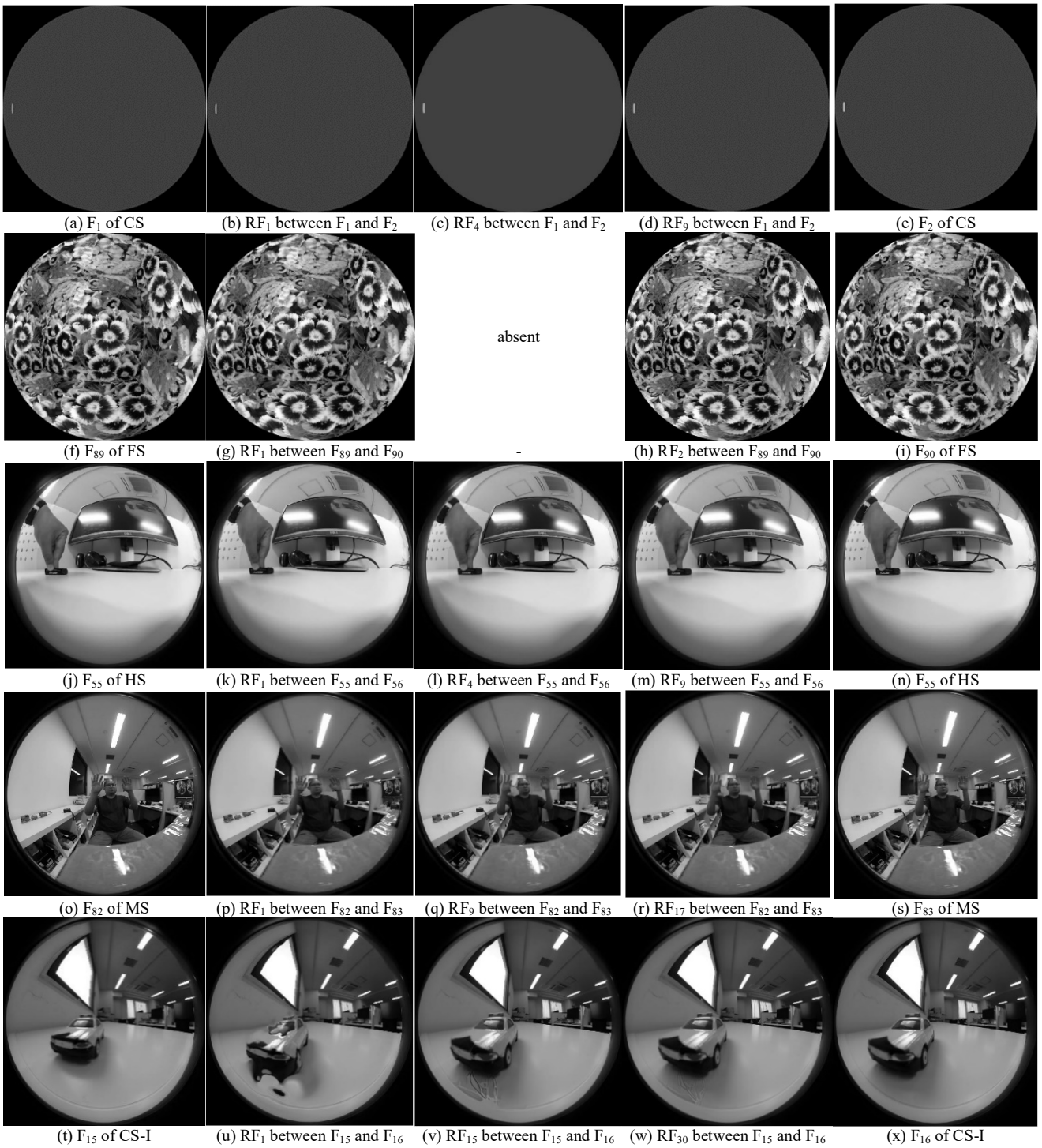


Fig. 5 The PSNR score for each set of samples of the RF. Each video is represented by the RF, which is most frequently occurs.

The two successive omnidirectional images are F_1 (a) and F_2 (e), while the three samples of RF consist of RF_1 (b), RF_4 (c), and RF_9 (d). The image quality of the three RFs is excellent, although the PSNR for each RF is different. The PSNR for RF_1, RF_2, and RF_3 are 55.67, 59.30, and 59.41 dB, respectively. It seems that the high value of PSNR for each RF confirms very well with the high visual quality performance.

Results from the Flowers Sequence (FS) show a similar observation with that of the CS; although, the maximum number of RF produced by the E-FRUC is only two. As depicted in Fig. 6 (f) to 6 (i), the E-FRUC can create RF1 (g) and RF2 (h) from two consecutive omnidirectional images, F89 (f) and F90 (i). Either RF_1 or RF_2 has a high visual quality performance, even though the PSNR value of the corresponding RF is not similar. The RF_1 obtains 48.24 dB, while the RF_2 gets 48.87 dB.

Moving Hand Sequence (HS) samples are shown in Fig. 6 (j) to 6 (n). From the two consecutive omnidirectional images, F_{55} (j) and F_{56} (n), the E-FRUC can generate up to 9 RFs, in which three of them (RF_1, RF_4, and RF_9) are displayed in (k), (l), and (m), successively.



(a) F_1 of CS (b) RF_1 between F_1 and F_2 (c) RF_4 between F_1 and F_2 (d) RF_9 between F_1 and F_2 (e) F_2 of CS

(f) F_{89} of FS (g) RF_1 between F_{89} and F_{90} (h) RF_2 between F_{89} and F_{90} (i) F_{90} of FS

(j) F_{55} of HS (k) RF_1 between F_{55} and F_{56} (l) RF_4 between F_{55} and F_{56} (m) RF_9 between F_{55} and F_{56} (n) F_{55} of HS

(o) F_{82} of MS (p) RF_1 between F_{82} and F_{83} (q) RF_9 between F_{82} and F_{83} (r) RF_{17} between F_{82} and F_{83} (s) F_{83} of MS

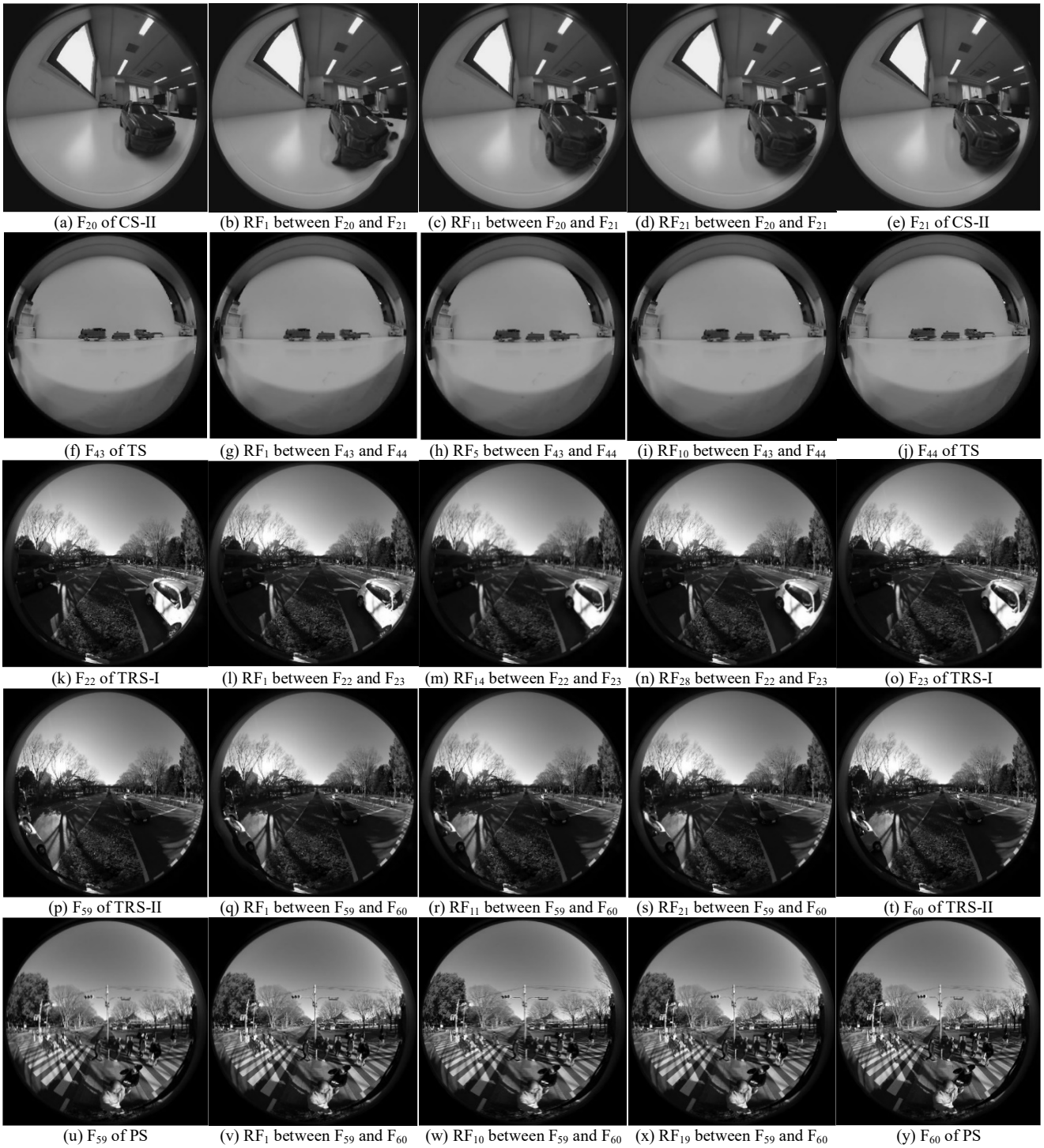
(t) F_{15} of CS-I (u) RF_1 between F_{15} and F_{16} (v) RF_{15} between F_{15} and F_{16} (w) RF_{30} between F_{15} and F_{16} (x) F_{16} of CS-I

Fig. 6 Five sets of samples of the RF coming from the CS (a - e), FS (f - i), HS (j - n), MS (o – s), and CS-I (t - x).

Based on Fig. 5., the RF1, RF4, and RF9 get PSNR of 48.78, 50.40, and 50.96 dB, respectively, and this is in conformity with the visual quality performance.

In the case of the Man Sequence (MS), the RF can be constructed until a maximum of 16 times, which is achieved for F_{82} and F_{83} as displayed in Fig. 6 (o) and 6 (s), respectively. There are three RFs put on the same figure, and they are located at 6 (p) for RF_1, 6 (q) for RF_9, and 6 (r) for RF_{17}. According to Fig. 5, the PSNRs of the RF_1, RF_9, and RF_{17} are 46.28 dB, 51.56 dB, and 52,57 dB, respectively. Once again, the high-quality visual images show consistent performance with the PSNR values.

The omnidirectional image sequence for Car-I Sequence (CS-I) is shown in Fig. 6 (t) to 6 (x). The examples include a pair of the input, F_{15} in 6 (t) and F_{16} in 6 (x), and three RFs, depicted by RF_1 in 6 (u), RF_{15} in 6 (v), and RF_{30} in 6 (w). Interestingly, RF_1, which has a PSNR score of 33.63 dB, suffers from small distortion (in front of the car area). Nevertheless, this distortion reduces gradually in RF_{15} and RF_{30}. The last two RFs have a score of PSNR of 45.90 dB and 46.94 dB, respectively. A comparable situation is also observed for omnidirectional images from the Car-II Sequence (CS-II) (See Fig. 7 (a) to 7 (e)).



(a) F_{20} of CS-II (b) RF_1 between F_{20} and F_{21} (c) RF_{11} between F_{20} and F_{21} (d) RF_{21} between F_{20} and F_{21} (e) F_{21} of CS-II

(f) F_{43} of TS (g) RF_1 between F_{43} and F_{44} (h) RF_5 between F_{43} and F_{44} (i) RF_{10} between F_{43} and F_{44} (j) F_{44} of TS

(k) F_{22} of TRS-I (l) RF_1 between F_{22} and F_{23} (m) RF_{14} between F_{22} and F_{23} (n) RF_{28} between F_{22} and F_{23} (o) F_{23} of TRS-I

(p) F_{59} of TRS-II (q) RF_1 between F_{59} and F_{60} (r) RF_{11} between F_{59} and F_{60} (s) RF_{21} between F_{59} and F_{60} (t) F_{60} of TRS-II

(u) F_{59} of PS (v) RF_1 between F_{59} and F_{60} (w) RF_{10} between F_{59} and F_{60} (x) RF_{19} between F_{59} and F_{60} (y) F_{60} of PS

Fig. 7 Five sets of samples of the RF coming from the CS-II (a - e), TSS (f - j), TRS-I (k - o), TRS-II (p – t), and PS (u - y).

A pair of omnidirectional images that achieved the highest RF number is between F_{20} (a) and F_{21} (e). There are 21 RFs from the process, and three of RFs are shown in figures 7 (b) for RF_1, 7 (c) for RF_{11}, and 7 (d) for RF_{21}. Like the CS-I case, RF_1 also has a small distortion on the car's front side, and the PSNR score is 36.08 dB. The condition gets improved, however, for RF_{11}, which has a PSNR value of 46.44 dB. Eventually, in RF_{21}, the error is visually undetected and unnoticeable. This image has a PSNR score of 47.00 dB.

Fig. 7 (f) to (j) shows a set of samples given by the Train Sequence (TS). A combination of F_{43} and F_{44} enables the E-FRUC to create a maximum of 10 RFs, and three of them i.e., RF_1, RF_5, and RF_{10} are shown in 7 (g), 7 (h), and 7 (i), respectively. As can be seen, the visual quality of the three RFs is impressively incredible, and this observation conforms to the PSNR values for RF_1, RF_5, and RF_{10} which are 52.12, 55.84, and 56.12 dB, respectively.

In the Traffic-I Sequence (TRS-I), there are 28 RFs produced by the E-FRUC, and three of the RFs, including the input pair, are shown in Fig. 7 (7 (k) to 7 (o)). This RF number is the second-highest number of RF in this experiment, and this happens between the omnidirectional images input of F_{22} (k) and F_{23} (o). The three RFs; RF_1, RF_{14}, and RF_{28}, have PSNR scores of 36.45 dB, 43.87 dB, and 44.13 dB. The visual quality of each RF is remarkable.

From Fig. 7 (p) to 7 (t), three RFs between the two consecutive omnidirectional image inputs F_{59} and F_{60} from the Traffic-II Sequence (TRS-II) are shown. It is evident that the observed quality of those RFs is outstanding. The PSNR of the generated RFs i.e., RF_1 in 7 (q), RF_{11} in 7 (r), and RF_{21} in 7 (s) are 40.60 dB, 45.51dB, and 45.61 dB, respectively.

Finally, the last experiment utilizes the People Sequence (PS), as shown in figures 7 (u) to 7 (y). The omnidirectional image pair used as the input is F_{59} in 7 (u) and F_{60} in 7 (y). With this pair, 19 RFs can be generated by the E-FRUC, and RF_1 in 7 (v), RF_{10} in 7 (w), and RF_{19} in 7 (x) are three examples of those RFs. The PSNR for RF_1, RF_{10} and RF_{19} are given as 42.35 dB, 43.44 dB and 43.52 dB, respectively and their visual quality does correspond excellently.

The above data shows that using the E-FRUC, moving objects' continuity can be preserved. The PSNR score for each RF confirms this condition. Even though the first few RF have low PSNR values, the subsequent RF will gradually improve and thus attain a higher PSNR score. This increase in the PSNR score will reach a maximum at the final RF, which is just before the second omnidirectional image input. In some instances, in which the E-FRUC creates more than two RFs from a pair of omnidirectional images, there is a possibility the first RF has undesirable visual quality due to too much distortion. This situation will happen if the PSNR score of the first RF is below 38 dB. Nevertheless, such a distorted figure will be eliminated in subsequent RFs with PSNR scores having higher values and having a visual perception of exceptional quality. Having said this, the first RF can be ignored if one wishes so.

IV. Conclusion

In conclusion, this paper has shown that the proposed elastic frame rate up-conversion method can be successfully applied in generating intermediate frames even to an omnidirectional image sequence with irregular motion and distorted figures. Since the motion characteristic of moving objects in an omnidirectional image is not only non-linear but also deforms irregularly, the E-FRUC, which is constructed from the optical flow concept, has proven to be capable of generating a varying amount of reconstructed omnidirectional images (RFs). The RF can be generated elastically depending on the motion condition of the omnidirectional images. Moreover, rapid changes in brightness also contribute to the production of the RF. The more complex the motion objects condition inside the omnidirectional images is, the more RFs the E-FRUC can create. In that condition, the PSNR of the RFs varies from about 33 to 59 dB, and the visual quality of the RFs ranges from acceptable to extremely good. As a result, the E-FRUC is an auspicious method for raising the frame rate of omnidirectional video obtained from any modest cameras and increasing smoothness of an incomplete omnidirectional video sequence played back in a receiver after grabbed from error-prone telecommunication networks.

Acknowledgment

The authors are grateful to LIPI under *Inisiasi Insentif Kolaborasi Riset Global*'s (Global Research Collaboration Incentive Initiation) program. This research was conducted in collaboration with UTM's researchers. Finally, the authors are obliged to *Lembaga Pengelola Dana Pendidikan* (Education Fund Management Institution) for funding this research.

References

[1] J. Zhou, Y. Fu, Y. Yang, and A. T. S. Ho, "Distributed video coding using interval overlapped arithmetic coding," *Signal Process. Image Commun.*, vol. 76, pp. 118–124, 2019, doi: https://doi.org/10.1016/j.image.2019.03.016.

[2] R. Yang, M. Xu, T. Liu, Z. Wang, and Z. Guan, "Enhancing Quality for HEVC Compressed Videos," *IEEE Trans. Circuits Syst. Video Technol.*, 2019, doi: 10.1109/TCSVT.2018.2867568.

[3] D. Checa and A. Bustillo, "A review of immersive virtual reality serious games to enhance learning and training," *Multimed. Tools Appl.*, 2020, doi: 10.1007/s11042-019-08348-9.

[4] A. Habibian, T. Van Rozendaal, J. Tomczak, and T. Cohen, "Video compression with rate-distortion autoencoders," 2019, doi: 10.1109/ICCV.2019.00713.

[5] W. Bao, X. Zhang, L. Chen, L. Ding, and Z. Gao, "High-Order Model and Dynamic Filtering for Frame Rate Up-Conversion," *IEEE Trans. Image Process.*, 2018, doi: 10.1109/TIP.2018.2825100.

[6] M. Zhang, W. Zhou, H. Wei, X. Zhou, and Z. Duan, "Frame level rate control algorithm based on GOP level quality dependency for low-delay hierarchical video coding," *Signal Process. Image Commun.*, vol. 88, p. 115964, 2020, doi: https://doi.org/10.1016/j.image.2020.115964.

[7] C.-H. Yeh, J.-R. Lin, M.-J. Chen, C.-H. Yeh, C.-A. Lee, and K.-H. Tai, "Fast prediction for quality scalability of High Efficiency Video Coding Scalable Extension," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 462–476, 2019, doi: https://doi.org/10.1016/j.jvcir.2018.12.021.

[8] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Blurry Video Frame Interpolation," 2020, doi: 10.1109/CVPR42600.2020.00516.

[9] G. G. Lee, C. F. Chen, C. J. Hsiao, and J. C. Wu, "Bi-directional trajectory tracking with variable block-size motion estimation for frame rate up-convertor," *IEEE J. Emerg. Sel. Top. Circuits Syst.*, 2014, doi: 10.1109/JETCAS.2014.2298923.

[10] A. Jiménez-Moreno, E. Martínez-Enríquez, and F. Díaz-de-María, "Bayesian adaptive algorithm for fast coding unit decision in the High Efficiency Video Coding (HEVC) standard," *Signal Process. Image Commun.*, vol. 56, pp. 1–11, 2017, doi: https://doi.org/10.1016/j.image.2017.04.004.

[11] H. Liu, R. Xiong, D. Zhao, S. Ma, and W. Gao, "Multiple hypotheses bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations," *IEEE Trans. Circuits Syst. Video Technol.*, 2012, doi: 10.1109/TCSVT.2012.2197081.

[12] Y. Yang, L. Shen, H. Yang, and P. An "A content-based rate control algorithm for screen content video coding," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 328–338, 2019, doi: https://doi.org/10.1016/j.jvcir.2019.02.031.

[13] Y. Chen, R. Hu, J. Xiao, and Z. Wang, "Multisource surveillance video coding with synthetic reference frame," *J. Vis. Commun. Image Represent.*, vol. 65, p. 102685, 2019, doi: https://doi.org/10.1016/j.jvcir.2019.102685.

[14] S. J. Yoon, H. H. Kim, and M. Kim, "Hierarchical Extended Bilateral Motion Estimation-Based Frame Rate Upconversion Using Learning-Based Linear Mapping," *IEEE Trans. Image Process.*, 2018, doi: 10.1109/TIP.2018.2861567.

[15] P. A. Brousseau and S. Roy, "Calibration of axial fisheye cameras through generic virtual central models," 2019, doi: 10.1109/ICCV.2019.00414.

[16] W. Gao and S. Shen, "Dual-fisheye omnidirectional stereo," 2017, doi: 10.1109/IROS.2017.8206587.

[17] S. Ji, Z. Qin, J. Shan, and M. Lu, "Panoramic SLAM from a multiple fisheye camera rig," *ISPRS J. Photogramm. Remote Sens.*, 2020, doi: 10.1016/j.isprsjprs.2019.11.014.

[18] P. Liu, L. Heng, T. Sattler, A. Geiger, and M. Pollefeys, "Direct visual odometry for a fisheye-stereo camera," 2017, doi: 10.1109/IROS.2017.8205988.

[19] L. F. Posada, A. Velasquez-Lopez, F. Hoffmann, and T. Bertram, "Semantic mapping with omnidirectional vision," 2018, doi: 10.1109/ICRA.2018.8461165.

[20] C. Won, J. Ryu, and J. Lim, "SweepNet: Wide-baseline omnidirectional depth estimation," 2019, doi: 10.1109/ICRA.2019.8793823.

[21] A. S. Satyawan, J. Hara, and H. Watanabe, "Automatic self-improvement scheme in optical flow-based motion estimation for sequential fisheye images," *ITE Trans. Media Technol. Appl.*, 2019, doi: 10.3169/mta.7.20.

[22] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *Int. J. Comput. Vis.*, 2004, doi: 10.1023/B:VISI.0000011205.11775.fd.

[23] Matlab, "Matlab 2016, Tutorial." www.mathworks.com/products/matlab.html (accessed Aug. 01, 2018).

[24] Blender, "Blender 2.81, Tutorial." www.blender.org (accessed Mar. 01, 2018).

[25] N. Asuni and A. Giachetti, "Testimages: A large-scale archive for testing visual devices and basic image processing algorithms," 2014, doi: 10.2312/stag.20141242.

[26] Ricoh, "Ricoh Theta S, User Manual." theta360.com/en/about/theta/s.html (accessed Jan. 01, 2017).