

Article

Time Sequential Motion-to-Photon Latency Measurement System for Virtual Reality Head-Mounted Displays

Song-Woo Choi , Siyeong Lee , Min-Woo Seo and Suk-Ju Kang * 

Department of Electronic Engineering, Sogang University, Seoul 04107, Korea; songwoo602@gmail.com (S.-W.C.); siyeong@sogang.ac.kr (S.L.); yoynok08@gmail.com (M.-W.S.)

* Correspondence: sjkang@sogang.ac.kr; Tel.: +82-02-705-8466

Received: 12 August 2018; Accepted: 29 August 2018; Published: 1 September 2018



Abstract: Because the interest in virtual reality (VR) has increased recently, studies on head-mounted displays (HMDs) have been actively conducted. However, HMD causes motion sickness and dizziness to the user, who is most affected by motion-to-photon latency. Therefore, equipment for measuring and quantifying this occurrence is very necessary. This paper proposes a novel system to measure and visualize the time sequential motion-to-photon latency in real time for HMDs. Conventional motion-to-photon latency measurement methods can measure the latency only at the beginning of the physical motion. On the other hand, the proposed method can measure the latency in real time at every input time. Specifically, it generates the rotation data with intensity levels of pixels on the measurement area, and it can obtain the motion-to-photon latency data in all temporal ranges. Concurrently, encoders measure the actual motion from a motion generator designed to control the actual posture of the HMD device. The proposed system conducts a comparison between two motions from encoders and the output image on a display. Finally, it calculates the motion-to-photon latency for all time points. The experiment shows that the latency increases from a minimum of 46.55 ms to a maximum of 154.63 ms according to the workload levels.

Keywords: head-mounted display; virtual reality; motion-to-photon latency

1. Introduction

Because the personal computer's performance has been greatly improved, real-time rendering of high-quality images has become possible, and virtual reality (VR) technology has become a reality [1]. According to this trend, a variety of VR devices utilizing 3D rendering and sensor-based technology have been released. The head-mounted display (HMD) device, which is a system designed to improve immersion by mounting a wide field-of-view display within the user's sight, has been gaining popularity [2]. However, because of the mismatch between visual and vestibular systems, users wearing HMD devices experience motion sickness and dizziness, which can be an obstacle to the VR market. The motion-to-photon latency, which is one of the causes for this mismatch, is the time delay for a user movement to be fully reflected on a display screen [3]. Generally, three steps are required to render an image, as shown in Figure 1. When head motion occurs, the motion detection unit samples the orientation data for the view generation. After the motion detection, the visual processing unit renders a 3D image. Finally, the rendered image is outputted to a display corresponding to the head orientation of the user measured by the sensor. As these steps take time, the delay results in motion-to-photon latency. In this case, the image does not exactly correspond to the actual head orientation of the user, thereby causing the user to experience motion sickness [4]. Therefore, many HMD makers such as Oculus VR have conducted studies to minimize the mismatch caused by the motion-to-photon latency.

Numerous studies such as prediction techniques based on the data acquired by inertial measurement unit (IMU) sensors [5] and asynchronous time warp (ATW) [6] have been conducted to overcome this limitation. The quantitative evaluation of the above methods required the authors to measure the motion-to-photon latency. They could then improve and evaluate these methods with reference data.

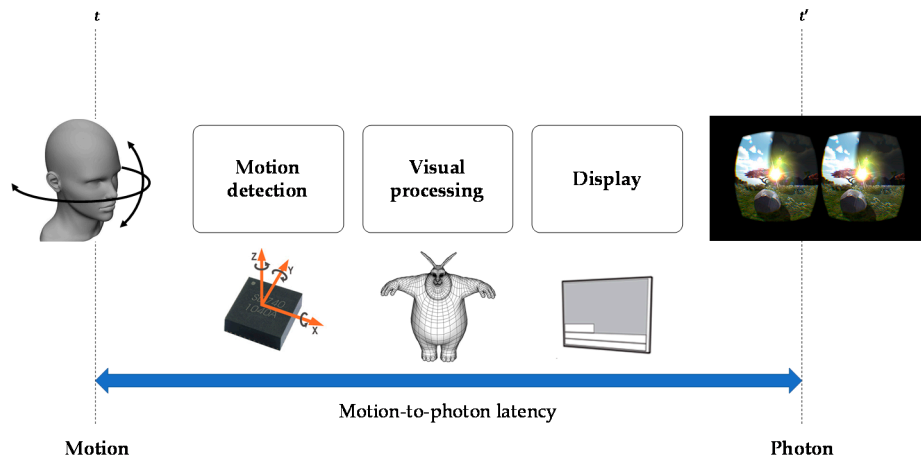


Figure 1. Image generation process in the HMD (head-mounted display) system.

To measure this motion-to-photon latency, the method of [7] presents a low-cost system with high accuracy which measures the latency. However, it can be used for the mobile device. The method of [8] proposes the measurement system and a technique to reduce the latency for the optical see-through display. The authors had previously proposed a new measurement system by using multiple sensors such as an optical sensor and encoders and by comparing the physical signal to the luminance signal from a VR scene reflected on the display [9]. Figure 2 shows the overall architecture of this measurement system. A rotary platform, which controls the physical motion, was proposed in that method to simulate the rotation of the neck. It measured the change of the image brightness when the physical motion commenced. The method could directly measure the motion-to-photon latency because it calculated the time difference between the point at which the brightness started changing and that at which the physical motion started. However, although this method provided accurate measurement results, its limitation was that the latency could only be measured when the physical motion began.

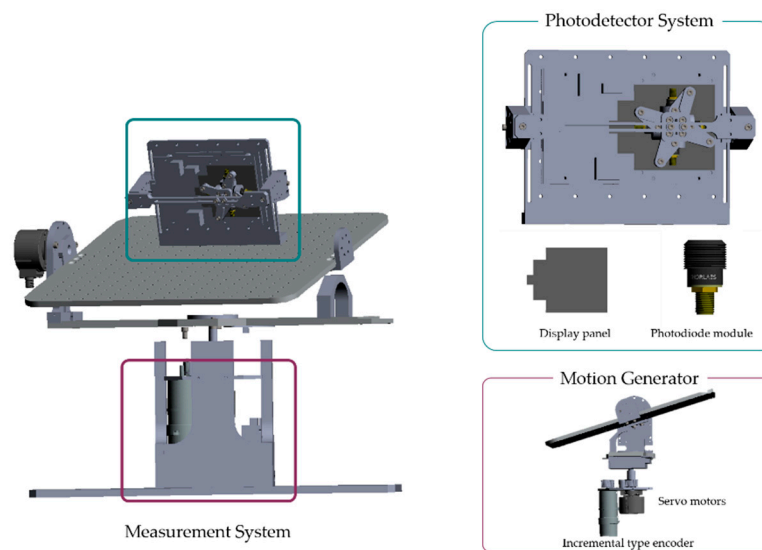


Figure 2. Overall architecture of the previously proposed motion-to-photon latency measurement system.

In this paper, the authors propose a novel time sequential measurement system for the motion-to-photon latency by comparing the physical motion and the motion reflected on the image by 3D rendering. Therefore, it is possible to measure and record the time sequential latency more precisely in real-time, while reflecting the change in workload over time, unlike the conventional method mentioned in [9]. Specifically, the proposed method renders the rotation data with the intensity levels of pixels on the measurement area, which is specially invented for the HMD system, and it can obtain the motion-to-photon latency data in all temporal ranges. Therefore, it has a great advantage over the existing method.

This paper is organized as follows. Section 2 describes the proposed latency measurement system with five subsections. Section 3 presents the experimental environment and the results using the proposed system. Section 4 presents the paper’s conclusion.

2. Proposed Method

Figure 3 shows the overall block diagram of the proposed system. The proposed method measured the actual angular change of the encoder when physical movement occurred. Concurrently, when the physical movement was measured by the IMU sensor, the rendered image was outputted to the display reflecting the measured angle, and the corresponding angle value was converted into the intensity image. The proposed method compared these two values. Specifically, a measurement area, which was constructed with multiple 2D objects mapped with an intensity level converting the rotation angle measured from the IMU, was implemented. Then, the photodetector converted the luminance reflected in the measurement area into a voltage. Finally, the oscilloscope measured this voltage value to determine the intensity level. At the same time, the pulse obtained from the encoder was converted into a position value and compared with the luminance-based position value obtained from the photodetector in real time. Finally, the motion-to-photon latency was calculated by measuring the time difference at which the angle measured from the display reached the same rotation angle of the encoder, as shown in Figure 3. The detailed processes are explained in following sections.

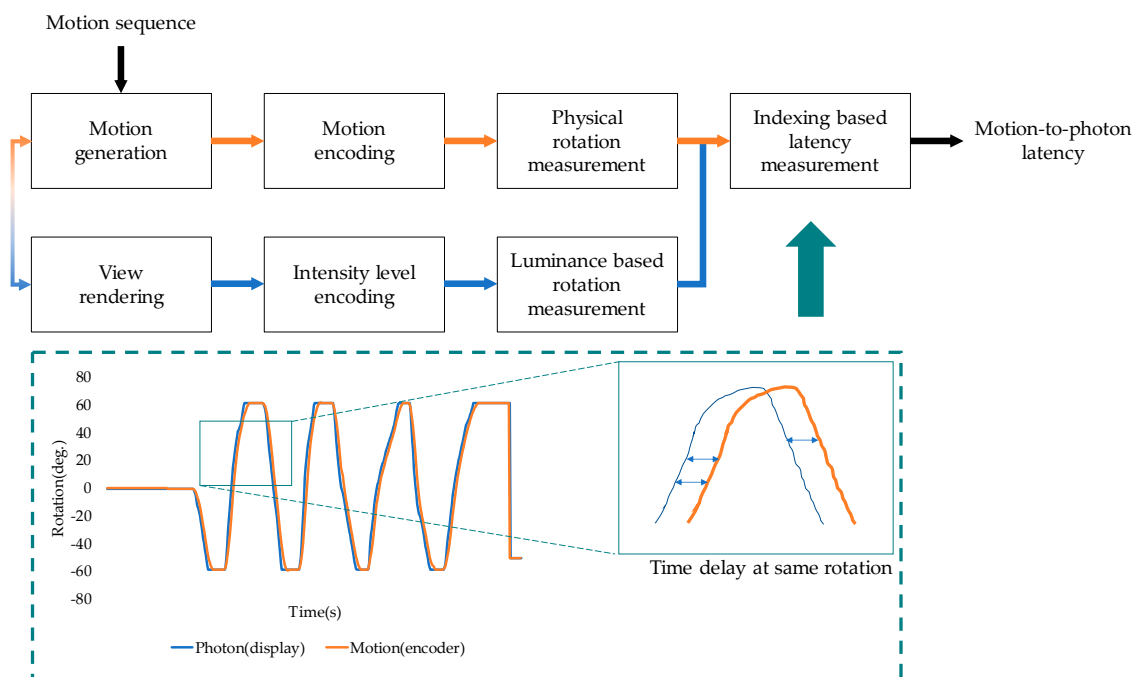


Figure 3. Overall block diagram of the proposed system.

2.1. Motion Generator-Based on Human Neck Movement

The authors used the rotary platform proposed in [9] to model the physical rotation of the user’s neck. Specifically, in order to simulate a human neck, a two-degree-of-freedom rotary platform was proposed, built with joints and links based on head kinematics. Therefore, the motion at the end-effector could be estimated through the motion in each joint based on forward kinematics [10,11], as shown in Figure 4.

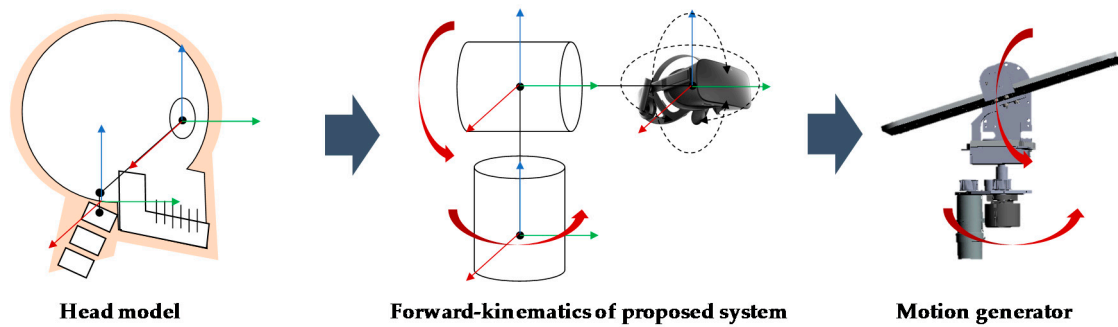


Figure 4. Kinematics of the head-model-based measurement platform.

For this paper, the authors implemented the kinematic analysis of the model using the Denavit–Hartenberg parameters [12], and the angle of the end-effector was estimated by calculating the angle of each joint [13]. In the proposed method, the rotation of the end-effector was estimated by performing rotation transformation on the displacement and rotation in the previous link, and coordinate transformation with the translation matrix. The transformation was carried out on n number of links with the following matrix multiplication:

$$[T] = [Z_1][X_1][Z_2][X_2] \cdots [X_n][Z_n], \tag{1}$$

where $[T]$ denotes the transformation matrix of the end-effector, $[Z_n]$ denotes a transformation matrix at the n -th joint, and $[X_n]$ denotes a transformation matrix of at the n -th link. The transformation matrix of the i -th joint is as follows:

$$[Z_i] = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where θ_i denotes the rotation angle between the previous joint and the next i -th joint along the z axis, and d_i denotes the displacement between each joint. The transformation matrix of the i -th link is as follows:

$$[X_i] = \begin{bmatrix} 1 & 0 & 0 & r_{i,i+1} \\ 0 & \cos \alpha_{i,i+1} & -\sin \alpha_{i,i+1} & 0 \\ 0 & \sin \alpha_{i,i+1} & \cos \alpha_{i,i+1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $\alpha_{i,i+1}$ denotes an angle between each link along the x-axis, and $r_{i,i+1}$ denotes the displacement between each link. Finally, the transformation matrix for the forward kinematics from the $(n - 1)$ -th link to the n -th link is as follows:

$$T_n = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & r_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & r_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

Therefore, the Euler angles [14] of the end-effector were calculated from the inner rotation matrix. The Euler angles were used in an interface of the proposed system.

2.2. Measurement Area Design

In the proposed method, photodetectors measured the rotation angles from the rendered image. The advantage of photodetectors is that they can be measured more accurately while modeling the human visual function owing to their fast response time. However, the photodetectors only measure the intensity of light within the specific range. Hence, the authors proposed specially invented objects that displayed the current rotation angle on a VR scene. Specifically, the objects did not have any colors but had grayscale intensities, and the intensity levels represented the rotation angle with the pre-defined conversion rule that converted the rotation angle with a float type into the intensity value with an unsigned integer type. Therefore, by measuring the luminance for the objects in the measurement area, the current rotation angle was obtained.

When designing the objects in the measurement area, the shapes of the object could influence not only the rendering performance but also the measurement performance. In 3D rendering, there is back-face culling in that the game engine does not calculate the region occluded by an object [15]. Therefore, due to occlusion in the measurement area, the rendering performance is changed. Since the back-face culling affects the latency measurement, the size of the measurement area to be projected into the display is optimally adjusted to minimize the back-face culling and to maximize the voltage that the active layer of the photodetector can measure.

The brightness of the measurement area was determined by the rotation value calculated from the IMU sensor of the HMD device when rendering the current frame. To enhance the measurement accuracy, two objects represented the current rotation angle at a single rotation axis. Arrangement of the four objects enabled both the yaw and pitch, which were the target rotation directions, to be calculated.

The scanline of the display for the area to be measured should also be considered. Since the entire image is not drawn at once but sequentially drawn along the scanline, there is a time difference depending on the arrangement of the measurement area [16]. In order to estimate the correct angle, two measurement areas must be synchronized considering the physical location.

2.3. Voltage Mapping Table

When a change in light is detected in the photodetector installed in the measurement area, the current (the output voltage as a result) emitted by the internal photodiode changes. If the experimental environment and parameters are controlled constantly, the output voltage of the photodetector to the image luminance is constant. This means that the output voltage can be converted back to the original luminance information. Therefore, if the conversion between the luminance and the voltage is known in advance, the rotation angle applied in the rendering process can be estimated. In the proposed method, the relationship between the voltage and luminance was acquired in advance, and the mapping table for the relationship was obtained by using these data. Figure 5 shows how to estimate intensity levels from the luminance of the measurement area on the display using the mapping table.

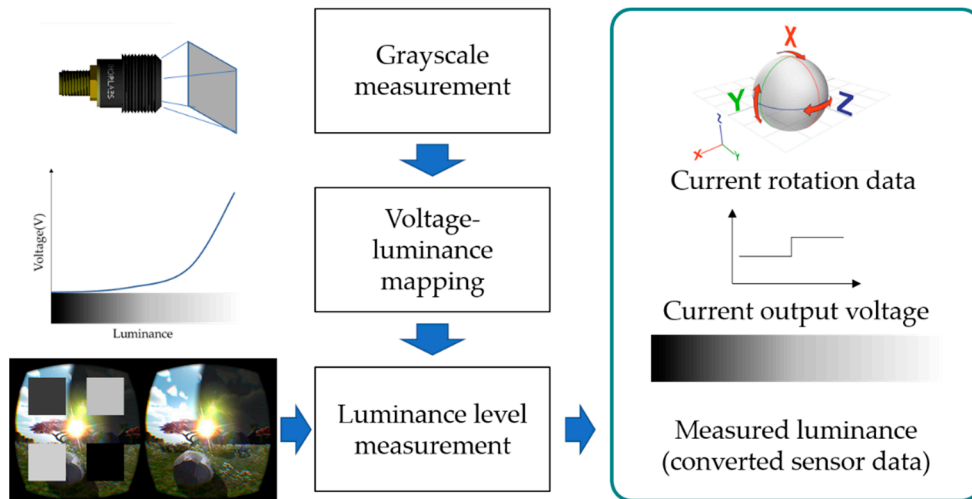


Figure 5. Luminance estimation from an object rendered by using the photodetector system and the voltage mapping table.

The voltage of the photodetector is proportional to the luminance, and therefore, the low luminance has a low output voltage. In this case, the amount of the voltage according to the brightness was not distinguishable from the noise when the voltage outputted from the measurement region was low at the stage of constructing the mapping table. Figure 6 illustrates the above problem. Therefore, low voltage levels are excluded from the measurement, and the mapping table is estimated from the upper level. Then, the current rotation value was inferred. There is a problem that if only the 128 levels of intensity are used for 256 gray levels with an 8-bit depth, the measurement resolution is halved. To compensate for this problem, an additional photodetector was used. An additional sensor restored the resolution to the original resolution by using double photodetectors to measure one rotation of the axis.

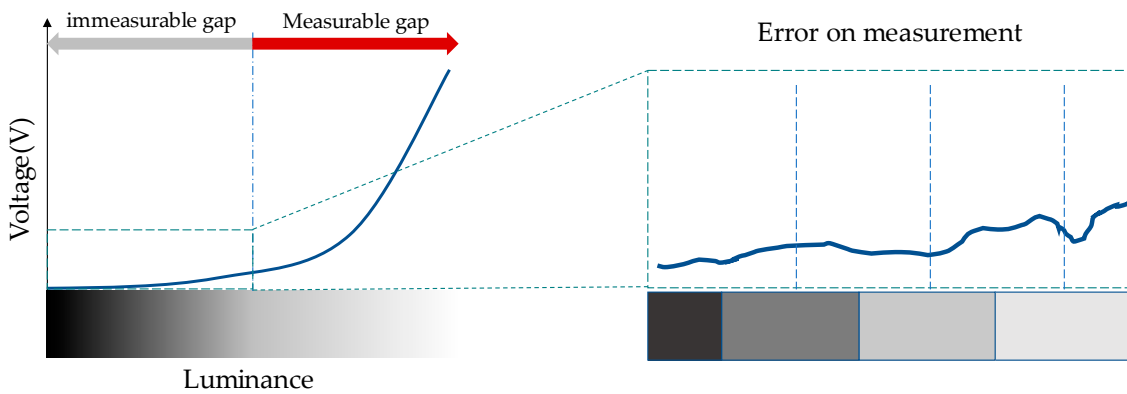


Figure 6. Measurement error and noise for an output voltage in the photodetector system.

For double photodetectors, a new formula was needed which converted a rotation value into two intensity levels. The rotation angle was split into two intensity levels, and these intensity levels were reflected on the measurement area so that the photodetectors measured each intensity level to convert them back to the rotation angle. The specific conversion formula is as follows:

$$\text{Channel} = \left\{ \begin{array}{l} MSB_{S7bit} \left[\frac{(\theta + M_{euler})}{2M_{euler}} \times 2^{14} \right] + 127 \\ LSB_{S7bit} \left[\frac{(\theta + M_{euler})}{2M_{euler}} \times 2^{14} \right] + 127 \end{array} \right\}, \text{ for } |\theta| \leq M_{euler}, \quad (5)$$

where M_{euler} denotes the maximum Euler angle in the measurement system, θ denotes the current angle measured from HMD, and $MSBs_{7bit}$ and $LSBs_{7bit}$ denote the upper and lower 7 bits of the converted value, respectively. In the measurement area, 14-bit data were mapped onto the objects.

For example, as shown in Figure 7, the current rotation of the yaw axis and maximum Euler angle are given. Then, these values were calculated with a conversion formula. The current rotation angle was converted into binary data and split into two parts. Each part was the intensity level of the measurement area. After estimating these intensity levels by the photodetectors, the reverse calculation was performed with the inverse of the conversion formula. Finally, the current rotation was obtained, which is used for the generation of a VR scene.

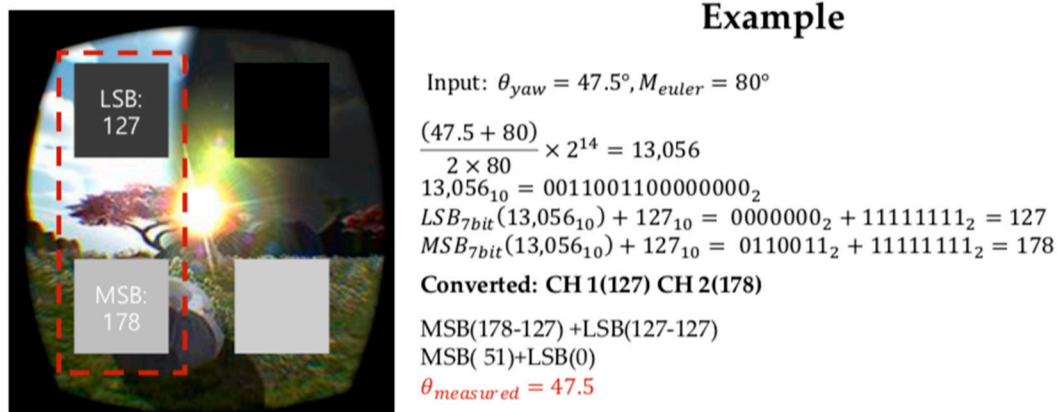


Figure 7. Example of the intensity level conversion and its inverse for measuring the current rotation angle.

However, in some cases, accurate mapping tables could not be formed because of changes in the external environment during the measurement process and changes in the output voltage due to the electrical noise. To solve this problem, a polynomial regression to store the data was performed from the voltage data obtained by repeated experiments. In the proposed method, an optimal curve was obtained by repeatedly performing a polynomial regression of less than third-order to avoid overfitting. Then, the obtained curve was used for the voltage estimation for a luminance level.

The third polynomial regression used in this paper is described below. The regression model is as follows:

$$v_i = a_0 + a_1 I_1 + a_2 I_2^2 + a_3 I_3^3 + \varepsilon_i \quad (i = 1, 2, 3, \dots, n), \tag{6}$$

where n denotes the number of points, v_i denotes an output voltage from the photodetector, I_i denotes an 8-bit unsigned integer value of the intensity level projected on the measurement area, a_i denotes a curvature parameter, and ε_i denotes a random error. The polynomial curve is composed as follows.

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & I_1 & I_1^2 & I_1^3 \\ 1 & I_2 & I_2^2 & I_2^3 \\ 1 & I_3 & I_3^2 & I_3^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}. \tag{7}$$

In (7), if the random error term is excluded, the curvature parameters can be estimated after the least-squares estimation [17]. The least-squares estimation of these parameters is as follows:

$$a = (I^T I)^{-1} I^T v, \tag{8}$$

where v denotes a vector composed of the output voltages from the photodetector, a denotes the curvature parameter vector of the voltage–luminance curve, and I denotes the matrix of intensity

levels from the repeated measurements. Using this regression model, the motion-to-photon latency was measured and is described in the following section.

2.4. Measurement of Motion-to-Photon Latency

The output image on the display was rendered by rotating the HMD using a motion generator designed to control the rotation of the HMD device. It controlled the HMD device with the desired rotation angle using a highly accurate DC servo motor. An internal encoder in the motor, which was used to control the motor, could also be used for measuring the current physical movement. However, in order to more precisely measure the physical movement over time, the incremental type encoders, additionally mounted on the axes, detected the rotation angles and outputted them as pulses. Concurrently, by measuring the luminance of the objects, the current angle data could be obtained by the conversion rule with the mapping table described above. The motion data about the current frame outputted from the display and the motion data outputted from the motion generator were compared. Due to the motion-to-photon latency, the rotation angle from the display was temporally lagged with respect to the angle outputted from the motion generator. To quantify the motion-to-photon latency, the proposed system calculated the time at which the angle, outputted from the display, matched the angle outputted from encoders of the motion generator. Finally, it calculated the difference, which was the motion-to-photon latency between the two time points that had the same rotation angle. When calculating the time points, the authors found the indices of two rotation buffers with the same rotation angles, as shown in Figure 8. To avoid finding multiple points of the same value, the searching range was limited. There were multiple points that had the same value in all ranges. By limiting the searching range, the method allowed the buffers to be searched in the limited range so that only one same value was selected.

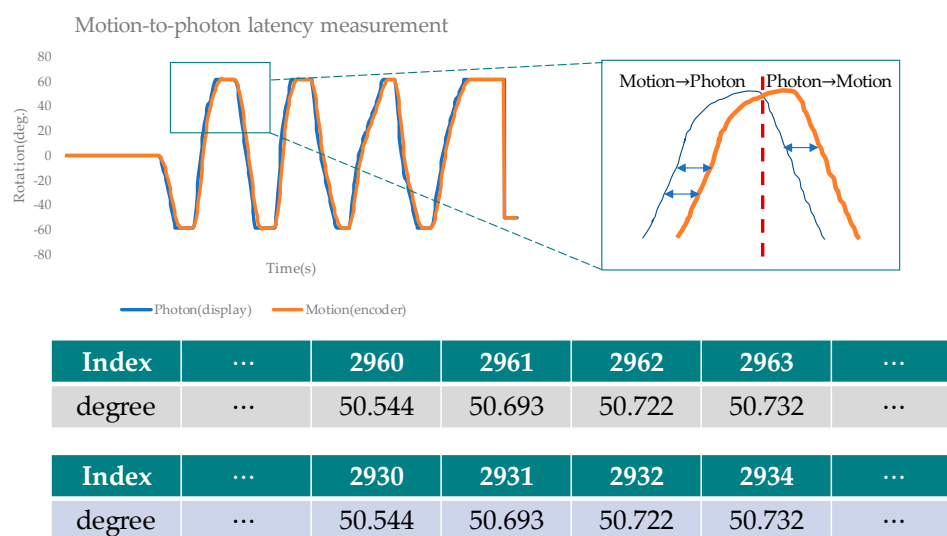


Figure 8. Example of the motion-to-photon latency calculation.

2.5. Real-Time Measurement and Interface

The proposed method used buffers to store the data from the encoders and the display. Specifically, it synchronized and stored all data in an oscilloscope, and the time delays were compared by searching points that had the same values. Once each time difference was found, it was converted to a time value to obtain the final motion-to-photon latency. In addition, a user interface was proposed to control the system and to integrate and run several modules. Figure 9 shows the configuration of the interface. The sequence-based motion generation part provided the information for controlling the

motion generator. In the data plotting part, the graphs for the rotation angles are plotted. The latency measurement part provides the current, mean, and max values for the motion-to-photon latency.

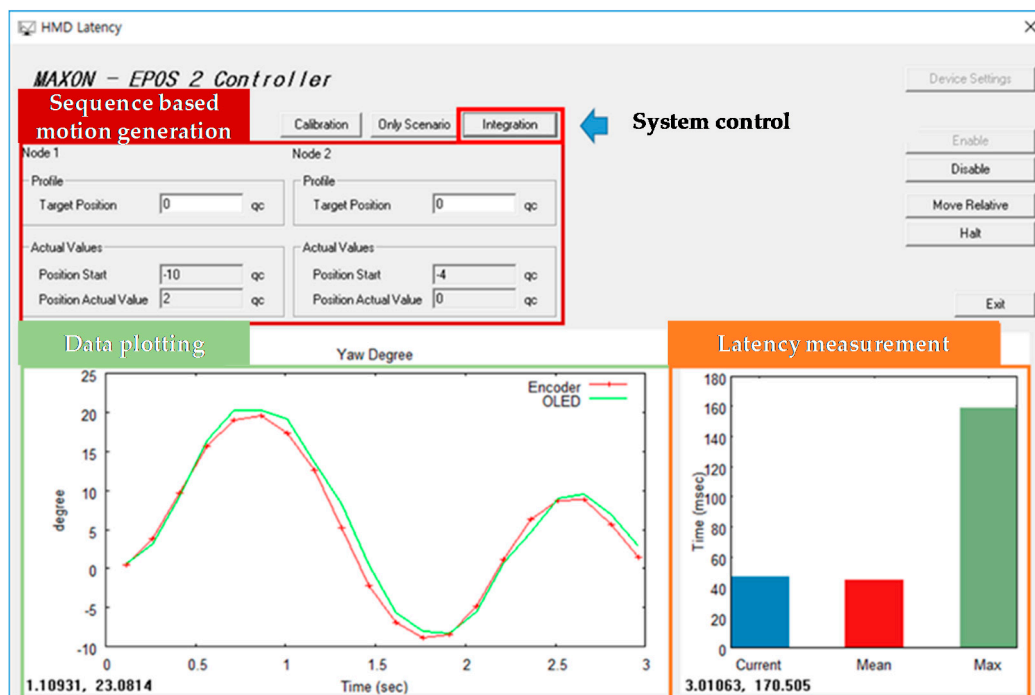


Figure 9. Proposed user interface of the motion-to-photon latency measurement system.

3. Experimental Results

The experimental environment is as follows. First, the authors used an Oculus Rift DK 2 HMD device (Oculus VR, Menlo park, CA, USA) for the VR implementation [18]. To drive a motion generator, two RE 40 (Maxon Motor, Sachseln, Switzerland) [19] were used as a DC motor along two axes, and the same number of EPOS2 50/5 (Maxon Motor, Sachseln, Switzerland) were used as controllers to control them [20]. However, since the position estimated from the motor itself had mechanical latency, additional encoders (EIL 580 Baumer, Frauenfeld, Switzerland), which are of the incremental type with 500 steps/turn and 300 kHz frequency, were used to measure more accurate current rotation angles [21]. SM05PDs (THORLABS, Newton, NJ, USA) were used as the photodetectors, which have a spectral range from 200 nm to 1000 nm [22]. A PicoScope 4824 (Pico Technology, Saint Neots, UK) was used to measure the voltages from encoders and photodetectors [23]. It measured the voltages of the encoder and photodetector with a sampling rate of 100,000 times per second and a bandwidth of 5.0 Gbps, thereby allowing nearly continuous data description and processing in real time. To render the VR scenes and drive the entire proposed system, a PC, which had an Intel i7-6700k @ 4.4 GHz CPU [24] and an NVIDIA Geforce GTX 1080 graphic card [25], were used. The experiment was conducted by the following two methods. First, the authors measured the changes in the latency according to the change in the position sequence while the initial workload was fixed. In this case, the workload was assigned the minimum level. The second method used only one of the motion sequences for different initial workload levels. Figure 10 shows the overall experimental process described above.

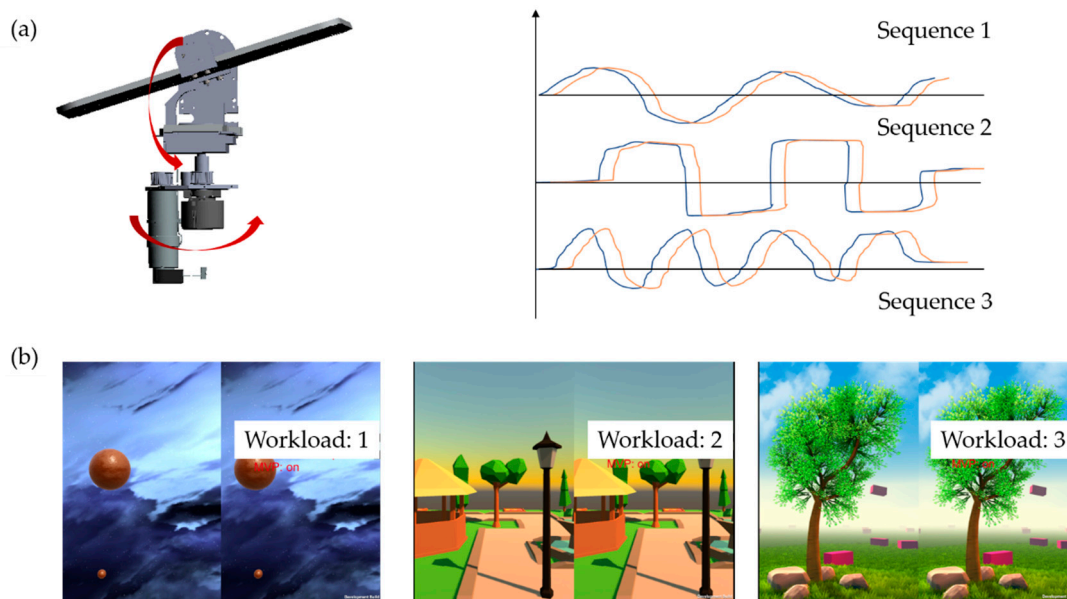


Figure 10. Overall experimental processes for calculating the motion-to-photon latency in (a) different sequences and (b) different workload levels.

3.1. Position Sequence

Table 1 lists the position sequences defined in the experiment for the yaw and pitch axes. Six peak points were inputted to the motion generator controlled by the input angle, and the system was controlled according to the sequence shown in Figure 11. In this case, as shown in Table 2, the average latency was up to 46.55 ms and the maximum latency was up to 63.72 ms. In this experiment, the positions and shapes of the 3D objects in a VR scene were changed according to the orientation of the HMD. By reflecting these changes of the objects, the rendering workload was changed with time, thereby changing the motion-to-photon latency. As shown in Table 2, the motion-to-photon latency was varied according to the motion sequence. These results show that the proposed system could measure the motion-to-photon latency in all temporal ranges when the position sequence changed.

Table 1. Peak points of simulated head orientations in yaw and pitch directions.

Orientation	Yaw		Pitch	
	1	2	1	2
Number of Sequences				
Peak point	1	0.00	0.00	0.00
	2	19.10	13.50	−14.00
	3	5.10	−14.20	−5.00
	4	0.00	−15.00	13.50
	5	−8.90	5.00	6.50
	6	0.00	0.00	0.00

Table 2. Minimum, maximum, and average motion-to-photon latencies by different position sequence.

Orientation	Yaw		Pitch	
	1	2	1	2
Number of Sequences				
Min. (ms)	21.83	15.78	23.42	32.10
Max. (ms)	63.72	69.33	53.75	65.36
Average (ms)	46.55	47.42	43.87	49.45

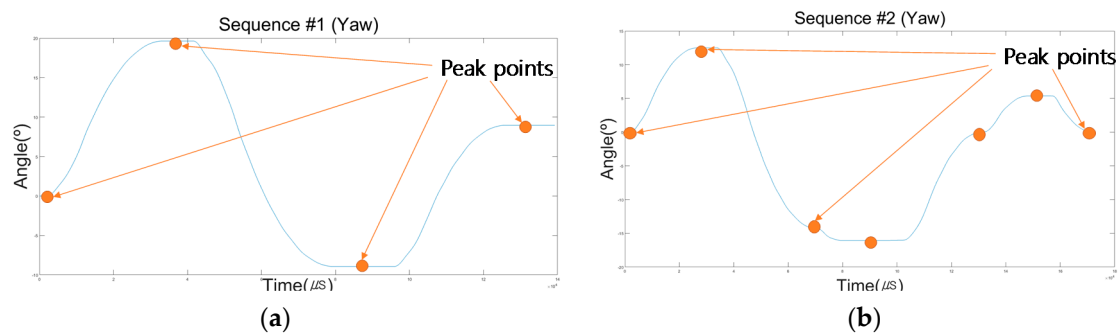


Figure 11. Sequences and peak points for simulated head orientations: the change of angles in (a) sequence #1 and (b) sequence #2.

3.2. Different Workloads

The authors also measured the latency change while varying the initial rendering workload with the fixed position sequence. The steps of each workload were determined by the number of vertices, which is the minimum unit that constitutes the mesh that has the greatest influence on the computation when rendering the image. In the workload of level 0, the minimum number of objects was two and the number of vertices was 800,000. In the workload of level 3, the number of objects was 17, and the number of vertices was 9.5 million. Table 3 summarizes the number of vertices per level. An increase in the number of vertices also increased the computation time, which directly affected the motion-to-photon latency. In the experiment, the minimum, maximum, and average motion-to-photon latency were measured according to the workload level as shown in Table 4. In this case, the average latency measured in the workload with level 0 was up to 46.55 ms, and the maximum latency was up to 63.75 ms. On the other hand, the average latency measured in the workload with level 3 was up to 154.63 ms, and the maximum latency was up to 198.24 ms. Figure 12 shows the results as a function of the workload level. In the figure, the blue points are the rotation angles from a VR scene, and the orange points are the angles from the motion generator. In the cropped image, the blue points were lagged behind orange points, which are the current angles of the motion. As shown in both of the cropped regions, the time delay of the two points was increased when the workload level was increased. Therefore, when the workload level was changed, the time delay also changed significantly.

Table 3. Various rendering workloads and vertices.

Rendering Workload	0	1	2	3
Number of objects	2	7	12	17
Vertices	0.8 M	2.4 M	4.8 M	9.5 M

Table 4. Minimum, maximum, and average motion-to-photon latencies by different rendering workload levels.

Rendering Workload	0	1	2	3
Min. (ms)	21.80	45.30	88.30	120.15
Max. (ms)	63.75	89.75	133.65	198.24
Average (ms)	46.55	63.22	101.29	154.63

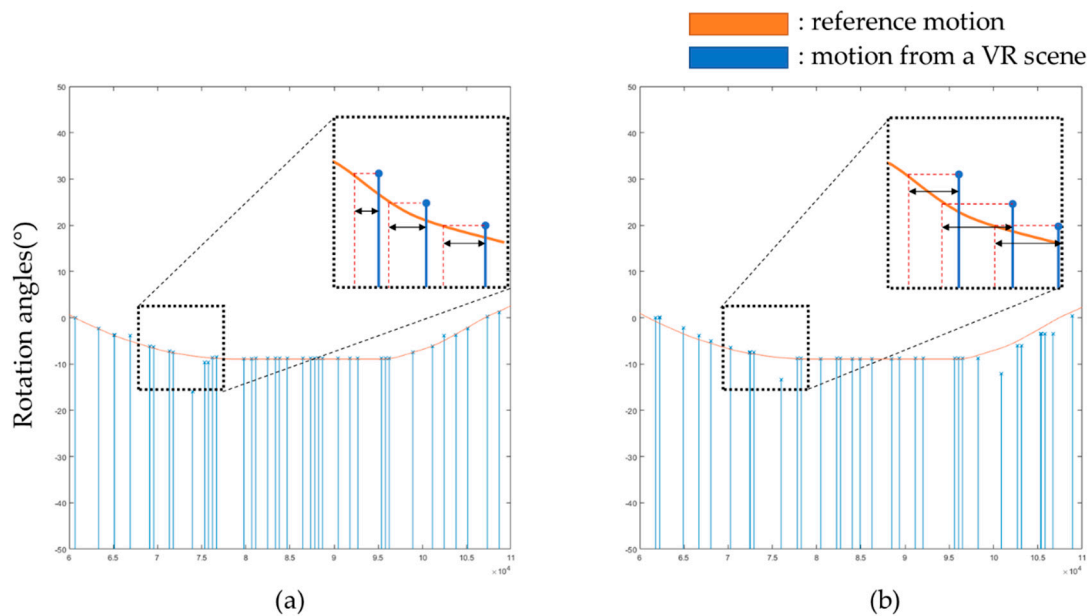


Figure 12. Profiled motion data measured from the physical motion and a virtual reality (VR) output image: (a) workload level 0 and (b) workload level 3.

4. Conclusions

This paper proposed a novel measurement system to visualize the motion-to-photon latency with time-series data in real time. The proposed system rotated the HMD device using a motion generator based on the kinematics of a human neck to control its motions. In addition, we proposed a measurement area, which includes four objects with intensity levels that are converted from an IMU sensor data of the HMD, to acquire the current motion from the display. Finally, the motion data obtained from the encoder were compared with the motion data obtained from the display to calculate the motion-to-photon latency and visualize it on the dedicated interface. In the experiment, the latency was changed from a minimum of 46.55 ms to a maximum of 154.63 ms according to the workload levels.

Author Contributions: S.-W.C., M.-W.S. and S.-J.K. conceived and designed the experiments; S.-W.C., S.L. and S.-J.K. performed the experiments, and analyzed the data; S.-W.C., M.-W.S., S.L. and S.-J.K. contributed the equipment development; S.-W.C. and S.-J.K. wrote the paper.

Acknowledgments: This research was supported by the National Research Foundation of Korea (NRF) through a grant funded by the Government of Korea (MSIT) (No. 2018R1D1A1B07048421), and supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-0-01421) supervised by the IITP (Institute for Information & Communications Technology Promotion).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chang, S.N.; Chen, W.L. Does visualize industries matter? A technology foresight of global Virtual Reality and Augmented Reality Industry. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017.
2. Azuma, R.; Baillot, Y.; Behringer, R.; Feiner, S. Recent advances in augmented reality. *IEEE Comput. Graph. Appl.* **2001**, *21*, 34–47. [\[CrossRef\]](#)
3. Iribe, B. Virtual reality—A new frontier in computing. In Proceedings of the AMD Announces 2013 Developer Summit, San Jose, CA, USA, 13 November 2013.
4. Justin, M.; Diedrick, M.; Stoffregen, T. The virtual reality head-mounted display Oculus Rift induces motion sickness and is sexist in its effects. *Exp. Brain Res.* **2017**, *3*, 889–901.

5. LaValle, S.M.; Yershova, A.; Katsev, M.; Antonov, M. Head tracking for the Oculus Rift. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014.
6. Evangelakos, D.; Mara, M. Extended TimeWarp latency compensation for virtual reality. In Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Redmond, WA, USA, 27–28 February 2016.
7. Tsai, Y.J.; Wang, Y.X.; Ouhyoung, M. Affordable system for measuring motion-to-photon latency of virtual reality in mobile devices. In Proceedings of the SIGGRAPH Asia 2017 Posters, Bangkok, Thailand, 27–30 November 2017.
8. Lincoln, P.; Blate, A.; Singh, M.; Whitted, T.; State, A.; Lastra, A.; Fuchs, H. From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. *IEEE Trans. Visual Comput. Graph.* **2016**, *22*, 1367–1376. [[CrossRef](#)] [[PubMed](#)]
9. Seo, M.; Choi, S.; Lee, S.; Lee, E.; Baek, J.; Kang, S. Photosensor-based latency measurement system for head-mounted displays. *Sensors* **2017**, *17*, 1112. [[CrossRef](#)] [[PubMed](#)]
10. Sharkey, M.; Murray, W.; Heuring, J. On the kinematics of robot heads. *IEEE Trans. Rob. Autom.* **1997**, *13*, 437–442. [[CrossRef](#)]
11. Funda, J.; Taylor, H.; Paul, P. On homogeneous transforms, quaternions, and computational efficiency. *IEEE Trans. Rob. Autom.* **1990**, *6*, 382–388.
12. Denavit, J.; Hartenberg, S. A kinematic notation for lower-pair mechanisms based on matrices. *J. Appl. Mech.* **1955**, *1*, 215–221.
13. Kim, J.; Kumar, R. Kinematics of robot manipulators via line transformations. *J. Rob. Syst.* **1990**, *7*, 649–674. [[CrossRef](#)]
14. Kucuk, S.; Bingul, Z. Robot kinematics: Forward and inverse kinematics. In *Industrial Robotics: Theory, Modelling and Control*; INTECH Open: London, UK, 2006.
15. Eberly, D.H. *3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics*; CRC Press: Boca Raton, FL, USA, 2006.
16. Jack, K.; Tsatsulin, V. *Dictionary of Video and Television Technology*; Gulf Professional Publishing: Houston, TX, USA, 2002.
17. Kariya, T.; Kurata, H. Generalized least squares estimators. In *Generalized Least Squares*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2004; pp. 25–67.
18. Oculus Rift DK 2 Overview of the DK2 and SDK 0.4. Available online: <https://developer.oculus.com/documentation/pcsdk/0.4/concepts/dg-intro-version/> (accessed on 2 September 2017).
19. Maxon RE 40 40 mm, Graphite Brushes, 150 Watt Specifications. Available online: https://www.maxonmotor.com/medias/sys_master/root/8825409404958/17-EN-132.pdf (accessed on 2 September 2017).
20. EPOS2 50/5, Digital Positioning Controller, 5 A, 11 - 50 VDC Specifications. Available online: https://www.maxonmotor.com/medias/sys_master/root/8821075705886/16-421-422-423-425-EN.pdf (accessed on 2 September 2017).
21. EIL580 Mounted Optical Incremental Encoders Specification. Available online: <http://www.baumer.com/es-en/products/rotary-encoders-angle-measurement/incremental-encoders/58-mm-design/eil580-standard> (accessed on 2 September 2017).
22. SM05PD Mounted Photodiodes – SM05 and SM1 Compatible Specifications. Available online: https://www.thorlabs.com/Images/PDF/Vol18_773.pdf (accessed on 2 September 2017).
23. PicoScope 4824 Data Sheet. Available online: <https://www.picotech.com/legacy-document/datasheets/PicoScope4824.en-2.pdf> (accessed on 2 September 2017).
24. Intel Core i7-6700K Processor Specifications. Available online: https://ark.intel.com/products/88195/Intel-Core-i7-6700K-Processor-8M-Cache-up-to-4_20-GHz (accessed on 2 September 2017).
25. Geforce GTX 1080 Graphic card Specifications. Available online: <https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080/> (accessed on 2 September 2017).

