An Implementation of Surveillance Systems with Dynamic Transaction Intervals under PQI Approach

Chaxiong Yukonhiatou[1,a)] Tomoki Yoshihisa[2,b)] Tomoya Kawakami[3,c)] Yuuichi Teranishi[4,2,d)]
Shinji Shimojo[2,e)]

Abstract: Surveillance systems (e.g., humans and cars detection systems) become popular and deployed in several places such as airports, banks, and parking spaces. In these systems, the cameras continuously send their recorded video data to the remote processing computers for real-time collection or analyses. Transaction rate is one of main factors to improve the performances of the applications for surveillance systems. For example, the probability to catch the intruders can increase as the processing computers analyze video data got from the cameras with a higher transaction rate. The previous simulation results shown that the transaction rates can be improved under our proposed method compared with the conventional approach. However, in the practical situation, it is necessary to confirm the transaction rates with some experimental results. Therefore, the goal of this paper is experimenting our proposed method in a surveillance system in order to get the experimental results. In this experiment, we measure the number frame rates instead of transaction rates of our proposed method. The experiment results show that the proposed method is able to improve the transaction rate in our implemented system.

Keywords: dynamic interval, transaction rate, surveillance system, progressive quality improvement

1. Introduction

Nowadays, the surveillance systems are widely used in various places such as airports, banks or supermarkets. In these systems, each camera continuously send its recorded video data to the collectors (e.g., storages, processing computers) for collecting or analyzing. The performances of applications of surveillance systems can improve by analyzing the data got from cameras with a higher transaction rate (a process for video data generation and analysis). For example, detecting an intruder in the surveillance systems, the processing computers can faster find the burglars with analyzing the video data got from cameras with a higher transaction rate. To improve transaction rates, we have proposed a method under a progressive quality improvement (PQI) approach. In this method, the processing computer dynamically changes the transaction intervals depending on the transaction rates. Here, a transaction includes a process for data generation and analysis. Moreover, the proposed method adopted a processing quality improvement (PQI) approach to reduce the communication and transaction times.

The transaction rates of surveillance systems can improve by using a higher computational power of processing computers or a higher communication bandwidth [1]-[5]. However, the existing surveillance systems send their video data with a constant transaction interval such as 30 [fps]. Hence, the transaction rate decreases. Therefore, transaction rates further improve by using our proposed method in order to dynamically change the transaction intervals.

In this paper, we implement a system to improve transaction rates in the surveillance systems under a PQI approach. In our implemented system, each recording computer can produce images that contain some qualities. The lowest quality has the highest priority to be sent to the processing computers. Only in the cases when the higher qualities are needed for analysis, the processing progressively collect them. For example, humans or cars recorded in the image.

The rest of this paper is organized as follows. In Section 2, we introduce some works that are related to our system. In Section 3, we briefly explain TRDI under PQI approach. Our implementation is explained in Section 4, and experimental results show in Section 5. Finally, we will conclude the paper in Section 6.

2. Related Work

In [6], a method to improve transaction rates has been proposed. In this method, the processing computer dynamically changes the transaction interval depending on the transaction rates. The results shown that the transaction rates can improve under this method. However, in the practical situations, it is necessary to confirm with the experimental results. In this paper, we further investigate the experimental results using the same

1 Graduate School of Information Science and Technology, Osaka University, Ibaraki, Osaka 567-0047, Japan
2 Cybermedia Center, Osaka University, Ibaraki, Osaka 567-0047, Japan
3 Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan
4 National Institute of Information and Communications Technology, Koganei, Tokyo 184-8795, Japan
a) chaxiong@ais.cmc.osaka-u.ac.jp
b) yoshihisa@cmc.osaka-u.ac.jp
c) kawakami@is.naist.jp
d) teranisi@cmc.osaka-u.ac.jp
e) shimojo@cmc.osaka-u.ac.jp

method. The difference is this paper shows the experimental results for a surveillance system but the previous results used simulated data.

Systems methods for changing transaction interval have been proposed [7]-[10]. Most of them adopt a simple approach in that processing computers request the changes of transaction intervals to data sources every a fixed number of transactions. However, excessively short transaction intervals cause high computational loads on processing computers and thus the transaction rates decrease. In our proposed method, the transaction intervals dynamically change depending the transaction rates.

In addition, some implementation method to improve the performances of surveillance systems have been proposed in [11]-[12]. However, in these method the transaction intervals are constant. In our implementation, the transaction intervals are dynamically change.

3. PQI-TRDI Method

In this section, we explain our PQI and PQI-TRDI approach. The PQI approach reduces the communication and transaction times by progressively collect data from the lowest quality to the highest quality. The PQI-TRDI approach is a method to improve the transaction rates under the PQI approach. This method dynamically changes transaction intervals depending on the transaction rates.

3.1 Summary of PQI Approach

In PQI approach, each camera or recording computer can generate its recorded images in the progressive JPEG format, which contains multiple images with different qualities (called scans in progressive JPEG). The lowest quality image data (first scan) has the highest priority to be sent to the processing computers. First, the processing computers receive the lowest quality image data and analyze the difference from the previous frame. The processing computer skips collecting and analyzing the next lowest quality image data (next scan) when it is meaningless. Only in the cases where higher-quality image data are needed for analysis (the processing computers judge to request higher quality when objects are recorded in the image data), the processing computers progressively collects the higher quality image data from the cameras or recording computers. Otherwise, the processing computer receives only the lowest quality image.

3.2 Communication Chart of PQI Approach

Figure 1 shows a timing chart for stream processing under the conventional approach and the PQI approach. In the PQI approach, the data $D_{n,a}(t)$ $(n = 1, 2,\ t = 1, \cdots)$ are divided into some qualities. Here, $D_{n,a}(t)$ is the observed original data of the camera n at the cycle t. $D_{n,q}(t)$ $(q = 1, \cdots, Q)$ is the generated data from $D_{n,a}(t)$ of that quality is q. Each transaction includes some processes for each divided data. In the figure, the number of the qualities is 2 and the transaction consists of two processes for the divided data $D_{n,1}(t)$ and $D_{n,2(t)}$. n is the camera number, t is the cycles for data collections, and q is the quality. In the cycle 1, the transaction finishes at the first quality in both streams. In the cycle 2, the processing computer requires $D_{1,2}(2)$ when it

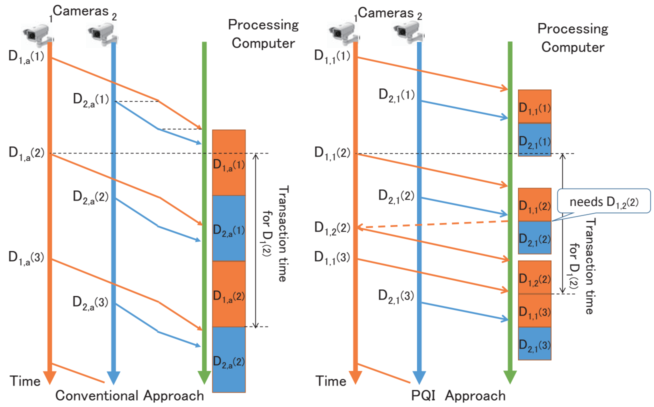

Fig. 1: Communication diagram for conventional approaches and of the PQI approach

finishes the process for $D_{1,1}(2)$. The camera 1 transmit the required $D_{1,2}(2)$ and the processing computer starts the process for $D_{1,2}(2)$. In this case, the transaction finishes when the processing computer finishes the process for $D_{1,2}(2)$ since the number of the qualities is 2. The transaction time in this case is reduced compared with that under the conventional approach as shown in the figure.

3.3 Transaction Rate-based Dynamic Interval (TRDI)

The PQI-TRDI method determines when and how long the processing computer changes the intervals under the PQI approach.

In the PQI-TRDI method, the processing computer changes the transaction interval when the average transaction rate for some previous transactions changes largely in order avoid frequent changes of transaction intervals.

3.4 Transaction Intervals Determination

In the PQI-TRDI method, the processing computer calculates the average transaction time for the previous C_n transactions $(AveTT_n(t) = \sum_{\tau=t-C_n+1}^{C_n} TT_n(\tau))$ and set the average value as the new transaction interval. This is because a longer transaction interval causes a less transaction rate and a shorter transaction interval has a large possibility to increase transaction times.

3.5 Data Generations in the Recording Computer

Figure 2 shows the flow chart of cameras. Each cycle comes every the transaction interval for the camera n (I_n) passes. When the t th cycle starts, each camera n $(n = 1, \cdots N)$ gets $D_{n,a}(t)$ from their sensors and temporarily stores it to their storages. First, they generate $D_{n,1}(t)$ from $D_{n,a}(t)$ and send $D_{n,1}(t)$ to the processing computer. When the camera n receives the request of $D_{n,q}(t)$, it generates $D_{n,q}(t)$ from stored $D_{n,a}(t)$ and sends $D_{n,q}(t)$ to the processing computer. When the camera n receives the request of changing interval to i, it changes its interval to i and rearranges the start of the next cycle.

3.6 Data Processing in the Processing Computer

Figure 3 shows the flow chart of the processing computer. When the processing computer receives $D_{n,q}(t)$, it processes $D_{n,q}(t)$. When $q = Q$, $D_{n,q}(t)$ is the final quality data and the transaction of t th cycle finishes. Otherwise, the processing com-

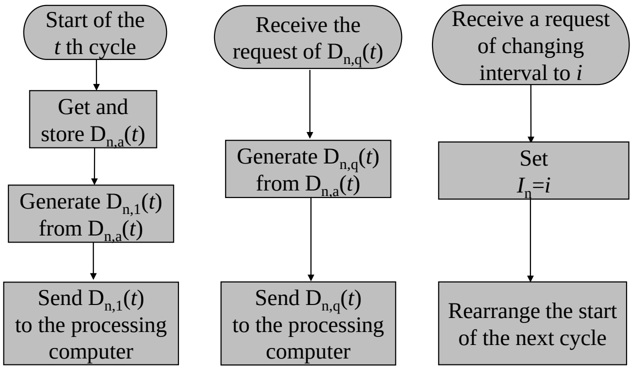

Fig. 2: The flowchart for the data generation

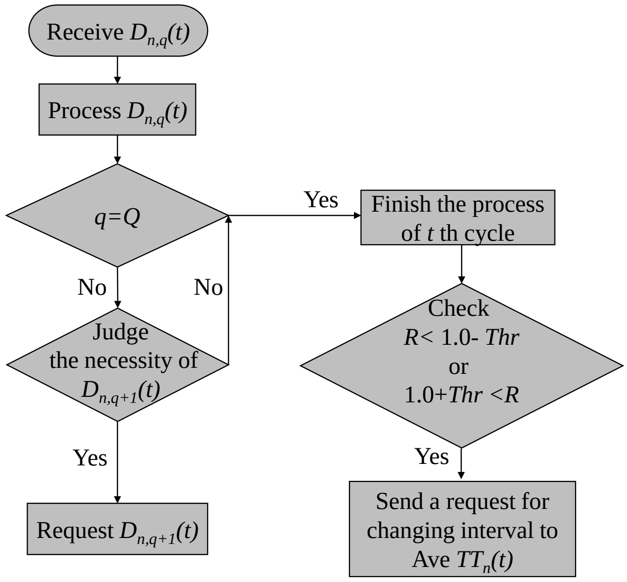

Fig. 3: The flowchart for the processing computer

puter judges the necessity of $D_{n,q+1}(t)$. In case that $D_{n,q+1}(t)$ is needed for the process execution, the processing computer requests $D_{n,q+1}(t)$ to the camera n. Otherwise, the transaction of t th cycle finishes. When a transaction finishes, the processing computer checks the average transaction rate for previous C_n cycles to judge whether the transaction interval should be changed or not. $TR_n(t)$ denotes the transaction rate of the t th cycle for the camera n and is given by the inverse value of the transaction time $TT_n(t)$, i.e., $TR_n(t) = 1/TT_n(t)$. The average value $AveTR_n(t)$ is given by the following equation.

$$AveTR_n(t) = \frac{\sum_{\tau=t-C_n+1}^{C_n} TR_n(\tau)}{C_n} \qquad (1)$$

The processing computer calculates R, which is the changing ratio of the average transaction rates of the current cycle and the previous cycle. That is:

$$R = \frac{AveTR_n(t)}{AveTR_n(t-1)} \qquad (2)$$

The value of R that largely differs from 1.0 means that the average transaction rate changes largely. Therefore, in the PQI-TRDI method, if the following inequality is satisfied, the processing computer requests the change of the transaction interval to the data source n. Here, Thr is the threshold for changing the transaction interval.

$$R < 1.0 - Thr \quad or \quad 1.0 + Thr < R \qquad (3)$$

The PQI-TRDI method use two parameters C_n and Thr for calculating the average transaction rate to change transaction intervals.

A smaller cycle C_n causes a more frequent change of transaction intervals since the number of the transactions for calculating the average value decreases. On the other hand, a larger cycle C_n causes a smaller transaction rate since the possibility to change transaction intervals decreases. Therefore, the middle value of C_n improves both a less change and a higher transaction rate.

A smaller Thr causes a more frequent change of transaction intervals since the processing computer requires the change of transaction intervals even if the changing rate of the transaction times is small. On the other hand, a larger Thr causes a smaller transaction rate since the possibility to change transaction intervals decreases. Therefore, similar to C_n, the middle value of Thr improves both a less change and a higher transaction rate.

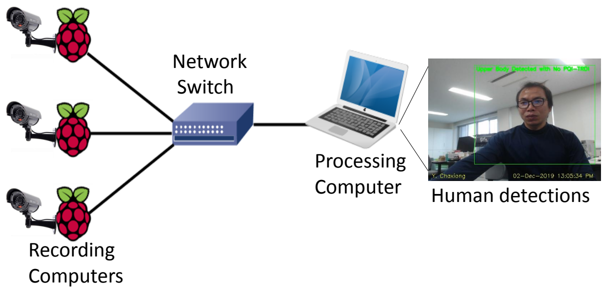

Fig. 4: System architecture for our implementation

4. Implementation

This section, we describe our implementation for PQI-TRDI approach.

4.1 Our System Architecture

Figure 4 shows our implemented system. The recording computers are based on the Raspberry Pi 3 Model B with Camera Module V2 and connect to a Laptop as a processing computer via 100BASE-TX/1000BASE-T network (Allied Telesis CentreCOM GS908GT switch). We used the Python programming language and implemented as a upper human body detection system. Each pi-camera gets every image frame with 480 × 340 resolution from its sensor and encodes into progressive JPG format, which contains 10 different qualities (called scans in progressive JPEG) as show in the Figure 5. These generated qualities are temporarily stored in the memory. Firstly, each recording computer send the lowest scans to the processing computer than other scans. The processing computer detects upper human bodies in the received scans. If the processing computer detects human bodies in the firstly received scan, the processing computer requests to the cameras to get the remaining scans (the higher quality image data) and progressively collects them. Otherwise, the processing computer skips collecting the higher scans. Table 1 shows the specifications of our implemented system.

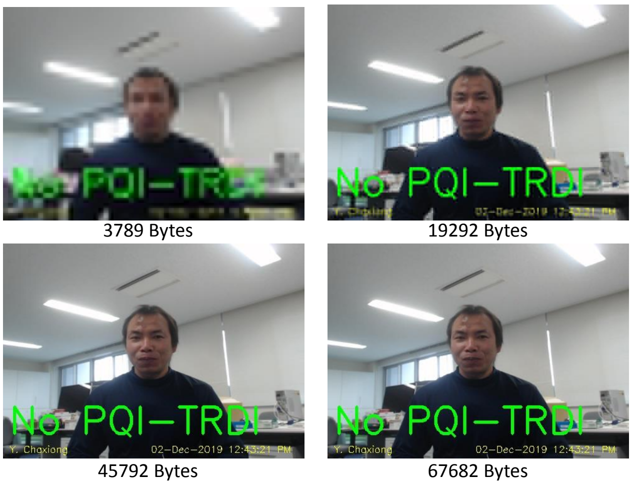

Fig. 5: Images with different qualities

Table 1: Specifications of our implemented system

Items	Details
Recording computer	Raspberry Pi 3 Model B (1.2GHz quad-core ARM Cortex-A53, 1GB memory, 100BASE-TX, Raspbian Ver 10. Sep. 2019)
Camera device	Raspberry Pi Camera Module V2
Processing computer (Windows 10 Pro 64bit)	Dell Latitude E7240 (2.10GHz dual-core Intel Core i7-4600U, 8GB memory, Intel HD Graphs 4600, 1000BASE-T
Network	Allied Telesis CentreCOM GS908GT (100BASE-TX/1000BASE-T), CAT5e cables
Cables	CAT5e, 1m length
Num. of recording computers	3
Num. of processing computers	1
Num. of qualities in prog. JPEG	10
Comparison methods	Without PQI approach
Evaluation items	Frame Rate [FPS]

4.2 System Communications

Figure 6 shows the communication of our implemented system. In this implementation, we use two channels for the communications, communication channel and data transfer channel. The common protocol TCP/IP is used for establishing of processing computer and cameras and requesting the higher quality data. The RTP ((Real-time Transport Protocol) is used for transferring the image data between recording computer and the processing computer. In our system, the processing computer first connects to the recording computers via a TCP/IP communication (TCP socket). The recording computers makes connection to the processing computer in second step. If the connection completed, the processing computer opens RTP channel and waiting for receiving RTP packets in the third step. If the connection confirmation is established, the processing computer requests for image data is started in step fourth. In step fifth, when the recording computer receives requests for image data, it gets image frames from its camera's sensor and generates them into 10 qualities (10 scans) for every single frame. The lowest quality (first scan) data is encoded into RTP packets and sends to the processing computer via data channel in step sixth. Finally, the processing computer receives RTP packet and decodes the image data for analysis. In the case that a higher quality image data is needed, the processing computer sends the requests to the recording computers via communication channel.

4.3 How to Generate and Request the Higher Quality

In this section, we explain how to generate different qualities and requesting the higher quality data of every single image frame.

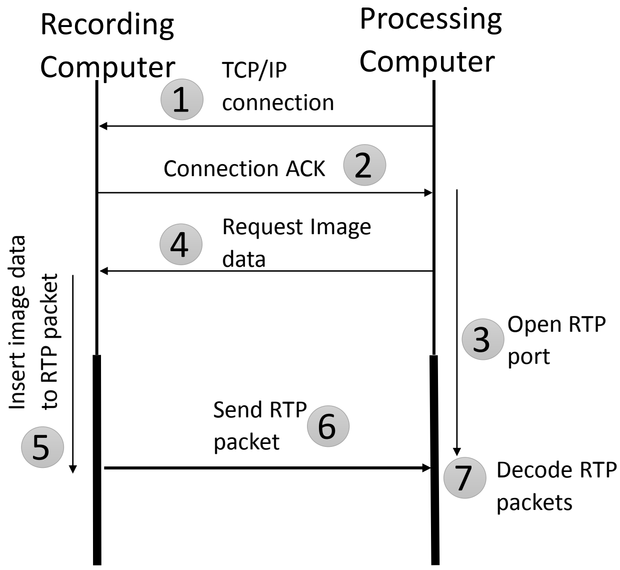

Fig. 6: The communication diagram for our implemented system

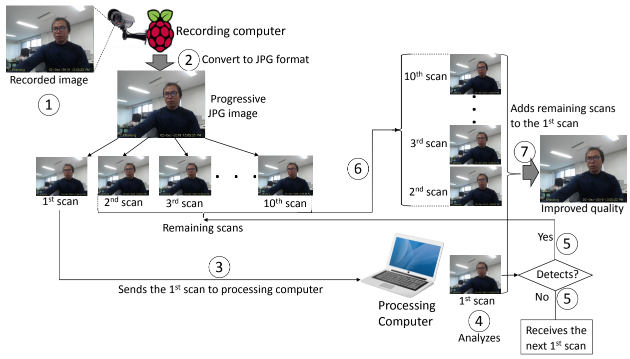

Fig. 7: How to generate and request a higher quality image

Figure 7 shows how to generate some qualities from every single image frame and requesting the higher quality image data. First, the recording computer gets the raw image frame from its camera's sensor. Second, the recording computer encodes the raw image frame into processing JPG format with 10 qualities (called scans in progressive JPG) using OpenCV (a popular programming library for the computer vision field) and temporarily stores in its buffer. Third, the first scan (first quality) is sent to the processing computer. Fourth, when the processing computer receives the first scan, it checks whether a human is detected or not. Fifth, in the case that a human is detected in the first scan, the processing computer requests to the recording computer in order to get the remaining qualities and progressively collect them in the step sixth and seventh, respectively. Finally, the processing computer adds the remaining scans to the first scan in order to get the improve quality image. In case there is not a human detected in the first scan in step fifth, the processing computer waits for receiving for the next frame (the first scan of the next image frame).

Figure 8 shows an example of data collection under our implementation using PQI-TRDI approach. Three cameras are there for detecting upper human body by using our approach. In PQI-TRDI approach, the processing computer collects and analyzes

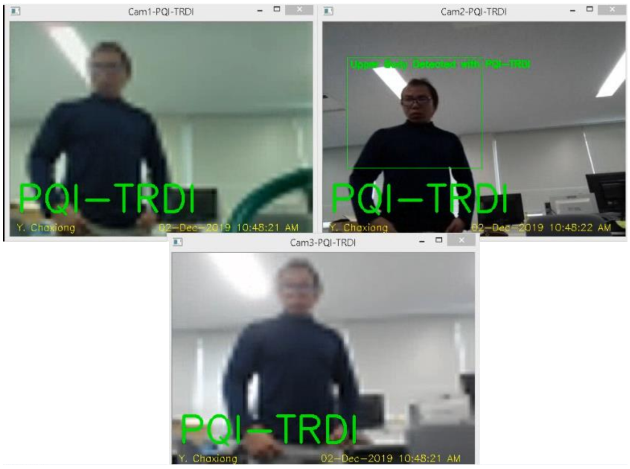

Fig. 8: Images under PQI-TRDI method

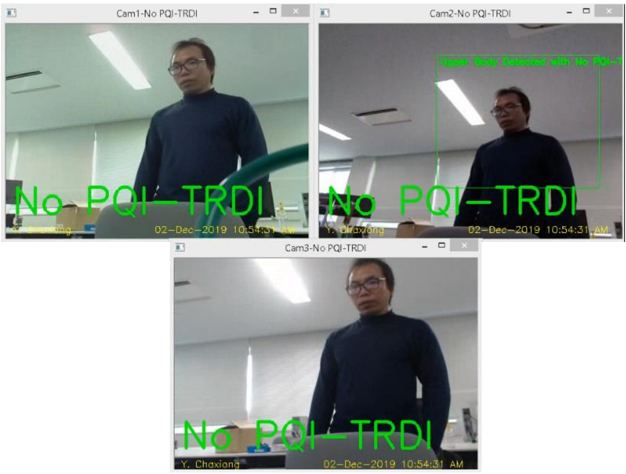

Fig. 9: Images under no PQI-TRDI method

image data in the lowest quality with dynamically changes the transaction intervals depending the transaction rates. In the cases that a human is detected in the lowest quality, the processing computer requests to get the remaining qualities and progressively collects them in order to improve the quality of image data. As shown the figure, Cam2-PQI-TRDI detects a human so that the quality is improved. The qualities image of other cameras are not improved since they are mis-detection.

Figure 9 shows an example of data collection under the conventional approach. Each camera send its original image data without dividing into some quality and transfer image data with a constant transaction interval such 30 fps. The processing computer analyzes the received image data with a clear image (a high quality image). As shown in the figure, each image is a clear image even there are not human detected such Cam1-No PQI-TRDI and Cam2-No PQI-TRDI.

4.4 Transactions

Each transaction includes some processes for each generated scans (quality). In our implementation, each image frame contains 10 scans (10 qualities). Therefore, the transaction consists 10 transactions when the higher qualities are need. In the case that the higher qualities are not needed, the transaction consists only 1 transaction. The transaction time is time from starting getting scans from a camera's buffer to finishing processing all the transmitted scans at the processing computer. The transaction rate is a inverse value of the transaction time.

Table 2: Parameter values

Communication Bandwidth	approx.100 [Mbps]
Communication Protocol	RTP over UDP
3 Raspberry Pi devices	live camera
Laptop	1
Image Resolution	480 × 340
Progressive JPG	10 scans
Initialize Frame Rate	30 fps
Threshold Rate	0.05
Image Analysis	upper human body detection (HAAR)

4.5 How to Change the Transaction Intervals

In this implementation, we use 4 parameters for changing the transaction intervals. Here, the parameters are cycle (C_n), average transaction rate $AveTR_n(t)$, threshold (Thr) and average transaction time $AveTT_n(t)$. We judge to change the transaction intervals according to Eq.3. The new transaction interval is equal to the average transaction time $AveTT_n(t)$.

5. Experimental Results

5.1 Experiment Setup

We use three Raspberry Pi devices connect to a laptop. Some parameters used in this implementation shows in Table 2.

5.2 Transaction Rate Experiment

The performances of applications increase as the processing computer analyzes image data with a higher transaction rate. We check the average interframe in the processing computer instead of transaction rates.

Figure 10 shows the average frame rate under 1 camera and 1 processing computer. The horizontal axis is the elapsed time [sec.] and the vertical axis is the average frame per second [fps]. The communication bandwidth is approx.100 [Mbps]. The image resolution is 480 × 340. The initialized frame rate is 30 fps. The threshold (Eq.3) for changing transaction interval is 0.05. We calculate the average transaction rate for every 1 cycle. The upper human body analysis is implemented in the processing computer. We can see that the average frame rate under PQI-TRDI and No PQI-TRDI are almost similar approx. 10 fps. When a human is detected, the average frame rate of both approaches reduce to approx. 8 and 9 fps. The average frame under PQI-TRDI is similar to under No PQI-TRDI because the communication bandwidth and processing computer has more extra capacities for processing the image data.

Figure 11 shows the average frame rate under 2 cameras and 1 processing computer. The horizontal axis is the elapsed time [sec.] and the vertical axis is the average frame per second [fps]. The other parameters are the same. We can see that the average frame under PQI-TRDI approach slightly increases compared with the average that under the No PQI-TRDI approach. This because the PQI-TRDI approach dynamically changes the transaction interval based on the average transaction time.

Figure 12 shows the average frame rate under 3 cameras and 1 processing computer. The horizontal axis is the elapsed time

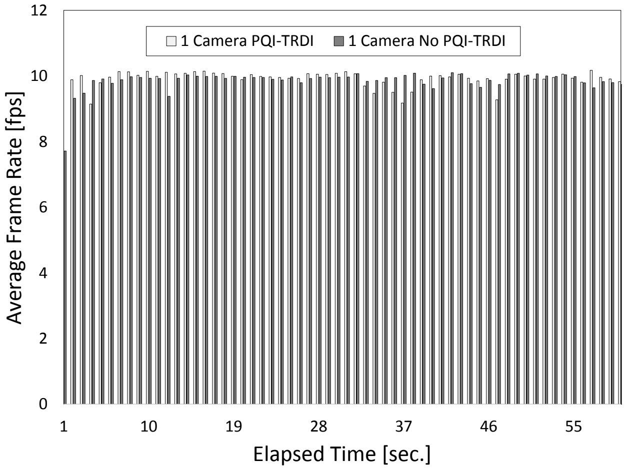

Fig. 10: Average frame rate when the number of the cameras is 1

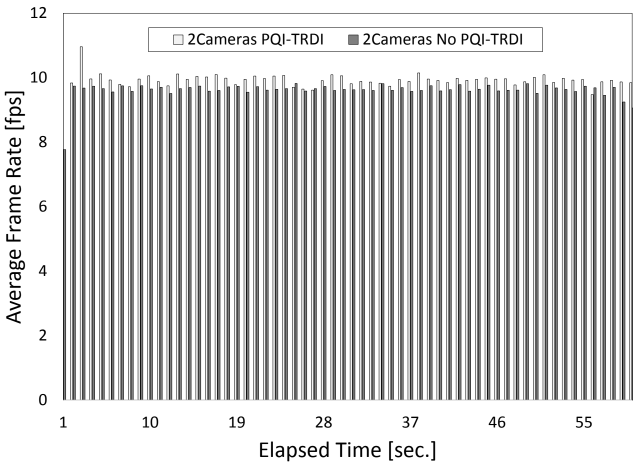

Fig. 11: Average frame rate when the number of the cameras is 2

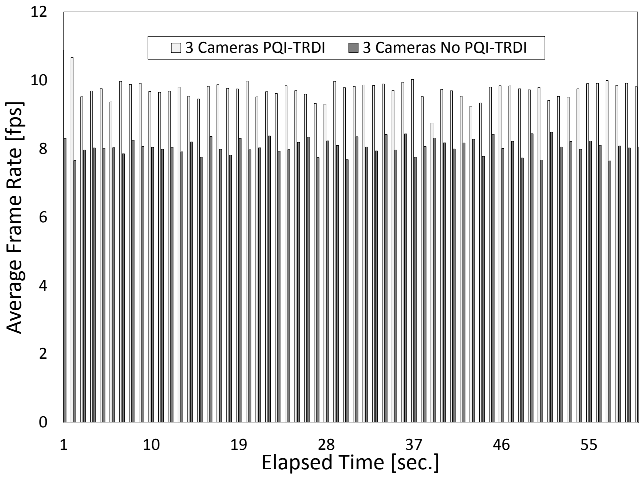

Fig. 12: Average frame rate when the number of the cameras is 3

[sec.] and the vertical axis is the average frame per second [fps]. The other parameters are the same. We can see that our proposed method gives higher frame rate than under the No PQI-TRDI approach since the PQI-TRDI dynamically changes the transaction intervals.

6. Conclusion

In this paper, we investigated the transaction rate with dynamic transaction interval. The experimental results shows that our proposed method PQI-TRDI can improve the transaction rates for the actual surveillance systems. We can see that the average frame rate under PQI-TRDI approach is higher than the average frame rate of No PQI-TRDI approach in case the number of cameras increase. This is because the PQI-TRDI dynamically changes the transaction intervals depending on the transaction rates.

In the future, we plan to implement a system with multiple processing computers using this proposed method and consider about other dynamic interval determination.

Acknowledgement

This research was supported by a Grants-in-Aid for Scientific Research(C) numbered JP17K00146 and JP18K11316, and by I-O DATA Foundation.

References

[1] Wahyono, A. Filonenko and K. Jo, "Designing interface and integration framework for multi-channels intelligent surveillance system," 2016 9th International Conference on Human System Interactions (HSI), Portsmouth, pp. 311-315, July 2016.

[2] R. Bansod, S. Kadarkar, R. Virk, M. Raval, R. Rashinkar and M. Nambiar, "High Performance Distributed In-Memory Architectures for Trade Surveillance System," 2018 17th International Symposium on Parallel and Distributed Computing (ISPDC), Geneva, pp. 101-108, June 2018.

[3] Hongliang Wang, Wenyu Cai, Junyi Yang and Qi Chen, "Design of HD video surveillance system for deep-Sea biological exploration," 2015 IEEE 16th International Conference on Communication Technology (ICCT), Hangzhou, pp. 908-911, Oct. 2015.

[4] T. Tsai and C. Chang, "A High Performance Object Tracking Technique with an Adaptive Search Method in Surveillance System," 2014 IEEE International Symposium on Multimedia, Taichung, pp. 353-356, Dec. 2014.

[5] A. Premadi, B. Ng, M. S. Ab-Rahman and K. Jumari, "Real Time Optical Network Monitoring and Surveillance System," 2009 International Conference on Computer Technology and Development, Kota Kinabalu, pp. 311-314, Nov. 2009.

[6] C. Yukonhiatou, T. Yoshihisa, T. Kawakami, Y. Teranishi and S. Shimojo, "A Dynamic Intervals Determination Method Based on Transaction Rates for Real-Time IoT Applications," 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), Milwaukee, WI, USA, pp. 7-12, July 2019.

[7] X. Wang, A. Jabbari, R. Laur and W. Lang, "Dynamic control of data measurement intervals in a networked sensing system using neurocomputing," 2010 Seventh International Conference on Networked Sensing Systems (INSS), Kassel, pp. 77-80, June 2010.

[8] J. Han, J. Wang and P. Wang, "On robust stability of dynamic interval systems with multiple time-delays," 2012 International Conference on Systems and Informatics (ICSAI2012), Yantai, pp. 239-243, May 2012.

[9] M. Li, W. Tang and M. Yuan, "Reliability-based topology optimization of interval parameters structures with dynamic response constraints," 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, pp. 469-474, Aug. 2014.

[10] S. Siddiqui, A. A. Khan and S. Ghani, "Investigating dynamic polling intervals for wireless sensor network applications with bursty traffic," 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, pp. 448-451, Nov. 2017.

[11] T. Tsai and C. Chang, "A High Performance Object Tracking Technique with an Adaptive Search Method in Surveillance System," 2014 IEEE International Symposium on Multimedia, Taichung, pp. 353-356, Dec. 2014.

[12] A. Premadi, B. Ng, M. S. Ab-Rahman and K. Jumari, "Real Time Optical Network Monitoring and Surveillance System," 2009 International Conference on Computer Technology and Development, Kota Kinabalu, pp. 311-314, Nov. 2009.

