An Implementation of PQI-based Surveillance Systems Changing Transaction Intervals Cyclically

CHAXIONG YUKONHIATOU[1] TOMOYA KAWAKAMI [3] TOMOKI YOSHIHISA[2]
YUUICHI TERANISHI[4,2] SHINJI SHIMOJO [2]

Abstract: Due to the widespread monitoring spaces such as banks, airports, building lobbies, stores, and other public places. In these systems, each camera continuously sends its recorded video data to the processing computers to detect the specific objects. The effectiveness of surveillance systems can be enhanced by speeding up the transaction rate which means the number of the processed video frames per unit time. In the case of burglary, for example, the rate of arrests increases as the processing computer can analyze more video data from the surveillance cameras. As our previous work, the simulation results show that the transaction rates can be improved under our proposed method compared with the conventional approach. The contribution of this paper is an implementation using our proposed method, called PQI-CDI (cycle-based dynamic interval). In the PQI-CDI method, the transaction interval dynamically changes depending on the transaction time. We adopt the PQI approach to further reduce the transaction time. In this paper, we describe an implementation of another PQI-based surveillance system. The experiment results show that the proposed method can improve the transaction rate in our implemented system.

1. Introduction

Nowadays, the monitoring systems (e.g., human monitoring systems, car monitoring systems) are widely used in various places such as banks, airports, building lobbies, stores, and other public places. In these systems, each camera continuously sends its recorded video data to the processing computers for analyzing to detect objects. The speed of surveillance systems to catch a thief can improve by analyzing the data got from cameras with a higher transaction rate (a process for video frame generation and analysis per unit time). For example, the probability to catch the burglars increases as the processing computer analyzes the video frame got from cameras with a higher transaction rate.

A powerful processing computer further improves the transaction rate. For example, larger computational and communication capacities. However, it is limited and give the limitation to the transaction rate. In the stream data processing for object detection with a powerful processing computer, it can accelerate the execution processes such as processing time. Generally, the video frames are sent from cameras via the computer network to the processing computers. Therefore, the transaction rate for object detection in the surveillance systems cannot improve even there is a high computational processing computer. This causes a possibility to miss detect since the objects are moving.

To improve the transaction rate for object detection in the surveillance systems, there are many studies [1]-[3]. However, in the existing surveillance systems, cameras send their video data with a constant transaction interval such as 30 [fps]. Hence, transaction rate improvement is required for object detection. As our previous work, we proposed a method to improve the transaction rate. This method the transaction interval dynamically changes depending on the transaction time. The processing computer requests for changing the transaction interval with a fixed period. The processing computer changes the transaction intervals of each camera every finishing some transactions. Dynamic transaction interval is one of the main factors to improve the transaction rates for surveillance systems. Therefore, the transaction rate further improves by using our proposed method (PQI-CDI) to change the transaction intervals dynamically.

In this paper, we implement a simple surveillance system using the PQI-CDI method. This aims to improve the transaction rate in the surveillance systems by changing the transaction intervals dynamically. In our implementation, each camera can generate 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, when humans or cars recorded in the image, processing computer requests to get the remaining qualities.

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 our proposed method. 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

To improve the performances of surveillance systems for object detection, some methods have been proposed.

[1] Graduate School of Information Science and Technology, Osaka University
[2] Cybermedia Center, Osaka University
[3] Graduate School of Engineering, University of Fukui
[4] National Institute of Information and Communications Technology

A method to reduce the amount of data to be transferred cameras to the processing computers was proposed in [4].

In [5], a surveillance system implementation Based on H.264, SIP, RTP/RTCP, and RTSP. This implementation aims to reduce traffic congestion on the network. It actually gives good video quality in the processing computers since the video images are encoded by H.264 video coding standards. However, this implementation doesn't consider the transaction rate. In our implementation, we further improve the transaction rate.

An implementation was proposed in [6]. This implementation focuses on detecting objects, grouping objects, and speed changing objects as suspicious moving objects or non-moving. They use a background subtraction technique to employ in their work. In most of this work, they focus on image processing in the processing computers side. However, the video frames are transmitted from cameras. If the video frames suffer delays on transmission, it causes miss detection for moving objects. In our implementation, we reduce the transaction time for sending video frames between cameras and the processing computer by using the PQI approach.

In [7], a video-encoding method was proposed. This method uses a standards-compliant video-encoding for object detection in order to improve the performance of compressed videos. However, this causes further delays since the video needs to decompress the video data before detecting objects.

A method to improve transaction rates was proposed in [8]. 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 practical situations, it is necessary to confirm the experimental results. In this paper, we further investigate the experimental results using the same method. The difference is this paper shows the experimental results for a surveillance system but the previous results used simulated data.

In [9], a method to improve the performance of application for object detection in the surveillance systems was proposed. This method uses the updated background model of inter-frame differences to obtain the moving objects in the client. This method is similar to the PQI-based surveillance system in the point that the background of the image is updated (progressively collects the remaining data) when an object has been detected. This method can reduce the transaction time so that the performances of the system get improve. However, in our implementation, the transaction interval dynamically changes in order to improve the transaction rate. Moreover, we adopt the PQI approach to reduce the transaction time.

In addition, some implementation methods to improve the performances of surveillance systems have been proposed in [10]-[11]. However, in these methods the transaction intervals are constant. In our implementation, the transaction intervals dynamically change.

3. Proposed Method

In this section, we explain our proposed method.

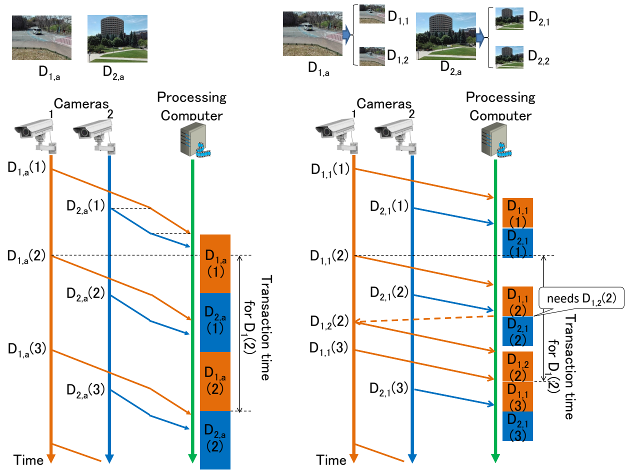

Fig. 1: Communication diagram for conventional approaches and of the PQI approach

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) [12]. 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 computer 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 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 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 PQI-CDI Method

In this section, we will explain the PQI-CDI method. As described in section 3.2, the PQI approach reduces the transaction time. Under the PQI approach, the transaction time dynamically changes due to the depends on quality data. For example, the transaction time is short when higher quality data is not needed and the transaction time is long when higher quality data is needed.

3.3.1 Cycle-based Dynamic Interval Determination

Figure 2 shows the difference between static transaction interval and PQI-CDI method. To make the example simple, we show the transaction rate of 1 camera and 1 processing computer under static transaction interval and PQI-CDI method.

We first explain the transaction rate under the static transaction interval. In the upper part of Figure 2, we suppose the static transaction interval is 0.1 seconds for starting each transaction. TT indicates the transaction time for each transaction. The first transaction starts at 0.0 seconds and finishes at 0.08 seconds as just an example. In this case, the transaction time is 0.08 seconds and the processing computer waits 0.02 seconds before receiving the next transaction. Each transaction starts when the transaction interval reaches 0.1 seconds For the second transaction, third transaction and fourth transaction start at 0.1 seconds, 0.2 seconds, and 0.3 seconds respectively and their transaction time is the same 0.02 seconds. In this case, the processing computer waits 0.08 seconds before receiving the next transaction. For the fifth transaction and sixth transaction start at 0.4 seconds and 0.5 sec. respectively and their transaction time are the same 0.03 seconds. In this case, the processing computer waits 0.07 seconds before receiving the next transaction. The number of transactions from the time 0.0 seconds to the time 0.5 seconds is 5 transactions as just an example.

Next, we explain the PQI-CDI method that shows in the lower part of Figure 2. In the PQI-CDI method, the processing computer calculates the average transaction time of the previous cycle C_n as the new transaction interval.

$$AveTT_n(t) = \sum_{\tau=t-C_n+1}^{C_n} TT_n(\tau) \qquad (1)$$

To make the example simple, in the Figure 2, we set the cycle length to 2 (2 transactions). The processing computer requests to change the transaction intervals of camera every finishing C_n transactions. For example, when the cycle length is 2, the first transaction time is 0.06 sec. and the second transaction time is 0.02 sec. In this case, the average transaction time returns from the processing computer is 0.04 sec. This value is arranged for the next transaction interval of the camera.

3.3.2 Data Generations in the Cameras

Figure 3 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

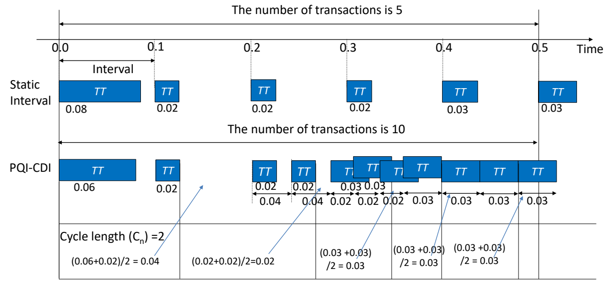

Fig. 2: PQI-CDI Method

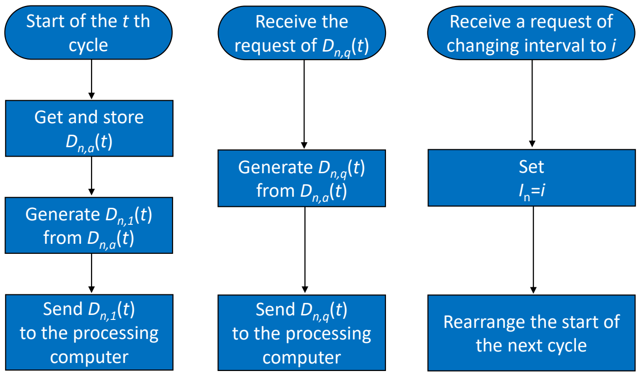

Fig. 3: The flowchart for the data generation

the request of changing interval to i, it changes its interval to i and rearranges the start of the next cycle.

3.3.3 Data Processing in the Processing Computer

Figure 4 shows the flow chart of the processing computer. When the processing computer receives $D_{n,q}(t)$, it processes $D_{n,q}(t)$. When $q = Q$ and $D_{n,q}(t)$ is the final level data, the process of t th cycle finishes. Otherwise, the processing computer 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 $D_{n,q}(t)$ to the camera n, otherwise, the process finishes. When the process finishes, in the PQI-CDI method, the processing computer checks whether c_n reaches C_n or not. c_n is the variable to count the number of transactions for the camera n. Here, again, C_n is the interval of transactions to change the interval of the data transmission of the camera n. In case that c_n reaches to C_n, the processing computer calculates the new interval and sends the request for changing the interval to the camera n. Then, initialize c_n. Here, $TT_n(t)$ is the transaction time of the t th cycle of the camera n, i.e., the time to get the original data at the camera n to the time to finish the process of the data at the processing computer.

4. Implementation

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

4.1 Our System Architecture

Figure 5 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 the 100BASE-TX/1000BASE-T network (Allied Telesis CentreCOM GS908GT switch). We used the Python programming lan-

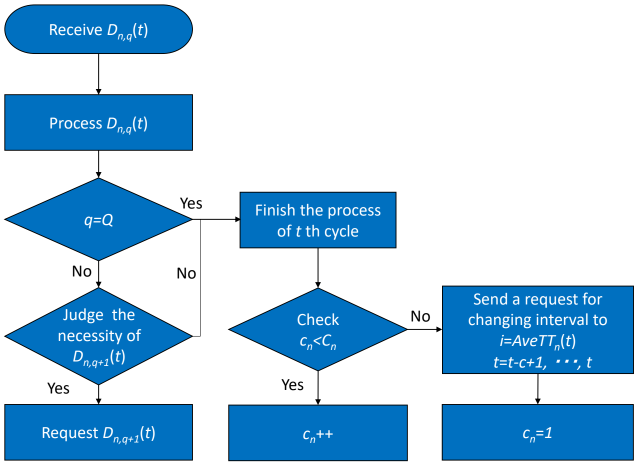

Fig. 4: The flowchart for the processing computer

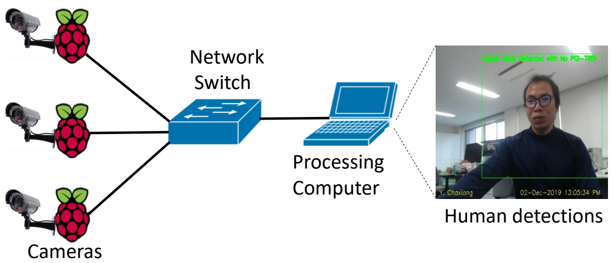

Fig. 5: Our implementation system architecture

Table 1: Specifications of our implemented system

Items	Details
Camera device	Raspberry Pi 3 Model B+ with Camera Module V2, 100BASE-TX
Processing computer	2.10GHz dual-core CPU, 8GB memory, onboard graphics, 1000BASE-T
Network	Gigabit Ethernet
Num. of cameras	3
Num. of processing computers	1
Num. of qualities in prog. JPEG	10
Comparison methods	Without the PQI approach
Evaluation items	Transaction Time & Transaction Rate

guage and implemented as an 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). These generated qualities are temporarily stored in the memory. Firstly, each recording computer sends 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.

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.

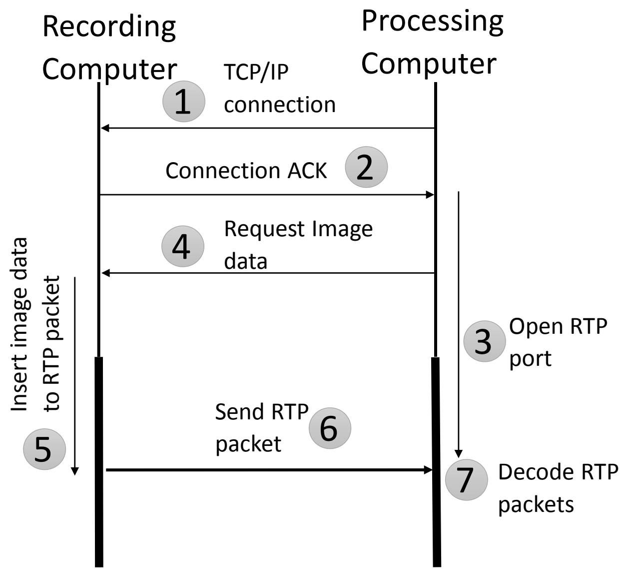

Fig. 6: The communication diagram for our implemented system

The common protocol TCP/IP is used for establishing processing computer and cameras and requesting the higher quality data. The RTP ((Real-time Transport Protocol) is used for transferring the image data between cameras and the processing computer. In our system, the processing computer first connects to the cameras via a TCP/IP communication (TCP socket). The cameras make the connection to the processing computer in the second step. If the connection completed, the processing computer opens the 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 camera 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 the 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 cameras via the communication channel.

4.3 How to Generate and Request High Quality Data

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

Figure 7 shows how to generate some qualities from every single image frame and requesting the higher quality image data. First, the camera gets the raw image frame from its camera's sensor. Second, the camera 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 re-

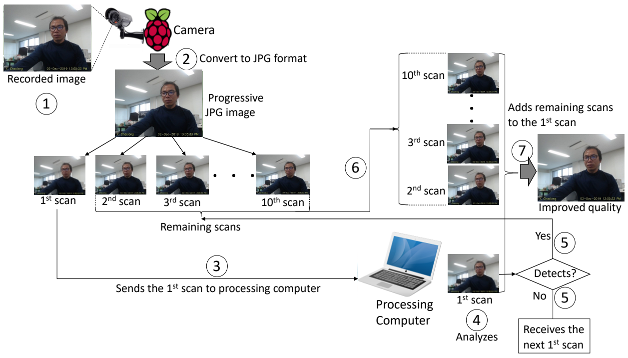

Fig. 7: How to generate and request a higher quality image

quests to the camera 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 images under the PQI-CDI method and No PQI-CDI method in our implementation. The top part of the figure shows the image under the PQI-CDI method. In PQI-CDI method, the processing computer collects and analyzes image data in the lowest quality for detecting objects. In the cases that an object is not detected in the lowest quality data (as shown in the left-hand side of the top part of the figure), the processing computer ignores requesting the remaining qualities and waits for the next transaction. In this case, the transmitted data for the lowest quality data is 6134 Bytes. Otherwise, the processing computer requests to get the remaining qualities and progressively collects them in order to improve the quality of image data (the improved image shows in the right-hand side of the top part of the figure). The transmitted data for the proved quality data is 104064 Bytes.

In the lower part of the figure shows the image under the No PQI method. In this case, the processing computer receives each image as the original image without extracting it into some qualities. The processing computer analyzes the received image data with a clear image (a high-quality image) even there is no object recorded as shown in the lower part of the figure. The transmitted image data for each transaction under the No PQI method is 104064 Bytes.

5. Experimental Results

In this section, we show some experimental results.

5.1 Experiment Setup

In this experiment, we use the implementation system explained in Section 4 and use the configuration parameters used in Table 2.

Communication Bandwidth is 100BASETX/1000BASE-T CAT5e cables that connect between Raspberry Pi devices and the processing computer (we connect the cable to 100BASETX Ethernet port on the Raspberry Pi). Its Bandwidth approximate 100 [Mbps].

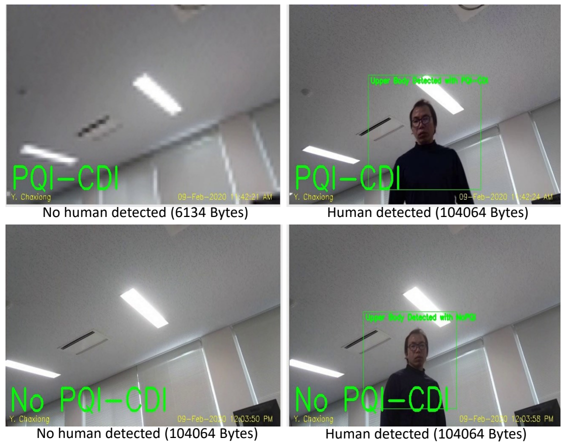

Fig. 8: Images under PQI-CDI and No PQI-CDI method

Table 2: Parameter values

Communication Bandwidth	approx.100 [Mbps]
Communication Protocol	TCP
Data Transfer Protocol	RTP over UDP
Image Resolution	480 × 340
Progressive JPG generation	OpenCV 3.2
Number of qualities	10 scans
Initialize Frame Rate	30 fps
Cycle length	2
Image Analysis	upper human body detection by Opencv (HAAR)

Communication Protocol is used for requesting quality data and changing the transaction intervals. In this implementation, we use common protocol TCP/IP. For Data Transfer Protocol, we use RTP over UDP. It is used for sending the image data.

We set the image resolution in the Raspberry Pi side to 480 × 340 pixels. Each recorded image frame is converted into Progressive JPG standard format by using Opencv (a popular programming library) which contains 10 qualities (10 scans in Progressive JPG format) and temporarily stored it in the memory of each Raspberry Pi. Each camera sends images frame 30 fps as the initial frame rate (or we can call the initial transaction interval is 0.03 seconds).

Cycle length is a counted number for each transaction C_n. The processing computer calculates the average transaction time $AveTT_n$ of each camera n when the number of transactions reaches 2 (C_n=2).

In this implementation, we implemented upper human body detection in the processing computer by using Opencv and its detection library called HAAR.

5.2 Transaction Time Evaluation

Since our implementation uses the PQI approach to improve the performances of the system by reducing the transaction time, we check the average transaction time.

Figure 9 shows the average transaction time under 1 camera and 1 processing computer. The horizontal axis is the transaction time of No PQI (conventional method) and the PQI-CDI method. The vertical axis is the average transaction time. In this implementation, we run the system for 60 seconds for upper human body detection and calculate the final probability. Here, the final probability is the number of image frames that detected the upper

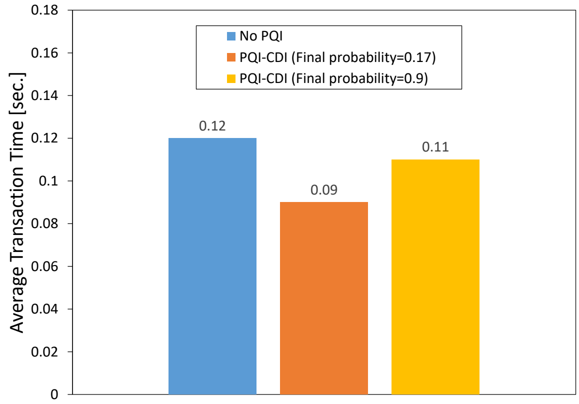

Fig. 9: Average transaction time when the number of the cameras is 1

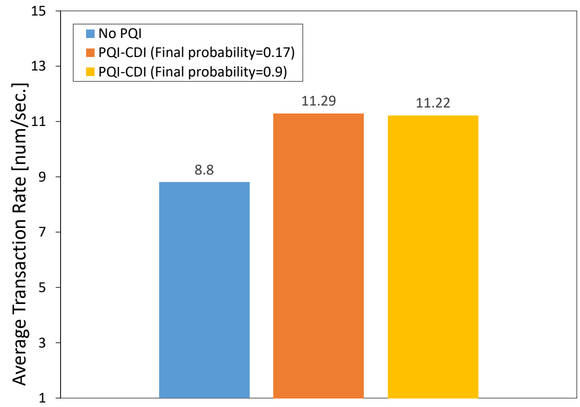

Fig. 11: Average transaction rate when the number of the cameras is 1

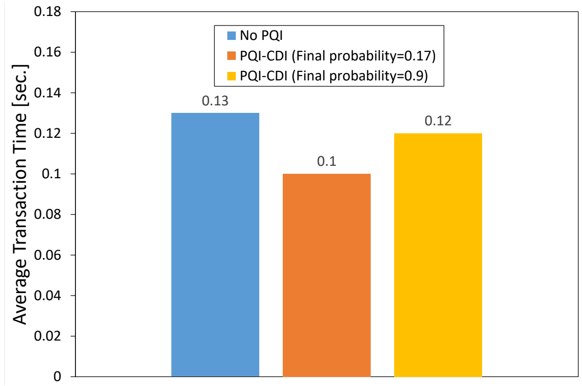

Fig. 10: Average transaction time when the number of the cameras is 3

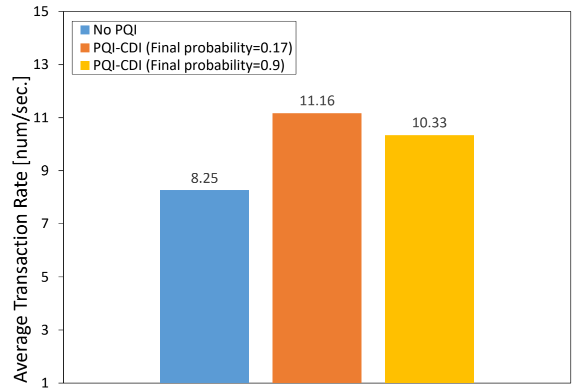

Fig. 12: Average transaction rate when the number of the cameras is 3

human body divide by the total of received image frames. The final probability value is 0 to 1. A higher final probability value gives a more detection occur in the system. We can see that the transaction time under the No PQI is longer than the transaction time that under the PQI-CDI method since the transmitted data amount under the No PQI is larger even there are detect human or not. The transaction time under the PQI-CDI method increases as the final probability increases since the transmitted data amount increases as the final probability increases.

Figure 10 shows the average transaction time under 3 cameras and 1 processing computer. The horizontal axis is the transaction time of No PQI and PQI-CDI method. The vertical axis is the average transaction time. We can see that the transaction time under the No PQI is longer than the transaction time that under the PQI-CDI method due to the same reason as Figure 9. The transaction times for both No PQI and PQI-CDI method is longer than under the transaction times shown in 9 since the number of cameras increase. This causes a larger communication traffic to occur in the system.

5.3 Transaction Rate Evaluation

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

Figure 11 shows the average frame rate under 1 camera and 1 processing computer. The horizontal axis is the transaction rates of the No PQI and PQI-CDI method. The vertical axis is the average transaction rate. The other settings are the same as Figure 9. We can see that our proposed method gives a higher transaction rate than under the No PQI since the transaction intervals under the PQI-CDI dynamically changes.

Figure 12 shows the average frame rate under 3 cameras and 1 processing computer. The horizontal axis is the transaction rates of the No PQI and PQI-CDI method. The vertical axis is the average transaction rate. The other settings are the same as Figure 9. We can see that our proposed method gives a higher transaction rate than under the No PQI due to the reason as mentioned in Figure 11.

6. Conclusion

In this paper, we implemented a video surveillance system with dynamic transaction intervals under the PQI approach (PQI-CDI method). We investigated the transaction time and transaction rate with PQI-CDI method. In our implementation, the transaction interval dynamically changes depending on the average transaction time by using a cycle to control the timing to change the transaction intervals. Moreover, our implementation adopted the PQI approach to reduce the transaction time. The experimental results show that the PQI-CDI method can improve the transaction rates for the actual surveillance systems compared with No

PQI (conventional approach).

In the future, we plan to implement a system with multiple processing computers using this proposed method.

Acknowledgement

This research was supported by a Grants-in-Aid for Scientific Research(C) numbered JP18K11316 and by G-7 Scholarship Foundation.

References

[1] M. A. Usman, M. R. Usman and S. Y. Shin, "An intrusion oriented heuristic for efficient resource management in end-to-end wireless video surveillance systems," 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, pp. 1-6, Jan. 2018.

[2] L. Han, Y. Zhao, S. Yu, B. Zhao, J. Li and J. Wu, "A general solution for multi-thread based multi-source compressed video surveillance system," Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), Wuhan, pp. 95-99, Oct. 2014.

[3] D. Whitman, "The need and capability of a Surveillance Data Distribution System," 2009 Integrated Communications, Navigation and Surveillance Conference, Arlington, VA, pp. 1-6, May 2009.

[4] I. R. Khan, A. Hassan, S. Ahsan, S. Alshomrani and G. Iqbal, "Remote surveillance with reduced transmission overhead," 2017 2nd International Conference on Frontiers of Sensors Technologies (ICFST), Shenzhen, pp. 435-439., April 2017.

[5] D. Chu, C. Jiang, Z. Hao and W. Jiang, "The Design and Implementation of Video Surveillance System Based on H.264, SIP, RTP/RTCP and RTSP," 2013 Sixth International Symposium on Computational Intelligence and Design, Hangzhou, pp. 39-43, Oct. 2013.

[6] K. C. Lai, Y. P. Chang, K. H. Cheong and S. W. Khor, "Detection and classification of object movement - an application for video surveillance system," 2010 2nd International Conference on Computer Engineering and Technology, Chengdu, 2010, pp. V3-17-V3-21.

[7] L. Kong and R. Dai, "Object-Detection-Based Video Compression for Wireless Surveillance Systems," in IEEE MultiMedia, vol. 24, no. 2, pp. 76-85, April 2017.

[8] C. Yukonhiatou, T. Yoshihisa, Y. Teranishi, Y. Ishi, T. Kawakami, and S. Shimojo, "A Scheme to Improve Stream Transaction Rates for Real-Time IoT Applications," International Conference on Advanced Information Networking and Applications (AINA), vol 926, p. 787-798. March 2019.

[9] L. Wu, Z. Liu and Y. Li, "Moving objects detection based on embedded video surveillance," 2012 International Conference on Systems and Informatics (ICSAI2012), Yantai, pp. 2042-2045, May 2012.

[10] 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.

[11] 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.

[12] Chaxiong Yukonhiatou, Tomoki Yoshihisa, Tomoya Kawakami, Yuuichi Teranishi and Shinji Shimojo, "A Method to Reduce Transaction Time for Real-time IoT Applications", IPSJ Journal of Information Processing (JIP), pp.701-710, Nov. 2019 (Recommended paper).