PAPER *Special Section on Emerging Communication Technologies in Conjunction with Main Topics of ICETC 2023*

Compensation of Communication Latency in Remote Monitoring Systems by Video Prediction*

Toshio SATO [†a)], **Yutaka KATSUYAMA** [†], **Xin QI** [††], **Zheng WEN** [†], **Kazuhiko TAMESUE** [†], ***Members***, **Wataru KAMEYAMA** [†], ***Senior Member***, **Yuichi NAKAMURA** [†††], **Jiro KATTO** [†], ***and*** **Takuro SATO** [†], ***Fellows***

SUMMARY Remote video monitoring over networks inevitably introduces a certain degree of communication latency. Although numerous studies have been conducted to reduce latency in network systems, achieving "zero-latency" is fundamentally impossible for video monitoring. To address this issue, we investigate a practical method to compensate for latency in video monitoring using video prediction techniques. We apply the lightweight PredNet to predict future frames, and their image qualities are evaluated through quantitative image quality metrics and subjective assessment. The evaluation results suggest that for simple movements of the robot arm, the prediction time to generate future frames can tolerate up to 333 ms. The video prediction method is integrated into a remote monitoring system, and its processing time is also evaluated. We define the object-to-display latency for video monitoring and explore the potential for realizing a zero-latency remote video monitoring system. The evaluation, involving simultaneous capture of the robot arm's movement and the display of the remote monitoring system, confirms the feasibility of compensating for the object-to-display latency of several hundred milliseconds by using video prediction. Experimental results demonstrate that our approach can function as a new compensation method for communication latency.

key words: *video prediction, zero-latency, remote video monitoring, PredNet, image quality, object-to-display latency*

1. Introduction

Video surveillance and remote monitoring over networks inevitably introduce a certain degree of latency in the transmission of images. Use cases where transmission latency is a problem, such as remote operation [2] and remote driving control [3], are increasing, and reducing latency in video monitoring has become a major issue.

Video monitoring over a network is influenced by several latency factors, including camera acquisition, encoding, network, decoding, and rendering [4]. Various studies have proposed methods to reduce this "end-to-end" latency in different areas, such as efficient coding [5], packet management of video data [6], and URLCC communications [7]. Although combining these methods can achieve "low-latency" communication, it is not possible to achieve "zero-latency".

In recent years, video prediction techniques have been studied to predict future video frames [8], [9]. Although there are still many challenges in predicting future frames for complex video content, relatively accurate predictions have been achieved under constant motion conditions, such as in-vehicle camera videos [9].

In [10], frame extrapolation towards zero-latency video transmission is evaluated. While this study reports that 100 ms of latency can be compensated, it does not provide information on the prediction processing time. A video prediction method, MCnet, evaluated in [10], has a computational efficiency of 500 GFLOPs, as analyzed in [11], which is significantly larger than 65 GFLOPs of PredNet. In general, the computational cost required for video prediction is high and becomes impractical when the processing time exceeds the prediction time. Zero-latency has also been discussed for cloud gaming platforms [12]. The process handles complex objects in high-resolution images, but it is not straightforward video prediction involving 3D block caching to reduce artifacts and LSTM prediction for foreground objects. This study suggests that 23 ms of latency can be compensated, but processing takes 252.6 ms ($n = 1$) using a GPU, resulting in processing time exceeding prediction time. In contrast to these studies, we will implement real-time video prediction to compensate for latency, including processing time.

Furthermore, these studies primarily evaluate image quality using quantitative metrics, such as PSNR for prediction frames, and did not confirm validity through subjective assessment methods like the Mean Opinion Score [13].

Remote video monitoring involves additional latency factors introduced by cameras and display devices [14]. Most previous studies have focused on transmission latency in networks but did not consider latency of these devices [12]. We extend the concept of communication latency to "object-to-display latency," which includes the latency of both the camera and the display device. In [10], devices are considered as factors of latency, but it is not evaluated that the actual latency can be compensated for by video prediction. We verify that this total latency can be compensated using video prediction through a prototype system.

In this paper, we identified video monitoring of remote robot operations [15] as our initial target to investigate a compensation method for the object-to-display latency using video prediction. The contributions of this study can be summarized as follows:

Manuscript received February 26, 2024.
Manuscript revised June 2, 2024.
Manuscript publicized August 22, 2024.

[†]Faculty of Science and Engineering, Waseda University, Tokyo, 169-0072 Japan.
[††]School of International Liberal Studies, Waseda University, Tokyo, 169-0051 Japan.
[†††]Graduate School of Engineering, Kyoto University, Kyoto-shi, 606-8501 Japan.

*This paper is an expanded version of the paper that was presented at 2023 International Conference on Emerging Technologies for Communications (ICETC2023) [1].

a) E-mail: toshio4.sato@aoni.waseda.jp
DOI: 10.23919/transcom.2024CEP0004

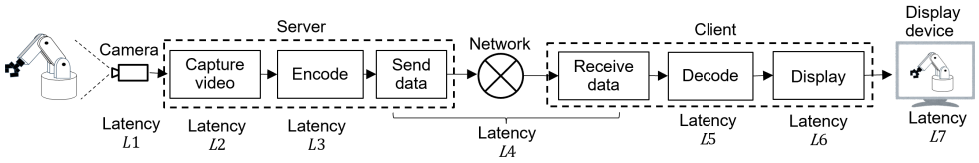


Fig. 1 Latency of video monitoring via networks.

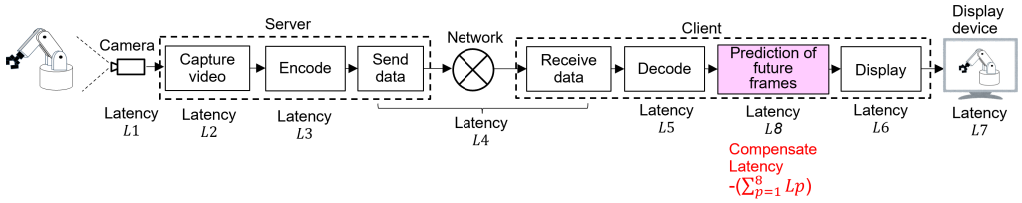
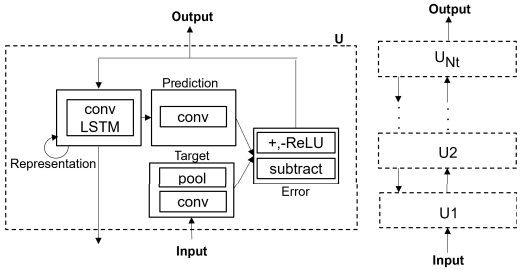


Fig. 2 Compensation of object-to-display latency by using video prediction.

- We implement a real-time remote monitoring system with video prediction capable of compensating for latency and processing time.
- Subjective assessment for image quality of the prediction frame is introduced for the remote control of the robot arm.
- We evaluate compensation for the object-to-display latency that includes the latencies of cameras and display devices.

The concept of compensating for latency in video monitoring is described in Sect. 2. The quality of prediction frames is evaluated by quantitative image quality metrics and subjective assessment to validate its effectiveness in Sect. 3. In Sect. 4, we outline the prototype design of the remote monitoring system utilizing real-time video prediction. Section 5 explains the processing time of the prototype system and provides results for the compensation of the object-to-display latency through video prediction.



(a) Configuration of one unit (b) Total configuration

Fig. 3 Configuration of PredNet [9].

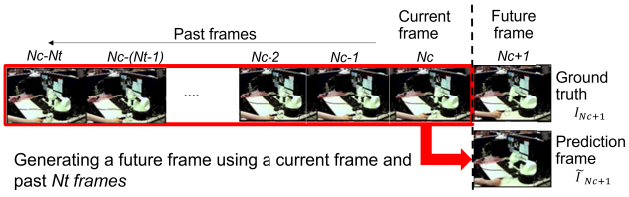


Fig. 4 Process of PredNet to generate prediction frames.

2. Zero-Latency for Video Monitoring

2.1 Latency of Video Monitoring over Networks

Figure 1 shows an example of a configuration for video monitoring captured by a camera via a network. There are various factors that contribute to transmission latency, including the camera, capturing, encoding, the network, decoding, rendering display images, and the display device. As described in Sect. 5, remote video monitoring systems have object-to-display latency of several hundred milliseconds.

2.2 Compensation of Latency Using Video Prediction

To compensate the latency, we introduce a new approach that involves adding video prediction to the video monitoring system, as shown in Fig. 2. In this approach, video prediction introduces additional latency ($L8$) due to processing, but compensates for all latency by generating and displaying future frames. To predict future frames, machine-learning techniques are applied. Numerous studies have proposed video prediction methods [8], including the Predictive Coding Network (PredNet) [9], which comprises multiple interconnected stacked convolutional LSTM units, enabling future frame prediction with a relatively simple configuration as shown in Fig. 3. According to comparison results for computational costs [11], PredNet is one of the lightweight video prediction models with lower GFLOPs requirements, making it suitable for real-time processing. In this study, we employ PredNet as the video prediction method in the video monitoring system to compensate communication latency.

As shown in Fig. 4 and Eq. (1), PredNet utilizes the current frame and the past N_t frames, represented as $I_{Nc}, I_{Nc-1}, \cdots, I_{Nc-Nt}$, to generate a prediction frame for the future frame number N_{c+1}, denoted as $\tilde{I}_{Nc+1}$:

$$\tilde{I}_{Nc+1} = f(I_{Nc}, I_{Nc-1}, \cdots, I_{Nc-Nt}) \quad (1)$$

where, $f(\cdot)$ represents the function of PredNet.

As shown in Fig. 4, the prediction frame $\tilde{I}_{Nc+1}$ can be compared to the ground truth frame I_{Nc+1} for evaluation of image quality.

3. Evaluation of Image Quality for Prediction Frames

3.1 Video Datasets for Evaluation

As shown in Fig. 5, two types of video datasets containing simple periodic movements of a robot arm and a human arm are prepared. These datasets consist of sequential images captured at a rate of 30 frames per second (fps). The "CG" dataset contains 74 video sequences, each comprising 1800 frames. Similarly, the "Real" dataset contains 41 video sequences, each comprising 1800 frames. Each frame is a 160x128 pixel RGB image. The video data in the "Real" dataset are captured by using a USB video camera. The "CG" dataset consists of a virtual robot arm operated by a haptic device (Haption Virtuose 6D), as seen in the "Real" dataset.

The moving and pause times of the arm vary depending on the operator's actions. The arm's reciprocating movement takes 5 to 10 s, with a one-way movement time of about 1.5 s, followed by a few seconds of pause at each end. The 1800 frames in one video sequence contain 9 to 10 reciprocating motions. For the 3 fps data shown in Fig. 5, the movement corresponds to 15 to 30 frames for a round trip and 4 to 5 frames for moving (as indicated by red squares).

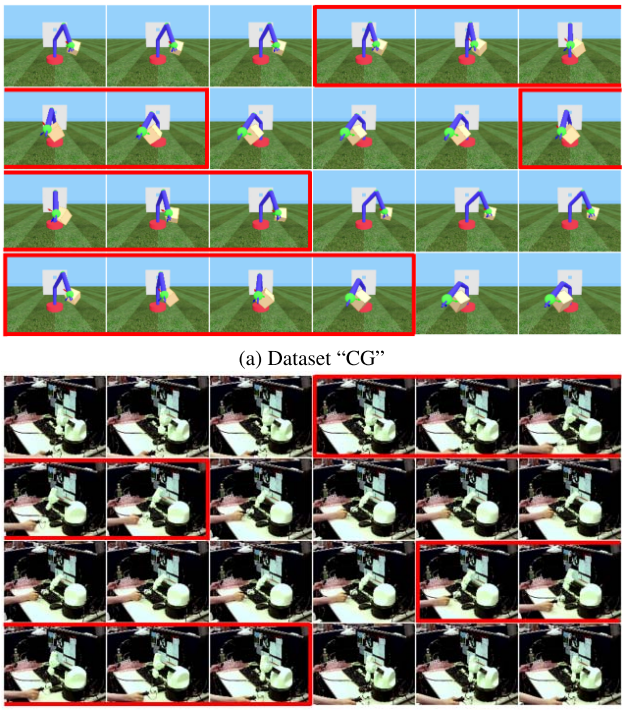
To evaluate at various prediction intervals, the captured 30 fps video data are selected and converted to 6 fps, 3 fps, 2 fps, 1 fps video data that corresponds to prediction times of 167 ms, 333 ms, 500 ms, and 1 s, respectively. This conversion creates ten evaluation datasets from two types of video datasets, as shown in Table 1.

Using these datasets, future frames are predicted by PredNet and compared to the corresponding ground truth frames. We set the number of past frames, $Nt = 8$, in Eq. (1). For the training of PredNet, we select 72 video sequences from the "CG" dataset and 39 sequences from the "Real" dataset. The remaining video data from both datasets are reserved for validation and evaluation purposes.

3.2 Evaluation of Quality by PSNR Metrics

Examples of prediction results are shown in Figs. 6 and 7. The positions of the robot arm and the human arm in the prediction frames, $\tilde{I}_{Nc+1}$, closely approximate the corresponding ground truth frame, I_{Nc+1}. In contrast, the positions in I_{Nc} differ from the ground truth frames. The comparison between the prediction frame, $\tilde{I}_{Nc+1}$, and the ground truth, I_{Nc+1}, is based on the Peak Signal-to-Noise Ratio (PSNR) metric using the OpenCV functions:

$$PSNR = 10 \cdot \log_{10}\left(\frac{\{MAX_I\}^2}{MSE}\right) \quad (2)$$



(a) Dataset "CG"

(b) Dataset "Real"

Fig. 5 Examples of sequences in datasets (converted to 3 fps from the original 30-fps dataset). The frame transitions from upper left to lower right. Frames with red rectangles indicate arm movements.

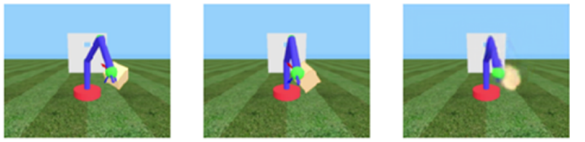
Table 1 Sequences and frames in datasets.

Datasets	Sequence for training	Sequence for validation	Sequence for evaluation	Number of frames in one sequence
CG 1-fps	72	1	1	60
CG 2-fps	72	1	1	120
CG 3-fps	72	1	1	180
CG 6-fps	72	1	1	360
CG 30-fps	72	1	1	1800
Real 1-fps	39	1	1	60
Real 2-fps	39	1	1	120
Real 3-fps	39	1	1	180
Real 6-fps	39	1	1	360
Real 30-fps	39	1	1	1800

$$MSE = \frac{1}{3 \cdot mn} \sum_{h=0}^{2} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \{I(i,j,h) - K(i,j,h)\}^2 \quad (3)$$

where MAX_I is the maximum pixel value among the ground truth $I(i,j,h)$, $K(i,j,h)$ represents a prediction frame, m and n are the dimensions of images, and h is for three color RGB channels.

Figure 8 illustrates average PSNR values of prediction frames for one evaluation sequence across ten datasets. In the case of the 3-fps dataset, the average PSNR is calculated by comparing the ground truth data to 171 prediction frames. This is obtained by subtracting the number of previous frames needed for video prediction ($Nt = 8$) and the last frame without the next ground truth frame from the total 180 frames in one evaluation sequence. According to studies on JPEG2000 [16], [17], the PSNR threshold for the minimum quality requirement is established at 25 to 30 dB. The results show that the PSNR is above 25 dB for the 3-fps dataset



(a)I_{Nc} (b)I_{Nc+1}(ground truth) (c)$\tilde{I}_{Nc+1}$(prediction)

Fig. 6 Example of prediction frame (CG, 3-fps).

(a)I_{Nc} (b)I_{Nc+1}(ground truth) (c)$\tilde{I}_{Nc+1}$(prediction)

Fig. 7 Example of prediction frame (Real, 3-fps).

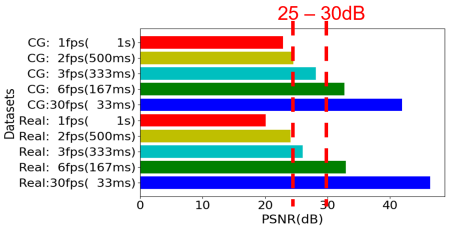


Fig. 8 PSNR values of prediction frames.

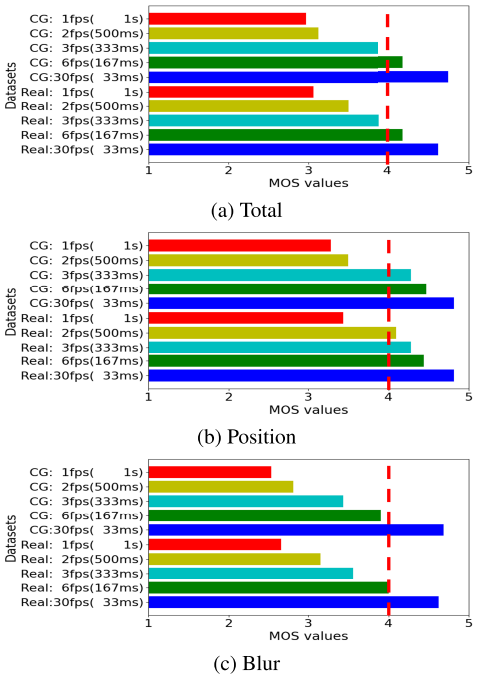
and above 30 dB for the 6-fps dataset that correspond to the future prediction of 333 ms and 167 ms, respectively.

3.3 Evaluation of Quality by Subjective Assessment

We also evaluate prediction frames using subjective assessment based on Mean Opinion Score (MOS) [13]. The evaluators rate the degree of "(a) total qualities", "(b) position of arms", and "(c) blur of arms" on a five-point scale comprising "5 (excellent)", "4 (good)", "3 (fair)", "2 (poor)", and "1 (bad)". Sixteen individuals are selected as evaluators, and their average MOS values are calculated. The evaluators compare the ground truth data and prediction frames by sub-sampling every 10 frames from the original prediction result that is used for PSNR evaluation in Sect. 3.2. Sub-sampling every 10 frames is employed because the total number of frames in the sequence is too large for visual inspection.

Figures 9(a), (b), and (c) show results of three categories for various datasets. In the case of setting a threshold of "4 (good)" in the MOS evaluation, the 3-fps datasets surpass the threshold level in the "position" category and become the borderline range in the "total" category. However, in the "blur" category, the 3-fps datasets receive lower scores than the threshold. These characteristics regarding positions and blurs can be also observed in Figs. 6(c) and 7(c), where arm positions are improved compared to the current frame, but regions of the arms exhibit blurs.

Comparing the results with Fig. 8, we find that the value of "4 (good)" is associated with PSNR values ranging from 25 dB (by comparison with Fig. 9(b)) to 35 dB (by comparison with Fig. 9(c)). For the overall evaluation, we will consider a PSNR of 30 dB or higher to represent acceptable image quality in subsequent sections, as indicated by comparing Fig. 8 with Fig. 9(a). Considered the evaluation results are reasonable based on two different metrics, we will continue to primarily use PSNR metrics in Sect. 4.



(a) Total

(b) Position

(c) Blur

Fig. 9 MOS assessment for prediction.

4. Real-Time Prediction for Video Monitoring Systems

4.1 System Configuration

We have integrated the video prediction function into the video monitoring system, comprising a server equipped with a USB camera and a client with a display device, as illustrated in Fig. 10. The server program is developed using Python 3.9 on a PC (CPU: Intel Xeon E-2124, memory: 16 GB, OS: Ubuntu 20.04). The USB camera captures images at 30 fps, and the server selectively skips frames to convert frame rates between 3 fps and 10 fps. The captured images are encoded as JPEG data with a resolution of 640x512 pixels. The JPEG image quality parameter in the OpenCV function is set to 80, resulting to an approximate 5% compression. A packet, consisting of header data and a JPEG image, is transmitted to the client PC through a TCP/IP socket connection. Clock data for T1 and T2 in Fig. 10 are stored in the header data.

The client system is implemented on a PC (CPU: Intel Core i7-9700, memory: 16 GB, OS: Ubuntu 20.04) equipped with a GPU unit (NVIDIA GeForce RTX 2060, memory: 6 GB). The program is written in Python 3.9 to receive packets, decode JPEG images, generate future frames through video prediction, and display the results. In PredNet, each frame is converted into a 160x128 pixel RGB image and

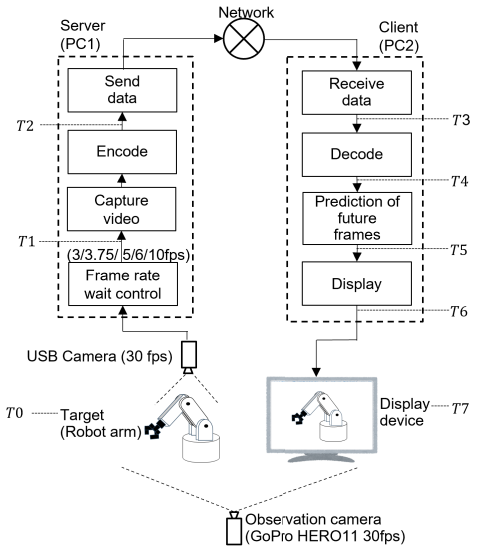


Fig. 10 Integration of video prediction in video monitoring system.

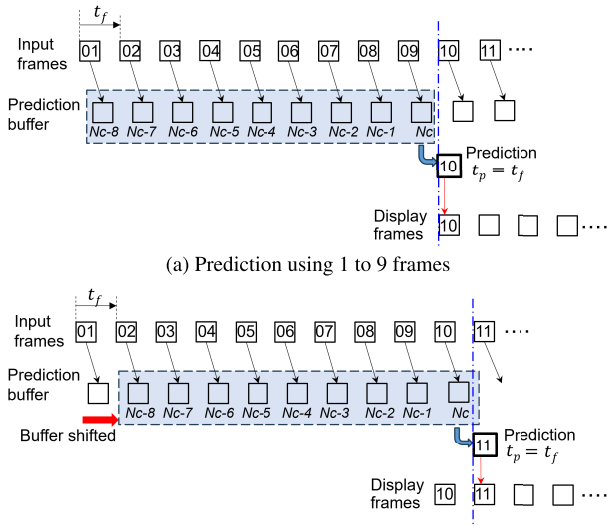


Fig. 11 Detailed description of real-time prediction process.

accumulated through past frames, as shown in Fig. 4. The PredNet is implemented using Python with PyTorch 1.8.1 and CUDA 11.4 utilizing the GPU. Time data from T3 to T6 described in Fig. 10, are recorded to measure the processing time of the client. The clocks of the two PCs are synchronized via an NTP server, allowing the measurement of the entire process time from T1 to T6.

Figures 11(a) and (b) illustrate the detailed operation of the real-time prediction process. In Fig. 11(a), the prediction buffer captures nine sequential frames and generates a prediction frame labeled "10", representing the future time t_p from the last frame labeled "09".

After inputting the next frame, No.10, the prediction buffer shifts by one frame and adds the newer frame to generate a prediction frame labeled "11", as shown in Fig. 11(b). The most straightforward setup involves setting the prediction time t_p, to be the same as the frame interval t_f. In this study, we examine this simple setup condition to explore the possibilities of achieving zero-latency remote monitoring in real-time.

Fig. 12 Example of video dataset to capture the robot arm (3-fps).

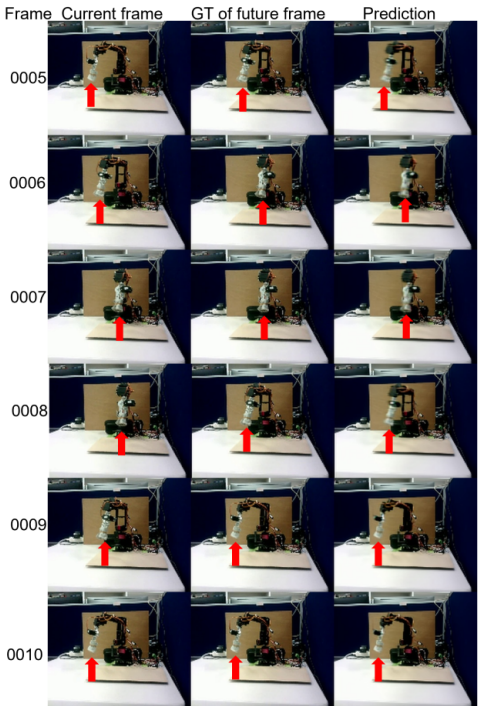


Fig. 13 Example results of video prediction (3-fps dataset).

4.2 Construction of Video Prediction Model

To construct a prediction model, training datasets are obtained through the video monitoring system. We acquire 12 video sequences, which include simple periodic movements of a robot arm (approximately two seconds for one swing), as shown in Fig. 12. Each sequence consists of 1500 frames of a 640x512 pixel RGB image captured at 30 fps. The 12 video sequences are divided into 10 data for training data, one for validation, and one for evaluation.

The captured 30 fps video data is converted to 10 fps, 6 fps, 5 fps, 3.75 fps, 3 fps video data that corresponds to prediction times of 100 ms, 167 ms, 200 ms, 267 ms, and 333 ms, respectively, to build models for various prediction times t_p. The image quality of predictions using these databases is assessed using the same methods explained in Sect. 3.

Examples of predictions for 3-fps datasets are illustrated in Fig. 13. In the ground truth (GT) frames, the arm moves to the right in frames 0005 and 0006, comes to a stop and turns

Table 2 Performances of prediction for moving robot arm.

Datasets	Prediction time (ms)	PSNR for current frames(dB)	PSNR for prediction frames (dB)
3-fps	333	23.7	33.5
3.75-fps	267	24.5	34.5
5-fps	200	25.1	35.9
6-fps	167	25.7	36.1
10-fps	100	27.3	36.7

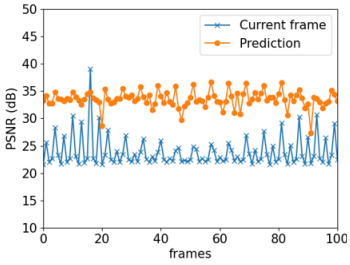


Fig. 14 PSNR for current and prediction frames (3-fps dataset).

back in the frame 0007, and then moves to the left in frames 0008 and 0009. Red arrows indicate the positions of the arm, and the ground truth data of future frames are almost one frame ahead of the current frames. The prediction frames exhibit some blurs, but its position is close to the ground truth.

The average PSNRs compared with the ground truth are shown in Table 2. The PSNRs for the prediction frames are above 33.5 dB, indicating an improvement of almost 10 dB compared to the current frames, which exceeds the quality criterion of 30 dB considered in Sect. 3.3. Although the objects and movements differ from those in the datasets of Sect. 3, the image quality is considered satisfactory. Based on these results, it is possible to apply the 333 ms video prediction with a PSNR of more than 30 dB for this subject. The PSNR decreases as the frame rate decreases and this trend is the same as results in in Sect. 3.

Figure 14 illustrates the PSNR transition for each frame in the 3-fps data set. The orange line in Fig. 14 represents PSNR values for the prediction frames compared with the ground truth. Conversely, the blue line indicates the PSNR values for the current frames compared with the ground truth. The prediction frames achieve higher PSNR (average 33.5 dB) compared to the current frames (average 23.7 dB). Notably, the higher PSNR values in the blue line correspond to situations when the arm has come to a stop position in the swing motion (see frames 0007 and 0010 in Fig. 13).

5. Evaluation of Remote Monitoring System with Video Prediction

5.1 Processing Time of Video Monitoring System

The remote monitoring system with video prediction in Fig. 10 is evaluated by using live input from USB camera is to assess the object-to-display latency. The object-to-display latency corresponds to the latency from T0 to T7 in Fig. 10. We also use the robot arm shown in Fig. 12 as the object for

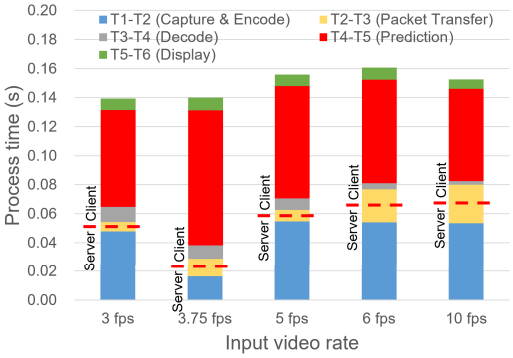


Fig. 15 Processing time for video monitoring using prediction.

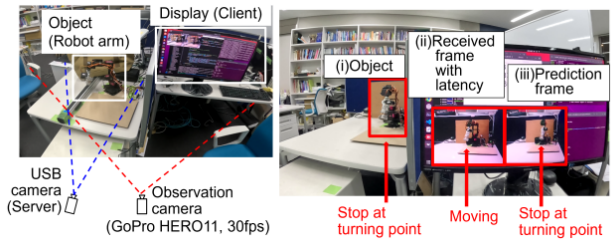
Table 3 Processing times for (T1-T2), (T2-T3) and (T4-T5).

Processing time		Input video rate 3 fps	3.75 fps	5 fps	6 fps	10 fps
T1-T2	Mean (ms)	47.8	16.3	54.7	54.1	53.5
	SD (ms)	7.0	7.1	2.6	1.9	1.3
T2-T3	Mean (ms)	6.5	12.5	7.9	22.6	26.5
	SD (ms)	12.9	17.0	14.5	22.8	17.8
T4-T5	Mean (ms)	66.8	93.0	77.4	71.3	63.6
	SD (ms)	17.5	35.0	30.3	36.2	37.2

measuring the object-to-display latency. The USB camera is installed in the same arrangement as during the capture of training datasets. The frame rates of the input are controlled at 3 fps, 3.75 fps, 5 fps, 6 fps, and 10 fps. Prediction models trained by 3-fps, 3.75-fps, 5-fps, 6-fps, and 10-fps datasets are selected to match the respective input frame rates.

Average process times from T1 to T6 are illustrated in Fig. 15. The time data from T1 to T6 corresponds to Fig. 10 and represent the average of 100 samples from the log file of the client program. T1 and T2 values are embedded in packet data from the server and transferred to the client. The total process times, including both the server and the client, are less than 160 ms for each input video rate. The major portions of the processing time correspond to video capture and encoding (T1-T2) of the server and prediction (T4-T5) of the client. The process time for packet transfer (T2-T3) increases with the frame rate. The client's processing time, measured from T2 to T6 for the 10 fps video input, is calculated as an average of 99.0 ms that is shorter than the frame interval of 10 fps videos. Based on these results, it can be concluded that this prototype system is capable of real-time processing for video inputs up to 10 fps.

In Fig. 15, some intervals exhibit variation across the five input video rates. The mean and standard deviation (SD) of the processing times for (T1-T2), (T2-T3), and (T4-T5) are shown in Table 3. The (T1-T2) interval represents the time from reading data from the USB camera to the end of encoding. It has been observed that the (T1-T2) varies over multiple trials, irrespective of the video rate. These variations are considered to be due to asynchronous time control between the camera and processing in the server. The (T2-T3) interval is thought to be affected by frame rates



(a) Arrangement of cameras (b) Image of observation camera

Fig. 16 Evaluation of object-to-display latency.

and fluctuations in network latency, which range up to 30 ms as observed using the ping command in Linux, between the server and the client. The (T4-T5) interval corresponds to video prediction processes, which exhibit large standard deviations as shown in Table 3, and the mean time may also vary. The variation in latency is caused by different conditions and needs to be adapted to in practical systems.

5.2 Measurement of Object-to-Display Latency

To measure the total latency of remote video monitoring, analyzing data from T1 to T6 in Fig. 15 is not sufficient. The actual latency for an operator should be defined as object-to-display latency, corresponding to from T0 to T7 in Fig. 10, where the latencies of the camera (T0 to T1) and the display device (T6 to T7) are always unknown.

We employ an observation camera (GoPro HERO11, 30 fps) to measure the object-to-display latency. The client's display device is positioned on the desk next to the object (robot arm), and the observation camera captures both areas, as depicted in Fig. 16(a). The object-to-display latency is determined by analyzing 30 fps frames from the observation camera and counting the frame numbers at the right turning point of the robot arm movement.

An example of images captured by the observation camera is shown in Fig. 16(b). Figure 16(b) illustrates three targets: (i) the object, (ii) the received frame with latency, and (iii) the prediction frame. The frames reaching the right turning point are measured through the operator's visual inspection for ten samples. The difference in frame numbers from (i) to (ii) corresponds to the actual object-to-display latency (L_a), and the difference in frame numbers from (i) to (iii) indicates the compensated latency by video prediction (L_b).

Table 4 shows the measurement results of the object-to-display latency for five different capture rates in the server. The object-to-display latency is calculated by averaging ten measurements of the difference in frame numbers (D_f) and then multiplying by the frame interval of the observation camera (1/30 s). The object movement is consistent with the training data conditions, which were selected as the prediction model for the given capture rate. The prediction time t_p is defined by the frame rates of the database, as shown in Fig. 11.

The object-to-display latency of received frames without prediction, L_a, varies from 163.3 ms in the 5-fps dataset to 330.0 ms in the 3.75-fps dataset. The object-to-display latency of prediction frames L_b becomes lower than that of received frames, from −120.0 ms to 93.3 ms, approaching $L_a - t_p$. The presence of negative latency for prediction frames indicates that our approach has the potential to achieve zero-latency. If the prediction time t_p is small relative to the latency L_a, as in the case of an input video rate of 10 fps, the effect of compensation is small.

Table 4 Object-to-display latency for moving robot arm. D_f means difference of frame numbers. L_a also includes prediction process times.

Input video rate (fps)	Prediction Model: Training dataset	Prediction time t_p (ms)	Average Object-to-display latency: (i) to (ii) Received frame D_f	(i) to (ii) Received frame L_a (ms)	(i) to (iii) Prediction frame D_f	(i) to (iii) Prediction frame L_b (ms)	Compensation $L_a - L_b$ (ms)
3	3-fps	333	6.9	230.0	-3.6	-120.0	350.0
3.75	3.75-fps	267	9.9	330.0	1.1	36.7	293.3
5	5-fps	200	4.9	163.3	-0.9	-30.0	193.3
6	6-fps	167	6.4	213.3	1.4	46.7	166.7
10	10-fps	100	6.3	210.0	2.8	93.3	116.7

To compare with conventional remote monitoring systems, the processing time of video prediction should be excluded from the results. For instance, the latency of 230 ms in the 3 fps input includes the processing time for video prediction of approximately 70 ms; the object-to-display latency of the conventional remote monitoring system is estimated as 160 ms. The prediction time of the video prediction process t_p, which accounts for this additional processing time, t_v compensates for the object-to-display latency L_a, including the latency of the conventional system and this processing time itself. If relationship of times as $t_v > t_p$, applying video prediction should be avoided.

Figure 16(b) is an example image of the observation camera captured when the robot arm reached the right turning point. In this example, the arm is in motion in the received frame (ii), while the prediction frame (iii) stops at the right turning point.

6. Discussions

Based on the results of this study, we can demonstrate the feasibility of achieving "zero-latency" remote monitoring by using video prediction. We developed real-time video prediction with buffer control mechanisms as shown in Fig. 11 and confirmed the possibility of compensating for the object-to-display latency, including the processing time for video prediction itself, which had not been considered in previous studies [10], [12]. The experimental results for the input video rate of 3 fps indicate that the object-to-display latency, approximately 160 ms (derived from subtracting 66.8 ms of the (T4-T5) interval in Table 3 from 230 ms of La in Table 4), can be compensated using video prediction of up to 333 ms.

In the prototype system, the prediction time t_p is fixed according to the interval time t_f as illustrated in Fig. 11. As latency varies with conditions, compensation for the latency using video prediction results in both positive and negative variations, as shown in Table 4. If the prediction time t_p can be set larger than the total latency, as seen in the cases of

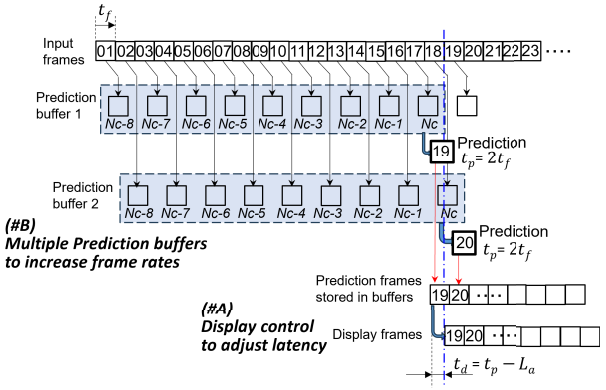


Fig. 17 Video prediction to process higher frame rates.

inputting video rates 3 fps and 5 fps in Table 4, the object-to-display latency becomes zero by controlling the display timing using buffers, as shown in function #A of Fig. 17.

According to the configuration of Fig. 11, if the prediction time t_p is increased, the frame rate must be set lower. However, a lower frame rate for the video will result in lower-quality remote monitoring. For high frame rates, it is possible to have the system predict future video several frames ahead, but this would require a large prediction buffer to capture movements and increase computational costs. To address this issue, t_p and t_f can be independently set by arranging the configuration of prediction buffers. The function #B in Fig. 17 illustrates an example that implements two prediction buffers to accommodate higher frame rates.

In this experiment, the latency in the network between the server and the client is less than 25 ms, as indicated by "T2-T3" in Fig. 15. Although this latency may vary depending on the distance and conditions of networks, it can also be compensated for using video prediction. If a larger latency needs to be accommodated, the prediction time can be increased by incorporating state-of-the-art video prediction algorithms and leveraging greater computational resources.

We have confirmed that the use of prediction frames results in higher PSNR values than using frames with latency. However, there are still image blurs present. Further improvements in image quality can be achieved by using the updated version of PredNet [18] and other state-of-the-art methods [8].

In this study, simple robot arm movements were targeted for monitoring. However, for broader applications, it is essential to focus on more complex movements. We believe that addressing this challenge can be also achieved by incorporating state-of-the-art video prediction algorithms. In such future scenarios, we anticipate that the framework for implementing video prediction will remain consistent with the remote monitoring system considered in this study.

7. Conclusion

We have investigated a practical method to compensate communication latency in the video monitoring system by using video prediction techniques. The prediction frames demonstrate higher PSNR and MOS values in the 167 ms prediction for two different video datasets, and higher PSNR in the 333 ms prediction for the simple movement of the robot arm. Implementation in the actual video monitoring system has indicated the potential of real-time processing to compensate communication latency using video prediction. The experimental results demonstrate that the object-to-display latency, which includes network latency as well as latency of devices (cameras and displays) and processing times (encoding, decoding, and video prediction), can be compensated using video prediction. Regarding the buffer control mechanism used to achieve real-time processing, we also considered a buffer mechanism for time adjustment to cope with variations in latency as a future concept. The next step is to study effective applications of online video systems that achieve "zero-latency" using video prediction, which can handle more complex motion.

Acknowledgments

These research results were obtained from the commissioned research (No. JPJ012368C03801) by National Institute of Information and Communications Technology (NICT), Japan.

References

[1] T. Sato, Y. Katsuyama, Y. Nakamura, X. Qi, Z. Wen, K. Tamesue, W. Kameyama, J. Katto, and T. Sato, "Compensation of communication latency using video prediction in remote monitoring systems," 2023 International Conf. on Emerging Technologies for Communications, Sapporo, Japan, Nov. 2023.

[2] T. Hatano, T. Shimada, and T. Yoshida, "Evaluation of end-to-end delay requirement for remote, precise pick-up action of miniature crane," IEICE Commun. Express, vol.12, no.5, pp.242–248, 2023.

[3] Y. Yu and S. Lee, "Remote driving control with real-time video streaming over wireless networks: Design and evaluation," IEEE Access, vol.10, pp.64920–64932, 2022.

[4] H. Wang, X. Zhang, H. Chen, Y. Xu, and Z. Ma, "Inferring end-to-end latency in live videos," IEEE Trans. Broadcast., vol.68, no.2, pp.517–529, 2022.

[5] M. Mody, P. Swami, and P. Shastry, "Ultra-low latency video codec for video conferencing," 2014 IEEE International Conf. on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, pp.1–5, Feb. 2014.

[6] Y. Liang and B. Girod, "Network-adaptive low-latency video communication over best-effort networks," IEEE Trans. Circuits Syst. Video Technol., vol.16, no.1, pp.72–81, 2006.

[7] H. Ji, S. Park, J. Yeo, Y. Kim, J. Lee, and B. Shim, "Ultra-reliable and low-latency communications in 5G downlink: Physical layer aspects," IEEE Wireless Commun., vol.25, no.3, pp.124–130, 2018.

[8] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Argyros, "A review on deep learning techniques for video prediction," IEEE Trans. Pattern Anal. Mach. Intell., vol.44, no.6, pp.2806–2826, 2022.

[9] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," International Conf. on Learning Representations (ICLR2017), Vancouver, Canada, April 2017.

[10] M. Vijayaratnam, M. Cagnazzo, G. Valenzise, A. Trioux, and M. Kieffer, "Towards zero-latency video transmission through frame extrapolation," 2022 IEEE International Conf. on Image Processing

(ICIP), Bordeaux, France, pp.2122–2126, Oct. 2022.
[11] X. Hu, Z. Huang, A. Huang, J. Xu, and S. Zhou, "A dynamic multi-scale voxel flow network for video prediction," 2023 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), pp.6121–6131, June 2023.
[12] J. Wu, Y. Guan, Q. Mao, Y. Cui, Z. Guo, and X. Zhang, "Zgaming: Zero-latency 3D cloud gaming by image prediction," Proc. ACM SIGCOMM 2023 Conf., New York, NY, USA, p.710–723, Sept. 2023.
[13] K. Hikichi, H. Morino, I. Arimoto, K. Sezaki, and Y. Yasuda, "The evaluation of delay jitter for haptics collaboration over the internet," IEEE Global Communications Conf. (GLOBECOM 2002), Taipei, Taiwan, pp.1492–1496, Nov. 2002.
[14] S. Ubik and J. Pospíšilík, "Video camera latency analysis and measurement," IEEE Trans. Circuits Syst. Video Technol., vol.31, no.1, pp.140–147, 2021.
[15] Y. Katsuyama, T. Sato, Z. Wen, X. Qi, K. Tamesue, W. Kameyama, Y. Nakamura, J. Katto, and T. Sato, "A predictive approach for compensating transmission latency in remote robot control for improving teleoperation efficiency," IEEE Global Communications Conf. (GLOBECOM 2023), Kuala Lumpur, Malaysia, Dec. 2023.
[16] X. Li and J. Cai, "Robust transmission of JPEG2000 encoded images over packet loss channels," 2007 IEEE International Conf. on Multimedia and Expo, Beijing, China, pp.947–950, Aug. 2007.
[17] N. Thomos, N. Boulgouris, and M. Strintzis, "Optimized transmission of JPEG2000 streams over wireless channels," IEEE Trans. Image Process., vol.15, no.1, pp.54–67, 2006.
[18] K. Sakama, S. Sekiguchi, and W. Kameyama, "Performance analysis of generated predictive frames using prednet with multiple convolution kernels," International Workshop on Advanced Imaging Technology (IWAIT), Jeju, Korea, Jan. 2023.

Toshio Sato received the B.E. degree in fine measurement engineering and the M.E. degree in systems engineering from the Nagoya Institute of Technology, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in systems design engineering from the University of Fukui, Japan, in 2003. He joined Toshiba Corporation, Japan, in 1987, where he developed video processing systems, including character recognition, facial recognition and vehicle detection. He was a Visiting Scientist with Carnegie Mellon University, USA, from 1996 to 1998. He is currently a Senior Researcher at Waseda Research Institute for Science and Engineering, Waseda University, Japan.

Yutaka Katsuyama received the B.E. and M.E. degrees in information engineering and the Ph.D. degree in information science from Tohoku University, Japan, in 1984, 1986, and 2014, respectively. From 1986 to 1992, he worked for Fujitsu Ltd., Japan. From 1992 to 2019, he worked for Fujitsu Laboratories Ltd., Japan, where he developed OCR systems. He is currently a Researcher at Waseda Research Institute for Science and Engineering, Waseda University, Japan. His main research interest includes pattern recognition. He is a member of the IEICE. He received the FIT2002 Award (Paper Award), in 2002.

Xin Qi received the M.E. and Ph.D. degrees from Waseda University, Japan, in 2016 and 2019, respectively. In 2019, he became a Research Associate in the Global Information and Telecommunication Institute, Waseda University. He is currently an Assistant Professor (a Lecturer) in the School of International Liberal Studies, Waseda University. His research interests include Intelligence Informatics, Artificial Intelligence and next-generation communication systems.

Zheng Wen received the B.E. degree in computer science and technology from Wuhan University, China, 2009, and the M.Sc. and Ph.D. degrees from Waseda University, Japan, in 2015 and 2019, respectively. In 2018, he became a Research Associate with Waseda University, Tokyo, Japan, where he is currently an Assistant Professor (a Lecturer) with the Department of Communication and Computer Engineering. His research interests include ICN/CCN for next-generation communication systems, AI, and the IoT.

Kazuhiko Tamesue received B.E. degrees in Electronic Engineering from Kumamoto University in 1988 and Received Ph.D. degrees in Wireless System on Global Information and Telecommunication Institute Studies from Waseda University in 2009. He joined RICOH from 1989 to 1991. From 1991 to 2021, he joined Panasonic Corporation. He has mainly developed the 2nd and 3rd generation cellular systems and the sensing recognition application for professional video systems. He is currently a Senior Researcher at Waseda Research Institute for Science and Engineering. His main research interests include wireless access systems.

Wataru Kameyama received Bachelor, Master and Doctor of Eng. from School of Sc. and Eng., Waseda Univ. in 1985, 1987 and 1990, respectively. He joined ASCII Corp. in 1992, and was transferred to France Telecom CCETT from 1994 to 1996 for his secondment. After joining Waseda Univ., as an associate professor in 1999, he has been appointed as a professor at Department of Communications and Computer Engineering, School of Fundamental Sc. and Eng., Waseda Univ. since 2014. He has been involved in MPEG, MHEG, DAVIC and the TV-Anytime Forum activities. He was a chairman of ISO/IEC JTC1/SC29/WG12, and a secretariat and a vice chairman of the TV-Anytime Forum. He received the best paper award of Niwa-Takayanagi in 2006, the best author award of Niwa-Takayanagi in 2009 from the Institute of Image Information and Television Engineers, and International Cooperation Award from the ITU Association of Japan in 2012. He is a senior member of IEICE and IPSJ, and a member of ITE, IIEEJ, ACM and IEEE.

Yuichi Nakamura received B.E., M.E., and Ph.D. degrees in electrical engineering from Kyoto University, in 1985, 1987, and 1992, respectively. From 1990 to 1993, he worked as an instructor at the Department of Electrical Engineering of Kyoto University. From 1993 to 2004, he worked for Institute of Information Sciences and Electronics of University of Tsukuba, Institute of Engineering Mechanics and Systems of University of Tsukuba, as an assistant professor and an associate professor, respectively. Since 2004, he has been a professor of Academic Center of Computing and Media Studies, Kyoto University. His research interests are on computer vision, multimedia, human-computer and human-human interaction including distance communication, and multimedia contents production.

Jiro Katto received B.S., M.E. and Ph.D. degrees in electrical engineering from University of Tokyo in 1987, 1989 and 1992, respectively. He worked for NEC Corporation from 1992 to 1999. He also a visiting scholar at Princeton University, NJ, USA, from 1996 to 1997. He then joined Waseda University in 1999. Since 2004, he has been a professor of the Department of Computer Science, Science and Engineering, Waseda University. From 2004 to 2008, he has also been a director of the Electronic and Information Technology Development Department, New Energy and Industrial Technology Development Organization. His research interest is in the field of multimedia communication systems and multimedia signal processing. He received the Best Student Paper Award at SPIE Conference of Visual Communication and Image Processing in 1991, and received the Young Investigator Award of IEICE in 1995. He is a member of IEEE, ACM, IPSJ and ITE.

Takuro Sato received B.E. and Ph.D. from Niigata University in 1973 and 1990. He joined Research Center, Oki Electric Company in Tokyo, where he worked on 1G, 2G and 3G cellular. He completed W-CDMA standardization IS-665 at JTC1 USA in 1995. He became Professor at Niigata Institute of Technology in 1995. He established the venture company Key Stream to provide 802.11 WiFi LSI in 2000. In 2004, He became a Professor of GITI and School of Fundamental Science and Engineering at Waseda University in 2004 and 2013. He is interested in future network and 6G. Now he is senior research professor at Waseda Research Institute for Science and Engineering. He is life fellow member of IEEE, IEICE and JSST.