IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 June 2023, accepted 9 July 2023, date of publication 13 July 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3294964

RESEARCH ARTICLE

Textural Detail Preservation Network for Video Frame Interpolation

KIHWAN YOON[1,2], JINGANG HUH[1], YONG HAN KIM[2], (Member, IEEE), SUNGJEI KIM[1], AND JINWOO JEONG[1]

[1]Korea Electronics Technology Institute (KETI), Seongnam-si, Gyeonggi-do 13488, Republic of Korea
[2]School of Electrical and Computer Engineering, University of Seoul, Seoul 02504, Republic of Korea

Corresponding authors: Jinwoo Jeong (jw.jeong@keti.re.kr) and Sungjei Kim (sungjei.kim@keti.re.kr)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00087, Development of high-quality conversion technology for SD/HD low-quality media).

ABSTRACT The subjective image quality of the Video Frame Interpolation (VFI) result depends on whether image features such as edges, textures and blobs are preserved. With the development of deep learning, various algorithms have been proposed and the objective results of VFI have significantly improved. Moreover, perceptual loss has been used in a method that enhances subjective quality by preserving the features of the image, and as a result, the subjective quality is improved. Despite the quality enhancements achieved in VFI, no analysis has been performed to preserve specific features in the interpolated frames. Therefore, we conducted an analysis to preserve textural detail, such as film grain noise, which can represent the texture of an image, and weak textures, such as droplets or particles. Based on our analysis, we identify the importance of synthesis networks in textural detail preservation and propose an enhanced synthesis network, the Textural Detail Preservation Network (TDPNet). Furthermore, based on our analysis, we propose a Perceptual Training Method (PTM) to address the issue of degraded Peak Signal-to-Noise Ratio (PSNR) when simply applying perceptual loss and to preserve more textural detail. We also propose a Multi-scale Resolution Training Method (MRTM) to address the issue of poor performance when testing datasets with a resolution different from that of the training dataset. The experimental results of the proposed network was outperformed in LPIPS and DISTS on the Vimeo90K, HD, SNU-FILM and UVG datasets compared with the state-of-the-art VFI algorithms, and the subjective results were also outperformed. Furthermore, applying PTM improved PSNR results by an average of 0.293dB compared to simply applying perceptual loss.

INDEX TERMS Video frame interpolation, textural detail preservation, perceptual loss, synthesis network.

I. INTRODUCTION

Temporal resolution is an important factor in video quality, because a low frame rate may cause temporal jittering, aliasing, and motion blur artifacts. This can be enhanced using video frame interpolation (VFI), which generates an intermediate frame between two consecutive frames. VFI has been utilized in various fields including slow-motion generation [3], [4], [5], [6], [7], novel view synthesis [8], [9], and video restoration [10], [11]. With recent advances in deep learning, there have been significant improvements in the performance of deep-learning-based VFI algorithms [12], [13], [14], [15], [16], [17], [18]. As VFI performance improves, the importance of VFI has increased, and recently, researches have been conducted to fuse it with various vision tasks. For example, STVSR [19], [20], [21], [22], which fuses super-resolution to restore spatial resolution and VFI to restore temporal resolution, is being studied, and various tasks such as VFI and deblurring [23], [24] are being studied.

The quality of an interpolated frame depends on image features such as edge, texture, and textural detail. Among these image quality factors, textural detail indicates film grain

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

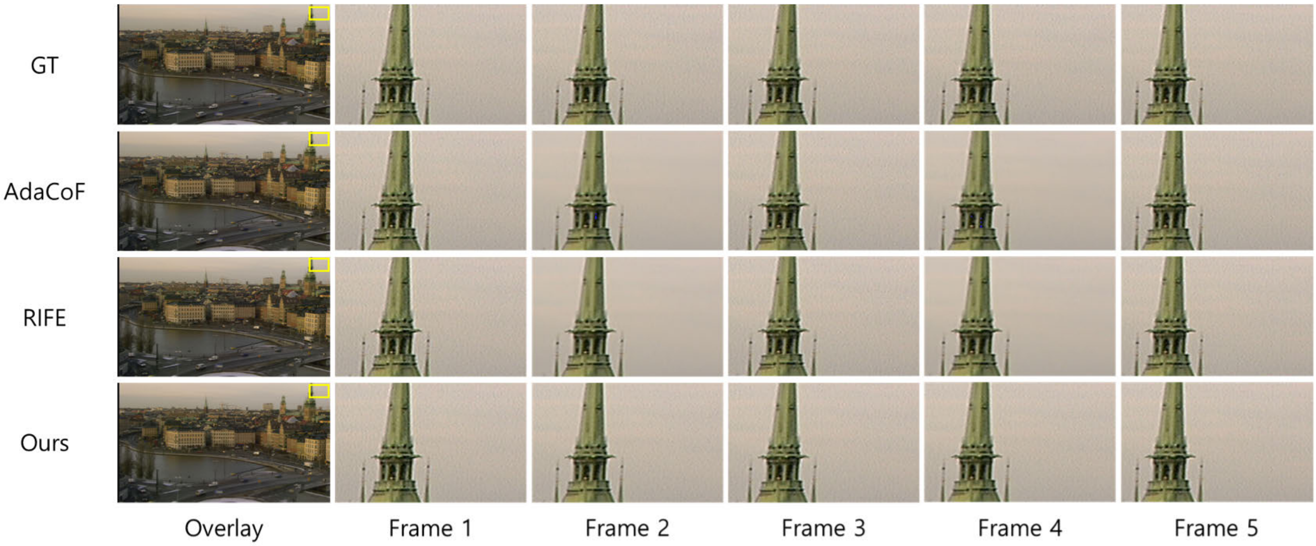

FIGURE 1. Visual comparison between the proposed algorithm and previous algorithms (AdaCoF [1], RIFE [2]). Frame 1, 3, and 5 represent the original image and frame 2 and 4 are the images restored by the VFI algorithm.

noise, weak texture, etc. Film grain noise is a type of noise that occurs during the process of digitizing analog film videos and weak texture represents water droplets, fine objects, etc. Textural detail is an important factor that represents the characteristics of a video because it affects the mood and vibe of the video. Therefore, the frames generated by VFI must retain the textural detail of the original content. If textural detail is not preserved in the VFI process, the resulting video contains both the original frame with textural detail and the interpolated frame without textural detail. This results in the flicker phenomenon, where the textural detail part flickers on and off when playing back the interpolated video, severely degrading the subjective quality (as shown in 2nd and 3rd rows in Fig. 1).

Several studies have been conducted to directly and indirectly preserve the textural detail of the original content. As the direct preserving method, there are video coding algorithms [25], [26], [27] for efficiently encoding film videos with film grain noise. These algorithms remove film grain noise before compressing it because compressing the noise is inefficient from a coding perspective. The film grain noise can be preserved by adding synthesized noise to the decoded video. These methods preserve film grain noise similar to the original method, but have the disadvantage that the noise is completely synthesized data and cannot be guaranteed to be the same as the original pattern.

As an indirect methods, there is a method using perceptual loss in the training process for deep-learning-based VFI [1], [12], [28]. Perceptual loss extracts the features of an image using existing classification networks such as VGG-Net [29], ResNet [30], and AlexNet [31], and calculates the difference of features between the original and restored image. When perceptual loss was used, the subjective image quality and textural detail were improved. However, previous studies focused on preserving the outline of the image instead of the textural detail, they failed to preserve the textural detail of the original contents in the interpolated frame. As shown in Fig. 1, although AdaCoF [1] was trained using perceptual loss, it failed to preserve the textural detail contained in the ground truth (GT) frame. As a result, previous methods could not avoid flickering in videos with textural detail.

To address this problem, this paper proposes a VFI algorithm that preserves the textural detail of the original content. We determined the reason why textural detail was not preserved by analyzing the change in textural detail for each output of each sub-network (e.g., flow estimation network and synthesis network). Through this analysis, we discovered that textural detail disappeared during warping and merging during the VFI process and confirmed that they could be restored in the synthesis network. Therefore, we propose a new synthesis network, textural detail preservation network (TDPNet), and a training scheme to preserve textural detail. In addition, we propose a novel training method to enhance the performance at various resolutions, addressing the problem of performance degradation at resolutions different from the training data.

Our contributions can be summarized as follows:

1) This paper presented the problem that the existing VFI algorithms do not preserve textural detail, and raised the possibility of flicker. To the best of our knowledge, we were the first to raise an issue with this.
2) We have mathematically analyzed the reason why textural detail is not preserved in the previous deep learning-based VFI algorithms.
3) Based on the analysis, we confirmed the importance of a synthesis network in the VFI algorithm for preserving textural detail, and we proposed an enhanced synthesis network called Textural Detail Preservation Network

(TDPNet). And propose a Perceptual Training Method (PTM) that can appropriately utilize perceptual loss to preserve textural detail with minimal PSNR degradation in VFI.

4) To address the problem of poor performance of the VFI algorithm at resolutions different from the training data, we propose Multi-scale Resolution Training Method (MRTM), which can improve performance at various resolutions.

In Section II, we discuss previous works on VFI and efforts in other fields to preserve the textural detail of images. In Section III, we present our proposed network and training methods. And we validate our proposed network and training method in Section IV. Finally, the conclusions of our study are provided in Section V.

II. RELATED WORK

A. VIDEO FRAME INTERPOLATION

VFI is a technique for generating one or several non-existent frames $I_t (0 < t < 1)$ between two frames I_0 and I_1. With the development of deep learning, VFI methods using deep learning have been widely studied, including flow-based methods [2], [3], [5], [6], [7], [13], [17], [18], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], kernel-based methods [1], [4], [12], [28], [42], [43], and hallucination methods [16], [44], [45].

The kernel-based VFI algorithms estimate an adaptive convolution kernel for every pixel value in the input frames, and synthesize intermediate frames by convolving local patches. This method was first proposed by Nikalus [42]. However, this method has a high computational cost and requires a large amount of memory, therefore the size of the kernel is limited due to GPU memory limitations. To address the GPU memory issues, SepConv [4] was proposed to estimate the kernel using a separate 1D kernel instead of the previous 2D kernel estimation method, which successfully resolved the GPU memory problem. However, the problem of a limited kernel size still exists, which means that the algorithm cannot estimate motions larger than the kernel size. Subsequently, various algorithms have been proposed [12], [28] to address the issue of the existing method, which always refers to a fixed area, such as an algorithm that uses Deformable Convolution to add offsets to each kernel position [1], [43]. However, there is still the challenge of difficulty in estimating a large motion due to the problem of the computation complexity and the limitation of the kernel size.

On the other hand, flow-based VFI algorithms can estimate large motions without limiting the kernel size. A typical flow-based VFI algorithms process is as follows. The flow estimation network [2], [17], [41], [46], [47] estimates the flows $F_{0\to1}$ and $F_{1\to0}$ between input frames I_0, I_1, and based on the estimated flow, the intermediate flows $F_{t\to0}$, $F_{t\to1}$ are calculated. The intermediate frame $\hat{I}_t$ is thereafter created by warping the input frames with the estimated intermediate flow using Eq. (1) and a mask (M) learned by the flow estimate network using Eq. (2).

$$\hat{I}_{0\to t} = warp(I_0, F_{t\to0}), \hat{I}_{1\to t} = warp(I_1, F_{t\to1}) \quad (1)$$

$$\hat{I}_t = M \odot \hat{I}_{0\to t} + (1 - M) \odot \hat{I}_{1\to t} \quad (2)$$

In addition, for the final frame synthesis, the residual (R) of the image is obtained through a synthesis network such as U-Net [48], GridNet [49], and the final frame $\tilde{I}_t$ is synthesized by adding it to the intermediate frame $\hat{I}_t$ as follows Eq. (3).

$$\tilde{I}_t = \hat{I}_t + R \quad (3)$$

Flow-based VFI algorithm was first proposed by Jiang et al. [5]. Subsequently, various algorithms have been proposed [2], [6], [32], [34], [37], such as adding context and depth information to estimate residuals more accurately through synthesis networks, learning intermediate flows directly through flow estimation networks [2], and using algorithms that remove synthesis networks [33]. Through these studies, the objective performance of the interpolation results has been improved. An algorithm that applies perceptual loss to enhance subjective image quality has been proposed [18]. However, when using perceptual loss, there is a problem in which objective performance is degraded while subjective quality improves. Moreover, there has been no analysis on the appropriate use of perceptual loss to address this issue in VFI. In this paper we analyze the appropriate use of perceptual loss in VFI and propose a novel network and training method.

B. IMAGE FEATURE PRESERVATION AND EVALUATION METRICS

Subjective image quality is an important factor for images and videos. In image restoration tasks such as VFI and Super-Resolution (SR), preserving image features such as texture, edges, blobs, textural detail, and structure of the original image has a significant effect on the subjective image quality.

Studies on preserving features in images have mainly been conducted in the field of SR. One of the methods for preserving the features of an image in SR is using a loss function. When using traditional pixel loss (e.g., $L1$ and $L2$ *loss*), objective metrics such as the PSNR can be improved. However, they do not consider the structural characteristics of the image, and calculating the average pixel value can lead to over-smoothing or blurring of the restored image, resulting in a limited representation of the edge, texture, and textural detail of the image. Perceptual loss was proposed to address this problem [50], [51]. Perceptual loss extracts the features of an image using existing classification and recognition networks VGG-Net [29], ResNet [30], and AlexNet [31], and calculates the difference in features between the original and restored images. However, there is a problem that PSNR is degraded when using perceptual loss.

Subsequently, the analysis was performed in a SR task to appropriately utilize the perceptual loss. This involves identifying the appropriate convolution layer of the pre-trained

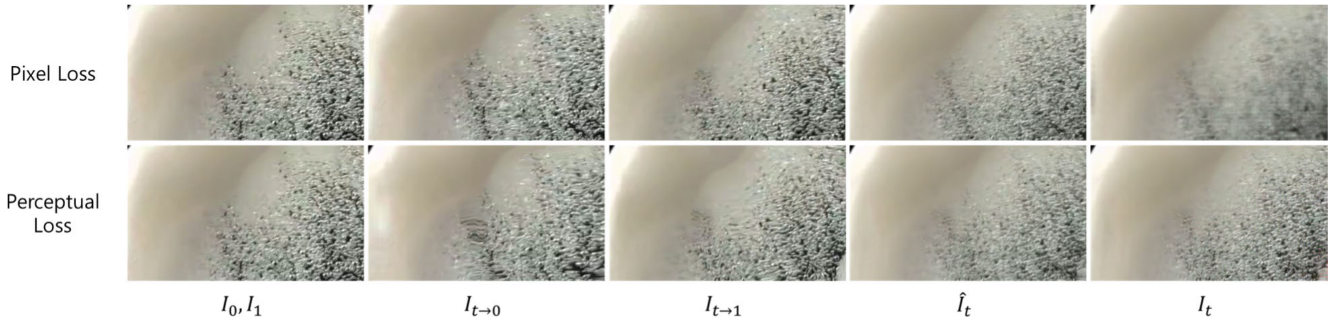

FIGURE 2. Visual quality changes when passing through sub-networks in a video frame interpolation algorithm. We use Inter4K [56] dataset for comparison.

CNN [52], [53]. A comparison of the features extracted by VGG-Net and ResNet suggested utilizing two features simultaneously for perceptual loss [54]. A method was proposed to appropriately use the perceptual loss for each segment such as the edge, boundary, and background of the images [55].

In addition, it is important to evaluate the restored images using metrics such as Video Quality Assessment (VQA) and Image Quality Assessment (IQA) that measure how well they align with the perception of human visual system. To quantify perceptual similarity between the restored and original images, metrics such as SSIM [57], MSSIM [58], and FSIM [59] have been proposed. Subsequently, deep learning-based evaluation metrics such as LPIPS [14] and DISTS [15] have been proposed to improve the accuracy of perceptual similarity evaluation. LPIPS utilize neural networks to extract features from images, which are then used to assess the perceptual similarity between the original and restored images. DISTS also utilizes neural networks, but is proposed to evaluate the sensitivity to structural distortion and texture resampling. In this paper, we use DISTS to evaluate the preservation of textural detail. These deep learning-based metrics have the advantage of being able to evaluate more complex and abstract image feature similarities.

III. PROPOSED METHOD

The proposed methods and analysis for preserving textural detail, which is part of image features are illustrated in this section. First, we analyzed the reasons for the lack of preservation of textural detail in previous VFI algorithms. Second, we introduce the proposed network architecture to preserve textural detail, TDPNet. Finally, we introduce a PTM that can preserve more textural details and an MRTM that can address the problem of performance degradation when there is a resolution difference between the training and test datasets.

A. WHY TEXTURAL DETAIL IS NOT PRESERVED IN VIDEO FRAME INTERPOLATION

We conducted experiments to analyze textural detail changes during VFI algorithm. For this analysis, we used the RIFE [2] algorithm. And we compared the image quality of the warped frames $\hat{I}_{0\to t}$ and $\hat{I}_{1\to t}$ generated using Eq. (1), merged frame $\hat{I}_t$ generated by Eq. (2), and the final interpolated frame $\tilde{I}_t$ generated by Eq. (3). Moreover, to analyze the impact of the VFI algorithm on the loss function, end-to-end training was performed using the traditional pixel loss ($L1$ *loss*) and perceptual loss. To train with perceptual loss, we used the 3rd convolution in the 4th layer of the VGG-16 network [29]. Conventional pixel loss improves the objective performance, but the subjective quality is poor because the image is over-smoothed or blurred. In contrast, perceptual loss is a loss function that can preserve image feature by utilizing the difference between feature maps. As a result, the subjective quality of the restored image is improved, but the objective quality is relatively poor compared to the pixel loss. This comparison of the two loss functions can identify structural issues in the VFI algorithm.

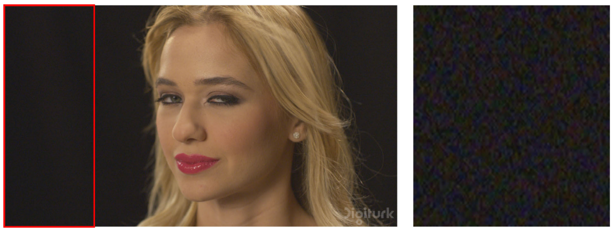

FIGURE 3. Contain Film Grain Noise in left 800 × 2160 area of the Beauty data in the UVG.

Fig. 2 shows the subjective results obtained using each loss function during VFI. When using pixel loss, it can be observed that the textural detail is preserved in $\hat{I}_t$ before the residual is added, however, in the final interpolated result $\tilde{I}_t$ where the residual is added, the textural detail becomes blurred. In contrast, when examining the results obtained using perceptual loss, we observed that textural detail was preserved not only in $\hat{I}_t$ before adding the residual, but also in the final interpolated frame $\tilde{I}_t$ where the residual is added. This is in contrast to the results obtained using pixel loss, where the textural detail becomes blurred in the final interpolated frame $\tilde{I}_t$ after adding the residual.

To quantitatively compare the preservation of textural detail, we utilized LPIPS [14] and DISTS [15] as the

TABLE 1. Change in quantitative metrics DISTS, LPIPS, and PSNR as video frame interpolation passes through each sub-network for HD and UVG datasets.

Dataset	Metric	Pixel Loss					Perceptual Loss				
		I_0, I_1, I_t	$\hat{I}_{0\to t}$	$\hat{I}_{1\to t}$	$\hat{I}_t$	$\tilde{I}_t$	I_0, I_1, I_t	$\hat{I}_{0\to t}$	$\hat{I}_{1\to t}$	$\hat{I}_t$	$\tilde{I}_t$
HD (Stockholm)	DISTS	-	0.036	0.045	0.057	0.069	-	0.041	0.052	0.063	0.032
	LPIPS	-	0.117	0.124	0.130	0.166	-	0.122	0.132	0.135	0.103
	PSNR	-	34.162	33.738	35.076	35.317	-	33.033	32.806	33.441	34.248
UVG (HoneyBee)	DISTS	-	0.039	0.039	0.064	0.127	-	0.041	0.045	0.067	0.022
	LPIPS	-	0.178	0.178	0.192	0.296	-	0.179	0.180	0.190	0.154
	PSNR	-	37.084	37.088	38.019	38.672	-	37.030	37.046	37.886	36.502
UVG (Beauty)	DISTS	-	0.090	0.086	0.112	0.186	-	0.096	0.090	0.110	0.054
	LPIPS	-	0.331	0.333	0.326	0.426	-	0.333	0.334	0.321	0.302
	PSNR	-	28.710	28.726	29.957	30.294	-	28.633	28.622	29.835	29.124
	Variance	29.594	16.851	16.872	9.303	3.297	29.594	16.826	16.838	9.437	24.459

metrics. LPIPS measures the perceptual similarity between the interpolated and original frame, whereas DISTS measures the texture similarity, which compares the similarity of the textural detail positions in the interpolated frame to the textural detail positions in the original frame. we use DISTS to evaluate the degree of preservation of film grain noise within the textural detail. To measure the degree of textural detail, the variance of the interpolated images was referenced. Furthermore, to assess the change in the objective result, we calculated the PSNR. Table 1 presents a quantitative comparison of the degree of textural detail preservation using the PSNR, LPIPS, DISTS, and variance. The variance was evaluated using only the left 800 × 2160 area of the beauty data in the UVG dataset to avoid the influence of the interpolation results depending on the estimated flow, as shown in Fig. 3.

As shown in Table 1, the variance decreases when the estimated flows $F_{t\to 0}$, $F_{t\to 1}$ and the input images I_0 and I_1 are backward warped to generate the warped frames $\hat{I}_{0\to t}$ and $\hat{I}_{1\to t}$. A decrease in variance indicates degradation of the textural detail. The reason for the degradation in textural detail is that, when performing backward warping, the position of each pixel in the warped frame is mapped to a pixel in the input frames, and the position of the mapped pixel is not always an integer. At this point, four integer pixels around the floating-point pixel are used for weighted averaging. A more detailed explanation is provided as follows.

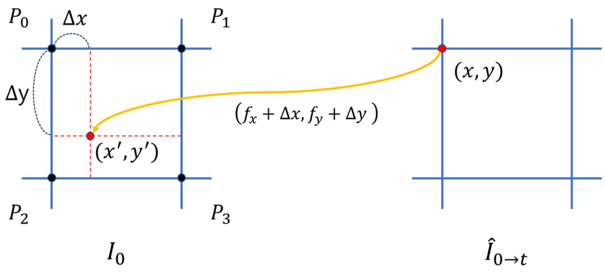

FIGURE 4. An illustration of the backward warping process.

For instance, as shown in Fig. 4, warped frame $\hat{I}_{0\to t}(x, y)$ is mapped to input frame $I_0(x', y')$ according to the estimated flow $F_{t\to 0}(x, y) = (f_x, f_y)$. The pixel positions in the input frame, denoted as x' and y', were obtained by adding the estimated flow values f_x and f_y, respectively, to the pixel coordinates in the warped frame (x, y). This can be expressed as $x' = x + f_x$ and $y' = y + f_y$. When the estimated flow values are non-integers, they can be calculated using Eq. (4), where ⌊⌋ represents rounding to the nearest integer smaller than the original number.

$$\begin{aligned} f_x &= \lfloor f_x \rfloor + \Delta x, \quad 0 \le \Delta x < 1 \\ f_y &= \lfloor f_y \rfloor + \Delta y, \quad 0 \le \Delta y < 1 \end{aligned} \tag{4}$$

If the pixel location x' and y' are not integers, bilinear interpolation is performed using the values of four neighboring integer pixels, namely P_0, P_1, P_2, and P_3, which are calculated using Equation (5). The pixel position corresponding to each pixel values are $(x + \lfloor f_x \rfloor, y + \lfloor f_x \rfloor)$, $(x + \lfloor f_x \rfloor + 1, y + \lfloor f_x \rfloor)$, $(x + \lfloor f_x \rfloor, y + \lfloor f_x \rfloor + 1)$, $(x + \lfloor f_x \rfloor + 1, y + \lfloor f_x \rfloor + 1)$.

$$\begin{aligned} \hat{I}_{0\to t} &= \Delta x \Delta y P_3 + (1 - \Delta x)\Delta y P_2 \\ &\quad + \Delta x(1 - \Delta y)P_1 + (1 - \Delta X)(1 - \Delta y)P_0 \end{aligned} \tag{5}$$

Assuming that each random variable is independent and identically distributed, neighboring pixels have the same variance, σ^2. Therefore, the variance of $\hat{I}_{0\to t}$ calculated by bilinear interpolation is equal to Eq. (6). $Var()$ denotes the variance. Within the range of Δx and Δy, the variance of $\hat{I}_{0\to t}$ has a minimum value of $0.25\sigma^2$ when $\Delta x = \Delta y = 0.5$, and a maximum value of σ^2 when $\Delta x = \Delta y = 0$. Therefore, the variance is reduced when backward warping because the

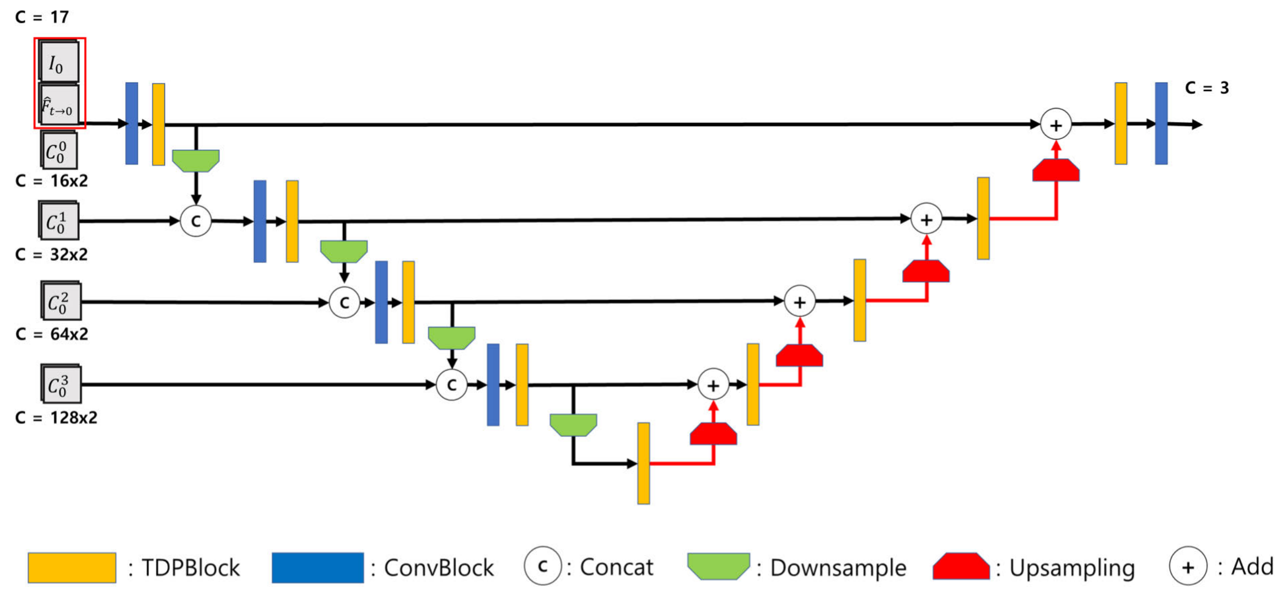

FIGURE 5. Overview of our proposed synthesis network architecture.

pixels are not always mapped to integer values.

$$Var(\hat{I}_{0\to t}) = \{2(\Delta x - 0.5)^2 + 0.5\}\{2(\Delta y - 0.5)^2 + 0.5\}\sigma^2 \tag{6}$$

As summarized in Table 1, the variance of the merged frame degrades. $\hat{I}_t$, which is obtained using Eq. (2), combines the warped frame $\hat{I}_{0\to t}$ and $\hat{I}_{1\to t}$ with a mask. In this process, the variance of $\hat{I}_t$ changes according to Eq. (7), assuming that the random variables are independent and identically distributed. By assumption, the values of $Var(\hat{I}_{0\to t})$ and $Var(\hat{I}_{0\to t})$ are equal to σ^2. Because the value of the estimated mask is $0 \leq M < 1$, the variance has a minimum value of $0.5\sigma^2$ when the mask is 0.5, and a maximum value of σ^2 when the mask is 0 or 1. As a result, when merging warped frames, the values of the mask are not always 0 or 1, which leads to variance degradation in the variance.

$$Var(\hat{I}_t) = M^2 \times Var(\hat{I}_{0\to t}) + (1 - M)^2 \times Var(\hat{I}_{1\to t}) \tag{7}$$

Therefore, the variance of the frame inevitably decreases when backward warping and merging of warped frames are performed. As summarized in Table 1, the variance of the warped and merged frames were degraded compared with that of the input frame. When perceptual loss is used, the variance value can be degraded in a manner similar to pixel loss because it involves backward warping and merging operations.

However, when the residuals trained by the synthesis network were added to $\hat{I}_t$, the resulting variance differed depending on the loss function used. Specifically, using pixel loss reduces the variance, whereas using perceptual loss increases the variance. This shows that the residual contains the textural detail of the image.

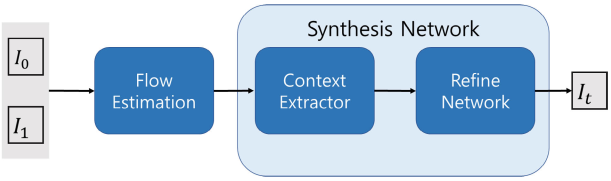

FIGURE 6. Overall Video Frame Interpolation algorithm.

Similar to the change in variance, we observed that both LPIPS and DISTS metrics exhibited similar trends. This analysis demonstrated the importance of residual training using a synthesis network to preserve textural detail. To generalize these results, we conducted experiments on additional datasets containing textural detail, and the results are summarized in Table 1.

Based on this analysis, it is clear that to preserve the textural detail effectively, an improvement in the performance of the synthesis network is necessary. Therefore, we propose an improved synthesis network, the Textural Detail Preservation Network, to preserve the textural detail of images. Furthermore, the results demonstrate that even if perceptual loss is used in the VFI algorithm, the textural detail is not restored until a residual is added. Therefore, it is inefficient to use perceptual loss end-to-end, and we propose a new learning method to address this problem.

B. TEXTURAL DETAIL PRESERVATION NETWORK

The overall architecture of the proposed algorithm is shown in Fig. 6. To estimate the flow, we employ IFNet from a pre-existing RIFE algorithm [2]. The proposed synthesis network comprises a context extractor that can extract context features from the input image and a refine network that learns

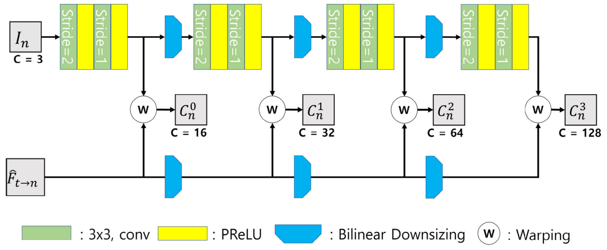

FIGURE 7. Overview of context extractor for generate context feature. Context extractor extracts context features at 1, 1/2, 1/4, and 1/8 resolution of the input frame.

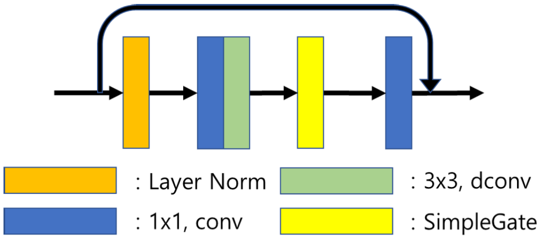

FIGURE 8. Overview of Textural Detail Preservation Block (TDPBlock).

the residual containing textural detail information using the extracted features, flow estimated from the input, and other input images.

The context extractor generates features by convolving the input image I_n (n=0,1) and warping it with the flow $F_{t\to n}$ estimated by the flow estimator. This generated contextual features correspond to the middle frame. To generate context features of varying scales, we employed bilinear downsizing by 1/2, resulting in a total of four context feature scales. Fig. 7 illustrates the detailed structure of the network.

The refine network uses the flow, mask, warped image, input images, and context features learned from previous networks, flow estimation network and context extractor, as inputs to learn the residual. The refine network adopted a U-Net structure that incorporated a skip-connection structure to utilize information at different resolutions. Within the U-Net structure, we utilize ConvBlock and Textural Detail Preservation Block (TDPBlock). The detailed network structure is illustrated in Fig. 5.

The TDPBlock in the proposed refine network is an effective structure for enhancing the performance and reducing the inference time. To improve the performance, we used layer normalization, which is insensitive to mini-batch size and data distribution, instead of batch normalization, which is used on a mini-batch basis. To reduce the computational complexity, we implemented SimpleGate as an activation function. SimpleGate is a simplified form of computation compared with GeLU, which was proposed in NAFNet [60]. SimpleGate is a method that divides the feature map into channel dimensions and performs element-wise multiplication, as shown in Eq. (8). Furthermore, we utilize depth-wise convolution, which is calculated separately for each channel to preserve the spatial information of each channel, while also having a lower computational complexity than 2D Convolution.

$$SimpleGate(X, Y) = X \odot Y \tag{8}$$

C. TRAINING METHOD

1) PERCEPTUAL TRAINING

Previous algorithms that used perceptual loss achieved an improved subjective quality. However, when calculating the loss, input frame passes through a deep convolution layer to calculate the difference between the original and interpolated frames with reduced resolution, which affects the flow estimation in end-to-end training. Consequently, the flow estimation is inaccurate, resulting in poor performance [1], [18], [28]. To address this problem, we propose a novel training method that improves the subjective image quality and objective results compared with a simple application of perceptual loss. Based on our analysis of the synthesis network preserving textural detail using perceptual loss, the proposed PTM comprises two stages. In the first stage, the our proposed network is trained end-to-end using the loss function $\mathcal{L}_{total}$ as shown in the following Eq. (9).

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \lambda_d \mathcal{L}_{dis} \tag{9}$$

We use the distillation loss $\mathcal{L}_{dis}$ in IFNet, which is a flow estimation network based on the RIFE [2] algorithm. This distillation loss accurately estimates the flow by leveraging the difference between the flow learned by the student and the teacher models. The pixel loss $\mathcal{L}_{rec}$ was also utilized to measure the difference between the interpolated and the GT frames. The detailed formula for $\mathcal{L}_{rec}$ is provided in the following Eq. (10).

$$\mathcal{L}_{dis} = \sum_{i \in 0,1} \left\| F_{t\to i} - F^{tea}_{t\to i} \right\|_2 \tag{10}$$

$$\mathcal{L}_{rec} = \left\| I_t - I_{gt} \right\|_1 \tag{11}$$

In the second stage, we perform fine-tunning to preserve the textural detail of the residuals trained by the synthesis network. In the second stage, we used a loss function that combined $\mathcal{L}_{fine}$, $\mathcal{L}_{total}$, and the perceptual loss $\mathcal{L}_{per}$ to train the model. The detailed equations for this loss function are shown in Eqs. (12). We set $\lambda_d = 0.01$, $\lambda_p = 0.1$.

When using perceptual loss, selecting an appropriate value for λ_p is important because textural detail may not be preserved to the maximum. To best preserve textural detail, we conducted experiments using the RIFE algorithm and employed Eq. (12) as the loss function. In this experiment, we fixed λ_d to 0.01 and varied only λ_p in the second stage of the PTM to obtain the results for DISTS. The dataset used to obtain the experiment results is the Beauty data in the UVG dataset. In Fig.9, we observe that DISTS is minimal when λ_p is 0.1, thus we select 0.1 as the value that preserves the most textural detail.

$$\mathcal{L}_{fine} = \mathcal{L}_{rec} + \lambda_d \mathcal{L}_{dis} + \lambda_p \mathcal{L}_{per} \tag{12}$$

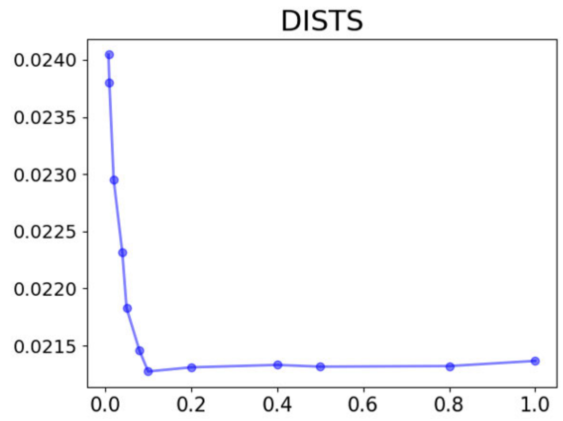

FIGURE 9. DISTS tendency according to the percentage of perceptual loss. The x-axis refers to λ_p, and the y-axis refers to DISTS values.

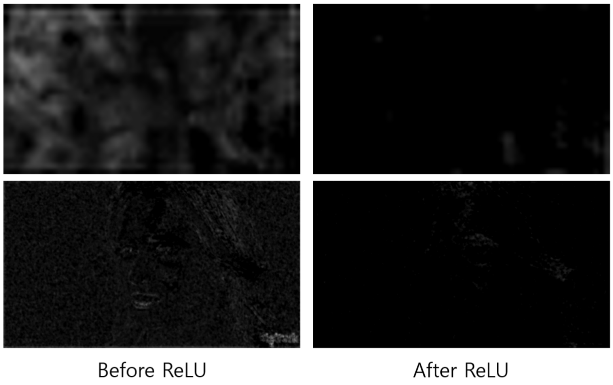
Before ReLU
After ReLU

FIGURE 10. Compare VGG feature before activation function and after.

The perceptual loss $\mathcal{L}_{per}$ is defined by Eq. (13), where Φ denotes the output of the 3rd covolution in the 4th layer of the VGG-16 network [29]. As shown in Fig. 10, most of the features are lost when passing through the activation function, thus we extract the features before the activation function after the convolution layer.

$$\mathcal{L}_{per} = \left\| \Phi(I_t) - \Phi(I_{gt}) \right\|_1 \tag{13}$$

2) MULTI-SCALE RESOLUTION TRAINING

A multi-scale resolution training method (MRTM) was proposed to address the problem of decreased performance when there is a resolution difference between the training and test datasets. This is achieved by adding various resolutions to the training dataset and training the network. To increase the resolution, the EDSR network was used to perform 2× upsampling. In each iteration of the training process, the training data were randomized to ensure that the network was trained at various image resolution. The detailed experimental results are discussed in Section IV-C.

IV. EXPERIMENT RESULT

In this section, we introduce the implementation details for training the proposed network and the evaluation metric, which is the benchmark used to evaluate our method. We thereafter quantitatively and qualitatively compare our method with recent state-of-the-art approaches. Finally, an ablation study is conducted to analyze the effectiveness of our proposed method.

A. IMPLEMENTATION DETAILS

1) TRAINING DATA

We used Vimeo90K and Vimeo90K (X2) upsampled by ×2 using EDSR [61]. The Vimeo90K data consists of 51,312 triplets and has a resolution of 448 × 256. We augment the datasets by randomly horizontal and vertical flipping and cropping 224 × 224 patches.

2) TRAINING STRATEGY

We implemented our proposed network using PyTorch 1.9.0 version. During training we used AdamW [62] to optimize our proposed model with a weight decay 10^{-3}, and the learning rate was gradually reduced from 3×10^{-4} to 3×10^{-5} using cosine annealing. For end-to-end training, we used a batch size of 64 for 230k iterations, and for fine-tuning, we used a batch size of 64 for 76k iterations. Our training was performed on a single NVIDIA RTX A6000, taking 48 hours for end-to-end training and 12 hours for fine-tuning.

3) EVALUATION METRICS

To evaluate the proposed model comprehensively, we evaluated it on datasets of various resolutions using PSNR for objective image quality evaluation and LPIPS [14] and DISTS [15] for perceptual similarity.

4) EVALUATION DATASETS

Vimeo90K [63]: 448 × 256 resolution with triplets frames per clip. It consists of 3782 clips in total.
SNU-FILM [44]: 1280 × 720 resolution with a total of 1240 frames, categorized into easy, medium, hard, and extreme according to the motion magnitude.
HD [35]: It consist of 11 videos. The HD dataset consists of four 1080p, three 720p and four 544p videos, and we used the first 100 frames of each video
UVG [64]: It consists of 7 videos with 3840×2160 resolution, and we used the first 300 frames of each video.

B. COMPARISONS WITH THE STATE-OF-THE-ART

To compare our proposed algorithm, we used recent flow-based algorithms, such as ABME [32], RIFE [2], and IFRNet [33], kernel-based algorithms AdaCoF [1] and CDFI [28], and the hallucination-based algorithm CAIN [44]. We measured the inference time, PSNR, LPIPS, and DISTS using NVIDIA RTX A6000 to ensure a justified comparison. Previous algorithms were compared using pre-trained weight files and publicly available code. In terms of inference time, we used the 720p HD dataset [35] to evaluate the performance of each algorithm.

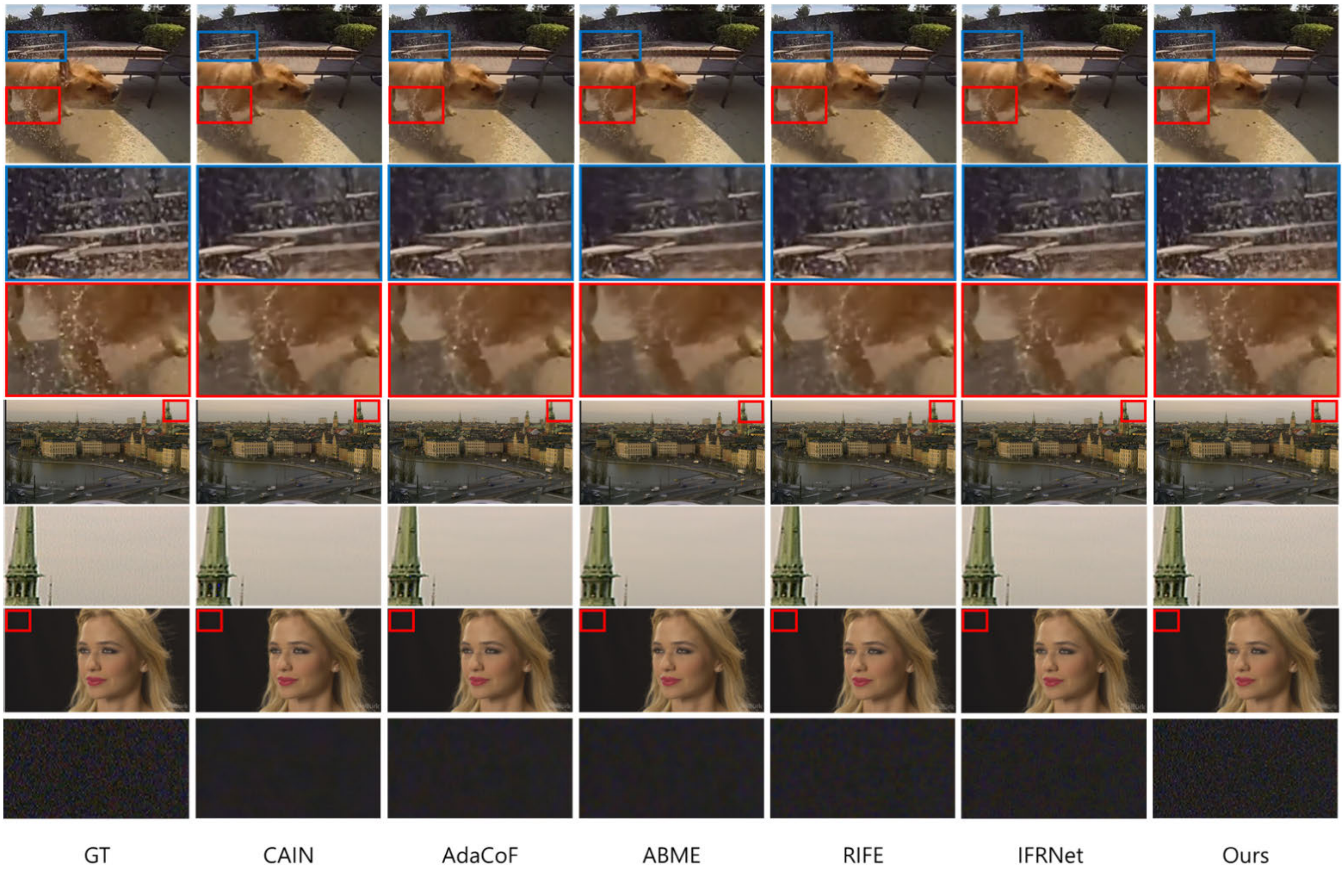

FIGURE 11. Visual results between the proposed algorithm and previous algorithm.

TABLE 2. Quantitative results comparison on benchmark datasets. (Red indicates best PSNR values within each dataset, and Blue indicates second best).

Algorithm	Vimeo90K	HD	SNU-FILM				UVG	Average	Param (M)	Inference Time (ms)
			easy	medium	hard	extreme				
	PSNR(↑)	PSNR(↑)	PSNR(↑)	PSNR(↑)	PSNR(↑)	PSNR(↑)	PSNR(↑)			
CAIN [44]	34.645	31.795	39.893	35.606	29.895	24.777	31.115	32.532	42.8	46.4
AdaCoF [1]	34.379	31.010	39.855	35.074	29.473	24.311	30.547	32.093	21.8	62.7
ABME [32]	36.201	31.665	39.640	35.797	30.590	25.426	32.167	33.069	18.1	326.7
CDFI [28]	35.194	31.470	40.120	35.526	29.753	24.543	OMM	OMM	4.9	431.6
RIFE [2]	35.558	32.140	39.984	35.761	30.116	24.869	31.191	32.802	9.8	39.1
IFRNet [33]	35.799	32.151	39.975	35.910	30.399	25.049	31.339	32.894	5.0	41.3
Ours_{nv}	36.045	32.659	40.144	36.069	30.571	25.401	32.226	33.301	12.4	70.2
Ours	35.669	32.108	40.008	35.831	30.307	25.075	31.271	32.894	12.4	70.2

Fig. 11 presents a subjective image quality comparison between the previous VFI algorithms and our proposed network for SNU, HD, and UVG datasets that contain textural detail. The comparison shows that the previous algorithms failed to preserve the textural detail of the video, such as the film grain noise shown in 5th and 7th rows of Fig. 11, while our proposed network preserves it. Moreover, the previous algorithms tended to over-smooth weak textures shown in 2nd and 3rd rows in Fig. 11. In contrast, our proposed network preserve the weak textures. In addition, when comparing AdaCoF [1] and CDFI [28] twhich use perceptual loss, the preservation of textural detail remains poor. However, our proposed network preserves both the film grain noise and weak texture due to the appropriate use of perceptual loss.

The PSNR, inference time, and parameter comparison results of recent and proposed networks are summarized in Table 2. In the table, $Ours_{nv}$ represents the results of end-to-end training with the loss function $\mathcal{L}_{total}$ without

TABLE 3. Quantitative results comparison on benchmark datasets. (Red indicates best LPIPS/DISTS values within each dataset and Blue indicates second best.).

Algorithm	Vimeo90K		HD		SNU-FILM								UVG	
					easy		medium		hard		extreme			
	LPIPS(↓)	DISTS(↓)	LPIPS(↓)	DISTS(↓)	LPIPS(↓)	DISTS(↓)	LPIPS(↓)	DISTS(↓)	LPIPS(↓)	DISTS(↓)	LPIPS(↓)	DISTS(↓)	LPIPS(↓)	DISTS(↓)
CAIN [44]	0.051	0.05	0.196	0.106	0.036	0.025	0.059	0.037	0.118	0.063	0.207	0.106	0.38	0.177
AdaCoF [1]	0.049	0.048	0.182	0.082	0.035	0.023	0.059	0.034	0.112	0.052	0.197	0.082	0.334	0.148
ABME [32]	0.036	0.037	0.177	0.094	0.035	0.025	0.055	0.037	0.093	0.052	0.165	0.077	0.360	0.175
CDFI [28]	0.030	0.027	0.142	0.062	0.025	0.013	0.042	0.021	0.091	0.036	0.171	0.066	OMM	OMM
RIFE [2]	0.039	0.037	0.174	0.085	0.032	0.021	0.051	0.030	0.094	0.045	0.176	0.076	0.338	0.139
IFRNet [33]	0.035	0.035	0.153	0.072	0.032	0.021	0.049	0.029	0.085	0.040	0.159	0.065	0.287	0.105
Ours	0.025	0.020	0.114	0.048	0.022	0.010	0.037	0.015	0.072	0.025	0.144	0.047	0.213	0.053

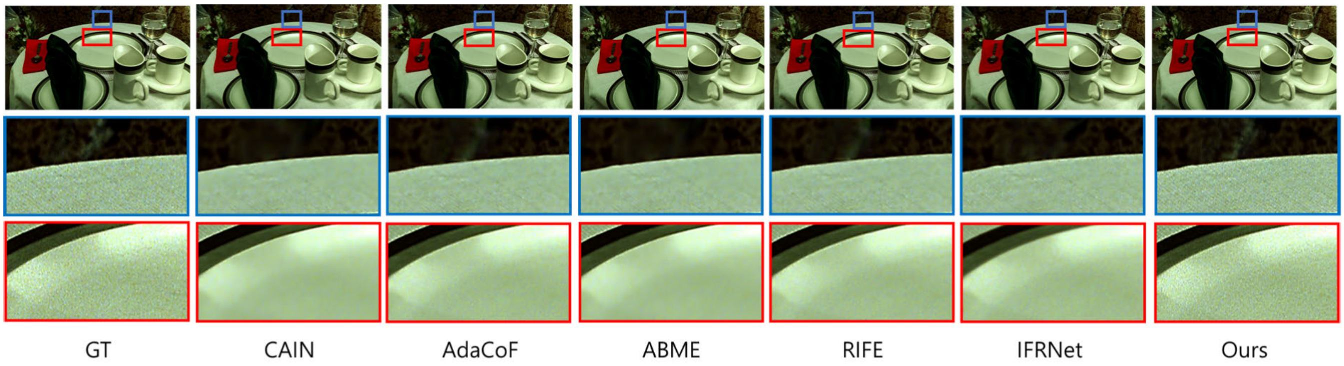

FIGURE 12. Film scanned real video [65] visual comparison between the proposed algorithm and previous algorithm.

applying the proposed PTM to measure the performance of the proposed synthesis network. OMM denotes out of memory, it was not possible to measure the UltraVideo data for the CDFI algorithm due to exceeding the memory available on out GPU. The results of the proposed synthesis network demonstrate that the PSNR performance is improved for most of the test data compared to existing algorithms. The proposed network has the best PSNR for the HD, SNU-FILM (easy), SNU-FILM (medium), and UVG datasets and the second-best performance for the remaining datasets. Moreover, the proposed network has an average PSNR improvement of 0.232dB compared to ABME [32], which has the best overall performance among existing algorithms. The inference time of the proposed network is approximately 6 times faster than that of ABME, and it has fewer parameters. When the PTM was applied, the PSNR was degraded compared to $Ours_{nv}$ because the PSNR was degraded when perceptual loss was applied. However, there is still an improvement in PSNR compared to AdaCoF and CDFI, which use perceptual loss for training.

In Table 3, the results of LPIPS and DISTS are summarized to determine whether the existing and proposed networks preserve textural detail. The results indicate that the proposed network preserves the textural detail for all test data. Among the existing algorithms, CDFI which uses perceptual loss exhibited the best performance on average for both LPIPS and DISTS. However, the proposed network outperformed CDFI on all datasets in terms of both LPIPS and DISTS.

TABLE 4. BMPRI results on film scanned real video.

	CAIN	AdaCoF	ABME	CDFI	RIFE	IFRNet	Ours
MBPRI	51.28	45.66	42.65	15.83	31.61	23.68	7.03

Finally, we performed a subjective image quality comparison using the film scanned real video [65] to compare the results to see if textural detail is restored in real-video. In the second column of Fig. 12, we can see that the textures of the tablecloth are restored in the proposed network. In the third column, we can see that the film grain noise of the film scanned real video is not restored by the existing networks, but it is restored by the proposed network. We measured the VFI results using BMPRI [66], an NR-IQA, as a image quality assessment for real video scanned to film. The results are shown in Table 4, which shows that our proposed algorithm outperformed the existing VFI algorithms.

C. ABLATION STUDY

To evaluate the effectiveness of our proposed training methods, PTM and MRTM, we conducted an ablation study. Table 5 compares the PSNR results based on the proposed training method. A comparison of LPIPS and DISTS results based on the proposed training method is listed in Table 6.

1) MULTI-SCALE RESOLUTION TRAINING METHOD

The baseline model was trained end-to-end using the proposed synthesis network without the proposed training

TABLE 5. PSNR results for ablation studies of the proposed training method.

Perceptual Loss	PTM	MRTM	Vimeo90K	HD	SNU-FILM				UVG	Average
					easy	medium	hard	extreme		
			PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR
X	X	X	36.069	32.250	39.999	35.859	30.166	24.953	31.189	32.926
X	X	O	36.045	32.659	40.144	36.030	30.571	25.401	32.226	33.301
O	X	O	35.189	31.908	39.926	35.598	30.102	24.869	30.612	32.601
O	O	O	35.660	32.108	40.008	35.831	30.307	25.075	31.271	32.894

TABLE 6. LPIPS and DISTS results for ablation studies of the proposed training method.

Perceptual Loss	PTM	MRTM	Vimeo90K		HD		SNU-FILM								UVG	
							easy		medium		hard		extreme			
			LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS	LPIPS	DISTS
X	X	X	0.037	0.036	0.172	0.086	0.032	0.021	0.051	0.032	0.096	0.048	0.176	0.080	0.338	0.137
X	X	O	0.038	0.038	0.174	0.088	0.034	0.023	0.054	0.035	0.092	0.050	0.168	0.077	0.345	0.146
O	X	O	0.026	0.021	0.115	0.048	0.022	0.011	0.037	0.015	0.074	0.025	0.151	0.051	0.223	0.059
O	O	O	0.024	0.020	0.114	0.048	0.022	0.010	0.037	0.015	0.072	0.025	0.144	0.047	0.213	0.053

method. When applying multi-scale resolution training, PSNR was improved for large resolution such as HD, SNU-FILM, and UVG datasets compared to the baseline by training with various resolutions. In particular, for the extreme data of SNU-FILM with large motion, the PSNR was improved by 0.447 dB, and for the UVG dataset with a 4K resolution video, the PSNR improvement was 1.037 dB. The average PSNR overall test dataset improved by 0.375 dB. It can be observed that the performance of the MRTM is improved for various resolutions, and the results for datasets with large motions are significantly improved. However, as summarized in Table 6, the results for DISTS and LPIPS with MRTM were similar to the baseline.

2) PERCEPTUAL TRAINING METHOD

The results indicate that using only perceptual loss improves the performance of DISTS and LPIPS compared with the baseline model, but the PSNR results are degraded. However, applying our proposed PTM enhanced the PSNR results compared with not applying our method. The proposed method improved the PSNR for the entire dataset by an average of 0.293 dB compared with a model that simply applied perceptual loss, with a specific improvement of 0.659 dB for the UVG dataset. Moreover, the performance of LPIPS and DISTS has also been improved. These findings confirm the importance of the residuals learned by the synthesis network in enhancing the subjective image quality, and demonstrate the effectiveness of the proposed Perceptual Optimal Training method in optimally preserving the residuals.

V. CONCLUSION

In this paper, we presented an analysis for preserving textural detail in VFI and proposed a synthesis network and learning method optimized for preserving such details based on this analysis. Our proposed synthesis network demonstrated improved objective performance compared with those of existing algorithms. Moreover, subjective and objective improvements were observed when using our proposed PTM compared with simply applying perceptual loss. We successfully resolved the problem of performance degradation at different resolutions of the training data using MRTM. However, the issue of PSNR degradation when perceptual loss is used, remains to be addressed.

REFERENCES

[1] H. Lee, T. Kim, T. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5315–5324.

[2] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 624–642.

[3] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4463–4471.

[4] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 261–270.

[5] H. Jiang, D. Sun, V. Jampani, M. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9000–9008.

[6] W. Bao, W. Lai, C. Ma, X. Zhang, Z. Gao, and M. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3698–3707.

[7] T. Peleg, P. Szekely, D. Sabo, and O. Sendik, "IM-Net for high resolution video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2393–2402.

[8] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proc. 14th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 286–301.

[9] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deep stereo: Learning to predict new views from the world's imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5515–5524.

[10] T. Brooks and J. T. Barron, "Learning to synthesize motion blur," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6833–6841.

[11] S. Y. Kim, J. Oh, and M. Kim, "FISR: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11278–11286.

[12] S. Niklaus, L. Mai, and O. Wang, "Revisiting adaptive convolutions for video frame interpolation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 1098–1108.

[13] P. Hu, S. Niklaus, S. Sclaroff, and K. Saenko, "Many-to-many splatting for efficient video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3543–3552.

[14] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[15] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2567–2581, May 2022.

[16] W. Shangguan, Y. Sun, W. Gan, and U. S. Kamilov, "Learning cross-video neural representations for high-quality frame interpolation," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2022, pp. 511–528.

[17] X. Jin, L. Wu, G. Shen, Y. Chen, J. Chen, J. Koo, and C. Hahm, "Enhanced bi-directional motion estimation for video frame interpolation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 5038–5046.

[18] F. Reda, J. Kontkanen, E. Tabellion, D. Sun, C. Pantofaru, and B. Curless, "FILM: Frame interpolation for large motion," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2022, pp. 250–266.

[19] M. Hu, K. Jiang, L. Liao, Z. Nie, J. Xiao, and Z. Wang, "Progressive spatial–temporal collaborative network for video frame interpolation," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 2145–2153.

[20] M. Hu, K. Jiang, Z. Nie, and Z. Wang, "You only align once: Bidirectional interaction for spatial–temporal video super-resolution," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 847–855.

[21] M. Hu, K. Jiang, L. Liao, J. Xiao, J. Jiang, and Z. Wang, "Spatial–temporal space hand-in-hand: Spatial–temporal video super-resolution via cycle-projected mutual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3574–3583.

[22] Z. Geng, L. Liang, T. Ding, and I. Zharkov, "RSTT: Real-time spatial temporal transformer for space-time video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17420–17430.

[23] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Blurry video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5113–5122.

[24] W. Shen, W. Bao, G. Zhai, L. Chen, X. Min, and Z. Gao, "Video frame interpolation and enhancement via pyramid recurrent framework," *IEEE Trans. Image Process.*, vol. 30, pp. 277–292, 2021.

[25] Z. Ameur, W. Hamidouche, E. François, M. Radosavljević, D. Menard, and C.-H. Demarty, "Deep-based film grain removal and synthesis," 2022, *arXiv:2206.07411*.

[26] B. T. Oh, S.-M. Lei, and C.-C.-J. Kuo, "Advanced film grain noise extraction and synthesis for high-definition video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1717–1729, Dec. 2009.

[27] I. Hwang, J. Jeong, S. Kim, J. Choi, and Y. Choe, "Enhanced film grain noise removal and synthesis for high fidelity video coding," *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. 96, no. 11, pp. 2253–2264, 2013.

[28] T. Ding, L. Liang, Z. Zhu, and I. Zharkov, "CDFI: Compression-driven network design for frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7997–8007.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[32] J. Park, C. Lee, and C. Kim, "Asymmetric bilateral motion estimation for video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14519–14528.

[33] L. Kong, B. Jiang, D. Luo, W. Chu, X. Huang, Y. Tai, C. Wang, and J. Yang, "IFRNet: Intermediate feature refine network for efficient frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1959–1968.

[34] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1701–1710.

[35] W. Bao, W. Lai, X. Zhang, Z. Gao, and M. Yang, "MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 933–948, Mar. 2021.

[36] S. Niklaus and F. Liu, "Softmax splatting for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5436–5445.

[37] J. Park, K. Ko, C. Lee, and C.-S. Kim, "BMBC: Bilateral motion estimation with bilateral cost volume for video interpolation," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Aug. 2020, pp. 109–125.

[38] H. Zhang, Y. Zhao, and R. Wang, "A flexible recurrent residual pyramid network for video frame interpolation," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 474–491.

[39] Y. L. Liu, Y. T. Liao, Y. Y. Lin, and Y. Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8794–8802.

[40] H. Li, Y. Yuan, and Q. Wang, "Video frame interpolation via residue refinement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2613–2617.

[41] H. Sim, J. Oh, and M. Kim, "XVFI: EXtreme video frame interpolation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14469–14478.

[42] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2270–2279.

[43] X. Cheng and Z. Chen, "Video frame interpolation via deformable separable convolution," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10607–10614.

[44] M. Choi, M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, "Channel attention is all you need for video frame interpolation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10663–10671.

[45] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3522–3532.

[46] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[47] D. Sun, X. Yang, M. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[48] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[49] D. Fourure, R. Emonet, E. Fromont, D. Muselet, A. Tremeau, and C. Wolf, "Residual conv-deconv grid network for semantic segmentation," 2017, *arXiv:1707.07958*.

[50] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

[51] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. 14th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 694–711.

[52] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, p. 10.

[53] F. Kong, M. Li, S. Liu, D. Liu, J. He, Y. Bai, F. Chen, and L. Fu, "Residual local feature network for efficient super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 765–775.

[54] J. Song, H. Yi, W. Xu, X. Li, B. Li, and Y. Liu, "Dual perceptual loss for single image super-resolution using ESRGAN," 2022, *arXiv:2201.06383*.

[55] M. S. Rad, B. Bozorgtabar, U. Marti, M. Basler, H. K. Ekenel, and J. Thiran, "SROBB: Targeted perceptual loss for single image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2710–2719.

[56] A. Stergiou and R. Poppe, "AdaPool: Exponential adaptive pooling for information-retaining downsampling," *IEEE Trans. Image Process.*, vol. 32, pp. 251–266, 2023.

[57] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[58] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, 2003, pp. 1398–1402.

[59] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[60] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. 17th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 17–33.

[61] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.

[62] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," 2017, *arXiv:1711.05101*.

[63] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[64] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proc. 11th ACM Multimedia Syst. Conf.*, May 2020, pp. 297–302.

[65] T. Suzuki, *Ad Hoc Group Report: Study of 4:4:4 Functionality*, document JVT-R009, JVT, Bangkok, Thailand, Jan, 2006.

[66] X. Min, G. Zhai, K. Gu, Y. Liu, and X. Yang, "Blind image quality estimation via distortion aggravation," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 508–517, Jun. 2018.

KIHWAN YOON received the B.S. degree from the Department of Electronic and Electrical Engineering, Dankook University, Yongin, South Korea, in 2019. He is currently pursuing the integrated M.S. and Ph.D. degree in electrical and computer engineering with the University of Seoul, Seoul, South Korea. He is a Research Engineer with the Intelligent Image Processing Research Center, Korea Electronics Technology Institute, Seongnam, South Korea. His research interests include multimedia signal processing and artificial intelligence processing.

JINGANG HUH received the B.S. degree in electronic engineering from Gangneung-Wonju National University, Gangwon, South Korea, in 2015, and the M.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2017. Since 2016, he has been a Research Engineer with the Intelligent Image Processing Research Center, Korea Electronics Technology Institute, Seongnam, South Korea. His research interests include multimedia signal processing, artificial intelligence processing, and distributed system development.

YONG HAN KIM (Member, IEEE) was born in South Korea, in 1959. He received the B.S. and M.S. degrees in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1982 and 1984, respectively, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. From 1984 to 1996, he was with the Electronic and Telecommunications Research Institute, Daejeon, South Korea. From 1991 to 1992, he was on leave from NTT Human Interface Laboratories, Yokosuka, Japan, as a Visiting Researcher. In 1996, he joined the University of Seoul, Seoul, where he is currently a Professor with the School of Electrical and Computer Engineering. His research interests include visual data compression and audio-visual communication systems. In 2017, he served as the President for the Korean Institute of Broadcast and Media Engineers (KIBME).

SUNGJEI KIM received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2004, 2006, and 2011, respectively. From 2011 to 2015, he was a Senior Video Signal Processing Engineer with Samsung Electronics Company Ltd., Suwon, South Korea. Since 2015, he has been a Principal Research Engineer with the Intelligent Image Processing Research Center, Korea Electronics Technology Institute, Seongnam, South Korea. His research interests include multimedia signal processing, artificial intelligence processing, and VR/AR technologies.

JINWOO JEONG received the B.S., M.S., and Ph.D. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2004, 2006, and 2011, respectively. From 2011 to 2015, he was a Senior Video Signal Processing Engineer with Samsung Electronics Company Ltd., Suwon, South Korea. Since 2016, he has been a Principal Research Engineer with the Intelligent Image Processing Research Center, Korea Electronics Technology Institute, Seongnam, South Korea. His research interests include multimedia signal processing, artificial intelligence processing, and VR/AR technologies.