MOTION ESTIMATION ALGORITHM BASED ON DECODED RESIDUAL COMPENSATION: A PROOF-OF-CONCEPT

Lee Sze Foo[⋆], *Anissa Mokraoui*[†], *Fangchen Feng*[†], *Yoong Choon Chang*[⋆]

[⋆] Universiti Tunku Abdul Rahman, Malaysia,
[†] L2TI, Université Sorbonne Paris Nord, France

ABSTRACT

In most video coding standards, the reduction of temporal redundancy in a video is based on the traditional block-matching algorithm (BMA). It first estimates the motion vectors that minimize the distortion between the original image and its predicted version. The difference between these two images, i.e. residual image, is then encoded and its decoded version compensates the predicted image. This paper proposes an algorithm that estimates the motion vectors while taking into account the impact of the decoded residual image on the quality of the compensated image. This algorithm provides a higher PSNR for a given bit rate compared to the traditional method. This proof-of-concept shows the importance of taking into account the compensated image in the motion vector estimation process and should help in the design of solutions based on deep neural networks.

Index Terms— Video compression, block-matching algorithm (BMA), motion estimation, residual compensation

I. INTRODUCTION

The recent studies show that video transmission represents more than 70% of internet traffic and is expected to continue to increase in the coming years [1]–[3]. It is, therefore, necessary to consider the performance improvement of the current conventional video coding systems. Commonly used and/or newer video codecs are based on the H.264, H.265, and VP9 standards [4], [5]. These video codecs rely on the same hybrid block-based architecture and aim to reduce both spatial and temporal redundancy in the video stream. Moreover, the state of the art shows that many issues remain to be explored, especially with the emergence of artificial intelligence (see e.g. [6]–[8]).

This paper focuses on the reduction of temporal redundancy in a video while guaranteeing significant gains compared to classical schemes. Most video coding standards (e.g. H264/AVC, HEVC) adopted the solution of mapping blocks of different sizes combining both prediction strategies, motion vector coding and, depending on the case (type P or B), residual error coding in order to compensate for the motion [7], [9].

The aim of this paper is to provide a proof-of-concept work showing that estimating motion vectors according to the compensated image, instead of the predicted image, yields increased rate-distortion performance. The performance of this strategy should also help the design of deep neural networks based video coding schemes, particularly for the motion estimation process.

II. BASIC CONCEPTS OF TRADITIONAL BMA

In a traditional video coding framework, a motion estimation algorithm is first used to predict the target frame I_t from the previous frame I_{t-1} through motion vectors v_t, producing the predicted image $\bar{I}_t$. While the motion vectors v_t are transmitted directly to the decoder through entropy coding, the residual R_t, which is the difference between I_t and $\bar{I}_t$, undergoes lossy compression before being transmitted. The decoder then uses the previously reconstructed frame $\hat{I}_{t-1}$ already available at the decoder and the decoded $\hat{v}_t$ to generate the predicted frame $\bar{I}_t$ before compensating it with the decoded residual $\hat{R}_t$, producing the final reconstructed target frame:

$$\hat{I}_t = \bar{I}_t + \hat{R}_t. \tag{1}$$

In the case of the traditional exhaustive BMA, the motion estimation is performed by first decomposing I_t into K non-overlapping square blocks. For each block $k \in \{1, 2, \ldots, K\}$ in I_t whose coordinates are denoted as (x, y), the block in I_{t-1} that most resembles the target block in I_t is determined through an exhaustive search that considers every possible candidate block in a search range S whose displacement from the target block is described by a motion vector $v_t(k) = \mathbf{s} = (\Delta\mathbf{x}, \Delta\mathbf{y})$. The best matching block with motion vector $\mathbf{s}_{\text{match}}$ is determined by minimizing the distortion D between the target block and the candidate blocks in I_{t-1}, often calculated as the Mean Squared Error (MSE):

$$D(x, y, \mathbf{s}) = \text{MSE}\left(I_t(x, y), I_{t-1}(x, y, \mathbf{s})\right), \tag{2}$$

$$v_t(x, y) = \mathbf{s}_{\text{match}} = \underset{\mathbf{s}\in S}{\text{argmin}}\, D(x, y, \mathbf{s}). \tag{3}$$

The block in I_{t-1} that minimizes the distortion would then be selected as the k-th block of the predicted frame, i.e. $\bar{I}_t(x, y) = I_{t-1}(x, y, \mathbf{s}_{\text{match}})$. It is interesting to notice that the equation (3) is equivalent to minimizing the distortion between $I_t(x, y)$ and $\bar{I}_t(x, y)$:

$$v_t(x, y) \equiv \underset{\mathbf{s}\in S}{\text{argmin}}\, \text{MSE}\left(I_t(x, y), \bar{I}_t(x, y)\right). \tag{4}$$

Therefore, the traditional BMA described above essentially selects the motion vectors based on minimizing the distortion between I_t and $\bar{I}_t$. However, it can be seen from equation (1) that the quality of the final reconstructed image depends not only on the predicted image but also on the decoded residual which is not taken into account in traditional BMA.

III. PROPOSED MOTION ESTIMATION ALGORITHM BASED ON RESIDUAL COMPENSATION

This section presents the strategy of the novel proposed motion estimation algorithm that takes the decoded residual into consideration as a proof-of-concept. The underlying idea was initiated for disparity map estimation for stereoscopic image coding by the authors of [10]. The proposed algorithm is organized into two main stages described in this section. The first stage (called Algorithm 1), considered as an initialization step, significantly improves the performance and can be sufficient in some cases. The second stage (called Algorithm 2) considers the optimization problem of the quality factors to further improve the performance.

A. Initialization step (Algorithm 1)

Similar to the traditional BMA, the proposed strategy relies on an exhaustive search of $\mathbf{s} \in S$ in I_{t-1} to identify the best match of the k-th block in I_t. We propose a new cost function of the search which is based on the MSE between the target

block in I_t and the compensated candidate block. Specifically, for each candidate block $I_{t-1}(x, y, \mathbf{s})$, the residual, i.e. $R_t(x, y, \mathbf{s}) = I_t(x, y) - I_{t-1}(x, y, \mathbf{s})$ undergoes a Discrete Cosine Transform (DCT), a quantization step controlled by a quality factor q, followed by a lossless coding (as in JPEG) and a subsequently decoding procedure, producing the following decoded residual:

$$\hat{R}_t(x, y, \mathbf{s}, q) = C_q^{-1}\left(C_q\left(R_t(x, y, \mathbf{s})\right)\right), \tag{5}$$

where C_q and C_q^{-1} denote respectively the JPEG encoding and decoding process with a corresponding quality factor q. Specifically, $C_q(I) = Q_q\left(\text{DCT}(I)\right)$ and $C^{-1}(I) = \text{IDCT}\left(Q_q^{-1}(I)\right)$ where Q_q and Q_q^{-1} denote respectively the quantization and its inverse operation, while IDCT denotes the inverse DCT.

The decoded residual $\hat{R}_t(x, y, \mathbf{s}, q)$ is then added to compensate the candidate block to produce the reconstructed block:

$$\hat{I}_t(x, y, \mathbf{s}, q) = \bar{I}_t(x, y, \mathbf{s}) + \hat{R}_t(x, y, \mathbf{s}, q). \tag{6}$$

The motion vector is then obtained through the minimization of the proposed cost function of the search which is the distortion between $I_t(x, y)$ and $\hat{I}_t(x, y, \mathbf{s}, q)$:

$$D(x, y, \mathbf{s}, q) = \text{MSE}\left(I_t(x, y), \hat{I}_t(x, y, \mathbf{s}, q)\right), \tag{7}$$

$$v_t(x, y) = \mathbf{s}_{\text{match}} = \underset{\mathbf{s}\in S}{\text{argmin}}\, D(x, y, \mathbf{s}, q). \tag{8}$$

We can see from the above formulas that by passing the residual through the encoding and decoding process for each candidate block, the decoded residual is taken into account during the motion estimation procedure. After completing the search for $k \in \{1, 2, \ldots, K\}$, the global bitrate is calculated as the summation of the entropy of the quantized residual $Q_q\left(\text{DCT}(\hat{R}_t)\right)$ and the entropy of the motion vectors v_t:

$$b = \text{entropy}\left[Q_q\left(\text{DCT}(\hat{R}_t)\right)\right] + \frac{\text{entropy}\left(v_{t,x} \frown v_{t,y}\right) \cdot K}{\text{no. of pixels in } I_t}, \tag{9}$$

where $\frown$ denotes the concatenation operation.

The quality of the reconstructed frame is evaluated using the peak-signal-to-noise radio (PSNR):

$$\text{PSNR} = 10 \cdot \log\left[\frac{255^2}{\text{MSE}\left(I_t, \hat{I}_t\right)}\right]. \tag{10}$$

The proposed algorithm is summarized in Fig. 1. The pseudo-code of this initialization step is given in Algorithm 1.

Algorithm 1 Initialization step

Input : I_{t-1}, I_t,q
Output : $\hat{I}_t$, b, PSNR
for $k \in \{1, 2, \ldots, K\}$ **do**
  cost $= \infty$
  for $\mathbf{s} \in S$ **do**
    Compute $\hat{R}_t(x, y, \mathbf{s}, q)$ with equation (5)
    Compute $\hat{I}_t(x, y, \mathbf{s}, q)$ with equation (6)
    Compute $D(x, y, \mathbf{s}, q)$ with equation (7)
    if $D(x, y, \mathbf{s}, q) <$ cost **then**
      Select $\mathbf{s}$ as the optimal motion vector
      $D(x, y, \mathbf{s}, q) =$ cost
    end if
  end for
end for
Compute b using equation (9)
Compute PSNR using equation (10)

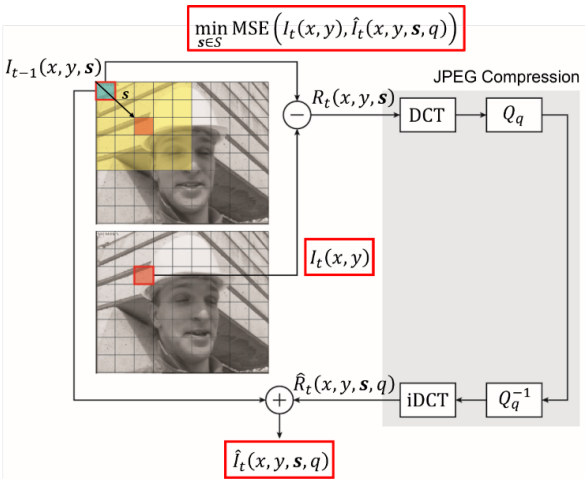

Fig. 1: Schematic of the proposed algorithm.

B. *Optimization step (Algorithm 2)*

The compression process of $R_t(x, y, \mathbf{s})$ in equation (5) depends on the quality factor q. A naïve way is to choose a pre-fixed value of this parameter for every block k. To further improve the performance of the proposed algorithm described in the previous section, we propose to implement the Algorithm 1 as an initialization step where the predicted image $\bar{I}_t$ and the corresponding motion vectors v_t are produced. We then perform an optimization step of the parameter q where the global bitrate is set as a target bitrate b_{target} which is pre-defined.

The optimization is carried out as follows: For every $k \in \{1, 2 \ldots, K\}$, a set of quality factors is tested, where for every $q \in \{q_1, q_2, \ldots, q_N\}$, an exhaustive search for the optimal motion vector according to equation (8) is performed based on the global distortion and the global bitrate. It is important to notice that, as the quality factor is fine-tuned and potentially different for every block, this information should also be transmitted and the number of bits required must be taken into account for the bitrate calculation. Thus, the global bitrate is calculated as follows:

$$b = \text{entropy}\left[Q_q\left(\text{DCT}(\hat{R}_t)\right)\right] + \frac{\text{entropy}\left(v_{t,x} \frown v_{t,y} \frown F\right) \cdot K}{\text{no. of pixels in } I_t}, \tag{11}$$

where F is the matrix which contains the quality factor of every block, e.g. $F(x, y) = q$. In practice, all elements of F are initialized with the quality factor used in Algorithm 1.

As the motion vector $\mathbf{s}_{\text{match}}$, the global MSE$(I_t, \hat{I}_t)$ and the global bitrate b_q depend on the quality factor q, to optimize this parameter for a particular block, we choose the q as the optimal quality factor q_{optim} that minimizes MSE$(I_t, \hat{I}_t)$, given that $b_q < b_{\text{target}}$. In the case where no q satisfies $b_q < b_{\text{target}}$, the q that produces b_q closest to b_{target} is chosen.

In this manner, the initialized F is updated for the k-th block, i.e. $F(x, y) = q_{\text{optim}}$. Similarly, the initialized v_t is updated as $v_t(x, y) = \mathbf{s}_{\text{match},q_{\text{optim}}}$, and the initialized predicted image is updated as $\bar{I}_t(x, y) = I_{t-1}\left(x, y, \mathbf{s}_{\text{match},q_{\text{optim}}}\right)$. The optimization is then repeated for every block.

The quality of the final reconstructed image is evaluated using the PSNR defined in equation (10). This optimization step is summarized in Algorithm 2.

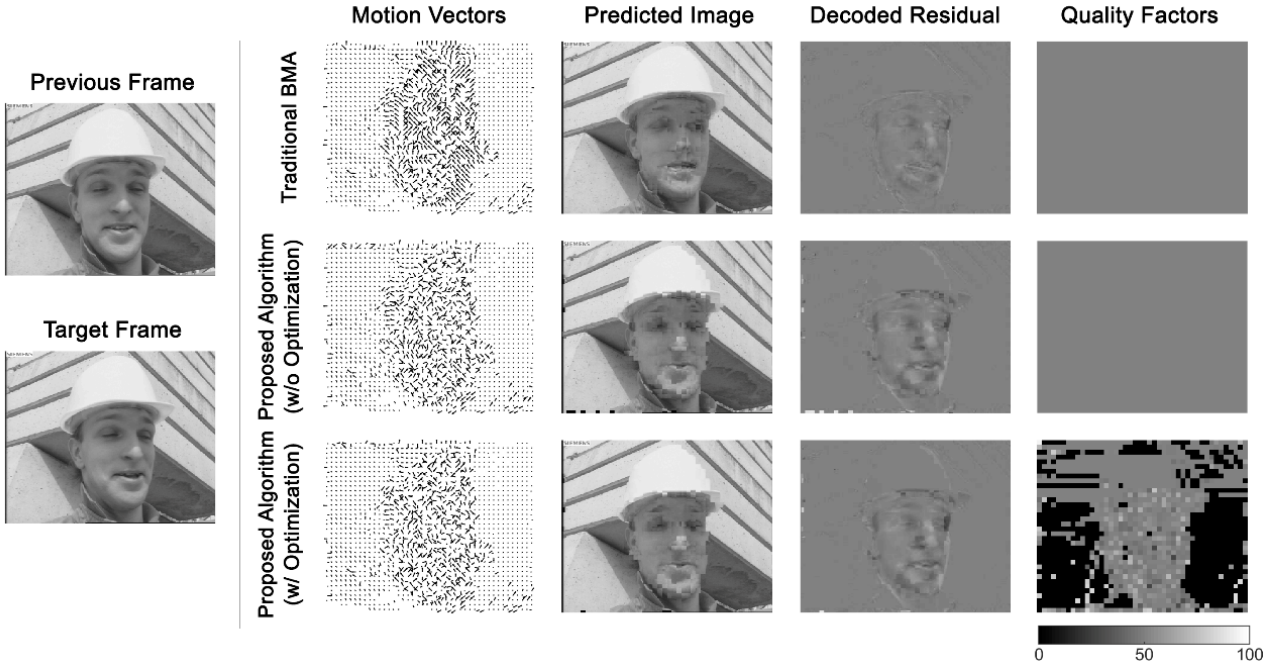

Fig. 2: Motion vectors, predicted image, decoded residual, and quality factors of the traditional BMA and proposed algorithms without and with optimization at bitrate of ~ 0.35 bpp (traditional BMA: 0.358 bpp; proposed algorithms: 0.351 bpp).

Algorithm 2 Optimization step

Input : I_{t-1}, I_t, $\{q_1, q_2, \ldots, q_N\}$
Output : $\hat{I}_t$, b, PSNR
Initialize $\bar{I}_t$ and v_t, and set b_{target} using Algorithm 1
Initialize F as the quality factor used in Algorithm 1
for $k \in \{1, 2, \ldots, K\}$ **do**
  for $q \in \{q_1, q_2, \ldots, q_N\}$ **do**
    cost $= \infty$
    for $\mathbf{s} \in S$ **do**
      Compute $\hat{R}(x, y, \mathbf{s}, q)$ with equation (5)
      Compute $\hat{I}_t(x, y, \mathbf{s}, q)$ with equation (6)
      Compute $D(x, y, \mathbf{s}, q)$ with equation (7)
      if $D(x, y, \mathbf{s}, q) <$ cost **then**
        Select $\mathbf{s}$ as $\mathbf{s}_{\text{match},q}$
        $D(x, y, \mathbf{s}, q) =$ cost
      end if
    end for
    Compute b_q using equation
  end for
  if any $b_q < b_{\text{target}}$ **then**
    Select q_{optim} as q that gives the smallest MSE $\left(I_t, \hat{I}_t\right) | b_q < b_{\text{target}}$
  else
    Select q_{optim} as q that gives b_q closest to b_{target}
  end if
  Update $F(x, y) = q_{\text{optim}}$
  Update $v_t(x, y) = \mathbf{s}_{\text{match},q_{\text{optim}}}$
  Update $\bar{I}_t(x, y) = I_{t-1}(x, y, \mathbf{s}_{\text{match},q_{\text{optim}}})$
end for
Compute b using equation (11)
Compute PSNR using equation (10)

IV. Performance of the Proposed Algorithm

The proposed algorithms were tested on three CIF (352×288) format sequences, namely "Foreman", "Stefan", and "Bus". Results are provided for the 1st and 2nd frames of "Foreman", "Stefan", and "Bus" as well as the 39-th and 40-th frames of "Foreman". The frames were chosen based on the significant movement between the two consecutive frames. Block size of 8×8 pixels and a search range of 16×16 pixels (4-pixel extension from each side of the position of the target block) was set. Each side of the previous frames were padded with zero by lengths of 4 pixels. Nine JPEG compression quality factors – 10, 15, 20, 30, 40, 50, 60, 70, 80 were tested, generating nine data points on the rate-distortion curve. For the proposed algorithm with optimization step, the range of quality factors tested during the optimization of each block was set as 1-100 with a pace of 1.

We show in Fig. 2 the motion vectors, the predicted image, the decoded residual, and the quality factors obtained from traditional BMA, the proposed initialization step (Algorithm 1), and the proposed algorithm with optimization (Algorithm 2). As expected, the motion vectors produced by traditional BMA and the proposed algorithms were found to be substantially different, giving rise to different predicted images, indicating that the decoded residual highly affects the choice of the motion vectors. The optimization step in the proposed Algorithm 2 further introduces slight differences in the selected motion vectors as the quality factor is also taken into consideration.

While the traditional BMA and the proposed algorithm without optimization (initialization) performed JPEG compression using the same quality factors throughout the whole image, the proposed optimization tunes the quality factor chosen for each block as seen in the right most column of Fig. 2. Most notably, the optimization allocates higher quality factors for regions with larger movement, e.g. the Foreman, and lower quality factors for regions with little movement, e.g. the background. This effectively allows the algorithm to allocate larger number of bits in the areas of the

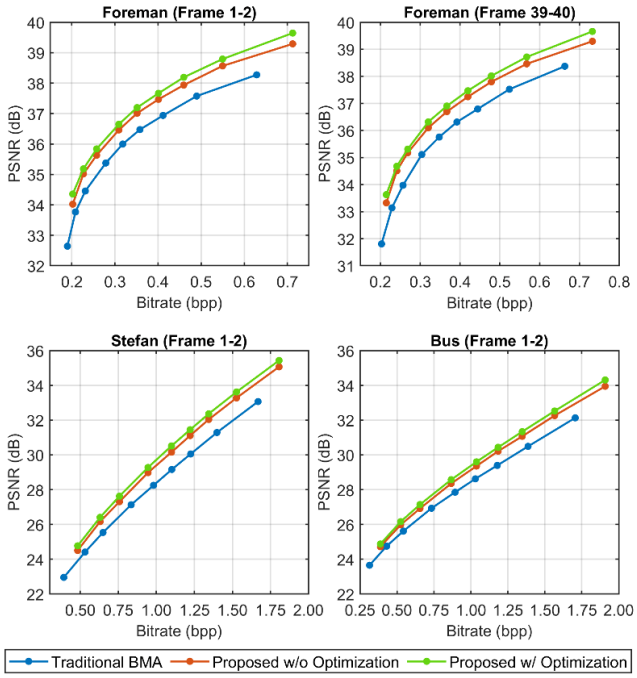

Fig. 3: Rate-distortion curves of tested frames in "Foreman", "Stefan", and "Bus".

image that require larger amounts of information thus producing an improved image quality.

The rate-distortion curves of the tested frames are presented in Fig. 3. The proposed algorithm without optimization outperformed the traditional BMA in all the tested videos and upon optimizing the quality factor, the proposed algorithm further increased the reconstructed image quality. The visual comparison of the reconstructed images of "Foreman" and "Stefan" are presented in Fig. 4 and Fig. 5, respectively. As expected, the proposed algorithm with optimization generated reconstructed images that had the highest visual quality among the three algorithms with less artifacts and noise, e.g. around the sign "WATCH" in "Stefan".

Although the proposed algorithm increases the complexity and is thus more time consuming compared to the traditional BMA, especially with optimization applied, this paper presents an early proof-of-concept demonstrating that the video reconstruction quality for a given bitrate can be improved when the decoded residual is considered.

V. Conclusion and Future Work

The proposed motion estimation algorithm which takes the decoded residual into consideration increases the reconstructed image quality for a given bitrate compared to traditional BMA. This demonstrates the importance of considering the decoded residual in the motion estimation process. However, due to its computational complexity, further improvement is needed to implement the proposed algorithms in a real-time scenario. Nevertheless, this proof-of-concept work should help in the design of solutions based on deep neural networks that should learn the motions taking into account the compensated image. The replacement of the motion estimation module in video coding standards by deep neural networks should take this into account.

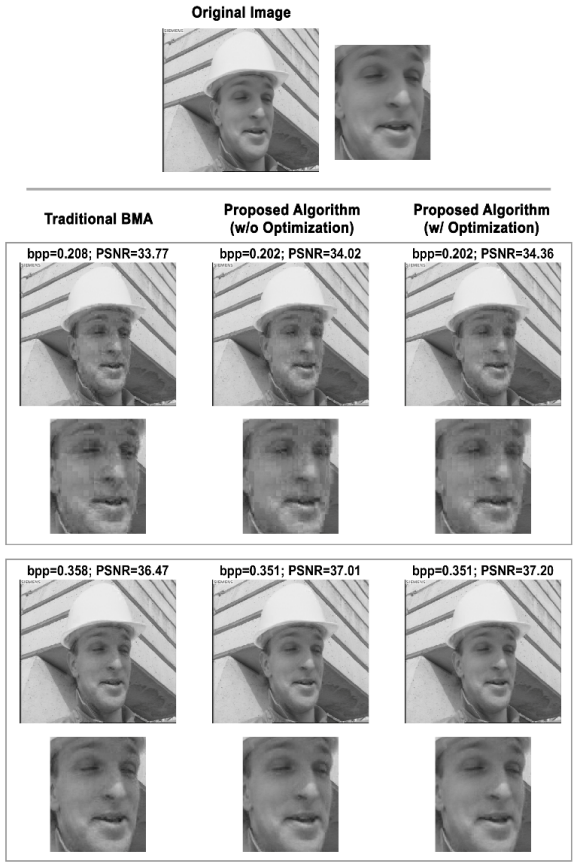

Fig. 4: Comparison of the reconstructed images of Frame 2 of "Foreman" at low bitrate ($\sim$ 0.20 bpp) and higher bitrate ($\sim$ 0.35bpp).

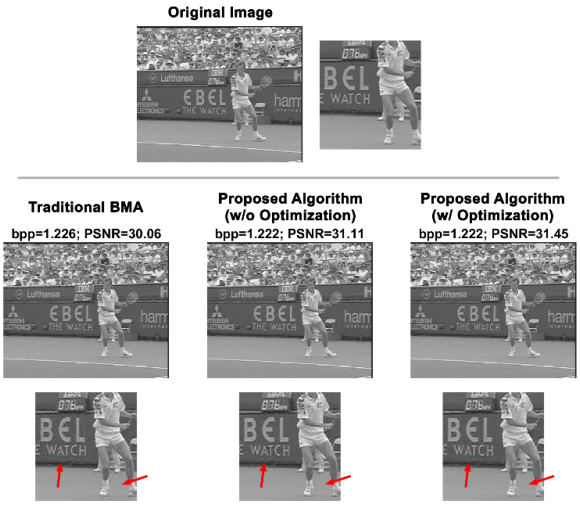

Fig. 5: Comparison of the reconstructed images of Frame 2 of "Stefan" at bitrate of $\sim$ 1.22 bpp.

REFERENCES

[1] Thomas Barnett, Shruti Jain, Usha Andra, and Taru Khurana, "Cisco visual networking index (vni) complete forecast update, 2017–2022," *Americas/EMEAR Cisco Knowledge Network (CKN) Presentation*, 2018.

[2] Trinh Man Hoang and Jinjia Zhou, "Recent trending on learning based video compression: A survey," *Cognitive Robotics*, vol. 1, pp. 145–158, 2021.

[3] Benjamin Bross, Jianle Chen, Jens-Rainer Ohm, Gary J Sullivan, and Ye-Kui Wang, "Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc)," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463–1493, 2021.

[4] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[5] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao, "Dvc: An end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11006–11015.

[7] Krishna Kumar, Krishan Kumar, and Rahul Mishra, "Block-based motion estimation in video frames using artificial neural networks: a selective review," *International Journal of Computer Applications*, vol. 975, pp. 8887, 2016.

[8] Jiahao Li, Bin Li, and Yan Lu, "Deep contextual video compression," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[9] Iain E Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia*, John Wiley & Sons, 2004.

[10] Imen Kadri, Gabriel Dauphin, Anissa Mokraoui, and Zied Lachiri, "Disparities selection controlled by the compensated image quality for a given bitrate," *Signal, Image and Video Processing*, vol. 14, no. 6, pp. 1143–1151, 2020.