Nokia Exhibit 2008, p. 1
Amazon.com v. Nokia
IPR2024-00605

A Multistage Motion Vector Processing Method for Motion-Compensated Frame Interpolation

Ai-Mei Huang, *Student Member, IEEE*, and Truong Q. Nguyen, *Fellow, IEEE*

***Abstract*—In this paper, a novel, low-complexity motion vector processing algorithm at the decoder is proposed for motion-compensated frame interpolation or frame rate up-conversion. We address the problems of having broken edges and deformed structures in an interpolated frame by hierarchically refining motion vectors on different block sizes. Our method explicitly considers the reliability of each received motion vector and has the capability of preserving the structure information. This is achieved by analyzing the distribution of residual energies and effectively merging blocks that have unreliable motion vectors. The motion vector reliability information is also used as a prior knowledge in motion vector refinement using a constrained vector median filter to avoid choosing identical unreliable one. We also propose using chrominance information in our method. Experimental results show that the proposed scheme has better visual quality and is also robust, even in video sequences with complex scenes and fast motion.**

***Index Terms*—Frame rate up-conversion, motion-compensated frame interpolation (MCFI), motion vector processing, residual energy.**

I. Introduction

MOTION-compensated frame interpolation (MCFI) that uses the received motion vectors (MVs) has recently been studied to improve temporal resolution by increasing the frame rate at the decoder. MCFI is particularly useful for video applications that have a low bandwidth requirement and need to reduce frame rate to improve spatial quality. However, MCFI that directly uses the received MVs often suffers from annoying artifacts such as blockiness and ghost effect. This is because the received motion vector field (MVF) is usually generated using block-based motion estimation at the encoder by minimizing prediction errors, rather than finding true motion. To solve this problem, a number of works have focused on producing a smoother MVF for better frame interpolation by finding true motion in motion estimation or processing the received MVs at the decoder.

In order to get true motion at the encoder, MVs are estimated by further considering spatial and temporal correlations in addition to block matching algorithm (BMA) [1]. Krishnamurthy *et al.* proposed using a multiscale optical-flow-based motion estimator to produce smooth, natural motion fields [2]. Motion estimation at the decoder has also been studied to re-estimate MVs, instead of relying on the encoder to produce a smooth MVF. Chen proposed estimating the forward and backward MVs between two frames reconstructed at the decoder by exploiting the spatial correlation of motion among adjacent blocks [3]. Ha *et al.* suggested using an overlapped block-based motion estimation to get more accurate motion trajectory [4]. The works in [5] and [6] reduced the computational complexity by selectively performing motion estimation with prior classification. Shinya and Akira proposed re-estimating MVs using BMA but with the concept that a larger block size should be used for global motion regions while a smaller block size is used for local motion regions [7]. It needs to perform motion estimation up to three times at the decoder in order to identify a global motion region and find its motion. To get more accurate and smoother MVFs, object-based motion refinement by performing image segmentation was presented in [8]. The method in [9] also used motion-based segmentation to find object boundaries and applied mesh-based motion compensation to interpolate the regions inside the objects.

Alternatively, instead of producing a smooth MVF using motion estimation, which requires higher complexity, MV processing techniques have been proposed by simply removing MV outliers and/or refining MVs from its neighborhood. A commonly used method is to apply vector median filter [10]. Alparone *et al.* proposed using adaptively weighted vector median filter based on prediction errors to obtain a smoother MVF *at the encoder* after motion estimation [11]. Dane *et al.* addressed the same concept and applied vector median filter *at the decoder* to correct irregular MVs [12]. The authors further presented another MV processing method in [13] that resamples MVs with smoothness measurement to decrease blockiness artifact. The work in [14] analyzed the reliability of the received MVF according to the number of intracoded macroblocks (MBs), isolated MB detection and MV variance. It does not interpolate the frame if the received MVF is not reliable to use. Sekiguchi *et al.* used weighted averaging of neighboring MVs based on their prediction errors to obtain a smooth motion field [15]. Zhang *et al.* proposed a method that detects isolated MVs and selects the best motion from adjacent blocks based on temporal modeling [16]. These MV processing techniques are also useful for other applications, such as error concealment and transcoding [17], [18].

In addition to MV processing, several works have also discussed how to reduce visual artifacts by adaptively choosing forward, backward, or bidirectional interpolations. This is because in areas where occlusion happens, unidirectional interpolation may provide better visual quality. A pixel-wise nonlinear

Manuscript received March 9, 2007; revised January 3, 2008. This work was supported in part by Conexant, Inc., and in part by the University of California Discovery program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitri Van De Ville.

The authors are with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093 USA (e-mail: aihuang@ucsd.edu; nguyent@ece.ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2008.919360

1057-7149/$25.00 © 2008 IEEE
Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 2
Amazon.com v. Nokia
IPR2024-00605

filtering approach for frame interpolation was first addressed in [19]. Krishnamurthy *et al.* proposed sending side information to inform the decoder which interpolation scheme should be used [2]. A similar concept of this encoder assisted approach was presented in [20], where the encoder decides how to interpolate the frame based on several predefined interpolation equations and sends that information as side information to the decoder. The work in [3] suggested choosing the forward and backward predictions adaptively based on their corresponding boundary absolute difference. Lee *et al.* also addressed this concept but used weighted averaging interpolation by considering multiple motion trajectories and prediction errors [21]. A theoretical analysis of adaptively choosing forward and backward interpolation was presented in [22].

In general, it is difficult for an encoder to accurately capture all the motion in a video frame using block-based motion estimation, or coding efficiency will significantly be reduced in order to send such detailed motion information. It is also unrealistic to assume that all the encoders are made aware of frame interpolation at the decoder. Even though MVs can be re-estimated at the decoder by considering spatial and temporal correlations, the true motion can easily be distorted due to coding artifacts such as blockiness and blurriness. The MV processing methods that remove outliers using vector median filter or refine MVs using smaller block sizes can only perform well in the areas with smooth and regular motion. That is, they are based on the assumption that the MVF should be smooth. However, this is usually not true as a video frame may contain complex motion, especially on the motion boundaries, where the true motion field is not smooth at all. As a result, irregular motion may appear in the received MVF and dominate the vector median filtering process to use those irregular MVs as the true motion. In addition, since many of the methods only operate on a smaller block size, they often fail to consider the edge continuity and the structure of the objects. We can often see broken edges and destroyed structures in an interpolation frame. Besides, MBs that are intracoded also make frame interpolation difficult as their MVs are not available. Some methods use object-based motion estimation and interpolation at the decoder to maintain the object structure and minimize the interpolation errors [8], [9]. However, high computational complexity may prevent them from being used in resource limited devices such as mobile phones. Therefore, frame interpolation still remains a very challenging problem as the artifacts due to the use of improper MVs can be very noticeable, or an extremely complex method has to be employed.

In [23], we have shown that the correlation between the residual information and the reliability of received MVs. We also observed that the interpolation usually fails at the areas where the density of inaccurate MVs is high. Therefore, instead of correcting MVs using smaller block sizes, we suggested finding a single MV for a group of adjacent MBs [24]. The preliminary results have shown that they are able to reduce visual artifacts in an interpolated frame. However, the approach in [23] performs better when the motion is relatively smooth because it corrects unreliable MVs based on 8×8 and 4×4 block size. On the other hand, the method proposed in [24] only addresses the problem of having fast and complex motion by grouping MBs that have unreliable MVs into a larger block size up to 32×32 to maintain object structures. It may not be able to capture all the detailed motion information inside a larger block.

In this paper, we further propose a hierarchical MV processing method that also exploits both of the residual information and bidirectional prediction difference by analyzing their energy distribution. We first work on correcting unreliable MVs using larger block size to maintain object structures and gradually refine the motion of smaller blocks to capture the detailed motion information. First, based on the received information, we identify MVs that are likely to produce visual artifacts during frame interpolation by exploiting the strong correlation between the reliability of a MV and the residual energy it produces. That is, residual energy of each block is analyzed to determine the reliability of the corresponding received MV. We also consider using chrominance information in MV reliability classification and in all MV processing stages, which is found very useful to identify and correct unreliable MVs and has not explicitly been considered in the literature. Then, before refining those unreliable MVs by further partitioning each block into smaller blocks, we propose to merge MBs that have unreliable MVs by analyzing the distribution of the residual energies. This MB merging process can effectively group MBs located on the motion boundaries. In order to prevent deformed structures, each merged group is assigned a single MV selected from its own and neighboring reliable MVs by minimizing the difference between the forward and backward predictions. This is different from the method in [7], which proposes only merging global motion regions into one larger block but uses smaller blocks in local motion regions. This is also different from the work in [24], since we exploit residual distribution to avoid gathering improper MBs as a merged group.

We further propose an effective MV refinement method that adaptively adjusts unreliable MVs in a smaller block size (of 8×8) by applying a *reliability and similarity constrained vector median filter* to their neighboring MVs. As MVF has been updated and residual information is no longer available, we propose using the bidirectional prediction difference (BPD) to obtain reliability information of each MV as a prior knowledge in the refinement process. Unlike the methods in [11], which uses prediction difference as the weighting factor for each MV in vector median filter, our method simply does not use those neighboring unreliable MVs. In addition, in order not to choose the same unreliable MV, we remove identical and similar MVs in the neighborhood from consideration. To further reduce blockiness effect, we adopt MV smoothing as in [13] on a even finer block size (of 4×4) as the last step in our MV processing method. During frame interpolation, we propose unidirectional frame interpolation for those MBs on the frame boundaries by adaptively selecting forward and backward predictions based on the motion.

Experimental results show that the proposed method significantly improves visual quality, especially in the areas with different motion or the motion boundaries. Our method can successfully maintain object structure and has less blockiness and ghost artifacts. It is also robust even in those video sequences with complex scenes and fast motion. Moreover,

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 3
Amazon.com v. Nokia
IPR2024-00605

low-complexity can be achieved since no motion estimation or object detection are involved at the decoder.

The rest of this paper is organized as follows. We first briefly review MCFI and illustrate the correlation between MV reliability and the energy of residual signals in Section II. In Section III, we describe the algorithms to create a MV reliability map and a MB merging map. The proposed MV processing method is described in details in Section IV. In Section V, we analyze the effect of chosen parameters for the proposed MV processing method. The simulation results and analysis on various compressed video sequences are demonstrated in Section VI. Finally, some conclusions are given in Section VII.

II. Motion-Compensated Frame Interpolation

In MCFI, the skipped frame is often interpolated based on the received MVF between two consecutive reconstructed frames, denoted by f_{t-1} and f_{t+1} respectively. Based on the assumption in which objects move along the motion trajectory, the skipped frame f_t can be interpolated bidirectionally using the following equation:

$$f_t(x,y) = w_f \cdot f_{t-1}\left(x + \frac{1}{2}v_x, y + \frac{1}{2}v_y\right) + w_b \cdot f_{t+1}\left(x - \frac{1}{2}v_x, y - \frac{1}{2}v_y\right) \quad (1)$$

where $\mathbf{v} = (v_x, v_y)$ is the received MVF in the bitstream for reconstructing the frame f_{t+1}. w_f and w_b are the weights for the forward and backward compensations, respectively, and are often set to 0.5. This frame interpolation method is also called the *direct* MCFI as it assumes that the received MVs can represent true motion and can directly be used.

The assumption used by the direct MCFI method does not always hold as the received MVs are often unreliable and do not represent the actual motion. This is because they are usually computed at the encoder by maximizing coding efficiency, instead of finding true motion. In addition, most video coding standards adopt block-based motion estimation, which usually has difficulties to represent finer motion inside a block. In such a case, the video encoder usually chooses the MVs that yield the smallest prediction errors and encodes the residues to compensate for the areas with different motion. Consequently, the estimated MVs are unreliable and unsuitable for the use of frame interpolation. Visual artifacts occur frequently when those unreliable MVs are used in (1). For example, Fig. 1(a) shows the interpolated frame 14 in the Foreman sequence using the direct MCFI method from reconstructed frames 13 and 15. The received MVs in frame 15 are used for interpolation. Artifacts such as blockiness and deformed structures appear in areas where the residual energies are high or in areas where no MV is available. These artifacts can also be observed even when the frame is interpolated by advanced MV processing methods such as in [10] and [12].

In order to illustrate our observation, Fig. 1(b) shows the residual energy of each 16×16 block of frame 15. We analyze where frame interpolation is likely to fail if the received MVs are directly used, and clearly, all the artifacts appear in

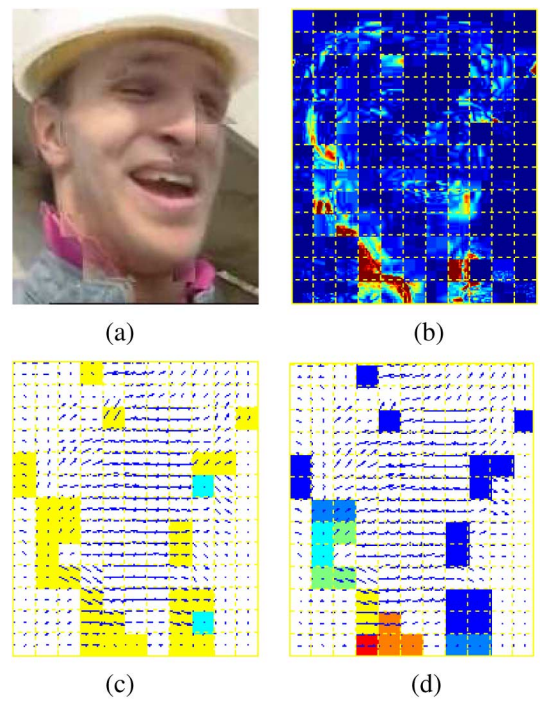

Fig. 1. (a) Interpolation result of the frame 14 of Foreman sequence using direct MCFI from reconstructed frames 13 and 15. (b) Residual energy of the reconstructed frame 15. (c) MV reliability classification map. Unreliable MVs are marked in yellow color and intracoded MBs are marked in cyan color. (d) MB merging map.

the MBs where residual energies are high, such as the boundaries of the face and the ear, and the edges of the collar and the neck. From Fig. 1(a) and (b), it is reasonable to argue that there exists a strong correlation between MV reliability and its associated residual energy. That is, when residual energy is high, it is likely that the corresponding MV is not reliable for frame interpolation.

From Fig. 1(a) and (b), we also observe that the edge and structure of the shirt collar are not maintained. This is because the MVs of those MBs on the edge of the collar are chosen to represent the motion of the cheek and the collar is compensated with higher prediction residues. In most existing methods, the MVs of those blocks are usually re-estimated or corrected separately. There is no guarantee that those blocks that belong to the collar will have the same MV to perfectly assemble the collar. However, if we look closely at how these high residual energies are distributed, we can roughly tell the boundary where the motion starts to differ. Then, for the portions that belong to the collar, we should assign them the same MV so that the structure can first be maintained before any MV refinement. The easiest way to do so is to properly merge those blocks by carefully analyzing the residual energy distribution and its *connectivity* between the MBs, and find a single MV. In such a way, we can avoid having disconnected structures by finding an object motion for a merged group. Please note that this object motion is still block-based in order to describe the global motion in that local neighborhood. It will further be refined as part of our MV processing method to capture the detailed motion inside each merged block.

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 4
Amazon.com v. Nokia
IPR2024-00605

III. Prediction Residual Energy Analysis and Its Application for Frame Interpolation

Based on these observations of received residual, we propose using residual energy to assist MV processing by creating a MV reliability map and a MB merging map. The MV reliability map is to determine the reliability level of each received MVs to make sure that unreliable ones are not used and are corrected. The MB merging map is to indicate whether the neighboring MBs should be grouped together in order to maintain the integrity of the entire moving object. The algorithms to generate the MV reliability map and the MB merging map are described below.

A. Motion Vector Reliability Classification

The first step is to determine if the MVs in the received bitstream are reliable. Let $\mathbf{v}_{m,n}$ denote the MV of each 8×8 block, $\mathbf{b}_{m,n}$. We classify $\mathbf{v}_{m,n}$ into three different reliability levels, *reliable*, *possibly reliable*, and *unreliable*, based on its residual energy, the reliability level of its neighboring blocks and the coding type. The reason why we choose a block size of 8×8 for MV reliability classification is that in MPEG-4 and H.263 coding standards, the prediction residues are generated and encoded based on 8×8 block size. For a MB with only one MV, we simply assign the same MV to all four 8×8 blocks.

For each block $\mathbf{b}_{m,n}$, we first calculate its residual energy, $E_{m,n}$, by taking the sum of the absolute value of each reconstructed prediction error of each pixel. In our algorithm, we consider both luminance and chrominance residues. This is because the motion estimation often uses pixel values in the luminance domain. This may result in an unreliable MV that minimizes the luminance difference but the colors are often mismatched. Therefore, we include chrominance information in residual energy calculation to identify those unreliable MVs. $E_{m,n}$ is computed as follows:

$$E_{m,n} = \sum_{(i,j)\in \mathbf{b}^{Y}_{m,n}} |r_Y(i,j)| + \alpha \cdot \left(\sum_{(i,j)\in \mathbf{b}^{Cb}_{m,n}} |r_{Cb}(i,j)| + \sum_{(i,j)\in \mathbf{b}^{Cr}_{m,n}} |r_{Cr}(i,j)| \right) \quad (2)$$

where $r_Y(i,j), r_{\mathrm{Cb}}(i,j)$, and $r_{\mathrm{Cr}}(i,j)$ are the reconstructed residual signals of Y, Cb and Cr components of the block, $\mathbf{b}_{m,n}$, respectively. α is the weight used to emphasize the degree of color difference. Please note that the residual signals have to be reconstructed during decoding process; therefore, there is no additional computation of using such information other than (2).

We then compare $E_{m,n}$ with a predefined threshold, ε_1, to determine if $\mathbf{v}_{m,n}$ is unreliable. If $E_{m,n}$ is greater than or equal to ε_1, it will be considered as *unreliable* and included into the reliability set L_1. For intracoded MBs, since they do not have MVs, we temporarily assign zero MVs and consider them as unreliable and place them in L_1.

Once an unreliable MV is identified, its neighboring MVs in the same MB and in its eight adjacent MBs will be classified as *possibly reliable* and placed into the second reliability set L_2, even if their residual energy levels are below the threshold. The reason is that when one MB contains at least one block with high residual energy, it is likely that this MB and the surrounding MBs are on the motion boundary. Hence, those MVs may not represent the actual motion, depending on how motion estimation is performed at the encoder. Thus, in order to ensure that all MVs used for frame interpolation are reliable, we mark these MVs as possibly reliable and they will be revisited in a later stage of the MV correction process for further verification. For example, for a MB with four MVs, if only one block exceeds the threshold, the other three blocks as well as all the MVs in the eight adjacent MBs will be considered to be possibly reliable and put into L_2. But if their residual energies are high, they will still be classified into L_1 instead of L_2.

For those MVs that are not classified yet and their $E_{m,n}$ are less than ε_1, they will be classified as *reliable* and placed into the third reliability set L_3. Therefore, we can create a MV reliability map (MVRM) by assigning the reliability level to each MV as follows:

$$\mathrm{MVRM}(m,n) = \begin{cases} L_1, & \text{if } E_{m,n} \geq \varepsilon_1 \\ L_2, & \text{if any MV in the same MB or in the adjacent MBs } \in L_1 \\ L_3, & \text{otherwise.} \end{cases} \quad (3)$$

Fig. 1(c) demonstrates the MV reliability map based on Fig. 1(b) using predefined values, α and ε_1. The selection of ε_1 is discussed in Section V. Since the luminance values of the collar and skin are very similar, some of the wrong MVs can only be detected by chrominance residues instead of luminance residues. The MVs in white color MBs are reliable MVs and those yellow color MBs contain at least one unreliable MV. We purposely mark intracoded MBs in cyan color so that we can differentiate their impact on the interpolation quality from intercoded MBs. However, in our reliability classification, they are considered unreliable as we initially assign zero MVs. As expected, we can successfully identify the regions where frame interpolation is most likely to fail by classifying the MV reliability.

B. Macroblock Merging Based on Motion Vector Reliability

After classifying the reliability of each MV, instead of correcting those unreliable MVs separately, we should consider to merge them by analyzing the connectivity of the residual energies. The merging process is performed on a MB basis, and all MBs that contain unreliable MVs will be examined in a raster scan order. For an intercoded MB that has unreliable MVs, we check if its unreliable MVs connect to other unreliable MVs in the adjacent MBs that have not yet been merged. That is, only those MBs that have unreliable MVs connecting to each other in vertical, horizontal and diagonal directions will be merged. If two adjacent MBs have unreliable MVs that are not next to each other in those three directions, those two MBs will not be merged. If there are no unreliable MVs in the neighborhood, this MB will remain as a single 16×16 block.

All possible shapes after MB merging are shown in Fig. 2. All the MBs that are merged together will be given a single MV in the first stage of the proposed MV processing. We choose 32×32 block size as the maximum block size after merging.

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 5
Amazon.com v. Nokia
IPR2024-00605

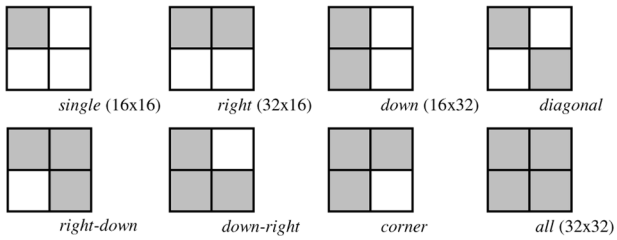

Fig. 2. Merging shapes for intercoded MBs that contain at least one unreliable MV and also for intracoded MBs except for the diagonal shape.

This is sufficient to obtain a good MV to describe the object motion to maintain the edges of an object on those MBs. Further increasing the block size to 48×48 or even larger is found to reduce the quality of interpolated frame as it is too large to represent all the motion inside. Moreover, the proposed MV processing method corrects MVs in a larger shape first and then refines them in smaller blocks later. Increasing the size of the merged block makes the motion refinement process difficult.

It is noted that intracoded MBs are automatically considered in this merging process as their MVs are unreliable. We assume that the intracoded MBs adjacent to unreliable MVs have higher prediction errors so that the encoder decides to encode those MBs using the intracoded mode. In addition, if there are adjacent intracoded MBs, these MBs are assumed to cover the same object in our method. That is, in the merging process, we have four types of MB merging: inter–inter, inter–intra, intra–inter, and intra–intra. However, the diagonal shape in Fig. 2 is not considered for intra–intra MB merging. It is because that the possibility for two diagonal intracoded MBs belonging to the same object is lower. Therefore, there are seven merging modes for intra–intra MB merging type and 8 merging modes for other MB merging types. We can create a MB merging map (MBMM) by assigning a unique number to the MBs that are merged, indicating that they should be considered together to find one MV in the MV processing stage.

Fig. 1(d) shows the MB merging map, where all the MBs in the same merged group is marked in the same color. Different colors are used to differentiate adjacent merged groups. Blue color is the default color if a merged group has no other merged groups next to it. Comparing Fig. 1(b) with Fig. 1(d), we observe that some blocks with high residual energies have been grouped together to form larger blocks such as those in the collar and ear areas. The MB merging map is created based on the predefined shapes of the residual energy distribution, which is very different from the work in [24]. In [24], the merging process simply groups all neighboring MBs as a larger block to find true motion.

IV. Proposed Multistage Motion Vector Processing Method

In this section, we propose a novel MV processing algorithm based on the MV reliability map and the MB merging map described in the previous section. In order to preserve edge information and maintain the integrity of moving object structure, we use MB merging to correct unreliable MVs by finding a best single MV, which is selected based on minimizing the difference of the forward and backward predictions. This new MV for each merged group is considered as a "global" MV in that merged group. However, there may still be smaller areas inside the MBs where this new MV cannot represent their motion well. Since the original residual energy information is no longer useful, we check the difference of forward and backward predictions resulting from the selected MV to reclassify those unreliable MVs. From the second classification, further MV refinement will be applied to those unreliable MVs based on the block size of 8×8. As a larger block size is considered to find a single MV in the first stage, this refinement process can be seen as a local motion adjustment within each merged group. In the final step, we use motion smoothing technique to reduce the blockiness artifact by increasing the number of MVs based on the block size of 4×4. In addition to the MV processing, we also adopt different interpolation strategy for those MBs on the frame boundaries. It is because mismatched bidirectional predictions often happen when moving objects appear along the frame boundary.

The block diagram of our proposed method is illustrated in Fig. 3. According to the MV reliability map (MVRM), the MB merging map (MBMM), and the originally received motion vector field (MVF^0), the first MV processing step is to select the best MV for each merged group, and meanwhile, this MV reliability reclassification can help subsequent MV refinement processing stages to differentiate improper motion. Our method can be considered as a hierarchical approach in the sense that the MVs in each merged group of block size up to 32×32 are first corrected and assigned a single MV, and then these selected MVs as well as other possibly reliable MVs are further refined and smoothed based on the block sizes of 8×8 and 4×4, respectively. In Fig. 3, we also demonstrate how the image quality can gradually be improved after each MV processing stage. In the following sections, we describe the proposed hierarchical, multistage MV processing method in greater details. As to the selection of parameters for corresponding motion vector processing stages is discussed in Section V.

A. Motion Vector Selection

Instead of re-estimating motion for each merged group, we propose MV selection. From the MB merging map, the MBs in each merged group have their own MVs and also the neighboring MVs in the adjacent MBs. These MVs are the candidates for our MV selection process. That is, we choose the best MV, $\mathbf{v}_b^*$, from these candidates by minimizing the averaged absolute bidirectional prediction difference (ABPD) between the forward and backward predictions

$$\mathbf{v}_b^* = \arg\min_{\mathbf{v} \in S}(\mathrm{ABPD}(\mathbf{v})) \tag{4}$$

where

$$\mathrm{ABPD}(\mathbf{v}) = \frac{1}{N_G} \sum_{x,y \in G} \left| f_{t-1}\left(x + \frac{1}{2}v_x, y + \frac{1}{2}v_y\right) - f_{t+1}\left(x - \frac{1}{2}v_x, y - \frac{1}{2}v_y\right) \right|.$$

S denotes the set of the MV candidates. G denotes the merged group in one of the 8 possible shapes in Fig. 2. It is noted that we

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 6
Amazon.com v. Nokia
IPR2024-00605

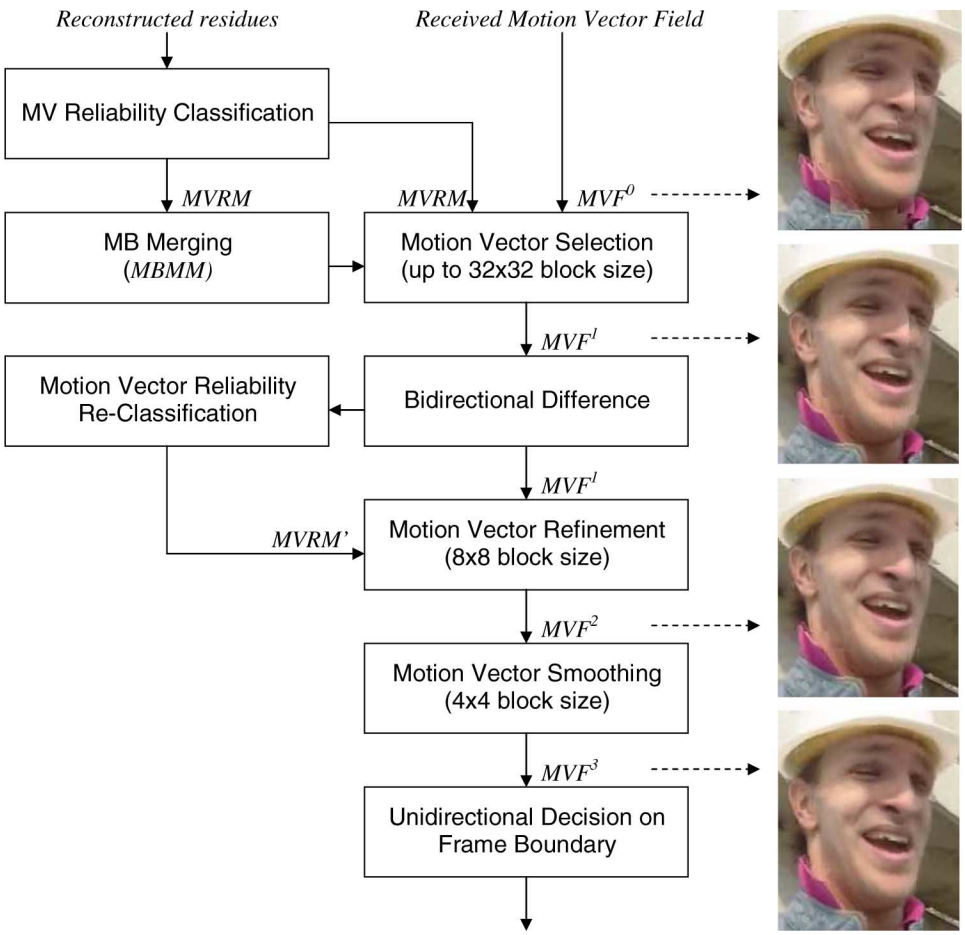

Fig. 3. Block diagram of the proposed algorithms. MVF^k denotes the updated motion vector field after each process.

consider both luminance and chrominance information in BPD calculation in (5) and use the same weighting factor as in (2).

Once the best MV is found, before we assign it to the merged MBs in G, we need to check if this selected MV is good enough by comparing its ABPD with a threshold ε_2. If it is less than ε_2, the MVs of the merged MBs in G will be replaced by the new MV $\mathbf{v}_b^*$ and marked done. However, if it is larger than or equal to ε_2, we drop the selected MV and skip this merged group temporarily to see if some of the neighboring MVs are updated to better ones when other merged groups are corrected. That is, we wait until a proper MV propagates to its neighborhood. If the ABPD of the selected MV is still higher than ε_2 and the neighboring MVs are no longer updated, we still assign the best MV $\mathbf{v}_b^*$ and refine it in the MV refinement stage. The MV selection process stops until all merged groups have been assigned new MVs.

To further illustrate how the proposed MV selection works, an example is shown in Fig. 4. Each MB has four MVs, which are denoted as v_{mn}, where m and n are the row and the column index, respectively. Assume that we have a moving object whose motion boundary is represented by the blue line as shown in Fig. 4(a). The left side of blue line is the moving object, and the opposite side is the background. We observe that since the motion along with the object edge differs, there should exist high residual energies around the blue line. Using our MV classification approach, high residual energy areas are identified and indicated in yellow color. The proposed merging

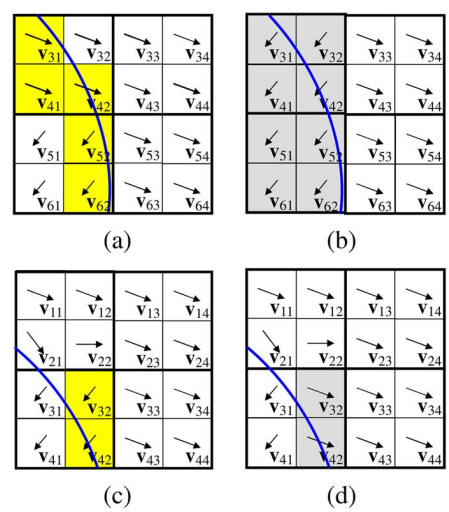

Fig. 4. (a) Motion vector field before the merging process. (b) The motion vector field after the merging process and MV selection. (c) The reclassification map for motion refinement. (d) The motion vector field after motion refinement.

algorithm groups the left two MBs, and a proper motion is assigned using (5). Fig. 4(b) demonstrates the updated MVF and modified MVs are in grey color. We can find the correct motion of the moving object and maintain the integrity of its structure after interpolation. However, as described previously,

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 7
Amazon.com v. Nokia
IPR2024-00605


```
output_MVF = zero array;

DO
{
   Finish_Flag = (output_MVF == input_MVF) ? TRUE : FLASE;
   output_MVF = input_MVF;
   FOR (all MB(i,j))
   {
      IF (MBMM(i,j) != 0)
      {
         Merged_Group(i,j) = MB(i,j) + neighboring MBs with the same index;
         new_MV =
         MV with min(ABPD(adjacent MVs and MVs in Merged_Group(i,j)));
         IF (ABPD(new_MV) < ε2 || Finish_Flag == TRUE)
         {
            output_MVF(Merged_Group(i,j)) = new_MV;
            MBMM(Merged_Group(i,j)) = 0;
            Merged_Blks(i,j) = 8x8 partitions of Merged_Group(i,j);
            MVRM (Merged_Blks(i,j)) = ABPD(new MV) < ε3 ? L3 : L1;
         }
      }
   }
}WHILE (Finish_Flag == FALSE)
```

Fig. 5. Pseudocode of the proposed MV selection.

this updated MV is still block-based and is limited to represent finer local details, such as $\mathbf{v}_{32}$ and $\mathbf{v}_{42}$, where the background motion should be chosen. Therefore, we need to perform MV re-classification described in the next section and motion refinement to further correct these unreliable MVs. The pseudocode of MV selection is shown in Fig. 5. According to MBMM, every merged group with the same nonzero index number will be examined and assigned a best MV. Please note that we will check if the new obtained MVF is identical to the input MVF. If yes, all unreliable MVs that are not corrected yet will be certainly corrected in the next pass.

B. Motion Vector Reclassification Based on Bidirectional Prediction Difference

As a consequence of the MV selection process, a MB in a high residual area will only have one single MV that presents major motion. As the MB consists of multiple motions, regions having different motion can be easily detected by high difference error between forward and backward predictions. Therefore, we propose using BPD to re-classify those unreliable MVs in MVF^1. In addition, we revisit those possibly reliable MVs and check their BPD to see if they are truly reliable. For those MVs that are reliable, they will still remain reliable in this stage. That is, in the MV re-classification, we only work on those MVs that are unreliable (L_1) and possibly reliable (L_2) in the first place.

The MV reclassification process is similar to the classification method described in Section III-A. Since we already use different weights on luminance and chroma while calculating residual energy in the previous classification and MV selection as well, here we simply sum up difference error based on 8×8 block size using same criteria to obtain the new energy distribution $\mathrm{BPD}(m,n)$

$$\mathrm{BPD}_{m,n} = \mathrm{BPD}^{Y}_{m,n} + \alpha \cdot \left(\mathrm{BPD}^{\mathrm{Cb}}_{m,n} + \mathrm{BPD}^{\mathrm{Cr}}_{m,n}\right) \quad (5)$$

where $\mathrm{BPD}^{Y}(m,n), \mathrm{BPD}^{\mathrm{Cb}}(m,n)$, and $\mathrm{BPD}^{\mathrm{Cr}}(m,n)$ are the sums of bidirectional prediction difference for Y, Cb, and Cr components of block $\mathbf{b}_{m,n}$ using the updated MV, $\mathbf{v}^*_{m,n}$, respectively.

If the $\mathrm{BPD}_{m,n}$ is higher than a threshold ε_3, then the MV $\mathbf{v}^*_{m,n}$ will be classified as unreliable and put in L_1. Those MVs with $\mathrm{BPD}_{m,n}$ lower than ε_3 will be classified as reliable in L_3. The classification can be written as

$$\mathrm{MVRM}(m,n) = \begin{cases} L_1, & \text{if } \mathrm{BPD}(m,n) \geq \varepsilon_3 \\ L_3, & \text{if } \mathrm{BPD}(m,n) < \varepsilon_3. \end{cases} \quad (6)$$

Please note that there is no more possibly reliable MV (L_2) in MVRM. All the L_2 MVs will be classified into L_1 or L_3 in this stage. L_3 MVs will remain L_3 in the updated MVRM.

From our observation, the difference between the forward and backward predictions usually has larger scale values than the received prediction error. As a result, we increase the threshold value to find improper MVs that will lead to noticeable artifacts. The updated classification map, MVRM, is the reference map in the MV refinement process. A simplified example is demonstrated in Fig. 4(c), where yellow blocks indicate unreliable MVs after the MV reliability re-classification. The reason why $\mathbf{v}_{31}$ is not classified as unreliable MV is that the high threshold value can only detect significant artifacts. We will leave minor difference error for the last stage of MV processing with a finer block size.

C. Motion Vector Refinement

From the updated classification map, we correct those unreliable MVs in L_1 by using a *reliability and similarity constrained vector median filter* as follows:

$$\mathbf{v}^*_{m,n} = \arg\min_{\mathbf{v} \in S} \sum_{i=m-1}^{m+1} \sum_{j=n-1}^{n+1} w_{i,j} \|\mathbf{v} - \mathbf{v}_{i,j}\|_2 \quad (7)$$

where

$$w_{i,j} = \begin{cases} 0, & \text{if } \mathrm{MVRM}(i,j) = L_1 \\ 1, & \text{if } \mathrm{MVRM}(i,j) = L_3 \text{ and } d_{i,j} > \varepsilon_4. \end{cases}$$

S contains the neighboring MVs of $\mathbf{v}_{m,n}$, and $d_{i,j}$ denotes the distance between $\mathbf{v}_{i,j}$ and $\mathbf{v}_{m,n}$ using the angular difference

$$d_{i,j} = 1 - \frac{\mathbf{v}_{m,n} \cdot \mathbf{v}_{i,j}}{|\mathbf{v}_{m,n}||\mathbf{v}_{i,j}|} = 1 - \cos\theta$$

where θ is the angle between $\mathbf{v}_{i,j}$ and $\mathbf{v}_{m,n}$. The distance is used for measuring the similarity of the candidate MVs and the original MV. Two MVs are considered to be similar if the distance is below a threshold, ε_4. Since we know those 8×8 blocks have different motion or belong to another object, we should prevent getting the same or similar MV. Hence, the vector median filter sorts the candidate MVs that have passed the similarity check and chooses the most probable one. Unlike [23], the filter is adopted to correct those unreliable MVs identified by the BPD classification.

When more than half of MVs of a MB have high difference error energy, motion refinement will not be applied in this case. It is because MV selection is supposed to select a major motion for this current MB, and it seems that the resultant high difference error is not caused by the motion boundary but other issues such as luminance or chrominance changes. In such case, performing motion refinement may break structures that have been well established by MV selection.

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 8
Amazon.com v. Nokia
IPR2024-00605


```
output_MVF = input_MVF;
Finish_Flag = FALSE;

WHILE (Finish_Flag == FALSE)
{
    FOR (all MB(i,j) )
    {
        IF (MBRM(i,j) != 0)
        {
            new_MV = RSCVMF(adjacent MVs);
            IF (ABPD(new_MV) < ABPD(input_MVF(i,j)))
            {
                output_MVF(i,j) = new_MV;
            }
        }
    }
    Finish_Flag = (output_MVF == input_MVF) ?  TRUE : FLASE;
    Input_MVF = output_MVF;
}
```

Fig. 6. Pseudocode of the proposed MV refinement. RSCVMF denotes reliability and similarity constrained vector median filter.

Before we update the MV, we need to perform energy check on BPD of the candidate MV whose error energy must be smaller than the original one. If the candidate MV, $\mathbf{v}^*_{m,n}$, fails to pass energy check, we will not update its unreliability level and try to correct it in the next iteration if a different MV can be found with an updated MVF. Similarly, the refinement MV process stops when the MVF no longer has changes. It is possible that MVs with reliability level L_1 remain unreliable after the refinement. Depending on how structure information distributes on a 8×8 block, the energy check decides if the candidate MV can represent the major motion. If not, we skip this refinement and further modify this unreliable MV with finer block size 4×4 in the MV smoothing process.

We again use Fig. 4 to illustrate MV refinement process. From Fig. 4(c) to (d), we can observe that due to the MV similarity constraint, we prevent using identical and similar MVs, $\{\mathbf{v}_{31}, \mathbf{v}_{41}, \mathbf{v}_{51}, \mathbf{v}_{52}\}$, to correct $\mathbf{v}_{42}$. Instead, this unreliable MV can be effectively corrected using one of $\{\mathbf{v}_{33}, \mathbf{v}_{43}, \mathbf{v}_{53}\}$, which have passed the similarity check. According to the distribution of the motion boundary in this area, these new obtained MVs should have lower difference error since they can better represent the major motion for these 8×8 blocks. As we can see, there still have high bidirectional difference in finer areas in Fig. 4(d) and this will cause blockiness artifact. Therefore, MV smoothing is used in the last stage of the MV processing method to reduce blockiness. For implementation issues, please see the pseudocode in Fig. 6.

D. Motion Vector Smoothing

In order to reduce the blocking artifact, we adopt the method in [13] as our final stage of the MV processing to create a motion field with a finer scale. In [13], each 8×8 block can be further partitioned into four 4×4 sub-blocks and the MVs of these four sub-blocks can be obtained simultaneously by minimizing a smoothness measure Ψ, which is defined as follows:

$$\Psi = \Psi_N + \Psi_S + \Psi_E + \Psi_W + \Psi_D + \Psi_C. \tag{8}$$

The subscripts $\{N, S, E, W, D, C\}$ of Ψ individually represents the smoothness measures between the centered MVs and their adjacent MVs in north, south, east, west, diagonal and center directions. For example, the smoothness measure of these four MVs in their north direction is defined as

$$\Psi_N = \left\|\mathbf{v}^1_{m,n} - \mathbf{v}^3_{m,n-1}\right\|^2 + \left\|\mathbf{v}^2_{m,n} - \mathbf{v}^4_{m,n-1}\right\|^2 + \left\|\mathbf{v}^3_{m,n} - \mathbf{v}^1_{m,n}\right\|^2 + \left\|\mathbf{v}^4_{m,n} - \mathbf{v}^2_{m,n}\right\|^2.$$

In the equation, the MV $\mathbf{v}_{m,n}$ in the block $b_{m,n}$ is partitioned into four sub-blocks $b^i_{m,n}, i = 1, 2, 3, 4$ in scan order, with initial MV $\mathbf{v}^i_{m,n} = \mathbf{v}_{m,n}$. Similarly, we can derive smoothness measures for all other directions. The optimal solution is obtained by combining different direction smoothness measures into a matrix form and minimizing Ψ in (8) with respect to the four MVs.

We only use this resampling approach on MVs with original reliability levels, L_1 and L_2, because they are the major cause of visual artifacts in the frame interpolation. Please note that we use corrected MVs in MVF^3 and produce a denser MVF MVF^4 during the smoothing process while the method in [13] uses the original received MVF MVF^0. As a matter of fact, smoothness measurement aims to lessen the difference among MVs, so using proper MVs can effectively decrease blockiness whereas improper motion can induce serious ghost artifact.

TABLE I
WEIGHT VALUES FOR FORWARD AND BACKWARD MOTION-COMPENSATION ON FRAME BOUNDARY

$n = 1$	w_f	w_b	$m = 1$	w_f	w_b
$v_y \le 0$	0	1	$v_x \le 0$	0	1
$v_y > 0$	1	0	$v_x > 0$	1	0
$n = N$	w_f	w_b	$m = M$	w_f	w_b
$v_y \le 0$	1	0	$v_x \le 0$	1	0
$v_y > 0$	0	1	$v_x > 0$	0	1

E. Motion Adaptive Unidirectional Interpolation on the Frame Boundary

MPEG4 and H.263+ allow motion estimation to search out of frame boundary by extending the boundary pixel values for better coding efficiency. However, for frame interpolation, it is difficult to get good interpolation results by using bidirectional interpolation in (1). For example, for MBs in the first row, if the MV in vertical direction, v_y, is less than zero, it implies that a new object appears in the next frame and the previous frame only has part of the content. Simply averaging the forward and backward predictions will cause visual artifacts. Hence, for those MBs on the frame boundary, we propose using unidirectional interpolation based on the directions of their MVs. That is, we adaptively change the weights of forward and backward predictions based on the MVs. Assume that each frame has N $\times$ M MBs, the weights can be summarized as in Table I.

V. PARAMETER ANALYSIS

In our implementation, MV reliability classification, MV selection, MV reclassification, MV refinement and even the iteration limitation number all make use of predefined threshold values for the proposed MV processing method. In this section, these parameters are analyzed and their impacts on the PSNR performance are also discussed in more details. By doing this,

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 9
Amazon.com v. Nokia
IPR2024-00605

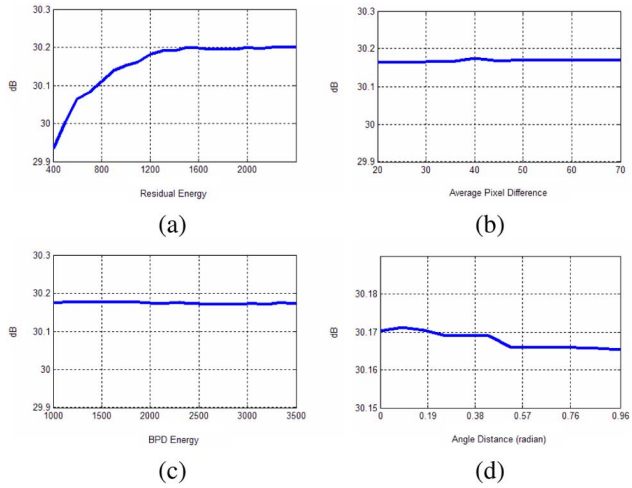

Fig. 7. (a)–(d) PSNR performances with corresponding residual energy threshold, MV selection threshold, MV reclassification threshold, and MV refinement threshold, respectively.

not only can we realize how sensitive the proposed method will be, but also we can select better thresholds and apply same values for all test video sequences so that the implementation can be simplified. The Foreman sequence of CIF size is used here for parameter analysis, which includes $\varepsilon_1, \varepsilon_2, \varepsilon_3$, ε_4, and iteration numbers for MV selection and MV refinement, respectively.

The iteration number is determined by comparing the input MVF and the resulting MVF. If the resulting MVF is static, MV refinement will simply stop MV processing. However, MV selection will continue MV processing once again to ensure that all merged groups have been assigned a single MV when the resulting MVF no longer changes. We compute the iteration number required for each frame and obtain average values, 1.47 and 1.30, for MV selection and MV refinement, respectively. As a result, the iteration varies between 1 and 2. In order to reduce the complexity, we set the iteration number to be 2 for MV selection and increase ε_2 to a very high value in the second iteration so that all merged blocks will certainly be assigned new MVs. With the same manner, we also set an iteration number to be 2 for the motion refinement process.

The overall PSNR performance with different corresponding threshold values is illustrated in Fig. 7. In Fig. 7(a), as the residual energy threshold increases, the PSNR performance increases as well. It is because for low ε_1, many correct MVs are identified as unreliable and are merged with improper neighboring MVs for MV selection. In such case, the MVF will be rearranged badly that makes it difficult to correct unreliable MVs in the MV refinement and MV smoothing steps. Therefore, the chosen parameter of the residual classification should be low enough to identify as many possibly unreliable MVs as possible, but at the same time, it should be high enough for not merging too many improper MBs as a merged group. Note that we do not show the simulation results of either extremely low or extremely high threshold values in Fig. 7(a). It is because extremely low value case means that the received MVF will be modified thoroughly by MV selection using 32×32 block size and extremely high value case means direct MCFI where the received MVF is used directly. Obviously, both scenarios cannot yield good interpolation results.

For ε_2 in Fig. 7(b), we observe that the chosen parameter does not affect the PSNR performance considerably. That is, as the resulting MVF becomes static, the MV with minimum ABPD will still be selected even its ABPD is greater than ε_2. Similarly, if ε_2 is too large, the MVs with minimum ABPD are also assigned to all merged blocks. The reason why we place a threshold value for MV selection is to check if there will be new MVs propagating to the neighborhood. In Fig. 7(c), it also shows that the scale of ε_3 does not have too much influence on MV reclassification since the energy check of MV refinement can avoid choosing unsuitable MVs. However, this threshold value has to effectively detect all significant visible artifacts introduced by MV selection and other possible unreliable MVs that are not identified yet during the residual classification. As to the angle threshold for the MV refinement, ε_4, we observe that the PSNR performance degrades gradually in Fig. 7(d) when the angle difference increases. The reason is that the MV choices for MV refinement become less. We also notice that zero angle difference does not obtain the best average PSNR. That is, similarity check can help us avoid getting same MVs to correct unreliable MVs especially on motion boundaries.

To get the best MCFI performance, these threshold values should be adjustable to different video sequences. However, based on the analysis of these parameters, we observe that the proposed MV processing method is only sensitive to ε_1 and ε_4. In order to reduce the complexity, we use same parameters to all test video clips in the simulation and set $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$ to be 1100, 45, 2000, and 0.15, respectively. These values are not chosen to optimize MCFI results of the Foreman sequence but to have the ability to process general video cases. Therefore, we select the values around saturated points in Fig. 7(a) and (d) for ε_1 and ε_4 and averaged values in Fig. 7(b) and (c) for ε_2 and ε_3.

VI. Simulations

In this section, we present simulation results to evaluate the performance of the proposed method. We compare our method with direct interpolation, vector median filter, adaptive vector median filter [11], MV smoothing [13], multisize block matching algorithm [7], and proposed MV selection from the neighboring MVs by minimizing the bidirectional prediction difference. Eight video sequences, Foreman, Formula1, Walk, Bus, Fast Food, Stephan, Rugby, and Football, of CIF frame resolution are used with original frame rate of 30 frame per second (fps). They are all encoded using H.263, where even frames are skipped to generate video bitstreams of 15 fps. The skipped frames are interpolated at the decoder and they are used to evaluate different interpolation schemes. The rate control function is disabled by fixing quantization parameter (QP) values. The averaged bit rates of all test sequences are 395.77, 474.50, 430.39, 509.43, 499.36, 503.10, 340.90, and 429.32 Kbps for Foreman, Formula1, Walk, Bus, Fast Food, Stephan, Rugby, and Football, respectively.

The visual comparisons are presented in Figs. 8–13. Figs. 8 and 9 show the visual comparisons for Foreman sequence. In Fig. 8, blockiness can easily be seen in Fig. 8(b) and (d)–(g). The blockiness artifacts can be removed by MV smoothing method

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 10
Amazon.com v. Nokia
IPR2024-00605

Fig. 8. Interpolated results of frame 186 of Foreman sequence using (a) original frame; (b) direct interpolation (PSNR: 24.43 dB, SSIM: 0.7259); (c) MV smoothing (PSNR: 24.59 dB, SSIM: 0.7233); (d) vector median filtering (PSNR: 24.72 dB, SSIM: 0.7431); (e) adaptive vector median filter (PSNR: 22.20 dB, SSIM: 0.5513); (f) multisize block matching algorithm (PSNR: 22.44 dB, SSIM: 0.5646); (g) proposed MV selection (PSNR: 25.23 dB, SSIM: 0.7811); and (h) the proposed multistage MV processing method (PSNR: 24.85 dB, SSIM: 0.8270), respectively.

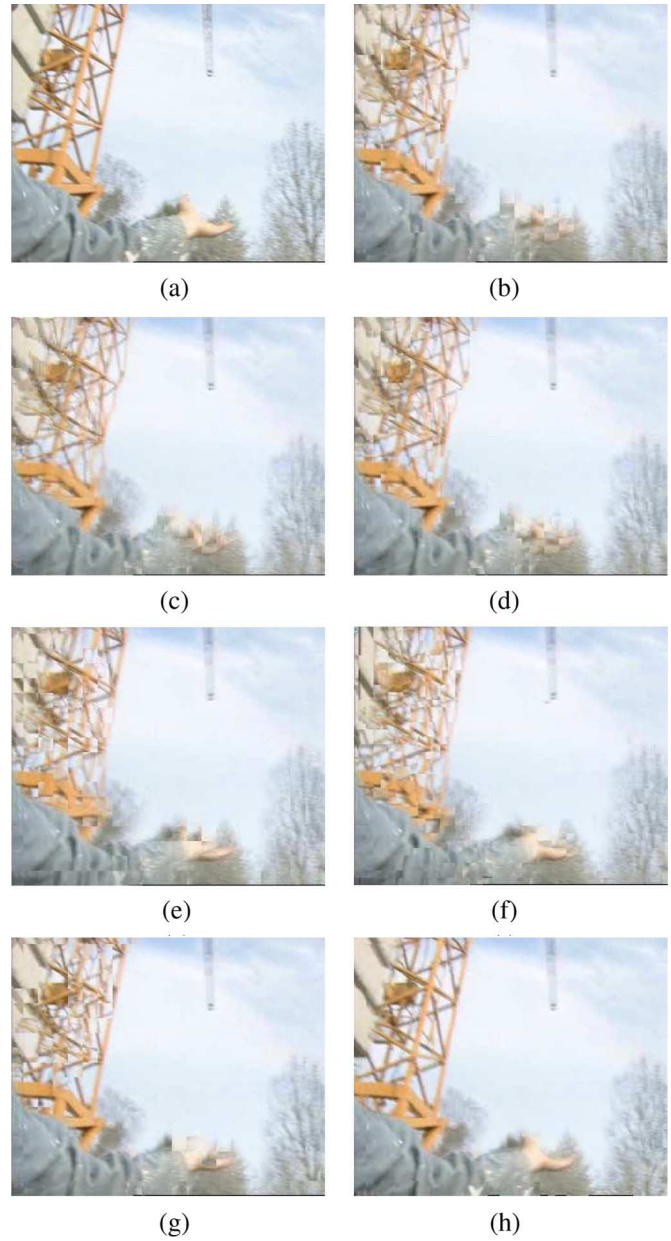

Fig. 9. Interpolated results of frame 196 of FOREMAN sequence using (a) original frame; (b) direct interpolation (PSNR: 24.43 dB, SSIM: 0.7353); (c) MV smoothing (PSNR: 24.42 dB, SSIM: 0.7328);, (d) vector median filtering (PSNR: 24.44 dB, SSIM: 0.7363);, (e) adaptive vector median filter (PSNR: 24.90 dB, SSIM: 0.7389); (f) multisize block matching algorithm (PSNR: 24.76 dB, SSIM: 0.7269); (g) proposed MV selection (PSNR: 25.70 dB, SSIM: 0.8316); and (h) the proposed multistage MV processing method (PSNR: 26.61 dB, SSIM: 0.9278), respectively.

as shown in Fig. 8(c). However, ghost artifacts are then generated because incorrect MVs have impacts on the smoothing process such as the areas around the face. Also, the structure of the building and the tower cannot be maintained. In Fig. 8(e) and (f), the performance of motion estimation is greatly degraded due to image distortion. MV selection can recover the contour of the face and some of edge information as shown in Fig. 8(g). However, it is more likely to fail when a frame has repeated structures or fairly smooth areas around the MBs that have edge information. This is because minimizing the bidirectional prediction difference may choose a MV that points to the smooth area, which will cause a broken edge as shown on the building in Fig. 8(g).

Fig. 9 shows another example of Foreman sequence. In Fig. 9(g), we can clearly see that the hand is deformed and replaced by the sky because the sky is a smoother area around the hand. Moreover, the structure of the yellow tower is destroyed since each MB searches its own motion without considering object structures. As to the results using the proposed method in Figs. 8(h) and 9(h), these artifacts have successfully been eliminated. The ghost effect in Fig. 8(c) and Fig. 9(c) does not appear in our results since our proposed method corrects those

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 11
Amazon.com v. Nokia
IPR2024-00605

unreliable MVs before smoothing. Comparing to MV selection, our results can better preserve edge information such as the yellow tower by taking object structures into consideration during the MB merging process. Moreover, due to the unidirectional interpolation scheme, we also perform better than other six methods on the frame boundary. It is because those MBs on the frame boundaries have motion moving to the right direction, and part of the content disappears in the subsequent frame. As we use unidirectional MCFI on the frame boundary, the content mismatched effect can be removed.

As one might notice, our PSNR value is lower than that of MV selection in Fig. 8, even though our method has better visual quality. This is because PSNR measures signal fidelity to its original frame, instead of perceived visual quality. Any pixel shifting may cause significant PSNR degradation but we may not see the difference. In addition, our proposed method attempts to refine motion by maintaining structure integrity in order to provide better visual experience, instead of recovering the original pixel values. Afterall, the users do not know the original frames that are skipped at the decoder as long as the motion is smooth and the images have no visible artifacts. Therefore, we adopt an alternative objective measurement, structure similarity (SSIM) index, for quality assessment [25], which has been used in [5]. It examines degradation of structure information and is less sensitive to the pixel shift effect. In this way, the quality numeric analysis can truly reflect structure integrity in interpolated frames. As we can see in Fig. 8, although our PSNR is lower than MV selection, our performance is better in terms of SSIM index. In Fig. 9, our performance is better in terms of both PSNR and SSIM.

It is noted that SSIM values are derived by averaging all index values within a frame, and, therefore, the difference between our method and MV selection may be seen smaller. That is, if we divide a frame into 11×9 units of size 32×32 for CIF video sequences, the obtained index value is an average of 99 local index values. Hence, the averaging process lowers the influence of local visual artifacts. Even though the difference of SSIM indexes is not much due to the averaging process, our SSIM still performs better than others.

Figs. 10 and 11 demonstrate two examples in Walk sequence. In Fig. 10, because the background has complex patterns, which is a challenge to find accurate MVs, the four methods that we compare with fail and all have deformed structures severely as shown in Fig. 10(b)–(g). Our method, however, can recover the background without any artifact even through the frame has many intracoded MBs and high-residual energy intercoded MBs. This proves that our MV reliability classification and MB merging can truly help MV processing to obtain a better motion as well as to maintain complex object structures. Our method also outperforms in terms of PSNR and SSIM. Our PSNR value is at least 1.39 dB higher and our SSIM value is at least 0.1299 higher than the other methods.

Another interpolation results of a different frame number are demonstrated in Fig. 11. As we can see, the MV selection method can recover many areas except the face where eyes and the nose have been replaced by smooth skin texture, not to mention that direct interpolation, vector median filtering, MV smoothing, adaptive vector median filter, and multisize block matching algorithm all result in many visual artifacts in both the face and body areas. The proposed method provides much better quality than these four methods by eliminating most of the artifacts as illustrated in Fig. 11(h).

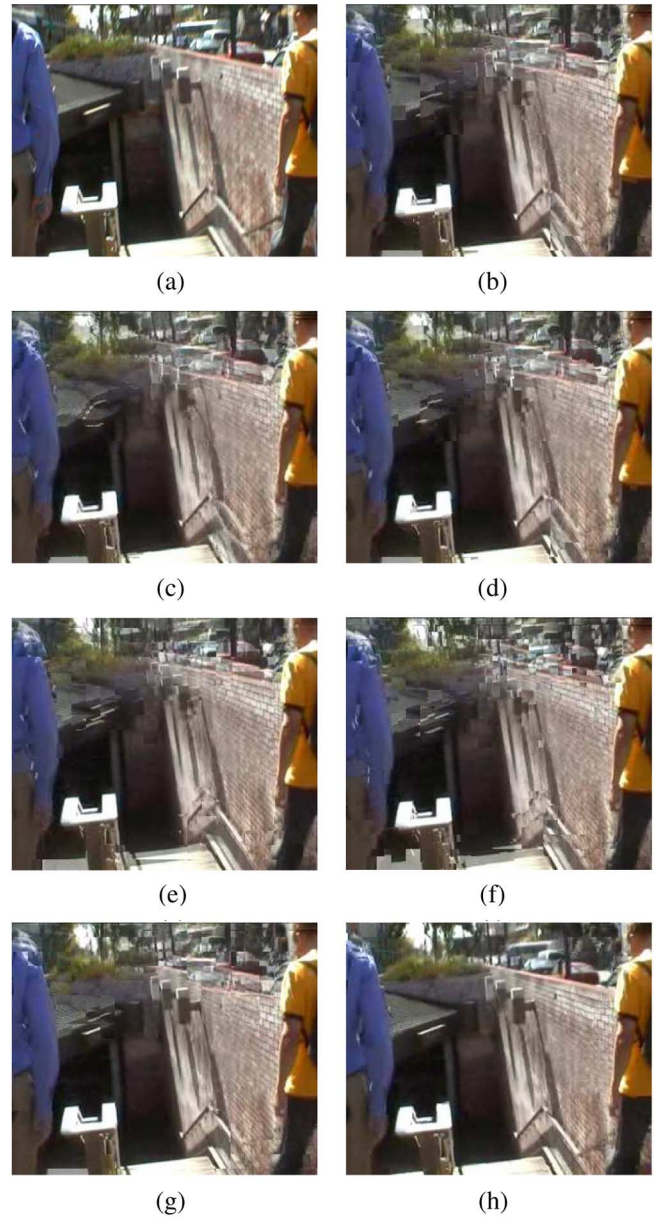

Fig. 10. Interpolated results of frame 18 of WALK sequence using (a) original frame; (b) direct interpolation (PSNR: 19.66 dB, SSIM: 0.6278); (c) MV smoothing (PSNR: 19.76 dB, SSIM: 0.6323); (d) vector median filtering (PSNR: 19.67 dB, SSIM: 0.6280); (e) adaptive vector median filter (PSNR: 19.26 dB, SSIM: 0.5934); (f) multisize block matching algorithm(PSNR: 19.34 dB, SSIM: 0.6234); (g) proposed MV selection (PSNR: 19.94 dB, SSIM: 0.6786); and (h) the proposed multistage MV processing method (PSNR: 21.33 dB, SSIM: 0.8085), respectively.

The interpolation results of the Formula1 sequence are shown in Fig. 12. Fast motion is involved as the camera tries to catch up with the race car and also the intensity of luminance between grass and pavement is very similar. These factors account for the failed interpolation on the white lines in the background as shown in Fig. 12(b)–(g). Bidirectional difference of color components and the MB merging algorithm assist us to find more

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 12
Amazon.com v. Nokia
IPR2024-00605

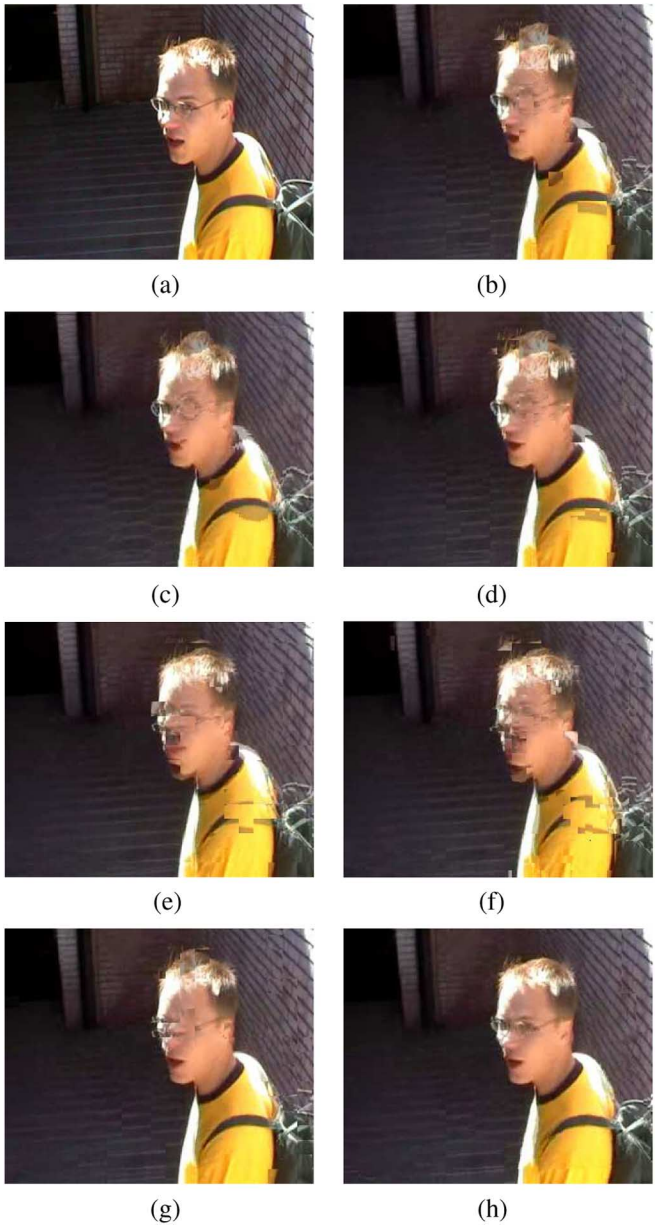

Fig. 11. Interpolated results of frame 288 of WALK sequence using (a) original frame; (b) direct interpolation (PSNR: 21.92 dB, SSIM: 0.7183); (c) MV smoothing (PSNR: 22.35 dB, SSIM: 0.7407); (d) vector median filtering (PSNR: 22.04 dB, SSIM: 0.7224); (e) adaptive vector median filter (PSNR: 21.37 dB, SSIM: 0.6874); (f) multisize block matching algorithm (PSNR: 21.58 dB, SSIM: 0.7024); (g) proposed MV selection (PSNR: 22.22 dB, SSIM: 0.7388); and (h) the proposed multistage MV processing method (PSNR: 22.39 dB, SSIM: 0.7606), respectively.

Fig. 12. Interpolated results of frame 56 of the Formula1 sequence using (a) original frame; (b) direct interpolation (PSNR: 29.68 dB, SSIM: 0.9050); (c) MV smoothing (PSNR: 29.46 dB, SSIM: 0.9013); (d) vector median filtering (PSNR: 29.80 dB, SSIM: 0.9067); (e) adaptive vector median filter (PSNR: 29.91 dB, SSIM: 0.9155); (f) multisize block matching algorithm (PSNR: 29.85 dB, SSIM: 0.9149); (g) proposed MV selection (PSNR: 30.23 dB, SSIM: 0.9225); and (h) the proposed multistage MV processing method (PSNR: 31.56 dB, SSIM: 0.9567), respectively.

suitable motion to represent this area so that our white lines are consistent, as illustrated in Fig. 12(h).

In order to prove the feasibility of our proposed method, we also apply our algorithm to video sequences with a larger frame size, 720×480, as shown in Fig. 13. This sequence is encoded with the bit rate of 1.7 Mbps. Obviously, our method still significantly outperforms other four motion vector processing methods. PSNR is 4.57 dB higher and SSIM is 0.1 higher.

We list averaged PSNR and SSIM values for these eight video sequences in Tables II and III. As observed, our SSIM performance is consistently better than others motion vector processing methods but PSNR performance is only better in Walk, Bus, and Fast Food. PSNR is derived from the quantity of fidelity difference while SSIM emphasizes on the similarity of object structures. According to their properties, SSIM is adequate to MCFI case since avoiding deformed structures and achieving video consistency are the most important requirement in MCFI application. A counter example is demonstrated in Fig. 8(h) where our result has better visual quality but PSNR is lower than Fig. 8(g). However, its SSIM index can more effectively reflect the subjective ratings. For the Bus sequence, we perform

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 13
Amazon.com v. Nokia
IPR2024-00605

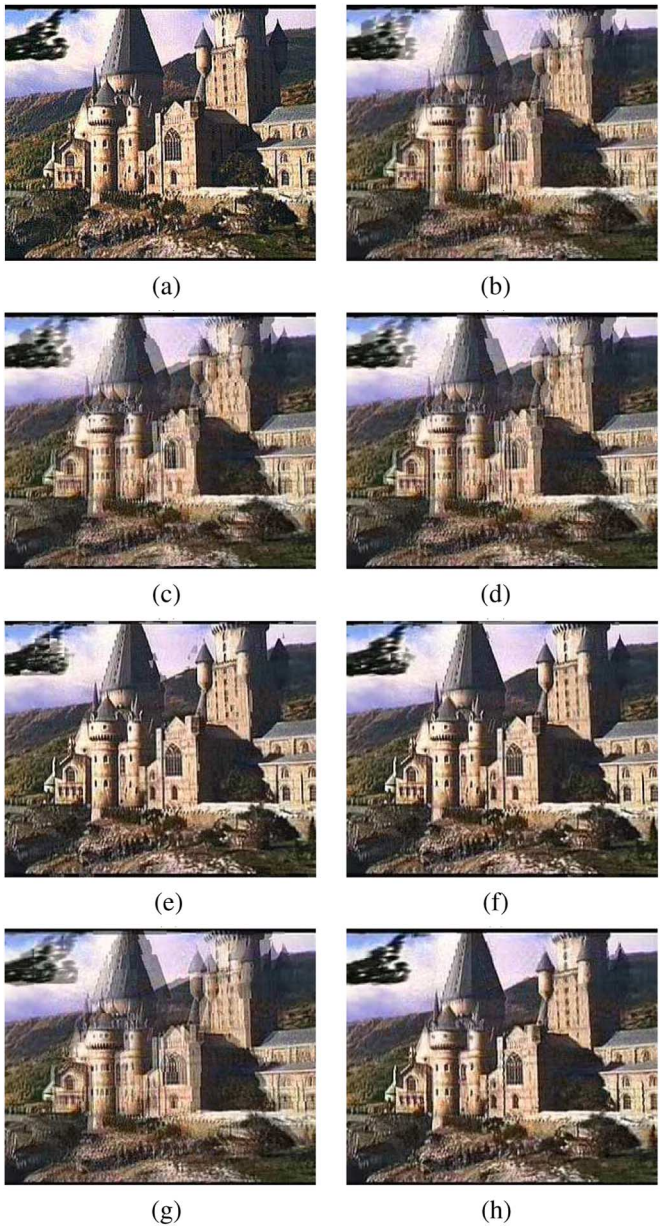

Fig. 13. Interpolated results of frame 12 of CASTLE AND TREE sequence using (a) original frame; (b) direct interpolation (PSNR: 21.43 dB, SSIM: 0.8289); (c) MV smoothing (PSNR: 21.80 dB, SSIM: 0.8415); (d) vector median filtering (PSNR: 21.57 dB, SSIM: 0.8342); (e) adaptive vector median filter (PSNR: 24.38 dB, SSIM: 0.9189); (f) multisize block matching algorithm (PSNR: 25.62 dB, SSIM: 0.9322); (g) proposed MV selection (PSNR: 21.89 dB, SSIM: 0.8452); and (h) the proposed multistage MV processing method (PSNR: 26.46 dB, SSIM: 0.9422), respectively.

slightly worse than adaptive vector median filter and multisize block matching algorithm. This is because the quality of the re-estimated MVF is much better than the received MVF. Comparing to Direct MCFI, the proposed motion vector processing approach has 1.24 dB improvement in average PSNR.

For further comparisons, we plot PSNR and SSIM values of interpolated frames using vector median filtering, MV selection, and the proposed method for the entire Bus sequence, as illustrated in Fig. 14. The overall performance of the proposed method is generally the best. Please note that in some difficult frames, where a lot of MVs are unreliable and there are many

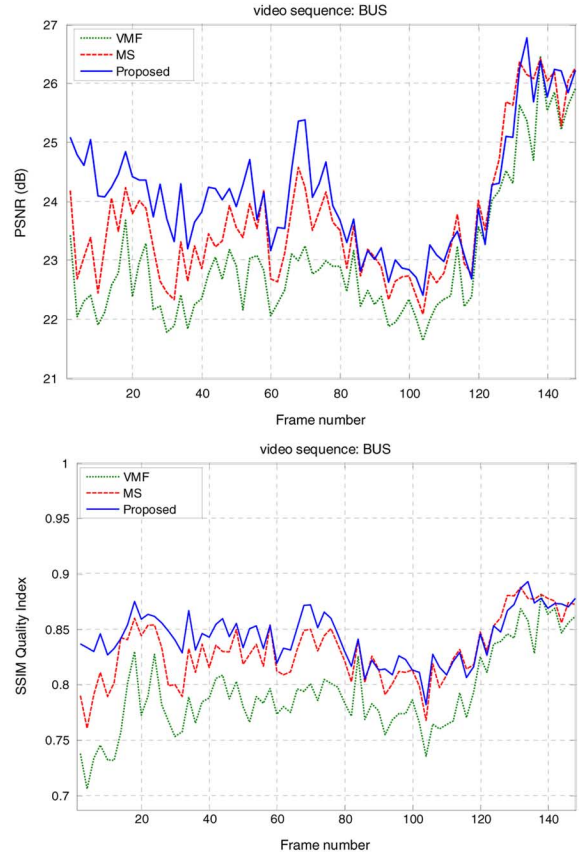

Fig. 14. PSNR and SSIM plots for Bus sequence. The green line denotes vector median filter, the red line denotes proposed MV selection with fixed searching size, and the blue line denotes the proposed multistage MV Processing.

intracoded MBs, the proposed method always has less structure degradation. It also seems that our method can keep the performance at a higher level so that the visual quality can be maintained relatively constant during video playback. More results can be found at http://videoprocessing.ucsd.edu/~aihuang/Paper1.htm.

VII. Conclusion

Frame interpolation that uses motion information in the received bitstream is an efficient and effective technique for compressed video to improve temporal quality by increasing frame rate at the decoder. However, not all received motion information is suitable for frame interpolation as block-based motion estimation at the encoder often fails to find true motion. In this paper, we have shown that those unreliable MVs can be identified by their prediction residual energies. We further use this MV reliability information and presented a hierarchical MV processing algorithm to produce a more reliable MVF for frame interpolation to remove artifacts. The proposed method classifies the MVs into different reliability levels and also analyzes the distribution of high residual energies to effectively merge MBs that are on the motion boundaries. This can avoid using complicated object-based segmentation or edge detection to maintain edge information.

We then propose refining unreliable MVs hierarchically by gradually reducing block size. In addition, for those MBs on

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 14
Amazon.com v. Nokia
IPR2024-00605

TABLE II
PSNR PERFORMANCE COMPARISONS AMONG FIVE FRAME INTERPOLATION METHODS AND THE PROPOSED METHOD FOR EIGHT VIDEO SEQUENCES

Sequences	Direct	VMF	AVMF	Smooth	MS BMA	Selection	Multi-Stage
WALK	22.88	22.95	22.90	22.99	23.00	23.04	**23.11**
FORMULA 1	28.09	28.24	28.17	27.85	28.34	**28.54**	28.36
BUS	22.92	23.03	24.69	22.64	**24.77**	23.72	24.16
FAST FOOD	25.67	25.84	25.10	25.69	25.41	26.31	**26.66**
FOREMAN	31.43	31.48	31.11	31.19	31.12	**31.75**	30.16
STEPHAN	23.80	23.87	23.73	23.68	23.77	**24.09**	23.93
FOOTBALL	24.88	24.95	22.97	24.99	23.05	**25.25**	25.21
RUGBY	24.58	24.75	24.28	24.76	24.45	**24.97**	24.81

TABLE III
SSIM PERFORMANCE COMPARISONS AMONG FIVE FRAME INTERPOLATION METHODS AND THE PROPOSED METHOD FOR EIGHT VIDEO SEQUENCES

Sequences	Direct	VMF	AVMF	Smooth	MS BMA	Selection	Multi-Stage
WALK	0.7869	0.7889	0.7862	0.7908	0.7911	0.7958	**0.8025**
FORMULA 1	0.8276	0.8305	0.8310	0.8251	0.8363	0.8388	**0.8432**
BUS	0.7893	0.7924	0.8651	0.7755	**0.8690**	0.8307	0.8438
FAST FOOD	0.8270	0.8304	0.7933	0.8275	0.8045	0.8566	**0.8819**
FOREMAN	0.9311	0.9319	0.9210	0.9295	0.9237	0.9406	**0.9467**
STEPHAN	0.7814	0.7818	0.7744	0.7743	0.7771	0.7993	**0.8116**
FOOTBALL	0.6752	0.6775	0.6580	0.6793	0.6634	0.7012	**0.7151**
RUGBY	0.7335	0.7397	0.7173	0.7416	0.7265	0.7585	**0.7694**

the frame boundaries, we should consider their motion and use unidirectional interpolation as the movement may go outside the frame. Besides, chrominance information should be consider explicitly as it provides valuable information to identify and correct unreliable MVs. Afterall, SSIM indexes and visual comparisons show that our MV processing algorithm can work with complicated MVF with complex texture, and more importantly, structure information is better preserved. The proposed method is very robust in eliminating artifacts for various video sequences. Moreover, our method is a low-complexity, standard compliant solution at the decoder since we only perform MV processing to accomplish the concept similar to object-based frame interpolation but without actual edge detection or motion estimation. The resulting simulation confirms the advantages of the proposed methods comparing to other conventional approaches. In the future works, we will apply the proposed MV processing method to H.264 decoder for MCFI and transcoding applications.

REFERENCES

[1] G. Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True-motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[2] R. Krishnamurthy, J. W. Woods, and P. Moulin, "Frame interpolation and bidirectional prediction of video using compactly encoded optical-flow fields and label fields," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 5, pp. 713–726, Aug. 1999.

[3] T. Chen, "Adaptive temporal interpolation using bidirectional motion estimation and compensation," in *Proc. Int. Conf. Image Processing*, Sep. 2002, vol. 2, pp. 313–317.

[4] T. Ha, S. Lee, and J. Kim, "Motion compnesated frame interpoaltion by new block-based motion estimation algorithm," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 752–759, May 2004.

[5] J. Wang, N. Patel, and W. Grosky, "A fast block-based motion compensated video frame interpolation approach," in *Proc. Asilomar Conf. Signals, Systems, Computers*, Nov. 2004, pp. 1740–1743.

[6] J. Zhai, J. L. K. Yu, and S. Li, "A low complexity motion compensated frame interpolation method," in *Proc. ISCAS*, Sep. 2005, vol. 2, pp. 4927–4930.

[7] S. Fujiwara and A. Taguchi, "Motion-compensated frame rate up-conversion based on block matching algorithm with multi-size blocks," in *Proc. Int. Symp. Intelligent Signal Processing and Communication Systems*, Dec. 2005, pp. 353–356.

[8] H. Blume, G. Herczeg, O. Erdler, and T. G. Noll, "Object based refinement of motion vector field applying probabilistic homogenization rules," *IEEE Trans. Consum. Electron.*, vol. 48, no. 3, pp. 694–701, Aug. 2002.

[9] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko, "Frame rate up-conversion using perspective transform," *IEEE Trans. Consum. Electron.*, vol. 52, no. 3, pp. 975–982, Aug. 2006.

[10] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," *Proc. IEEE*, vol. 78, pp. 678–689, Apr. 1990.

[11] L. Alparone, M. Barni, F. Bartolini, and V. Cappellini, "Adaptively weighted vector-median filters for motion-fields smoothing," in *Proc. ICASSP*, May 1996, vol. 4, pp. 2267–2270.

[12] G. Dane and T. Q. Nguyen, "Motion vector processing for frame rate up conversion," in *Proc. ICASSP*, May 2004, vol. 3, pp. 309–312.

[13] G. Dane and T. Q. Nguyen, "Smooth motion vector resampling for standard compatible video post-processing," presented at the Asilomar Conf. Signals, Systems, Computers, 2004.

[14] H. Sasai, S. Kondo, and S. Kadono, "Frame-rate up-conversion using reliable analysis of transmitted motion information," in *Proc. ICASSP*, May 2004, vol. 5, pp. 257–260.

[15] S. Sekiguchi, Y. Idehara, K. Sugimoto, and K. Asai, "A low-cost video frame-rate up conversion using compressed-domain information," in *Proc. Int. Conf. Image Processing*, Sep. 2005, vol. 2, pp. 974–977.

[16] J. Zhang, L. Sun, S. Yang, and Y. Zhong, "Position prediction motion-compensated interpoaltion for frame rate up-conversion using temporal modeling," presented at the Int. Conf. Image Processing, 2005.

[17] T. Shanableh and M. Ghanbari, "Loss concealment using b-pictures motion information," *IEEE Trans. Multimedia*, vol. 5, no. 2, pp. 257–266, Jun. 2003.

[18] C. C. A. Vetro and H. Sun, "Video transcoding architectures and techniques: An overview," *IEEE Signal Process. Mag.*, pp. 18–29, Mar. 2003.

[19] O. A. Ojo and G. Haan, "Robust motion-compensated video upconversion," *IEEE Trans. Consum. Electron.*, vol. 43, no. 4, pp. 1045–1056, Nov. 1997.

[20] G. Dane, K. El-Maleh, and Y.-C. Lee, "Encoder-assisted adaptive video frame interpolation," in *Proc. ICASSP*, 2005, vol. 2, pp. 349–352.

[21] S.-H. Lee, O. Kwon, and R.-H. Park, "Weighted-adaptive motion-compensated frame rate up-conversion," *IEEE Trans. Consum. Electron.*, vol. 49, no. 3, pp. 485–492, Aug. 2003.

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.

Nokia Exhibit 2008, p. 15
Amazon.com v. Nokia
IPR2024-00605

[22] G. Dane and T. Q. Nguyen, "Optimal temporal interpolation filter for motion-compensated frame rate up conversion," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 978–991, Apr. 2006.

[23] A. Huang and T. Nguyen, "A novel motion compensated frame interpolation based on block merging and residual energy," in *Proc. Multimedia Signal Processing Workshop*, Sep. 2006, vol. 4, pp. 353–356.

[24] A. Huang and T. Nguyen, "Motion vector processing based on residual energy information for motion compensated frame interpolation," in *Proc. Int. Conf. Image Processing*, Sep. 2006, vol. 4, pp. 353–356.

[25] Z. Wang, A. C. Conrad, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structure similarity," in *Proc. Multimedia Signal Processing Workshop*, Apr. 2004, vol. 13, no. 4, pp. 600–612.

Ai-Mei Huang (S'06) received the B.S. degree in communications engineering from the National Chiao Tung University, Hsien Chu, Taiwan, R.O.C., in 1999, and the M.S. degree in electrical engineering from National Taiwan University, Taipei, in 2001. She is currently pursuing the Ph.D. degree at the University of California at San Diego, La Jolla.

Her research interests include signal/image processing, video coding techniques, and video applications in broadcasting and TV display.

Truong Q. Nguyen (F'06) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the California Institute of Technology, Pasadena, in 1985, 1986, and 1989, respectively.

He was with the Massachusetts Institute of Technology (MIT) Lincoln Laboratory, Cambridge, from June 1989 to July 1994, as a member of technical staff. During the academic year 1993–1994, he was a Visiting Lecturer at MIT and an Adjunct Professor at Northeastern University, Boston, MA. From August 1994 to July 1998, he was with the the Electrical and Computer Engineering Department, University of Wisconsin, Madison. He was with Boston University from August 1996 to June 2001. He is currently a Professor at the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla. He is the coauthor (with Prof. G. Strang) of a popular textbook, *Wavelets & Filter Banks* (Wellesley-Cambridge Press, 1997) and the author of several Matlab-based toolboxes on image compression, electrocardiogram compression, and filter bank design. He has authored over 200 publications. His research interests are video processing algorithms and their efficient implementation.

Prof. Nguyen received the IEEE Transaction in Signal Processing Paper Award (Image and Multidimensional Processing area) for the paper he co-wrote with Prof. P. P. Vaidyanathan on linear-phase perfect-reconstruction filter banks (1992). He received the National Science Foundation Career Award in 1995 and is currently the Series Editor (Digital Signal Processing) for Academic Press. He served as Associate Editor for the IEEE TRANSACTION ON SIGNAL PROCESSING from 1994–1996, the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING from 1996–1997, the IEEE SIGNAL PROCESSING LETTERS from 2001–2003, and the IEEE TRANSACTION ON IMAGE PROCESSING from 2001–2004.

Authorized licensed use limited to: Emily Tannenbaum. Downloaded on August 11,2024 at 18:08:18 UTC from IEEE Xplore. Restrictions apply.