MOTION COMPENSATED TEMPORAL INTERPOLATION WITH OVERLAPPING

Chi-Kong Wong, Oscar C. Au* and Chi-Wah Tang**
Department of Electrical and Electronic Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong.
Email: eea@ee.ust.hk*, eetommy@ee.ust.hk**

Abstract
Common techniques such as frame repetition or linear interpolation for reconstructing skipped frames in temporally subsampled video sequence tend to introduce undesirable artifacts. A previously proposed technique, motion compensated temporal interpolation (MCTI), can interpolate video frames in time domain with good interpolated image quality. However, the reconstructed frames tend to be blocky. In this paper, we propose to incorporate an overlapping technique to solve the problem. The resulting technique, overlapped motion compensated temporal interpolation (OMCTI), was found in simulation to give significantly reduced blocking artifacts.

I. INTRODUCTION
A postprocessing technique called motion compensated temporal interpolation (MCTI) was proposed in [1] to generate the skipped frames from a temporally subsampled videoconferencing sequence. MCTI can potentially be combined with other compression schemes such as ITU-T H.261 or H.263 to achieve very large compression ratio so as to satisfy the very-low-bit-rate (VLBR) requirement of the telephone channels. Due to the block-based processing, the interpolated frames of MCTI tends to be blocky. In this paper, we propose a method called overlapped motion compensated temporal interpolation (OMCTI) which reduces the blocking artifacts of the interpolated frames estimated by MCTI. The overlapping idea comes mainly from H.263.

II. MOTION COMPENSATED TEMPORAL INTERPOLATION ALGORITHMS
2.1. Review of MCTI [1]
For any k, our goal is to generate a frame to be inserted between the (k-1)th and kth received frames so that object motions would be smoother. We divide the (k-1)th frame, the kth frame and the inserted frame into blocks of size $N \times N$ such that a grid of blocks is formed in each frame. Firstly, we perform forward motion estimation. For a block $B_1$ located at $(x, y)$ in the $k$th frame, we define a search area of size $(2W + 1) \times (2W + 1)$ in the (k-1)th frame and perform an exhaustive motion search using mean absolute difference (MAD) as the distortion measure. If the best match is the block $B_2$ located at $(x + dx, y + dy)$ in the (k-1)th frame, then $B_1$ or $B_2$ should also appear at $(x + dx/2, y + dy/2)$ in the inserted frame due to the assumed linear translational motion. However, the $N \times N$ blocks $B_3$ at $(x + dx/2, y + dy/2)$ in the inserted frame usually would not fit into the block grid exactly. Instead it would usually cover four $N \times N$ blocks $C_1$, $C_2$, $C_3$ and $C_4$ in the block grid. To solve this problem, we set up a list of motion vector candidates for each $N \times N$ block in the inserted frame. For the case discussed here, the forward motion vector $(dx/2, dy/2)$ is added to the candidate lists of all $C_1$, $C_2$, $C_3$ and $C_4$ together with the area of overlap between $B_2$ and corresponding blocks $C_1$, $C_2$, $C_3$ and $C_4$. For any block in the inserted frame, the motion vector candidates with larger overlapping areas should be more reliable than those with smaller overlapping areas. We choose the "best" motion vector for each block in the inserted frame with the largest overlapping areas from the motion vector candidate list of that block. Next, we perform backward motion estimation. For each block $B_1$ located at $(x, y)$ in the (k-1)th frame, a search area is defined in the kth frame. The same exhaustive search is performed. If the block of the best match, $B_2$, is at
(\(x + dx, y + dy\)) of the \(k^{th}\) frame, the backward motion vector \((dx/2, dy/2)\) is added to the candidate list of the blocks covered by the \(N \times N\) block \(B_3\) located at \((x + dx/2, y + dy/2)\) in the inserted frame.

2.2 Overlapped Motion Compensated Temporal Interpolation (OMCTI)

In MCTI, each block in the inserted frame has only one motion vector for interpolation. So each block is synthesized independent of the surrounding blocks. We propose to use overlapped motion compensated temporal interpolation (OMCTI) similar to that found in H.263 [2] to the interpolation algorithms, such that each block in the inserted frame makes use of its own motion vector as well as those of the surrounding blocks.

![Diagram of block-based overlapping](image)

Fig. 1: The idea of block-based overlapping.

Shown in Fig 1 is a 16x16 block \(A\) surrounded by four other 16x16 blocks \(B, C, D\) and \(E\). These are some of the blocks in the block grid of the inserted frame to be generated by MCTI. Assume that MCTI has already been performed such that each of the blocks \(A, B, C, D,\) and \(E\) has the associated motion vector \(V_a, V_b, V_c, V_d, V_e\).

To perform OMCIT on \(A\), it is subdivided into four 8x8 subblocks: \(A_1, A_2, A_3\) and \(A_4\). Each subblock is synthesized by the weighted average of three predicted subblocks: one according to its own motion vector \(V_a\) and two according to the motion vectors of the neighboring blocks. Weighting masks are designed for each of the three predicted subblocks such that the sum of the weights for each pixel within the subblock is unity. In general, larger weights are given to the predicted subblock using \(V_a\).

For instance, the predicted subblocks for \(A_1\) use \(V_a, V_b\) and \(V_c\) with corresponding 8x8 weighting matrices \(W_{A_1}, W_{B_1}, W_{C_1}\). Subblock \(A_2\) uses the motion vectors \(V_b, V_c\) and \(V_d\) with corresponding weighting matrices \(W_{A_2}, W_{C_2}, W_{D_1}\). Subblock \(A_3\) uses the motion vectors \(V_c, V_d\) and \(V_e\) with corresponding weighting matrices \(W_{A_3}, W_{B_1}, W_{D_1}\). Subblock \(A_4\) uses the motion vectors \(V_d, V_e\) and \(V_e\) with corresponding weighting matrices \(W_{A_4}, W_{D_2}, W_{E_1}\). Let

\[X_{i,j}^A,\] be the \(ij\)th element of matrix \(W_{A_1}\) and other matrix elements similarly defined. Then the properties of the weighting matrices are

\[X_{i,j}^A = X_{i-7,j}^A = X_{i-7,j}^A = X_{i-7,j}^A\]

\[X_{i,j}^B = X_{i-7,j}^B = X_{i-7,j}^B = X_{i-7,j}^B\]

\[X_{i,j}^C = X_{i-7,j}^C = X_{i-7,j}^C = X_{i-7,j}^C\]

By using motion information of neighboring blocks, blocking artifacts should be much reduced.

![Example weighting matrices](image)

Fig. 2: Example weighting matrices

(a) \(W_{A_1}\), (b) \(W_{C_1}\), (c) \(W_{B_1}\)
III. SIMULATION RESULTS AND DISCUSSION

Both MCTI and OMCTI are simulated on the luminous component of the “Miss America” and “Salesman” sequences which are in CIF (288 × 352) format. The parameters $W$ and $N$ are both 16 pixels. The example matrices $W_{A1}$, $W_{B1}$ and $W_{C1}$ are used in OMCTI.

![Graph of PSNR for Salesman sequence](image)

Fig. 3: PSNR of the “Salesman” sequence.

Fig.3 shows the PSNR of the “Salesman” sequence using MCTI and OMCTI. The average PSNR of OMCTI is 35.31dB which is higher than the 35.07dB of MCTI. Fig. 5.1 and 5.2 show the simulation results of frame 10 of the “salesman” sequence using OMCTI and MCTI, respectively. The characteristics of this part of the sequence is that the left hand of the salesman undergoes large vertical motion between frame 9 and 11, making it very difficult to generate the left hand in the interpolated frame 10. Also, the right-hand of the salesman that is holding a rectangular box moves rapidly from left to right such that the prism on the shelf is covered in the frame 9 but uncovered in frame 11. This makes it very difficult to interpolate the prism and the rectangular box in frame 10. As a result, in MCTI, the base of the prism and the left side of the rectangular box are misaligned and blocky, and there is a “ghost” of the prism tip. Using OMCTI, the base of the prism is aligned properly though the ghost of the prism tip remains. The box rectangular box is still misaligned but the smoothing in OMCTI makes it visually much more pleasing.

![Graph of PSNR for Miss America sequence](image)

Fig. 4: PSNR of the “Miss America” sequence.

Before the 60th frame, the improvement by OMCTI is not much. This is because the object motion in the sequence before the 60th frame is slow and the accuracy of the motion vectors in MCTI is very high. As the degree of block mismatching is low, the room for improvement by OMCTI is very limited. However, the motion of MissA after the 60th frame is considerably larger and less linear making the motion vectors in MCTI less accurate, and thus considerable blocking artifacts appear. By using OMCTI, the PSNR of the interpolated frames are improved significantly which suggested that OMCTI is effective in the large-motion region by increasing the correlation between the blocks in the inserted frames. Fig.6.1 and 6.2 show the interpolated frame 72 of the “Miss American” sequence using OMCTI and MCTI, respectively. The interpolated frame using MCTI tend to be blocky, especially around the mouth of MissA. However, the frame generated by OMCTI is blurred such that it is visually more pleasing.

Due to its simplicity, overlapping does not require much computation, particularly when compared with the motion estimation. The total computation needed is about $6.0 \times 10^5$ and $4.2 \times 10^5$ shifting per interpolated frame in CIF format.

The PSNR improvement of OMCTI over MCTI seems to be largest in the frames with low PSNR which are exactly the frames in greatest need of improvement. Overlapping seems to work well in overcoming the weakness of MCTI.
IV. CONCLUSIONS

In this paper, we proposed a novel combination of overlapping and MCTI to form overlayered MCTI (OMCTI) to interpolate images in temporally subsampled video sequence. Simulation results suggest that OMCTI can reduce object misalignment and blocky artifacts in MCTI resulting in visually more pleasing interpolated frames. Additional computation requirement of overlapping over MCTI is very mild.

V. ACKNOWLEDGEMENT

This work is supported by HKTIT93/94.EG02 and HKUST695/95E.

VI. REFERENCE