ABSTRACT

Error concealment is important for video transmission over noisy channels. This paper presents a new spatial error concealment algorithm with low complexity to recover the lost information corrupted by channel errors, especially by burst errors, which are very common in noisy channels. Firstly, we detect the edge components in neighboring boundary pixels surrounding the corrupted macroblock (MB) group. Then for every MB in the corrupted MB group, we can judge which edges may have affected this MB. Finally, each pixel in this MB can be interpolated from boundary pixels according to the corresponding edges. Experimental results show that our low complexity algorithm is capable of restoring high-frequency (edge) information as well as low-frequency information corrupted by channel errors, especially by burst errors, and provides better performance compared with conventional methods.

1. INTRODUCTION

In video communications, the compressed video signal is very vulnerable to channel errors when transmitted over noisy channels. Error concealment provides a good way to conceal the transmission errors without incurring much overhead and delay. The most common concealment techniques use spatial and temporal interpolation, which attempt to estimate a lost macroblock (MB) from the neighboring ones in the same frame or in the adjacent frames respectively. The temporal interpolation critically depends on the availability of the motion vectors and temporal replacement. Thus it is usually simpler. However, there is no motion information available in intra-frames in the video. Thus the errors can only be concealed with the spatial interpolation [1, 2].

In this paper, we propose a new spatial error concealment algorithm with low complexity. It can reconstruct the high frequency information (edge components) as well as low frequency information corrupted by errors, especially by burst errors, which are very common in noisy channels. Firstly, we detect the edge components in neighboring boundary pixels across the corrupted MB group. Then for every MB in the corrupted MB group, we can judge which edges may have affected this MB. Finally, each pixel in this MB can be interpolated from boundary reference pixels according to the corresponding edges. Experimental results prove that our low complexity algorithm can provide better performance compared with conventional methods.

This paper is organized as follows. In Section 2, we give a review of the existing methods. In Section 3, we propose our spatial error concealment algorithm especially for the burst errors case. Then we evaluate the proposed algorithm by simulations results in Section 4. Finally, in Section 5, we draw the conclusions.

2. EXISTING METHODS

Recently, many spatial error concealment methods have been proposed. Among them, a typical method, quadrilinear border interpolation (QBI), can interpolate the value of a missing pixel $p$ in a MB from the closest top, left, bottom and right closest pixels surrounding it [3]. Basically, the interpolated value for pixel $p$ is given by

$$p = \frac{p_Td_B + p_Bd_T + p_Ld_R + p_Rd_L}{d_T + d_B + d_L + d_R}$$  \hspace{1cm} (1)$$

where $p_T, p_B, p_L$ and $p_R$ respectively denote the pixel values of the closest top ($T$), bottom ($B$), left ($L$) and right ($R$) neighbors. And $d_T, d_B, d_L$ and $d_R$ respectively denote the distances from the pixel $p$ to four neighbors. However, this method only works in the homogeneous areas and it will result in blurred image if the lost MB contains high frequency information (edge components) [4].

Other methods began to consider the high frequency components. Lee used the fuzzy logic reasoning to recover the information loss in a block-based image coding system in [5]. Sun and Jung proposed multi-directional interpolation and projections onto convex sets (POCS) in [6, 7]. Although they can recover the high frequency components and provide satisfactory results, they are too computationally expensive for real-time applications.

Kung proposed a low complexity spatial-domain error
Concealment method in [8]. For this algorithm, the computational load is low and the performance is good, but it assumes that the missing MB is surrounded by four correctly decoded MBs. Since burst errors are very common for the noisy transmission channels, the right and left neighbor MBs are most likely corrupted by burst errors and there are usually several consecutive error-corrupted MBs. Kung’s method cannot work for this case. Therefore a new technique is required.

3. PROPOSED ALGORITHM

In this section, we will propose a new spatial-domain error concealment algorithm with low complexity. This algorithm can conceal several consecutive missing MBs corrupted by burst errors with good performance.

3.1. Edge detection

In order to conceal the transmission errors, we need to estimate from the decoded image whether the missing MB to be restored belongs to a monotone or edge area of the image. If the MB belongs to a monotone area of the image, it can be directly concealed with the QBI as presented in Section 2. If the MB belongs to an edge portion of the image, we must further estimate the orientation of the edge. The information obtained from the surrounding valid decoded MBs is used as an aid to determine the edge angle. To accomplish it, the presence and absence of edges on the boundary pixels in the neighboring MBs must be determined. A reasonable simple and effective method for edge detection is through the use of gradient measures in the spatial domain. The edge gradient components are computed by applying the $3 \times 3$ Sobel mask operators for the image $P(i,j)$.

$$
G_x(i,j) = P(i,j) \bigotimes S_x(i,j) \quad (2)
$$

$$
G_y(i,j) = P(i,j) \bigotimes S_y(i,j) \quad (3)
$$

where $S_x$ and $S_y$ are

$$
S_x = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix} \quad S_y = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{bmatrix}
$$

The magnitude and angular direction of the gradient at coordinate $(i,j)$ are

$$
G(i,j) = \sqrt{G_x(i,j)^2 + G_y(i,j)^2} \quad (5)
$$

$$
\theta(i,j) = \tan^{-1}(G_y(i,j)/G_x(i,j)) \quad (6)
$$

Consider a group of MBs corrupted by burst errors, which are surrounded by the shaded area as shown in Fig. 1. Firstly, the edges can be detected by calculating the gradient field on the boundary pixels in the neighboring MBs of the corrupted MB group, which are indicated by the shaded area in Fig. 1. Then we can get the amplitude $G(i,j)$ for every boundary pixel $(x(i,j))$. In order to get the edge points, we predefine the threshold $G_{th}$ for the amplitude of the gradient. If $G(i,j)$ is not smaller than $G_{th}$, $x(i,j)$ is declared to be the edge pixel. Usually several consecutive pixels are declared to be the edge pixels. In this case, only the pixel with the largest gradient amplitude is the true edge pixel.

Assume there are $N$ edge pixels detected in the boundary pixels surrounding the corrupted MB group. Then the set of the $N$ edge pixels can be described as:

$$
P_e = \{(x_{e,i},y_{e,i})|1 \leq i \leq N, i \in Z\} \quad (7)
$$

The edge passing through each edge pixel can be represented by a line equation. The edge slope $k_i$ of the $i$th edge line can be obtained from:

$$
k_i = -G_{x,i}/G_{y,i} \quad (8)
$$

where $G_{x,i}$ and $G_{y,i}$ are the gradient components for the $i$th edge pixel obtained from (2) and (3) respectively.

Then the $i$th edge line $l_{e,i}$ can be represented by a line equation, i.e.

$$
y - y_{e,i} - k_i(x - x_{e,i}) = 0 \quad (9)
$$

Thus the set of the edge lines can be described as:

$$
L_e = \{l_{e,i}|1 \leq i \leq N, i \in Z\} \quad (10)
$$

For example, in Fig. 2, there are five consecutive MBs corrupted by burst errors and seven edge lines detected in this MB group, i.e. $N = 7$. Thus $L_e$ is described as:

$$
L_e = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7\} \quad (11)
$$

3.2. Error concealment

After getting all the $N$ edge lines, we can reconstruct each pixel in the corrupted MB group.
Firstly, for a lost pixel \( p(x, y) \), we need to count the number of edge lines passing through the MB, which the lost pixel lies in. The number is represented by \( M \). These \( M \) edge lines passing through the MB can construct a set of lines described as:

\[
L_{eMB} = \{ l_{e,i} | MB \neq \phi, l_{e,i} \in L_e \} \tag{12}
\]

where \( l_{e,i} \cap MB \neq \phi \) means that the edge line \( l_{e,i} \) passes through this MB. In order to conceal the lost pixel \( p(x, y) \), we can get two reference pixels to interpolate for each edge line in the set of \( L_{eMB} \). Thus we can retrieve \( 2M \) reference pixels for each lost pixel to interpolate. For example, in order to conceal the lost pixel \( p \) in \( MB_3 \) in Fig. 2, we first find out that there are two edge lines, \( L_2 \) and \( L_4 \) passing through \( MB_3 \), i.e. \( M = 2 \). Thus \( L_{eMB} \) is described as:

\[
L_{eMB} = \{ l_2, l_4 \} \tag{13}
\]

Then along the edge directions of the two edge lines, we can draw the reference lines passing through the lost pixel \( p \), i.e. line \( AB \) and Line \( CD \). Then four reference pixels \((A, B, C, D)\) can be retrieved in the boundary pixels in neighboring MBs as shown in Fig. 2.

Then we need to consider which reference pixels can be used for interpolation. Since there are edges in the frame, only the reference pixels on the same side of every edge lines as the lost pixel \( p(x, y) \) can be adopted for interpolation. Actually, we need not consider all the edge lines and we only need to consider the edge lines in the corresponding MBs. The corresponding MBs are defined as Passed MBs (PMB), which indicate the MBs in the corrupted MB group that the reference lines pass through. For example, the PMB is comprised of \( MB_2 \) and \( MB_3 \) to conceal the missing pixel \( p \) as shown in Fig. 2, since the reference lines \( AB \) and \( CD \) only pass through these two MBs in the corrupted MB group.

Suppose there are \( Q \) edges passing through the PMB. Then the set of these \( Q \) edge lines can be described as:

\[
L_{ePMB} = \{ l_{e,i} | l_{e,i} \cap PMB \neq \phi, l_{e,i} \in L_e \} \tag{14}
\]

where \( l_{e,i} \cap PMB \neq \phi \) means that the edge line \( l_{e,i} \) passes through this PMB. For example, in order to conceal the missing pixel \( p \) in Fig. 2, \( Q = 3 \) and the set of \( L_{ePMB} \) is comprised of \( l_2, l_3 \) and \( l_4 \).

\[
L_{ePMB} = \{ l_2, l_3, l_4 \} \tag{15}
\]

Then for each reference pixel, we need to judge if it is on the same side of each edge line in the set of \( L_{ePMB} \) as the lost pixel \( p(x, y) \). Actually, it’s not necessary for a reference pixel to consider the edge line, by which the reference pixel is obtained. For example, for the reference pixel of \( A \) in Fig. 2, the edge line of \( l_4 \) is not necessary to be considered, since \( A \) is obtained by \( l_4 \). Therefore only the reference pixels, which succeed for \((Q - 1)\) edge lines in \( L_{ePMB} \), can be declared as the valid reference pixels and can be used for interpolation. For example, in order to conceal the missing pixel \( p \) in Fig. 2, the valid reference pixels are pixels of \( A, B \) and \( C \). Pixel \( D \) is dropped due to the edge of \( l_3 \).

After getting all the valid reference pixels, the lost pixel \( p \) can be concealed as:

\[
p = \frac{\sum p_i}{\sum d_i} \tag{16}
\]

where \( p_i \) means the ith reference pixel and \( d_i \) means the distance between the lost pixel \( p \) to \( p_i \). For example, in Fig. 2, the lost pixel of \( p \) can be concealed as:

\[
p = \frac{p_A + p_B + p_C}{d_A + d_B + d_C} \tag{17}
\]

It should be noted that if there are no edge lines in a MB, which is in a group of corrupted MBs, then each pixel in this MB can be interpolated by the nearest edge pixels along the vertical direction.

It should also be noted that our proposed algorithm has low complexity. As mentioned above, there are two reference pixels for each edge line in order to conceal a lost pixel in the corrupted MB group. Then there should be \( 2N \) reference pixels, since \( N \) edges are detected. In order to determine if a reference pixel is valid or not, it should be judged that if it is on the same side of other \((N - 1)\) edge lines as the lost pixel. Thus the computation time for every reference pixel is \((N - 1)\). For all the \( 2N \) reference pixels, the total computation time is \( 2N(N - 1) \) to conceal a pixel. But in our proposed algorithm, we only need to consider the \( 2M \) reference pixels to conceal a lost pixel, which is determined by the number of the edge lines passing through the MB as mentioned above. In order to decide if the reference pixel is valid, we only need to consider the \((Q - 1)\) edge lines which pass through the PMB as defined above. Therefore in order to conceal a lost pixel, the total computation time is only \( 2M(Q - 1) \). Since \( M << N \) and \( Q << N \), then \( 2M(Q - 1) << 2N(N - 1) \). For example, in order to conceal the missing pixel \( p \) in Fig. 2, the computation time for conventional method is \( 2N(N - 1) = 2 \times 7 \times (7 - 1) = 84 \). But in our proposed algorithm, the total computation time is only \( 2M(Q - 1) = 2 \times 2 \times (3 - 1) = 8 \). Therefore our proposed algorithm has low complexity.

4. SIMULATION RESULTS

Simulations have been done in order to test the performance of our proposed spatial error concealment algorithm. Although the proposed algorithm is general and can be applied to any video compression method, in this paper, MPEG-4 is used as our video coding framework. The "Foreman" CIF (352 × 288) image is compressed and decompressed at the bit rate of 384 kbps with a MPEG-4 codec in the simulation as an example, since it exhibits many sharp edges and small homogeneous areas in the background. The size of the MB is \( 16 \times 16 \).
Fig. 3. An intra-coded frame of the "Foreman" sequence at 11% packet loss rate (a) error-free frame; (b) error-damaged frame; (c) concealed with QBI; (d) concealed with our proposed algorithm

Table 1. PSNR comparison for the "Foreman" video sequence at different PLRs.

<table>
<thead>
<tr>
<th>PLR</th>
<th>1%</th>
<th>3%</th>
<th>6%</th>
<th>8%</th>
<th>11%</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBI</td>
<td>33.31</td>
<td>32.37</td>
<td>31.40</td>
<td>30.69</td>
<td>29.31</td>
</tr>
<tr>
<td>Proposed</td>
<td>35.17</td>
<td>34.63</td>
<td>34.44</td>
<td>33.82</td>
<td>33.38</td>
</tr>
</tbody>
</table>

To simulate the channel burst errors, we group the coded video information into packets, where each video packet consists of several MBs. Then the video packets are multiplexed with audio information. Since the audio information is inserted between two video packets, the burst errors will most likely not corrupt two consecutive video packets.

In this simulation, we employ different packet schemes, assigning different number of MBs to a packet. Then we can get the simulations results at different packet loss rates (PLRs). Table 1 provides the PSNR values for QBI and our proposed algorithm at different PLRs. The table shows that our proposed algorithm significantly outperforms the QBI.

Fig. 3 shows the simulation results for a "Foreman" intra-coded frame with two missing packets corrupted by burst errors. In this packet scheme, each packet contains 22 MBs. So the PLR is about 11%. Fig. 3(a) shows the error free frame coded and decoded with the MPEG-4 codec. Fig. 3(b) shows the error damaged frame with two missing packets corrupted by burst errors. Fig. 3(c) shows the results obtained using the QBI. Fig. 3(d) shows the results concealed by our proposed algorithm. By comparing these figures, the performance of our proposed algorithm is perceptually superior to QBI. This advantage can be demonstrated around the edges in the missing MBs. And the image quality is consistent with the PSNR measurement shown in Table 1.

5. CONCLUSIONS

In this paper, we have proposed a spatial-domain error concealment algorithm for video transmission over noisy channels. Compared with other existing error concealment methods, the novelties of our approach lies in: 1) The algorithm proposed by Kung can only conceal a missing MB surrounded by four correctly decoded MBs [8]. However our proposed algorithm can conceal the consecutive MBs corrupted by burst errors. It’s very important since burst errors are very common in noisy channels. 2) Our algorithm works with low complexity compared with others, which is important for real-time applications.

Experimental results have demonstrated the effectiveness of our proposed algorithm to conceal the consecutive erroneous MBs corrupted by burst errors. In conclusion, the algorithm proposed in this paper is a highly effective and efficient error concealment method for visual communications over noisy channels.

6. REFERENCES