Robust transmission of packet video through dispersive packetization and error concealment

Ivan V. Bajić, John W. Woods, and Ali Mohammad Chaudry

Center for Image Processing Research and ECSE Department
Rensselaer Polytechnic Institute, Troy, NY 12180, USA
{ivanb, woods, chaudm}@cipr.rpi.edu

Abstract

In this paper we present a new coding and packetization approach for intraframe coded video transmission, based on space-frequency dispersion. Subband-decomposed video data is packetized so that data appearing in any one packet is highly dispersed in both space and frequency. Each packet is a description of the frame at various points in space and frequency. Any errors arising from packet loss are spread across the frame and this allows easy error concealment. We compare our method to the more conventional block-based packetization scheme and illustrate its advantages in both PSNR and subjective visual quality.

1 Introduction

The lack of quality of service (QoS) guarantees in today’s most widely used networks makes real-time video communication a challenging task. Hence, the need exists for robust video codecs that are able to deliver reasonable quality video even under high packet loss. Robustness has been recognized as one of the important factors in recent image and video coding standards [1].

Methods proposed so far for robust image/video transmission can be broadly classified into two groups. One group consists of the methods for layered transmission, e.g. [2, 3, 4]. Layers are protected either by assigning different priorities (if the network supports this feature), or by assigning different amount of FEC, to different layers. The amount of protection depends on the importance of the layer for the overall image quality. Both approaches, however, have been criticized, either because not all network protocols support prioritization, or because of the bandwidth inefficiency of FEC.

Recognizing the limitations of layered transmission, the second group of methods for robust image/video transmission emerged recently [5, 6, 7, 8]. These methods are particularly suitable for networks that do not support prioritization. Since in this case all packets are treated equally, the goal is to produce packets that are equally important for visual quality. Several ways of achieving this have been proposed. In multiple description scalar quantization (MDSQ) in [5], each coefficient is quantized using several interleaved quantizers, thereby producing several coarse descriptions. Combining all the received descriptions results in a finer quantization scale and gives better image quality than any single description. In [6], a method called packetized
zerotree wavelet (PZW) for packetizing the bitstream produced by zerotree coders such as [9, 10] was proposed for ATM cells. Each tree of subband/wavelet coefficients is compactly encoded and stored in the cell along with several other trees. The trees that appear in any given cell are separated in space so that loss of one cell produces only one missing tree locally and some coefficients from the missing tree can be interpolated from their neighbors. In [7], a similar concept is employed by making the data that appears in any packet separated in frequency. Here, each subband is partitioned into several blocks of spatially clustered coefficients and then blocks from different subbands are combined into packets so that blocks in a packet correspond to different spatial locations. Robustness comes from the fact that the loss of a packet destroys the information about any spatial location from only one frequency band.

The method we propose here is the extension of dispersive packetization (DP) from [8] to the case of intraframe coded video. It has its roots in [11], and represents the combination of good features from the methods of [6] and [7] described above. Data that appears in any packet is dispersed in both space and frequency. This enables both easy error concealment (due to dispersion in space), as well as preservation of high frequency information in the areas affected by the loss of low frequency information (due to dispersion in frequency). Our method can also be treated as a multiple description coding approach, since each packet describes any spatial location in different frequency bands and any frequency band at different spatial locations. Here, multiple descriptions are created in the space-frequency domain, rather than the ‘amplitude’ domain, as in MDSQ [5].

In the following section we describe coding packetization and error concealment steps in the proposed method. In Section 3 we briefly discuss the network simulation used during the experiments and in Section 4 we present and discuss experimental results. The paper ends with a conclusion in Section 5.

2 Dispersive packetization

2.1 Dispersion in space and frequency

The concept of dispersive packetization (DP) can be explained using Figure 1. The figure shows a subband-decomposed image with two levels of decomposition. One particular subband coefficient is shown in black, and its neighborhood in the space-frequency domain is shown in gray. In DP, data is packetized in such a way that none of the coefficients in the gray region appears in the packet where the coefficient shown in black is stored. If this packet is lost, the space-frequency neighborhood of the missing coefficient remains intact, and the coefficient value may be estimated from its neighbors.

To simplify further explanations, we introduce the following notation: each subband is denoted by \( XY_n \), where \( X, Y \in \{L, H\}, n \in \{1, 2, \ldots, N\} \), and \( N \) is the number of decomposition levels. \( X \) and \( Y \) denote the horizontal and vertical direction respectively, while \( L \) and \( H \) stand for low-pass and high-pass filtering. Hence, for example, \( LH_3 \) denotes the subband that has been low-pass filtered horizontally and high-pass filtered vertically at the 3rd decomposition level.
Now consider a subband-decomposed image with \( N \) levels of decomposition. Each coefficient in the interior of the \( LL_N \) band has a total of 8 nearest neighbors in the same band. These are its neighbors in space. The number of its neighbors in frequency is simply the number of the coefficients of the tree originating in that coefficient. There are three other subbands at the \( N \)-th decomposition level (\( LH_N \), \( HL_N \) and \( HH_N \)), each containing one member of the tree, and three subbands in any of the higher frequency decomposition levels. As we go towards higher frequency subbands, the number of the elements of a tree in any subband increases by a factor of 4 compared to the corresponding subband in the previous decomposition level. Hence, the total number of its neighbors is

\[
N_{nb} = 8 + 3 \sum_{k=0}^{N-1} 4^k = 7 + 4^N.
\]

If we want each coefficient from a given neighborhood to appear in a different packet, the number of packets \( P \) must be at least \( N_{nb} + 1 \), i.e.

\[
P \geq N_{nb} + 1 = 8 + 4^N.
\]

Hence, to achieve full dispersion, the required number of packets may be relatively large. This is acceptable in case of ATM networks where, due to the small packet size, the number of packets needed to transmit the image is large. But in the case of Internet, small packets are considered to be inefficient, and with large packets it may be impossible to achieve full dispersion. In this case we suggest high degree of dispersion in the subbands where error concealment is applied (usually in the low-frequency bands), while dispersion in other bands may be sacrificed.

In our experiments we used a 336 × 240 football sequence, compressed at 0.4bpp per frame, which at 30fps is equivalent to 1 Mbps. Each compressed frame is about 4 KBytes in size and can be transmitted in 9 packets of about 450 Bytes. This size is not crucial, as long as it does not exceed 576 Bytes, which guarantees that the packet will not be fragmented. Four levels of subband decomposition are performed on each frame, using a 9/7 Daubechies filter. The size of subbands at the fourth level of decomposition is 21 × 15.
Dispersion is achieved in the following way. $LL_4$ subband is covered with a packetization mask shown in Figure 2. The figure shows a $21 \times 15$ matrix, where each block in the matrix corresponds to a coefficient in $LL_4$. The number in the block is the number of the packet where the coefficient is stored. No two neighboring coefficients are stored in the same packet, hence dispersion in space is achieved in $LL_4$. This dispersion is illustrated by highlighting the coefficients that are placed in packet number 5.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**Figure 2:** Dispersive packetization (DP) mask for 9 packets

Packetization masks for higher frequency bands are also $21 \times 15$. Subbands from first three decomposition levels, whose size exceeds $21 \times 15$, are covered with several instances of their packetization mask. Construction of packetization masks for higher frequency bands proceeds as follows. Subbands are considered in sequence from lowest frequency band ($LL_4$) to highest frequency band ($HH_1$). In our case we have 13 subbands in the sequence, say $S_1, S_2, \ldots, S_{13}$, with $S_1 \equiv LL_4$ and $S_{13} \equiv HH_1$. A block in the mask for subband $S_i$ is filled with the number which was not used in that block in subbands $S_1, S_2, \ldots, S_{i-1}$, if possible. Since we only have 9 available numbers and 13 subbands, some numbers will have to be repeated after $S_9$, but dispersion is achieved in lower frequency bands. In particular, space-frequency dispersion is achieved in $LL_4$, $LH_4$ and $HL_4$, where error concealment is performed.

Using this approach, each packet contains the same number of coefficients from each subband, and coefficients from any single band are dispersed in space. Hence, no particular region of the image is favored in any packet, which makes the packets roughly equally important for both objective and subjective quality. Also, resulting packets are of approximately equal sizes.

In our experiments, we compare the dispersive packetization (DP) method with a block-based packetization (BP) method. In BP, each $336 \times 240$ frame is divided into nine $112 \times 80$ blocks, and all data from a given block are stored in the same packet. This results in image quality being highly dependant on the actual packets that are lost, and also prevents successful error concealment, as will be demonstrated in Section 4.
2.2 Coding and packetization

In this section we describe the coding and packetization steps in the proposed method for robust video transmission. Our coder is the modification of the baseline wavelet coder [12], which is based on context modeling from [13]. The operational block diagram is shown in Figure 3.

![Block diagram of the coder/packetizer](image)

**Figure 3:** Block diagram of the coder/packetizer

Input to the coder is the set of transform coefficients of the subband-decomposed image. Coding starts from the lowest frequency subband and proceeds to the higher frequency subbands. Each coefficient is applied to the quantizer represented by a sequence of decreasing thresholds levels. Comparing the coefficient to a particular threshold level produces a symbol at a layer corresponding to that threshold level. This symbol is then arithmetically encoded using the probability table for its context, where the context is determined from the symbols corresponding to the same coefficient at previously encoded layers. Probability tables are created in the first pass of the quantization process, and the actual encoding is done in the second pass. These probability tables are transmitted in a separate packet since they are needed for decoding. This overhead can be reduced if the same set of tables is used for encoding a group of consecutive frames, as in our case, where we have used one set of tables per 10 frames. Overhead can be completely avoided if universal tables are used, as was done in [13].

A separate context-based arithmetic coder is used for each subband. The output of the arithmetic coder is switched from one packet to another according to the packetization mask for the subband that is currently being processed. While a coefficient is being encoded, the output of the coder is directed to a packet into which the coefficient should be stored, according to the packetization mask. After that coefficient has been encoded, the output of the coder is redirected to a packet where the following coefficient should be stored, and so on. For example, with reference to
Figure 2 in the previous section, when the third coefficient in the first row of the matrix is being encoded, the bits produced are stored in packet number 3, when the fourth coefficient is being encoded, the bits are stored in packet number 7, etc. In this way, all the bits produced by encoding a given coefficient are stored in the same packet, and the context information needed for decoding that coefficient is also available in the same packet. This means that packets are individually decodable, i.e. decoding of any packet does not require the presence of any other packet.

Decoding proceeds in reverse order from encoding. Packetization tables that are used for encoding are assumed to be present at the decoder, so decoder can determine which subband coefficients are stored in which packet. During decoding, as the symbols corresponding to more significant layers of the coefficient are being decoded, the context information for less significant layers becomes available and decoding proceeds until the coefficient value has been fully decoded. The value is then placed in the appropriate location in the subband structure, according to the packetization tables. Once all the coefficients from the received packets are decoded, error concealment is performed on the missing coefficients, as described in the following section.

2.3 Error concealment

Simple error concealment is performed in $LL_4$, $LH_4$ and $HL_4$ bands, since in these bands there is a lot of correlation remaining between the neighboring coefficients. In $LL_4$ band, a missing coefficient is interpolated bi-linearly from the four nearest available neighbors, in the horizontal and vertical directions. In the $LH_4$ and $HL_4$ bands, missing coefficients are interpolated linearly from the two nearest available neighbors in the direction in which the subband has been low-pass filtered. Missing coefficients in other bands are assumed to be zero.

It should be noted that a more advanced concealment algorithms can be applied to DP data. Due to space-frequency dispersion, adaptive interpolation algorithms, such as the one proposed in [14], can be used. In the method of [14] interpolation in the lowest frequency band is adapted according to the data behavior in the higher frequency bands at the same spatial location. Nevertheless, we found that even a simple linear concealment can produce useful reconstructed images.

3 Network simulation

Network topology used in the simulation is shown in Figure 4. The network consists of 52 nodes, with our sender and receiver indicated. Our sender/receiver pair uses RTP as the application layer protocol, with UDP and IP as transport and network layer protocols, respectively. Other nodes in the network generate TCP traffic, so our RTP data flow competes against multiple TCP flows.

For the purposes of this paper, topology shown in Figure 4 is able to simulate sufficiently realistic environment for video transmission over IP. Simulations were performed using the ns-2 network simulator [15].
4 Results and discussion

We performed a set of experiments on a $336 \times 240$ football sequence, compressed at 0.4bpp per frame, as described before. Results of comparing the proposed dispersive packetization (DP) method to the block-based method for three values of fixed packet loss are summarized in Table 1. DPC stands for ‘dispersive packetization with concealment’, BPNC stands for ‘block-based packetization with no concealment’, etc. Results were generated by simulating the transmission of 60 frames of the sequence, with packet loss fixed at 11.1%, 22.2% and 33.3%, respectively. Average PSNR for the sequence with no packet loss was 27.80dB with standard deviation of 0.90dB.

<table>
<thead>
<tr>
<th>Method</th>
<th>Avg PSNR</th>
<th>Std Dev</th>
<th>EC Gain</th>
<th>Avg PSNR</th>
<th>Std Dev</th>
<th>EC Gain</th>
<th>Avg PSNR</th>
<th>Std Dev</th>
<th>EC Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPC</td>
<td>25.14</td>
<td>0.85</td>
<td>5.31</td>
<td>23.42</td>
<td>0.90</td>
<td>6.22</td>
<td>22.07</td>
<td>0.89</td>
<td>6.44</td>
</tr>
<tr>
<td>BPC</td>
<td>24.35</td>
<td>1.54</td>
<td>4.22</td>
<td>22.44</td>
<td>1.54</td>
<td>4.90</td>
<td>20.71</td>
<td>1.19</td>
<td>4.89</td>
</tr>
<tr>
<td>DPNC</td>
<td>19.83</td>
<td>0.52</td>
<td></td>
<td>17.20</td>
<td>0.45</td>
<td></td>
<td>15.63</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>BPNC</td>
<td>20.13</td>
<td>1.15</td>
<td>17.54</td>
<td>15.82</td>
<td>0.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: PSNR performance in dB for (a) 11.1% loss, (b) 22.2% loss and (c) 33.3% loss

In the case when no concealment is applied, BP results in a slightly higher average PSNR than DP, but with a significantly higher variation in PSNR. This is caused by the fact that in BP different blocks have different importance to the image quality, so the image quality is highly dependent on the actual blocks that are lost. On the other hand, when concealment is applied, DP outperforms BP by about 1dB, again with a much lower variation in PSNR. Dispersion in DP

![Network topology used in the simulation](image-url)
makes error concealment more powerful than in the case of BP. This is also indicated by the error concealment (EC) gain in the table.

Figure 5: Results of simulating the transmission of 120 frames of the football sequence

Figure 5 shows the results obtained by simulating the transmission of 120 frames (4 seconds at 30fps) of the football sequence, over the network described in Section 3. Again, we compare DP to BP, with and without concealment. Average packet loss for the entire segment was 25.4%, and average PSNR results were 23.05 dB for DPC, 22.02 dB for BPC, 16.99 dB for BPNC and 16.84 dB for DPNC.

Results in the figure confirm the behavior indicated in Table 1. With no concealment, both DP and BP have similar average performance, with BP having more variation in quality on a frame-by-frame basis. When concealment is applied, DP has an advantage of about 1dB on average, with a more consistent quality i.e. less variation in PSNR.
DP also provides a better visual quality than BP. To illustrate this, we show in Figure 6 one frame from the sequence with 33.3% packet loss. Both BP and DP cases are shown, with and without concealment. In BP case, blocks that were lost happen to contain important details and due to the size of the blocks, error concealment cannot recover this data which results in the visually poor quality. DP on the other hand has preserved most of the details and error concealment produces, for a given amount of loss, a reasonable quality image.

Figure 6: Frame 6 at 33.3% (3 out of 9) packet loss (from top left to bottom right): original frame, coded frame with no loss (PSNR = 29.2dB), DPNC (PSNR = 15.3dB), DPC (PSNR = 23.5dB), BPNC (PSNR = 14.8dB), BPC (PSNR = 20.3dB)
5 Conclusions

A dispersive packetization (DP) method was presented and compared against a more conventional block-based packetization (BP). Main feature of DP is dispersion of image/video data in both space and frequency, which provide its robustness. Dispersion in space enables error concealment to achieve higher reconstruction gain. Dispersion in frequency ensures that the high frequency information (features) are preserved in the area that are affected by the loss of low frequency information. It was found that DP performs better than BP in both PSNR and subjective image quality, producing a consistent image quality at a fixed packet loss rate.

Future research will include the extension of the concepts of DP to the motion compensated video coding and incorporation of the more advanced interpolation algorithms.

6 References