HAL
open science

A Novel Video Packet Loss Concealment Algorithm & Real Time Implementation

Abraham Suissa, Jennifer Mellor, Frantz Lohier, Patrick Garda

To cite this version:

Abraham Suissa, Jennifer Mellor, Frantz Lohier, Patrick Garda. A Novel Video Packet Loss Concealment Algorithm & Real Time Implementation. DASIP 2008, Nov 2008, Bruxelles, Belgium. hal-00406145

HAL Id: hal-00406145

https://hal.science/hal-00406145

Submitted on 21 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Novel Video Packet Loss Concealment Algorithm & Real Time Implementation

Abraham Suissa
Pierre and Marie Curie University – Paris 6, EA 2385 – BC 252, 4 Place Jussieu, 75252 PARIS CEDEX 05, France
Abraham.Suissa@gmail.com

Jennifer Mellor
New Platforms and Technologies, ICBU, Logitech Inc., Fremont, California, USA
Jennifer_Mellor@Logitech.com

Frantz Lohier
New Platforms and Technologies, ICBU, Logitech Inc., Fremont, California, USA
Frantz_Lohier@Logitech.com

Patrick Garda
Pierre and Marie Curie University – Paris 6, EA 2385 – BC 252, 4 Place Jussieu, 75252 PARIS CEDEX 05, France
Patrick.Garda@upmc.fr

Abstract— Streaming video data over a digital Radio Frequency (RF) link operating in the ISM band (e.g., 2.4GHz/5GHz) is known to be a challenging task that has recently captured considerable industry attention. Various sources of air interference cause packets to be dropped despite retries and forward error correction (FEC) schemes found either at the transport or video encoding layer. However, there is an industrial interest for a concealment technique suitable for wireless cameras, featuring very low processing power. This article presents therefore an alternative to the conventional wisdom: a novel video packet loss concealment scheme that operates after the decoding/ decompression of a truncated video bitstream. The approach is therefore codec-independent and does not incur the overhead associated with typical FEC techniques. We demonstrate recovery up to 20dB using a real-time implementation of our proposed approach.

I. Introduction

In order to ensure proper Quality of Service (QoS) when streaming audio/video data over an error prone transmission link, such as an RF channel challenged by interference sources including the hopping nature of certain technologies (e.g., Bluetooth), multi-path fading challenges, and competing air-time access of devices, a combination of techniques is typically necessary both at the transport layer and data encoding/compression layer. At the transport layer, we can cite coexistence schemes that have been promoted around the Bluetooth technology or convolutional data coding techniques at the modulation layer (e.g., viterbi). For the case of video encoders such as H.264 or JPEG2000 (e.g. of Chapter 11), standards typically discuss the insertion of resynchronization markers in the compression loop as well as a structural bitstream arrangement to allow the decoder to continue the decoding process despite erasures encountered in the compressed image, which essentially prevents frame skipping.

However, there is an industrial interest for a concealment technique suitable for wireless cameras [1], sharing the electromagnetic spectrum with some wireless networks. The very low complexity encoder processing power of these devices limits the video coding to very low complexity encoder. The Wyner-Ziv video coding algorithm [2] is attractive for its low complexity, but it is not yet standardized. Thus we relied on the Motion JPEG (MJPEG), which also features low complexity encoder. Then, the concealment task is performed by the decoding receiver. This receiver is actually a PC or a server (Figure 1); it has thus far more computing power that the wireless camera.

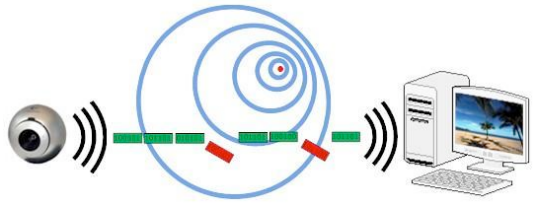

Figure 1 : Video PLC difficulty overview

Therefore, in this paper, we pursue an approach where a video packet loss concealment (VPLC) technique is executed after the truncated stream is decompressed. The technique is independent of the underlying transport or video encoding technology, the main assumption being that the video decoder is capable of identifying when an erasure occurs and can continue the video decoding process of an image as opposed to dropping the frame. Once the partially decoded image is reconstructed, our algorithm, by using spatial and temporal information of the stream, is able to recover the non-truncated frame with minimal distortion.

This article is structured as follows. In section 2, we describe our unique approach for VPLC, which leverages several known techniques operating in the spatial and temporal domain. At the algorithm's heart are the following techniques: motion estimation of deleted blocks, block copy and spatial interpolation of missing data, a deblocking filter and a sophisticated decision tree which selects the execution of each of the above building blocks. In section 3, we discuss the key aspects of a sample implementation based on the motion JPEG (MJPEG) industry standard. We review the provision that such a standard offers for a decoder to resynchronize on a truncated stream and propose extensions to the baseline mechanism to further optimize the performance of our VPLC algorithm. Finally, in section 4, we present the performance gain of our

approach based on our experimental MJPEG-over-wireless streaming demonstrator. The benefits of our approach are discussed in terms of video quality improvement, processor execution time impact and recovered frame rate.

II. Video Packet Loss Concealment Scheme

In this section, we present our new approach to video packet loss concealment. The VPLC scheme uses motion vector estimation after decoding and a decision tree employing several methods of error concealment, including block copy and spatial interpolation, to achieve recovery of lost video data with minimal visual artifacts.

In order to characterize the size and frequency of truncations to be concealed, we consider a straightforward compression and packetization scheme for streaming wireless video data. For example, with a compressed JPEG image transmitted over an 802.11 connection, a single lost transport packet could realistically result in up to 4 consecutive 8x8 block rows of missing uncompressed video data. Depending on the frequency of errors in transmission, each decoded video frame may incur several lost packets of several rows each. This nature of video data loss is inherently challenging for recovery mechanisms and serves to demonstrate the applicability of our VPLC algorithm for a variety of codecs and/or packetization schemes for low to moderate probabilities of transport packet loss.

A. Error Concealment Using Motion Vector Estimation

1) Definition: For our proposed VPLC algorithm, there are several core concepts. First, we maintain a reference image for motion detection based on a valid decoded video frame (e.g., the previous frame). Then, we calculate motion vectors for the regions surrounding lost video data in the current frame based on the reference image. A block matching technique [7] is used for this purpose. We interpolate the values to estimate motion vectors for the missing data itself, and finally we use that motion vector information in our decision tree to apply the most appropriate techniques for replacing the missing blocks.

One key point is that motion vector estimation is not only used to gauge whether there is motion in the region but to determine where in the reference image the now-missing data would have been. At 15-30 frames per second (fps), there is generally little enough motion between frames that the likelihood of locating the missing video data in the reference image is high.

To obtain motion vectors of the missing blocks we perform block matching on the closest neighbor blocks, calculating motion vectors for the nearest valid blocks on the top and bottom of the missing blocks. To estimate the motion vectors for the missing blocks, we use an interpolation between these two motion vectors as shown in Figure 2.

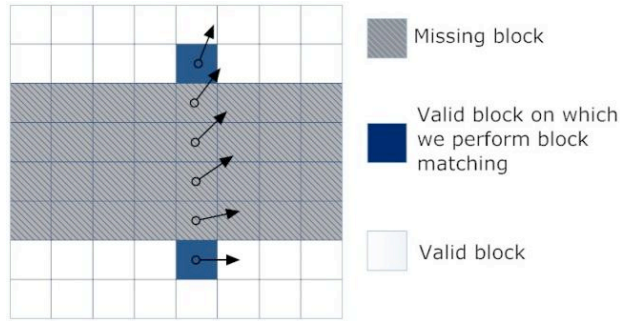

Figure 2 : Motion vector interpolation

2) Block Matching: Block matching techniques [7] match blocks from the current frame with blocks from a reference frame. In the proposed mechanism the block matching process is done on the luminance plane; processing in the chrominance plane is not required here because the luminance plane stores in average 80% of the signal energy.

The displacement in block location from the current frame to the location in the reference frame is the motion vector. Block matching techniques can be divided into three main components:

- Block determination
- Search method
- Matching criteria

The first component, block determination, specifies the position and size of blocks in the current frame, the start location of the search in the reference frame, and the scale of the blocks. We focus on fixed size, disjoint blocks spanning the frame, with initial start location at the corresponding location of the block in the reference frame.

The search method is the second component, specifying where to look for candidate blocks in the reference frame. A fully exhaustive search consists of searching every possible candidate block in the reference frame. This search is computationally expensive and another search method has been proposed to reduce the error in the matching (section *II.B.2. Search Method*).

The third component is the matching criteria. The matching criteria are a similarity metric to determine the best match among the candidate blocks.

SAD The sum of the absolute values of the differences in the two blocks:

$$M(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{n} |I_c(i,j) - I_r(i+p, j+q)|$$

MAD The mean of the absolute values of the differences in the two blocks:

$$M(p,q) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} |I_c(i,j) - I_r(i+p, j+q)|$$

MSD The mean of the square of the differences in the two blocks:

$$M(p,q) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(I_c(i,j) - I_r(i+p,j+q)\right)^2$$

MPC The sum of the non-matching pixels in the two blocks; a match is determined by the absolute value of the difference being less than a threshold, t_{MPC}.

$$M(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{n} D\left(I_c(i,j), I_r(i+p,j+q)\right)$$

$$D(a,b)\begin{cases} = 0 & if\ |a-b| \le t_{MPC} \\ = 1 & \text{otherwise} \end{cases}$$

SSD The square of the differences in the two blocks:

$$M(p,q) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left(I_c(i,j) - I_r(i+p,j+q)\right)^2$$

SAD and MAD only differ by a constant in the case of fixed size blocks and can be used interchangeably in our comparison. In practice, SAD is faster due to the removal of the divide operation. While MAD incorporates large differences, MSD penalizes blocks with one or more large differences. MPC on the other hand equally weights any difference above a threshold. In our implementation the MSD criteria is optimized, making it faster than the other criterions and the best choice for us to use.

3) Error Concealment Procedure: The concealment procedure is as follows. If there is no motion in the missing block region, we conceal the missing block using *Block Copy Concealment*. This common method exploits the statistical assumption that a lost block in the previous frame is unlikely to be lost in the next frame. Replacing a lost block with the similarly positioned block in the previous frame is the easiest and fastest concealment method [4] [5] and works well in still regions of the image, such as backgrounds and inanimate objects.

If there is motion in the region, however, we estimate the location of the missing block in the reference image and use that for copying to conceal the missing block. This works well with animated objects or persons or in situations where the camera itself is in motion.

The final path is when there is motion but the results of motion vector estimation cannot be used for concealment purposes. There are three cases in which this could occur: when the block on which we perform block matching does not exist in the reference frame, when the displacement of the missing block is larger than the search area, or when the interpolated motion vector indicates a block outside the boundary of the reference image. In any of these cases, we use *Spatial Concealment* [4] [5] [6], a method which conceals each pixel of the missing block with a weighted interpolation of the surrounding four pixels of the adjacent undamaged blocks.

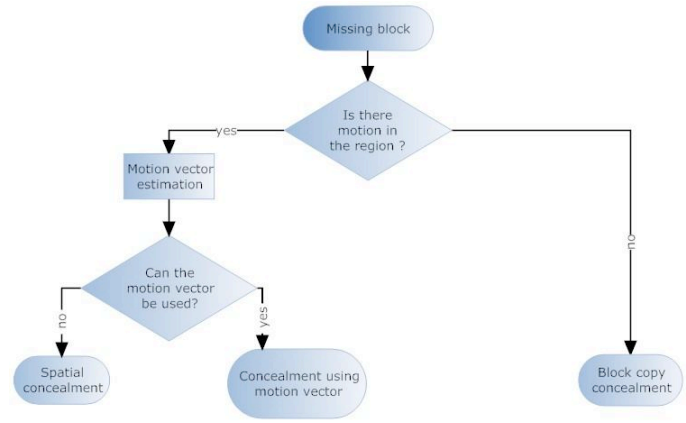

Figure 3 : Block diagram of the algorithm decision tree

Typically, we use the previous frame as the reference image for block matching; we refer to this as *Simple Concealment*. For better precision when performance constraints allow, we use both the previous and next frames, which we call *Double Concealment*, as shown in Figure 4.

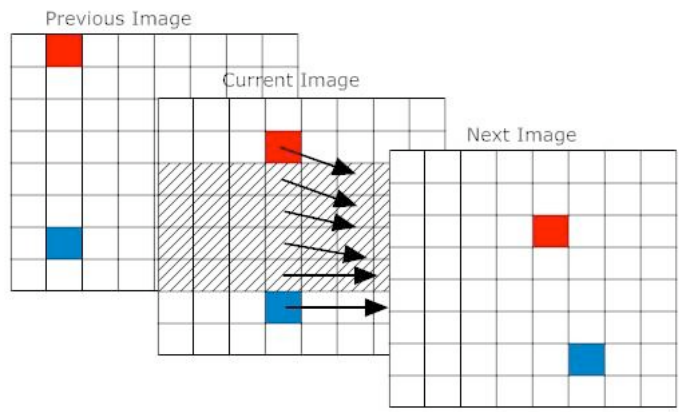

Figure 4: Motion vector estimation with two reference images

4) Post Processing: In the process of concealing missing blocks with those from other images we often introduce an undesired blocking artifact, not unlike that seen with high JPEG compression ratios. By applying a deblocking filter [9] to the concealed section, we are able to significantly reduce this artifact and improve the image quality of concealed regions.

Figure 5: Blocking Artifact

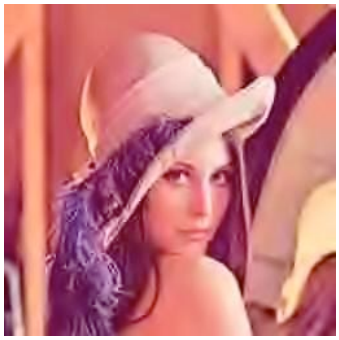

Figure 6: After Deblocking Filter

B. *Multi Resolution and Varying Block Size Concealment*

1) Uniform Regions: In uniform color regions, the process of estimating and compensating blocks' motion is challenged and can yield incorrect results. Issues can appear in situations where objects of interest have a uniform color. Blocks do not appear to be moving because all the blocks around them have the same color. We propose two methods and an improvement of the search method used for block matching in order to deal with this issue.

2) Search Method: We propose an additional search algorithm that reduces the Uniform Region Problem. In a classic search method like the Full Search method (Figure 7), the reference block is indicated in blue and the search window is carried out around this block. In full search, we start the scan with the top left corner and finish with the bottom right corner following the arrow in Figure 7. The color of the reference block is light blue, and block matching is going to return all the light blue blocks as matched. As the figure suggests, there are multiple source candidates.

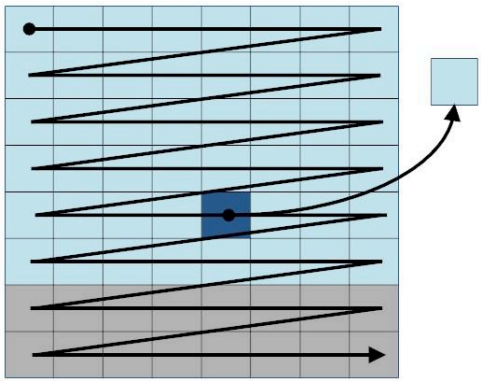

Figure 7: Full search method

In the Full search method the corresponding block is the first or last minimum, but as we can see in Figure 6 these are the farthest blocks from the reference block. We can assume that the movement of a block between two images is not large. We are more likely to find a block near its original position. This suggests it would make sense to start the search in the middle of the search window, in the nearest position to the reference block.

The Spiral search method used in some MPEG-2 implementations is exactly what we need in this case. In the Spiral search method (Figure 8) we start with the nearest block and we finish with the farthest block. With the spiral search method in a uniform color region the returned corresponding block is always the closest block to the reference block. In this work we favored the image quality of the block searching algorithm over its complexity.

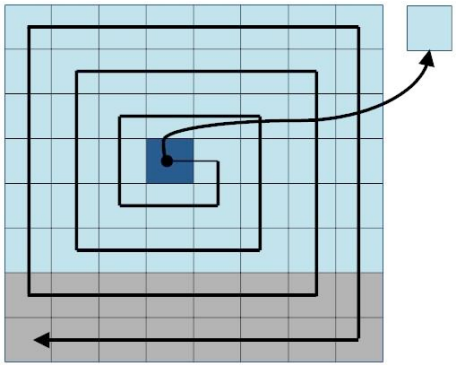

Figure 8: spiral search method

3) Multi Resolution Error Concealment: In order to improve the visual concealment and further mitigate the issues with uniform regions, it helps to perform block matching with high amplitude gradient blocks. Our suggested approach is to use the same image at a lower resolution. In this case the same block size as before contains more information, as shown in Figure 9. The lower the resolution is, the larger the search area in an 8 by 8 block and the more likely we are to get blocks with high amplitude gradient pixels.

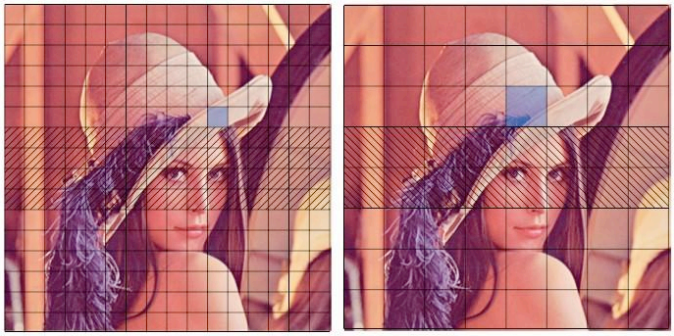

Figure 9: Multi Resolution Error Concealment

4) Varying Block Size Error Concealment: Another method to deal with motion vector estimation inaccuracies in uniform regions is to use bigger blocks, i.e., 16×16 or 32×32, to perform block matching. This method gives the highest results in terms of image quality, but it increases execution time considerably and violates the real-time constraint of our implementation.

III. Real Time Implementation Using M-Jpeg

In order to validate the VPLC algorithm described in the previous section, we propose a real-time PC implementation of our approach using the low complexity, mature and widely

accepted Motion-JPEG (M-JPEG) industry protocol. Leveraging the JPEG still image standard, this format offers a basic error detection scheme for corrupted bit-streams based on the concept of restart markers. We propose an extension to the restart marker concept to better support our VPLC approach while introducing very limited overhead in terms of transport size. Finally, we discuss how the decoding process integrates the video packet loss concealment algorithm and the experimental results we saw.

A. Error Detection and Recovery With JPEG

1) JPEG Restart Markers and a Proposed Extension: The JPEG standard defines a mechanism to partition a compressed stream into independently decodable segments called *restart intervals* [3]. A restart interval is composed of a fixed number of MCUs (Minimum Coded Units) identified in JPEG headers by a DRI (Define Restart Interval) marker. The restart marker allows a decoder to identify a bit-stream segment that can be processed independently of what was decoded up to that point. The error recovery procedure is not explicitly defined by the JPEG standard, but the restart marker combined with the DRI provides the necessary information for a decoder to handle errors in the stream. Below is an example of 16-bit restart markers segmenting compressed data:

```
FFD0 data FFD1 data FFD2 ... FFD7 data
FFD0
```

As shown in the example, the restart marker uses a modulo-8 count of the restart interval. The counter resets to 0 for each new scan, and it can wrap around after or within a scan. Thus, depending on the truncation error rate, it is quite possible to get a situation where the decoder is unable to uniquely identify exactly where, within an image extracted from a video stream, the (x, y) coordinate of an erasure is located. Alternatively, to avoid the issues caused by the counter wrapping around, we could impose the rule that only 8 legacy restart markers are used in each image. For a color YUV 422 VGA image (640×480 resolution), which corresponds to 4800 8×8 blocks or 2400 MCUs, each restart marker would cover 7.5×8 lines of the image. This translates to a very large data loss, which would challenge any error concealment algorithm.

Instead, we suggest an extension of the JPEG Restart Marker syntax by placing an additional byte next to FFDx, which we call the *resync marker*. This *resync marker* wraps around only after reaching 0xFE. We do not use 0xFF for the extra byte because of JPEG syntax restrictions. The sequence becomes:

```
FFD000 data FFD100 data FFD200 data ...
FFD700 data FFD001 data FFD101 ...
```

With this improved scheme, the maximum number of uniquely identifiable restart markers becomes 2039. For a QVGA image we can insert a resync marker every MCU, and in VGA every 2 MCUs, which greatly improves the decoder's ability to detect loss and resynchronize on a truncated bit-stream.

2) Error Detection and Recovery with Resync Markers: While parsing a JPEG bit-stream featuring the proposed resync markers, we first determine if there are any missing resync markers within the image and then calculate the size and location of the erasure. Once all the erasures of a frame have been identified, we reconstruct the truncated bit-stream by adding "blank" compressed data at the location of erasures and converting our resync syntax to that of the legacy restart markers. After this operation, the bit-stream is ready to be decoded by any legacy JPEG decoder, with truncated areas appearing as black regions in the decoded image. The frame is then ready to be processed by the video packet loss concealment algorithm presented in section 2.

B. Experimental Results

To validate our VPLC approach, we first focused on a pure PC software implementation. Various test sequences were collected and, for initial validation, erasures were artificially inserted before invoking our optimized software VPLC implementation. Then, we expanded the validation and benchmarking of our approach by using an integrated video over wireless streaming demonstrator challenged by interference in the ISM band. The test clips and test environment were selected to cover a number of usage scenarios and data content ranging from high global motion to more static scenes, with various levels of detail in the captured videos.

The main metrics collected for all these experiments were the CPU load of the PLC algorithm on a given PC configuration, the recovered frame rate, and the image quality improvement. To measure image quality, in addition to using the typical PSNR (Peak Signal-to-Noise Ratio), we measured the VQM (Video Quality Metric) scores [8]. This metric collects and aggregates various statistical attributes of images (percentage of blurriness/blockiness, percentage of unnatural motion, etc.) that are important to the human perception of image quality to report an overall Mean Opinion Score (MOS). Considering the video PLC problem space, we find this metric significantly more relevant than PSNR alone. The error rate for simulating random erasures was set at 14%, with resync markers introduced every MCU for QVGA images. The PC configuration used was a desktop PC running Windows XP Pro on an Intel Pentium 4 3.2GHz with 1GB of SDRAM.

Table 1 shows a comparison between common VPLC methods (*Block Copy Concealment* [4] [5], *Spatial Concealment* [4] [5] [6]) and our methods (*Simple Concealment, Double Concealment, Multi Resolution, Varying Block Size*) for VPLC. The image quality is improved as our concealment algorithm gets refined and features more CPU-demanding algorithmic blocks. It is intuitive that the higher the image quality, the more CPU time is needed to conceal a frame. As the table suggests, varying the block size during VPLC does not allow our implementation to run in real-time as the execution time exceeds 33ms and was thus excluded from the implementation.

Method	VQM	PSNR (dB)	Time (ms)
Truncated clip	74,8191	16,1353	0
Block copy Concealment	36,2835	29,2143	0,0487013
Spatial Concealment	48,7362	27,7853	0,0974026
Simple concealment	28,4018	31,8446	10,1461039
Double Concealment	24,8869	32,4958	19,6428571
Multi resolution Double	25,4259	35,7346	24,4480519
Varying block size 16x16	16,2172	34,6087	144,074675
Varying block size 32x32	14,0224	35,2812	582,282468

Complexity

Table 1: Image Quality vs. Execution Time (the smaller the VQM score, the higher the image quality)

As the table 2 suggests, in an office environment with significant interference in the ISM band, we have observed that using MJPEG compression, we can easily face close to half the received video frames being corrupted, even when the distance between the sender and receiver is limited (less than a meter). For example, table 2 shows that without our VPLC algorithm, 14fps would be achieved at QVGA resolution. With our VPLC solution, even by dropping frames exceeding 30% erasures (1.4fps on average in this test) we are able to conceal up to ~12 frames while maintaining a high PSNR (35.7dB using the multi resolution scheme) for most scenes, including those featuring a high degree of motion.

Perturbations	Reconstructed FPS	FPS with Errors	FPS with more than 30% Errors	FPS displayed after PLC	FPS displayed before PLC
Office environment	28.76	1.47	0.4	28.36	27.29
Heavy interference + Office environment	25.85	11.72	1.4	24.45	14.12

Table 2: Frame rate improvement using our PLC algorithm and a hardware prototype

IV. Conclusion

In this paper, we presented a new video packet loss concealment (VPLC) technique which is suitable for wireless transmission but independent of the encoding and transport mechanisms used. Video PLC is traditionally handled at the encoder and/or transport level. Our approach is unique in that video PLC is addressed after the decoding process. More specifically, we performed the motion estimation after the image decoding. We used Motion-JPEG and a proposed improvement to the JPEG syntax for error detection. We validated our VPLC algorithm in terms of real-time performance on a PC platform as well as from the standpoint of image quality and frame rate improvement. Frame rate was improved by 70% to 80% on a wireless hardware streaming demonstrator while video quality showed a 100% improvement over a wide range of reference sequences.

This VPLC algorithm is therefore well-suited to the wireless cameras, as it was performed in real-time by the decoding PC, and requires minimal modifications of the MJPEG codec. Observe that this algorithm could also fit the constraints of forthcoming wireless cameras operated on batteries or with green energy sources [1] and forthcoming low complexity encoders [2].

Our work could be further extended to characterize how our approach compares or complements video PLC tools offered by recent encoding techniques such as H.264 (data partitioning, reversible Huffman encoding, flexible macro block ordering) or the various tools offered by JPEG2000 and the recent Chapter 11 focusing on error resiliency.

References

[1] IF Akyildiz, T Melodia, KR Chowdhury, "A survey on wireless multimedia sensor networks", *Computer Networks, Elsevier*, 2007

[2] C. Brites, J. Ascenso, F. Pereira, "Improving Transform Domain Wyner-Ziv Video Coding Performance", *IEEE ICASSP*, 2006

[3] WB Pennebaker, JL Mitchell, *JPEG Still Image Data Compression Standard*, ISBN 0442012721, Van Nostrand Reinhold, New York, 1993.

[4] C Dovrolis, D Tull, P Ramanathan, "Hybrid spatial/temporal loss concealment for packet video", *Proc. 9th International Packet Video Workshop*.

[5] A Raman, M Babu, "A low complexity error concealment scheme for MPEG-4 coded video sequences", *Tenth Annual Symposium on Multimedia Communications*, IEEE Bangalore Section, 2001.

[6] Trista Pei-chun Chen and Tsuhan Chen, "Second-generation error concealment for video transport over error-prone channels", *ETRI Journal*, Carnegie Mellon University, Pittsburgh, 2005.

[7] Nicole S. Love and Chandrika Kamath, "An Empirical Study of Block Matching Techniques for the Detection of Moving Objects", Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore.

[8] ITS, Video Quality Research, ANSI T1.801.03-2003 http://www.its.bldrdoc.gov/n3/video/

[9] International Standard ISO/IEC 14496-2. Information Technology. Coding of Audio-Visual Objects - Part 2: Visual.

Figure 10: Concealment examples – Erroneous image

Figure 11: Concealment examples – Block copy concealment

Figure 12: Concealment examples – Spatial concealment

Figure 13: Concealment examples – Simple concealment

Figure 14: Concealment examples – Double concealment

Figure 15: Concealment examples – Multi resolution concealment

Figure 16: Concealment examples – Varying block size concealment (32x32)