INPAINTING-BASED ERROR CONCEALMENT FOR LOW-DELAY VIDEO COMMUNICATION

Ahmed Aldahdooh, Marcus Barkowsky*, David R. Bull†, and Patrick Le Callet**

*LUNAM Universite de Nantes, IRCCyN UMR CNRS 6597, Nantes, France
Email: firstname.lastname@univ-nantes.fr
†Department of Electrical, Electronic Eng., University of Bristol, BS1 8UB, UK
Email:Dave.Bull@bristol.ac.uk

ABSTRACT

Error concealment (EC) is one of the target applications of inpainting techniques. Some methods combine the estimated lost motion vectors (MVs) with the exemplar-based inpainting technique to recover the lost regions. Due to the erroneous motion vectors that might indicate a moving object as background object and vice versa, these methods are still showing visual artifacts in the recovered regions. In this paper, a concept of motion map that can be easily generated in the decoder side is introduced and it is combined with the exemplar-based inpainting technique. The proposed method introduces an adaptive search window size that trades-off the quality and complexity. Moreover, an optional blending technique is proposed to limit the spatio-temporal artifacts. Experiments show that the proposed method improves the visual quality with 5dB on average relative to the state-of-the-art inpainting-based EC method.

***Index Terms*—** Error concealment, inpainting, video communication

1. INTRODUCTION

Ensuring error resilience has become more complex in recent video coding standards due to the increased complexity of their prediction processes [1]. Conventional error resilience techniques [2] can be categorized into three main classes depending on where the processing is performed: forward-error-concealment (encoding side), post-processing error concealment (decoder side), and interactive error concealment (encoder and decoder sides). In this paper we focus on error concealment by post-processing. In post-processing techniques, the decoder utilizes the spatial and/or temporal redundancies to reconstruct the damaged/lost area in a video frame. Spatial techniques [3, 4], utilize available surrounding pixels to reconstruct the missing pixels. They are not efficient for large areas, non-constant areas, and in terms of complexity. They usually reconstruct the texture but not the structure. The work in [5] is an extension of [4] in which a spatio-temporal selective extrapolation strategy is used to reconstruct the missing area. Temporal techniques use available motion information to predict the missing motion vectors (MVs), for instance, by interpolating [6] or by selecting the MV that minimizes the side match distortion [7]. Despite providing information about whether the current area is moving or not, this technique is efficient only for low-motion and smooth sequences and for small areas since the precision of predicted MVs is not guaranteed. Thus, the structure (of copied data) is reconstructed but not the texture.

The target of any error concealment algorithms is twofold: reconstructing a satisfying reconstruction of a lost area and reducing the miss-match between the encoded and the reconstructed blocks which yields reducing the error propagation effect. To achieve that we need to reconstruct the texture and the structure of a missing area and that can be done using inpainting techniques. A review of inpainting techniques can be found in [8]. In this paper we focus on exemplar-based inpainting in which each lost patch is reconstructed by copying the best match from the known area. Specifically, inpainting algorithms have many target applications and in this work, we are interested in error concealment as a target application.

Inpainting-based error concealment algorithms are introduced in [9, 10]. The algorithms have three main steps: inpainting the moving foreground object, inpainting the stationary background temporally and spatially as in [11]. In this paper, a modified version of [10] is introduced with the following contributions:

- The quality of the results depend on the input M_c which indicates whether a pixel p is moving $M_c(p) = 1$ or not $M_c(p) = 0$. The quality also depends on the strategy that replaces the simple copy strategy of the best patch match by other strategies like LLE [12] and NMF [13]. The latter factor is investigated in [14] and it is shown that the performance of the inpainting algorithm is improved. In this paper we will investigate the former factor by introducing a concept of motion map M_c that includes the predicted motion vectors M_{mv}, the pixel-based motion intensity M_{pi} and the motion vector of interests (MVI) that relate to camera motion M_{cm}. It will be shown that the proposed motion map will significantly improve the performance of the inpainting.
- The algorithm in [10] works on the sequence level. The process does not start once the error occurs, but it waits

This work is supported by the Marie Skłodowska-Curie under the PROVISION (PeRceptually Optimised VIdeo CompresSION) project bearing Grant Number 608231 and Call Identifier: FP7-PEOPLE-2013-ITN.

until more frames are available. Then, it first searches for the highest priority frame to start with. That means that there may occur more than one error and it also means that the concealment might be performed out of temporal order. This strategy is not practical for some video applications, since, in video communication, once the error is detected in a frame, especially those that are used as reference for coming frames, it must be concealed before the decoder continues. In this paper, the error concealment strategy is optimized for low-delay configuration.

- Using a full search strategy or fixed window size is not efficient in terms of complexity and quality respectively. In this paper an adaptive search window size for temporal and spatial inpainting is introduced.
- Trying to reduce the spatio-temporal artifacts, a simple blending strategy is employed using Poisson blending [15] with the proposed mask strategy.

The rest of this paper is organized as follows: in Section 2, the proposed algorithm is demonstrated. The experimental results are shown in Section 3. Finally, we sum up with the conclusions in Section 4.

2. INPAINTING-BASED ERROR CONCEALMENT STRATEGY

2.1. Motion Map

In this subsection, the concept of the motion map is illustrated. The motion map M_c is computed as: $M_c = M_{mv} \vee M_{pi} \vee M_{cm}$, where $(\vee)$ is the logical OR operator.

2.1.1. Motion Vector Map

As in [10], the lost MVs are predicted using Bilinear Motion Field Interpolation (BMFI) [6]. MV components V_x and V_y are threshold to determine whether the pixel p belongs to a moving object $M_{mv}(p) = 1$ or not $M_{mv}(p) = 0$. In this paper, a threshold of 1 is used such that $M_{mv}(p) = 1$ if $V_x(p)$ or $V_y(p) > 1$.

2.1.2. Motion Intensity Map

In order to measure the pixel-based motion intensity, the pixel change ratio map (PCRM) [16] strategy is used. This algorithm assumes that that a high intensity of motion yields a large change in pixel intensities over a video shot. In this paper the shot is represented by up to 8 previous frames. $M_{pi}(p) = 1$ if $PCRM(p) > th_i$ and $M_{pi}(p) = 0$ if $PCRM(p) \leq th_i$, where th_i is motion intensity threshold. In this paper it is set to 0.25 to exclude the pixels that have low intensity changes over the video shot.

2.1.3. Camera Motion Map

In [17], the MVs of up to 8 previous frames are analyzed to obtain motion vectors of interest (MVI). MVIs identify the spatial region where the motion information has a direct relationship with the camera movements [17]. In this paper, the MVI of each frame is computed and assigned to M_{cm}.

2.2. Inpainting Process

In this section, the three main steps of inpainting-based error concealment method will be illustrated. Let some blocks be lost in frame F at time t (F_t). The frame F_t has known/source area Φ and lost/target area Ω to be filled. The fill front $\delta\Omega$ is defined as a contour that separates known and lost areas. A key elements of exemplar-based inpainting algorithms are the filling order (or patch priority) of lost area and the texture synthesis, i.e. finding the best match of the current processed patch. These two key elements will be illustrated in the following steps.

2.2.1. Inpainting Moving Objects

Once the error occurs, the error concealment (EC) process starts filling the lost area patch by patch using the following steps: for each pixel p of the fill front $\delta\Omega$, compute the patch Ψ_p priority $P(p) = C(p)D(p)$ as in [9, 10], where the $C(p)$ is the confidence term and $D(p)$ is the data term. The confidence term $C(p) = \frac{\sum_{q \in \Psi_p \cap (F-\Omega)} C(q)}{|\Psi_p|}$ represents the ratio of known data to the patch area. While the data term $D(p) = \frac{|\nabla M_c^{\perp} . n_p|}{\alpha}$ gives more priority to the patches that have orthogonal motion direction ($\nabla M_c^{\perp}$) to the fill front $\delta\Omega$. n_p is the normal to the fill front $\delta\Omega$ at p, and α is a normalizing constant ($\alpha = 255$).
The next step now is to synthesize the patch that has the highest priority $\Psi_{\hat{p}}$, where $\hat{p} = \arg\max_{p \in \delta\Omega} P(p)$. The block matching algorithm is used to find the best matching patch Ψ_q of the known part of $\Psi_{\hat{p}}$ within a search window w in the previous/reference frame using the sum of squared differences (SSD) of color and MV components (R, G, B, V_x, V_y) of known pixels of $\Psi_{\hat{p}}$ and all candidates in the search process. Where w is equal to the double of the largest value of MV components of the surrounding area. The pixels values of Ψ_q are copied to the co-located unknown pixels of $\Psi_{\hat{p}}$.
The aforementioned steps are repeated until all moving and damaged pixels are concealed, the confidence term of the copied pixels $\Psi_{\hat{p}}$ is updated and the motion map M_c is also updated by copying the M_c of Ψ_q to the $\Psi_{\hat{p}}$.

2.2.2. Inpainting The Stationary Background Temporally

In the previous step, all the moving pixels are concealed. In this section, the steps for inpainting the stationary background temporally are demonstrated.
Following almost the same process of filling-in the moving pixels, the priority term, $D(p) = C(p)D(p)$, for each patch centered at p, where $p \in \delta\Omega$ needs to be computed. First, the confidence term of each pixel that is either damaged or moving is set to $C(p) = 0$ and $C(p) = 1$ otherwise. The data term is defined to measure the amount of available temporal information ($M_t(p)$) in the up to 8 previous frames. Hence,

the data term is defined [9] as $D(p) = \frac{\sum_{p \in \delta\Omega, t=-\delta n \ldots 0} M_t(p)}{\beta}$, where $M_t(p)$ is 0 if p is either a moving or a damaged pixel, else $M_t(p)$ is 1. The time index t indicates the relative position of up to 8 previous frames from $t = 0$ (the current frame) and β is a normalizing factor that represents the number of previous frames used to compute the data term.
The next step is to copy the patch Ψ_q from the nearest frame to the unknown part of the patch $\Psi_{\hat{p}}$ that has the highest priority. Then, the confidence terms of previously damaged pixels are updated. The process iterates until no more temporal information needs copying, i.e. $D(p) = 0, \forall p \in \delta\Omega$. That means that the remaining pixels of the stationary background have to be inpainted spatially.

2.2.3. *Inpainting The Stationary Background Spatially*

In this section, the steps for spatially inpainting the remaining pixels of the stationary background will be demonstrated. This process follows the algorithm that is described in [11] exactly except for the search window size w_p. The search window size is adaptively changed for each process patch as follows. First, the minimum and maximum allowed window size is computed as $min_w = 2 * patchSize$, and $max_w = 2 * max(d)$, where $d(p)$ is the nearest distance between unknown pixels to the known pixels. Second, for each patch the search window size is set to $w_p = max(min_w, d(p) * \frac{max_w}{min_w})$. This adaptive procedure is to trade-off the quality and the complexity of the spatial inpainting.

2.3. Blending Step

In [18], the Poisson blending [15] is used to reduce the artifacts of the inpainting process and it was shown that it improves the performance since the inpainting algorithm is based on frames registration. In this work, we first demonstrate the blending of the inpainted frame temporally using the motion-compensated frame of the lost frame and using the lost area as a mask. It is observed that this process will not improve the quality if not making it worse since the structure of the lost area is not respected. This is because of the predicted motion vectors. Therefore, the blending mask M_{blend} is changed to blend only the pixels that are far enough from the edges d_{edge} and have a low motion magnitude MV_{mag}. Hence, the pixel will be blended if $M_{blend}(p) = \frac{d_{edge}}{MV_{mag}} > th_{blend}$. In this work the th_{blend} is set to 2. It was observed that this blending mask gives better results than the former method. Unfortunately, in general, this blending process is not improving the inpainted frame as assumed since the proposed motion map maintains the structure and the texture of the inpainted frame.

3. EXPERIMENTAL RESULTS

The proposed algorithms and other state-of-the-art algorithms [6, 10] are implemented using MATLAB. Spatial-only method [11] is compared in [10] and for the sake of complexity, it is not tested in this work. Eight 1280×720 video sequences are used in the experiment, Figure 1. In each frame, 5%, 10%, and 20% of the 64×64 blocks are randomly lost and inpainted using different error concealment methods. For the sake of fair comparison, each source share the same error pattern. The patch size should be greater than the thickest structure (e.g., edges) in the source region [11]. In this work, it is set to 9 for all sequences. Figures 2, 3, and 4 show the results of the recovered areas from sequence 1. It can be noticed that the proposed method improves the visual quality of the recovered areas. Table 1 shows the performance of the different methods, in terms of difference of quality (PSNR). Method of [10] is used as reference of comparison. The results are the average of the first 45 frames in the video shot. It can be noticed that the proposed method achieves 1 to 6 dB of quality improvements depending on the video shot characteristics. In terms of complexity, the proposed algorithm is faster than the algorithm in [10] by a factor of two on average.

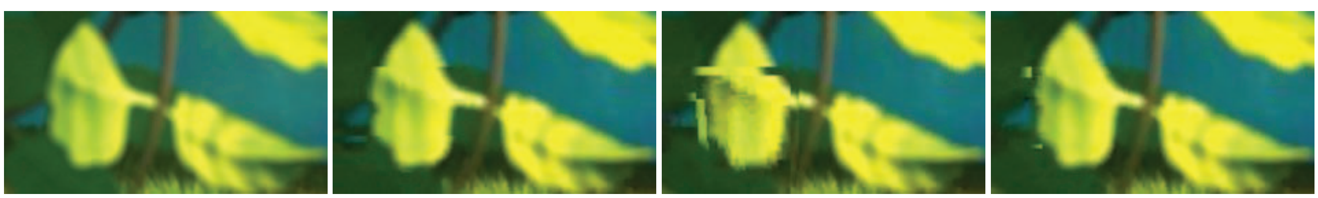
Table 1: Quality performance of the different EC methods.

Sequences	%lost	Δ PSNR (dB) compared to [10]		
		[6]	Proposed	Proposed with blending
Seq. 1	5	-0.2	2.0	2.0
	10	-1.1	2	1.9
	20	-2.4	1.5	1.4
Seq. 2	5	-0.6	6.0	5.6
	10	-0.7	6.0	5.7
	20	-0.7	6.3	6.2
Seq. 3	5	-8.2	1.0	0.3
	10	-9.7	0.6	-0.1
	20	-9.5	0.9	0.2
Seq. 4	5	-13.3	1.4	0.8
	10	-13.5	1.2	0.9
	20	-13.7	1.1	0.7
Seq. 5	5	-3.7	6.3	6.0
	10	-4.4	6.3	6.0
	20	-5.1	5.3	4.9
Seq. 6	5	-8.7	5.1	5.1
	10	-8.3	6.2	6.1
	20	-8.7	5.9	5.9
Seq. 7	5	-6.9	15.6	12.2
	10	-6.8	15.4	11.7
	20	-6.9	14.3	10.3
Seq. 8	5	-6.8	4.7	4.4
	10	-7.2	4.9	4.3
	20	-7.1	4.5	4.1
average	5	-6.0	5.3	4.5
	10	-6.5	5.3	4.6
	20	-6.8	5.0	4.2

Fig. 1: Thumbnails of the eight 1280x720 Video Sequences that are used in the experiment

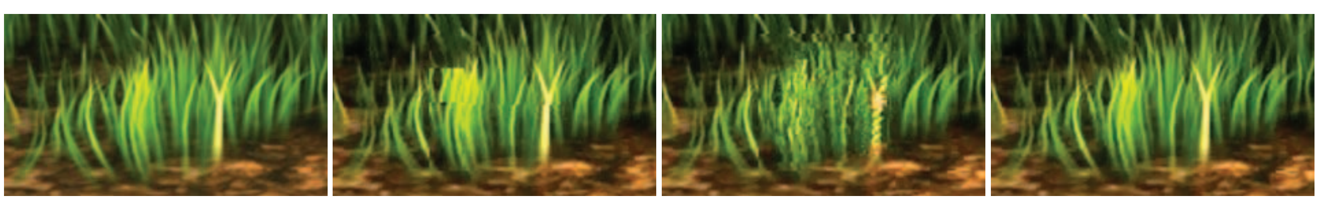
(a) original (b) motion comp. [6] (c) inpaint [10] (d) inpaint (proposed)

Fig. 2: Example 1: Comparison of different error concealment methods for 10% of lost of sequence 1 (pink rectangle).



(a) original (b) motion comp. [6] (c) inpaint [10] (d) inpaint (proposed)

Fig. 3: Example 2: Comparison of different error concealment methods for 10% of lost of sequence 1 (yellow rectangle).



(a) original (b) motion comp. [6] (c) inpaint [10] (d) inpaint (proposed)

Fig. 4: Example 3: Comparison of different error concealment methods for 10% of lost of sequence 1 (white rectangle).

4. CONCLUSION

This paper proposed a modified version of the inpainting-based error concealment [10] by introducing several enhancements. First, the concept of the motion map is introduced. Second, the adaptive search window size plays an important role to trade-off between the quality and the complexity. Third, the method is customized for low delay video communication systems. Finally, an optional step for blending the recovered areas is introduced. The experimental results show that the proposed methods improve the visual quality and hence, this will reduce the error propagation. More investigations are required to know when the blending technique might be used. Moreover, running the proposed method in a real network environment and real coding environment is planned as future work.

5. REFERENCES

[1] B. Oztas, M. T. Pourazad, P. Nasiopoulos, and V. C. M. Leung, "A study on the hevc performance over lossy networks," in *Electronics, Circuits and Systems (ICECS), 2012 19th IEEE International Conference on*, Dec 2012, pp. 785–788.

[2] Yao Wang, Jörn Ostermann, and Ya-Qin Zhang, *Video processing and communications*, vol. 5, Prentice Hall Upper Saddle River, 2002.

[3] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Transactions on Communications*, vol. 41, no. 10, pp. 1544–1551, Oct 1993.

[4] K. Meisinger and A. Kaup, "Spatial error concealment of corrupted image data using frequency selective extrapolation," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, May 2004, vol. 3, pp. iii–209–12 vol.3.

[5] K. Meisinger and A. Kaup, "Spatiotemporal selective extrapolation for 3-d signals and its applications in video communications," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2348–2360, Sept 2007.

[6] M. E. Al-Mualla, N. Canagarajah, and D. R. Bull, "Error concealment using motion field interpolation," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, Oct 1998, pp. 512–516 vol.3.

[7] W. M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, April 1993, vol. 5, pp. 417–420 vol.5.

[8] C. Guillemot and O. Le Meur, "Image inpainting : Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, Jan 2014.

[9] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting of occluding and occluded objects," in *IEEE International Conference on Image Processing 2005*, Sept 2005, vol. 2, pp. II–69–72.

[10] M. Ebdelli, O. Le Meur, and C. Guillemot, "Loss concealment based on video inpainting for robust video communication," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, Aug 2012, pp. 1910–1914.

[11] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, Sept 2004.

[12] Sam T. Roweis and Lawrence K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[13] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *In NIPS*. 2000, pp. 556–562, MIT Press.

[14] M. Ebdelli, C. Guillemot, and O. Le Meur, "Examplar-based video inpainting with motion-compensated neighbor embedding," in *2012 19th IEEE International Conference on Image Processing*, Sept 2012, pp. 1737–1740.

[15] Patrick Pérez, Michel Gangnet, and Andrew Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, July 2003.

[16] Haoran Yi, Deepu Rajan, and Liang-Tien Chia, "A new motion histogram to index motion content in video segments," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1221 – 1231, 2005.

[17] M. A. Hasan, M. Xu, X. He, and C. Xu, "Camhid: Camera motion histogram descriptor and its application to cinematographic shot classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 10, pp. 1682–1695, Oct 2014.

[18] M. Ebdelli, O. Le Meur, and C. Guillemot, "Video inpainting with short-term windows: Application to object removal and error concealment," *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3034–3047, Oct 2015.