ERROR CONCEALMENT VIA 3-MODE TENSOR APPROXIMATION

Dzung T. Nguyen, Minh D. Dao, Trac D. Tran

Department of Electrical and Computer Engineering,
The Johns Hopkins University,
Baltimore, MD 21218 USA
{dzungnguyen, minh.dao, trac}@jhu.edu

ABSTRACT

This paper presents a novel video error concealment method, which is essentially a combination of non-local grouping of image patches and low-rank tensor approximation. The proposed method, though does not require the knowledge of Motion Vectors (MVs) as in traditional video concealment techniques such as Boundary Matching Algorithm (BMA) and its derivations, gives striking results in restoration of sequences, especially in the challenging cases when the error rate is high and/or key frames are not available. Our proposed framework can also be customized to deal with single images as well (for example, in image inpainting tasks).

Index Terms— Error concealment, inpainting, tensor, low-rank approximation, BMA

1. INTRODUCTION

Data recovery is an essential process in any modern image/video communication system. Visual data encoders often work in block-based fashion, hence any lost in data transmission or damage in storage media results in corruptions of blocks or group-of-blocks. Numerous techniques had been proposed for data restoration. Different methods require different involvement of the source coder, the sender/receiver, the network or combinations of those entities [1]. Similar to recent works in video concealment [2, 3, 4], in this paper we focus on the development of source coder-independent receiver-based techniques: we only use information available at the receiver to recover lost data. This can also be classified as a post-processing technique that can be employed to recover left-over errors which other schemes failed to deal with.

Video coders often use block motion compensation techniques to eliminate redundancy in video sequences. In most cases (predictive frames), the actual information sent to the decoder are Motion Vectors (MVs) and the corresponding block residues. Therefore it is natural to exploit MVs for concealment. At an erroneous block (that loses its MV), the Boundary Matching Algorithm (BMA) [2] tries to recover the MV by selecting the best candidate from the set of neighbouring MVs and the zero MV, in terms of minimum total variation of pixels values at the block's boundary. Variations of BMA [4, 3] focus either on finding a better MV or developing more sophisticated boundary matching criteria. Such methods have the advantage of fairly low complexity, but on the other hand restrict themselves to one MV and only one reference frame in the search for best recovering data.

We propose an error concealment method that does not rely on recovering MV, therefore are also liberated from one reference frame limitation as well as the Macro-Block (MB) grid decided by the encoder. We are free to redefine the block size, the grid and its displacement in each corrupted frame. The number of reference frames is also a free parameter. Infact, more reference frames typically yield better recovery results and this number can be increased as much as the computational capacity allows. In a nutshell, the proposed method groups image patches in the corrupted frame with similar patches searched from the entire reference frames to form 3-mode tensors with missing data. This approach is inspired by the success of denoising methods such as non-local mean or 3-D transform-domain collaborative filtering [5, 6]. Those tensors are then approximated using an algorithm derived from tensor n-rank (Tucker) decomposition [7] to exploit the low-rank nature of the common patches in the 3rd dimension.

The pioneering work on visual data restoration from a tensor completion viewpoint is presented in [8]. In that paper, the authors follow a different path by relying on tensor canonical rank, rather than the n-rank. The paper shows some impressive preliminary results on image/video restoration, but it implicitly assumes that data must have global low rank structure. In general, one can not make such restrictive assumption about natural images and video sequence. In our work, we explicitly show how to form tensors with certain low-rank structure from patches of visual data, and exploit that structure in our proposed tensor approximation algorithm customized for this type of tensors.

Our paper is organized as follows: the next section explains the key elements of our concealment framework. Sec-

tion 3 shows pseudo code of the algorithm while Section 4 presents some experimental results in video concealment and image inpainting. Finally, Section 5 concludes and discusses some ideas for future work.

2. TENSOR CONSTRUCTION AND CONCEALMENT

Before going into the details, let us adopt some important notations in tensor algebra from [7]. An N$^{\text{th}}$-order (or N-mode) tensor $\boldsymbol{\mathcal{X}}$ (in calligraphic letter) is an N-dimensional array $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$. Unfolded matrix of $\boldsymbol{\mathcal{X}}$ along the n$^{\text{th}}$ mode is denoted $\boldsymbol{X}_{(n)}$. A tensor can be decomposed to a sum of rank-1 tensors (canonical decomposition) or a product of a core tensor with matrices corresponding to its modes (Tucker decomposition). In the later form, a tensor is represented as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \ldots \times_N \boldsymbol{A}^{(N)} \tag{1}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times \ldots R_N}$ is the core tensor, and $\boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ is a matrix whose columns are eigen vectors in mode-n. The mode-n product (operator $\times_n$), between a tensor $\boldsymbol{\mathcal{X}}$ and a matrix $\boldsymbol{U} \in \mathbb{R}^{R_n \times I_n}$ is defined elementwise as $(\boldsymbol{\mathcal{X}} \times_n \boldsymbol{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \ldots i_n \ldots i_N} u_{j i_n}$.

2.1. Tensor Construction via Block Matching

We slice each corrupted frame using a new MB grid, which is shifted half a block size in each spatial dimension. Each missing MB is hence divided into 4 quarters (subblocks), and all missing subblocks are put in a queue. The order in which subblocks are concealed is important. A smart queueing mechanism is implemented where missing subblocks with more number of clean neighbors and closer to the missing boundary are prioritized. As a result, subblocks in a missing area are processed from its boundary towards the center.

Once a missing subblock is selected, an image patch (MB $\boldsymbol{P}^0$) of size $N \times N$ is formed that contains the missing quarter and its clean/concealed neighbors (Figure 1). This MB is used to search for similar MBs ($\boldsymbol{P}^i$) in the entire stock of R reference frames. Figure 2 illustrates the grouping process. Various criteria of MB similarity had been consider in [5, 6]. We chose Block Matching criteria because of its simplicity.

$$S_i = \left\| \boldsymbol{P}^0_\Omega - \boldsymbol{P}^i_\Omega \right\|_1 \tag{2}$$

where Ω contains the indices of clean/recovered pixels in $\boldsymbol{P}^0$ (we will use $\overline{\Omega}$ later for indices of missing pixels).

$\boldsymbol{P}^0$ and several MBs with best matching scores S_i are grouped into a 3-mode tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{N \times N \times K}$ where $\boldsymbol{P}^0$ is on top. Those $\boldsymbol{P}^i$ should be scaled to have the same l_2-norm as $\boldsymbol{P}^0$ (at indices Ω). A good practical choise of K is $K = R + 1$ (if we assume one good match is found in each reference frame).

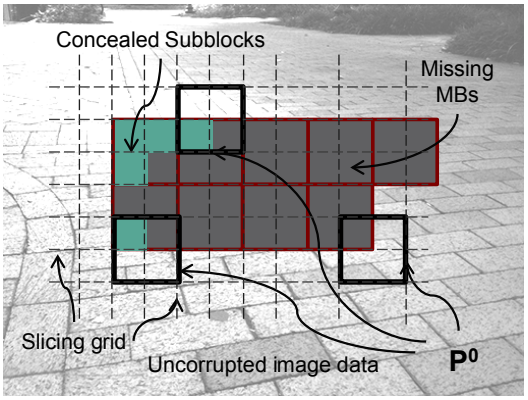


Fig. 1. New MB grid in corrupted frame and $\boldsymbol{P}^0$ selection

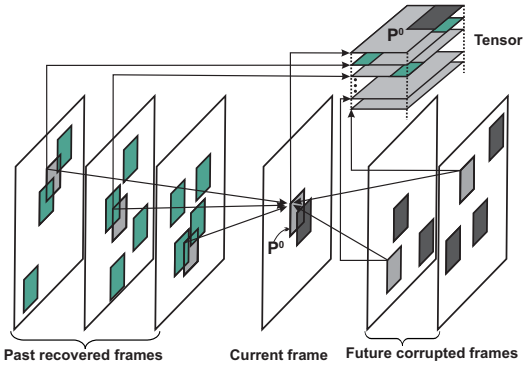


Fig. 2. Tensor build by grouping similiar MBs

2.2. Tensor n-rank Approximation

Ideally, if the video sequence follows the block-translational model, then $\boldsymbol{\mathcal{X}}$ is formed by K identical patches, therefore

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}_l + \boldsymbol{\mathcal{E}} = \boldsymbol{P} \times_3 \mathbf{1} + \boldsymbol{\mathcal{E}} \tag{3}$$

where $\boldsymbol{\mathcal{X}}_l$ has mode-3 rank equals 1 (and hence can be represented as an outer product between an image patch and a constant vector $\mathbf{1}$), and $\boldsymbol{\mathcal{E}}$ is an error tensor (which is in general sparse). If $\boldsymbol{\mathcal{X}}_l$ is factorized using the Tucker decomposition

$$\boldsymbol{\mathcal{X}}_l = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \boldsymbol{A}^{(3)} \tag{4}$$

then we can equate (3) and (4) to arrive at

$$\boldsymbol{P} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \text{ and } \boldsymbol{A}^{(3)} = \mathbf{1}. \tag{5}$$

In practice, patches in $\boldsymbol{\mathcal{X}}$ (being grouped from different frames) are slightly different, or the actual number of identical patches may be less than K. Therefore, $\boldsymbol{\mathcal{X}}_l$ may have mode-3 rank larger than 1, but we hypothesize that it should still have a low-rank structure.

These observations suggest us to use the n-rank decomposition technique to approximate $\boldsymbol{\mathcal{X}}_l$. To achieve this, we employ the well-known High-Order SVD (HOSVD) [9]

Algorithm 1 Concealment of an image patch w/ missing data

1. Form $\boldsymbol{\mathcal{X}}$ from $\boldsymbol{P}^{0,\ldots,K-1}$ using Block Matching criteria
2. $(\boldsymbol{\mathcal{X}}(:,:,1))_{\overline{\Omega}} = \left(\frac{1}{K-1}\sum_{i=1}^{K-1}\boldsymbol{P}^i\right)_{\overline{\Omega}}$
3. Choose mode ranks $\{\mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3\}$, tolerance σ; Initialize $\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}$
4. $\boldsymbol{A}^{(3)}(:,1) = [1,\ldots,1]^T/K$
5. for n = 1,2,3
   $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_1 \ldots \times_{n-1} \boldsymbol{A}^{(n-1)T} \times_{n+1} \boldsymbol{A}^{(n+1)T} \ldots$
   $\boldsymbol{Y}_n \leftarrow$ unfold $\boldsymbol{\mathcal{Y}}$ in mode n
   $\boldsymbol{A}^{(n)} \leftarrow$ first R_n principal component of $\boldsymbol{Y}_n$
   end
6. $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^{(1)T} \times_2 \boldsymbol{A}^{(2)T} \times_3 \boldsymbol{A}^{(3)T}$
7. $\boldsymbol{\mathcal{X}}_l = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \boldsymbol{A}^{(3)}$
8. If $\|\boldsymbol{\mathcal{X}}_l - \boldsymbol{\mathcal{X}}\|_F \leq \sigma$ STOP, otherwise return to Step 4.
9. Recover missing area in $\boldsymbol{P}^0$: $\left(\boldsymbol{P}^0\right)_{\overline{\Omega}} = (\boldsymbol{\mathcal{X}}_l(:,:,1))_{\overline{\Omega}}$

while enforcing the most important eigenvector in mode-3 to the constant vector $\mathbf{1}$ (normalized to unit l_2-norm) in each HOSVD's iteration.

3. PROPOSED ALGORITHM

After the tensor $\boldsymbol{\mathcal{X}}$ is formed, the missing area $\left(\boldsymbol{P}^0\right)_{\overline{\Omega}}$ is firstly filled with the mean area from similar patches. This is in fact a good starting estimate of the original data and helps reduce the energy of the noise $\boldsymbol{\mathcal{E}}$ significantly compared to approximating $\boldsymbol{\mathcal{X}}$ directly with a zero (missing) subblock in its $\boldsymbol{P}^0$ patch.

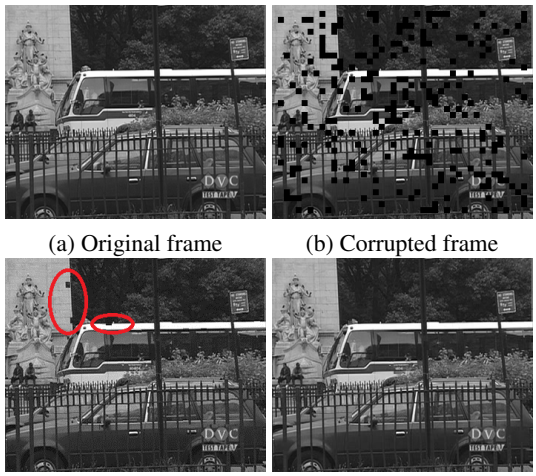
The algorithm then finds the best rank-(R_1, R_2, R_3) approximation $\boldsymbol{\mathcal{X}}_l$ of $\boldsymbol{\mathcal{X}}$ in a process call Alternating Least Square (ALS) [7, 9]. In short, this is an iterative process that tries to solve for one subset of free parameters ($\boldsymbol{A}^{(i)}$ i=1,2,3 or $\boldsymbol{\mathcal{G}}$) at a time while the others being fixed. At each iteration, the first column of $\boldsymbol{A}^{(3)}$ is enforced to the constant norm-1 vector to guide the optimization to converge to the desired low-rank estimate. Since we only assume that the mode-3 rank of $\boldsymbol{\mathcal{X}}_l$ is low, R_3 should be set quite low (3 in practice), while (R_1, R_2) can be as large as N. Algorithm 1 elaborates all the steps of the proposed concealment algorithm in details.

4. EXPERIMENTS

This section illustrates the validity of the theory via several experiments on video error concealment. In the first one, the Bus CIF sequence is corrupted randomly block-wise at a missing rate of 15%. The second one deals with missing slices in a single frame of Foreman CIF sequence, with error rate at 50%. Both are compared with standard BMA method. Finally, an experiment on object removal demonstrates the applicability of our proposed framework to image inpainting as well.

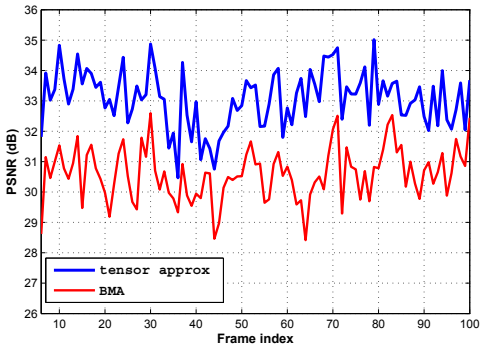
4.1. Corrupted sequence without key frames

Experiment is performed on the first 100 frames of Bus CIF sequence. All frames are quantized in the DCT domain with step size 8, therefore uncorrupted frames at the decoder would have the average PSNR of 42dB. Both BMA and our method are implemented in Matlab. At each frame, MVs refer to previous frame and BMA is allowed to use previously recovered frame as reference. Our method uses 5 nearest previously recovered frames and 5 corrupted future frames in the tensor building step. MB size (and missing block size) is 8×8. Overall, our method shows a 2.47 dB PSNR improvement over BMA on average. The PSNR curve is shown in Fig. 3(e), while Fig. 3(a,b,c,d) shows the visual results at frame #21. The red ovals indicate areas where BMA's failures are obviously visible.



(a) Original frame (b) Corrupted frame

(c) BMA PSNR = 29.19 dB (d) Tensor approx. PSNR = 33.05 dB



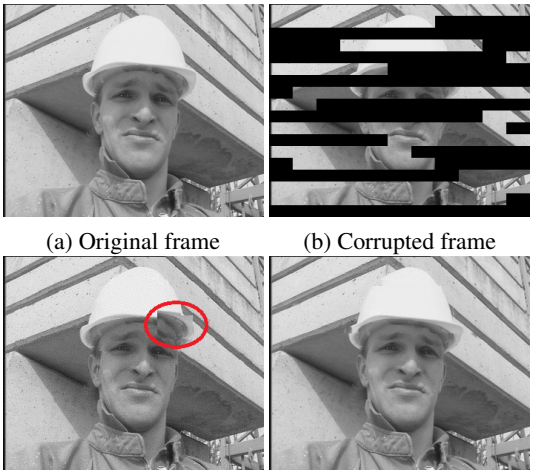
(e) PSNR comparison

Fig. 3. Recovery of Bus CIF sequence, with 15% random block missing in every frame without key frames.

4.2. High error rate in an individual frame

This experiment attack the slice-missing situation. A frame is divided into slices (each slice contains 18 MBs) and 50%

number of slices are missing at random. BMA is using clean previous frame that is refered to by MVs, while tensor approximation method uses 2 frames before and 2 frames after (all clean) for tensor building. Blocksize is 16×16. The restoration visual quality and PSNRs are shown in Fig. 4.



(a) Original frame (b) Corrupted frame

(c) BMA PSNR = 24.68 dB (d) Tensor approx. PSNR = 33.24 dB

Fig. 4. Recovery of frame #37 in Foreman CIF sequence with 50% random slice corruption.

4.3. Image inpainting

(a) Objects selection (b) Healing Brush tool (c) Tensor approx.

Fig. 5. Object removal experiment.

In this inpainting experiment, the objects (in red contours in Fig. 5(a)) are selected in Adobe's Photoshop and feeded as 'missing' maps to our algorithm. The image itself is used as the only reference frame for tensor building. The image is sliced using an 8×8 grid. The result is compared with output from the Healing Brush Tool in Photoshop CS4. There is no ground-truth for objective PSNR calculation, but our result is visually very competitive.

5. FUTURE WORK

In this paper, we propose a novel successful method for error concealment/inpainting using tensor approximation. There are several aspects that we plan to explore in the near future.

Firstly, the issues of how to select the best block size (according to frame resolution) or how to build the best wrapping MB around missing area should be investigated. More robust tensor approximation technique will be developed to exploit the sparse nature of the error.

To contrast our technique with BMA and its derivations, in our experiment we have ignored MVs and any smooth boundary constraints. Adopting those methods in the initialization stage and incorporating these constraints into the low-rank tensor approximation will certainly improve our performance.

6. REFERENCES

[1] B.W. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the internet," in *Multimedia Software Engineering, Proc. Int. Symposium on*, 2000, pp. 17–24.

[2] Y. Wang, M.M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Image Processing, Proc. Int. Conf. on*, 2002, vol. 2, pp. 729–732.

[3] Y. Chen, Y. Hu, O.C. Au, H. Li, and C.W. Chen, "Video error concealment using Spatio-Temporal boundary matching and partial differential equation," *Multimedia, IEEE Trans. on*, vol. 10, no. 1, pp. 2–15, 2008.

[4] W. Lie and Z. Gao, "Video error concealment by integrating greedy suboptimization and kalman filtering techniques," *Circuits and Systems for Video Technology, IEEE Trans. on*, vol. 16, no. 8, pp. 982–992, 2006.

[5] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, Jan. 2005.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D Transform-Domain collaborative filtering," *Image Processing, IEEE Trans. on*, vol. 16, no. 8, pp. 2080–2095, 2007.

[7] T. G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, Sept. 2009.

[8] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *Computer Vision, IEEE 12th Int. Conf. on*, 2009, pp. 2114–2121.

[9] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, Mar. 2000.