Sequential Error Concealment for Video/Images by Weighted Template Matching

Ján Koloda[†], Jan Østergaard[‡], Søren H. Jensen[‡]

Antonio M. Peinado[†] and Victoria Sanchez[†]

[†]Dpt. Signal Theory, Telematics and Communications
Universidad de Granada, Spain

[‡]Multimedia Information and Signal Processing
Dpt. of Electronic Systems
Aalborg University, Denmark

Abstract

In this paper we propose a novel spatial error concealment algorithm for video and images based on convex optimization. Block-based coding schemes in packet loss environment are considered. Missing macroblocks are sequentially reconstructed by filling them with a weighted set of templates extracted from the available neighbourhood. Moreover, a fast approximation of the optimization method is proposed. The technique produces high quality reconstructions that outperforms the state-of-the-art algorithms both in terms of PSNR and MS-SSIM.

1 Introduction

Block-based video coding standards, such as MPEG-4 or H.264/AVC, are widely used in recent multimedia applications. Video signals are split into macroblocks that are coded using inter- or intraframe prediction. Quantization is carried out in the DCT domain and lossless arithmetic compression is applied [1]. This leads to low distortions at moderate bitrates. However, achieving high quality reception is a challenging task since data streams are usually transmitted over error-prone channels.

The H.264/AVC standard has introduced several error resilience tools, such as arbitrary slice order or flexible macroblock ordering. Macroblocks within a frame are split into one or more slices. A slice forms the payload of a network abstraction layer unit (NALU), which is a data sequence that can be decoded independently [1]. Video streams are packetized by NALUs so the loss of a packet would lead to the loss of, at least, one macroblock.

Error concealment (EC) techniques form a very challenging field, since QoS is of utmost importance for the users. In many cases, retransmission of lost data is not possible due to real-time constraints of the application or lack of bandwidth. In contrast to error resilience, which is carried out at the encoder, EC is applied at the decoder. EC algorithms can be classified into two categories: Spatial EC (SEC), which relies on the information provided within the current frame and Temporal EC (TEC), that utilizes temporal information such as motion vectors (MV) and previous/future frames. Both categories exploit redundancy due to high spatial and temporal correlation within a video sequence. Temporal correlation tends to be higher than the spatial correlation, so TEC techniques usually

This work has been supported by the Spanish MEC/FEDER project TEC 2010-18009.

1068-0314/12 $26.00 © 2012 IEEE
DOI 10.1109/DCC.2012.24

provide better results. This would be the straightforward choice when concealing a P/B-frame (intercoded). However, utilizing temporal information for the recovery of I-frames (intracoded) is not always possible, since these are inserted mainly to reset the prediction error when a change of scene occurs. Thus, when all the available temporal information belongs to different scene or there is no temporal information available, SEC algorithms are preferred. Every I/P-frame in the video sequence usually serves as a prediction template for, at least, one intercoded frame. Thus, high quality concealment is required since any reconstruction error will be propagated until the next I-frame arrives.

Several SEC techniques have been proposed for block-coded video/images. In [2], a simple spatial interpolation is used. In [3] a directional extrapolation algorithm was proposed, which exploits the fact that high frequencies and especially edges are visually the most relevant features. More advanced techniques including edge detectors combined with a Hough transform were utilized in [4]. Modelling natural images as Markov random fields for error concealment purposes was treated in [5]. The authors in [6] combined edge recovery and selective directional interpolation in order to achieve a more visually pleasing texture reconstruction. Patch-based texture recovery was introduced in [7]. Inpainting-based methods can also be adopted for SEC purposes [8] [9]. Sequential pixel-wise recovery based on orientation adaptive interpolation is treated in [10]. In [11], sequential Bayesian restoration is combined with DCT pyarmid decomposition. Recently, SEC techniques in transform domains have shown promising results [12].

In this paper we propose a spatial error concealment technique, where the lost regions are recovered sequentially using templates that are extracted from the available neighbourhood and combined according to a proper set of weights. First, we formulate the problem as a convex optimization problem and then we derive a fast approximation. We compare our proposals to the existing state-of-the-art algorithms on a wide selection of images and show that both in terms of PSNR and MS-SSIM our proposals provide better results.

The paper is organized as follows. In Section 2 we formulate the problem. The proposed algorithm is described in Section 3. Simulations results and comparisons with other SEC techniques are presented in Section 4. The last section is devoted to conclusions.

2 Problem Formulation

Our aim is to conceal the lost region, $\mathcal{L}$, by exploiting only the correctly received pixels in the neighbouring support area, $\mathcal{S}$.

2.1 Spatial model of an image

We define the image as a quasi-stationary signal that is locally generated by means of a stationary AR process [13]. Thus, the pixel $z(i,j)$ located at position (i,j) is generated as a linear combination of those in its neighbourhood: [1]

$$z(i,j) \;=\; \sum_{(k,l)\in\mathcal{N}} w_{(k,l)} z(k,l) + \nu(i,j), \tag{1}$$

where $\mathcal{N}$ is the stationary surrounding area of $z(i,j)$ and $\nu(i,j)$ is the residual error, which is often modelled as independent and identically distributed Gaussian noise $N(0,\sigma_N^2)$. In

[1] Note that the intracoding scheme in H.264/AVC [1] can be seen as particular case of (1).

this work, however, we do not impose any specific type of distribution upon the noise.

We will assume that the pixel values are in the range $[0; 255]$ for each colour space component. Thus, the autocorrelation is non-negative regardless of the lag, i.e. $w_{(k,l)} \geq 0$, where $w_{(k,l)}$ denotes the scalar weight associated with pixel $z(k, l)$, and that will be used for reconstruction of the pixel $z(i, j)$.

2.2 Weighted template matching (WTM)

Let $\boldsymbol{z}$ be an arbitrarily shaped group of stationary pixels. Writing $\boldsymbol{z}$ as a column vector we have $\boldsymbol{z} \in \Psi^n$, where Ψ is the $[0; 255]$ subset of integers. We will now extend the AR formulation of (1) to vectors $\boldsymbol{z}$. First, let $\mathcal{Z}$ be the set of all shifted versions of $\boldsymbol{z}$ within the stationary neighbourhood so that $\bigcup_{j=1}^{|\mathcal{Z}|} \boldsymbol{z}_j = \mathcal{N}$ and $|\boldsymbol{z}_j \bigcap \boldsymbol{z}_k| < |\boldsymbol{z}_j|$ for any $j \neq k$, i.e. no two $\boldsymbol{z}$'s completely overlap. Thus, a group of lost pixels $\boldsymbol{z}_i$ can be expressed as a linear combination of neighbouring vectors as,

$$\boldsymbol{z}_i \;=\; \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{z}_j + \boldsymbol{\nu}_i. \tag{2}$$

The (unpredictable) residual error $\boldsymbol{\nu}_i$ is independent of the surrounding pixels and it is therefore not possible to exactly recover $\boldsymbol{z}_i$. In order to perform the concealment, the vector $\boldsymbol{w} = [w_1, ..., w_{|\mathcal{Z}|}]^T$ of scalar weights must first be computed in a way the residual error, $\boldsymbol{\nu}_i$, is minimized. Since the image is, in general, non-stationary, the support area $\mathcal{S}$ may include more pixels than those contained in the stationary neighbourhood $\mathcal{N}$ of $\boldsymbol{z}_i$, so the Yule-Walker equations cannot be directly applied. However, the missing block, $\boldsymbol{z}_i$, depends only on the set of $\boldsymbol{z}_j$'s belonging to the stationary neighbourhood $\mathcal{N}$. Since the location of these $\boldsymbol{z}_j$'s is unknown the entire support area needs to be searched. The set $\mathcal{Z}$ is therefore extended, so $\bigcup \boldsymbol{z}_j = \mathcal{S}$. Including more information than required for the generating AR process will, by definition, not lead to better results. Thus, given the search area $\mathcal{S}$, we are seeking the sparsest solution among all the $\boldsymbol{w}$'s that minimizes the squared prediction error. Specifically, we need to solve the following optimization problem:

$$\begin{aligned} f(\delta) = &\quad \min_{\boldsymbol{w} \in \mathcal{W}(\delta)} \|\boldsymbol{z}_i - \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{z}_j\|_2^2 \\ \mathcal{W}(\delta) = &\quad \{\boldsymbol{w} |\; \|\boldsymbol{w}\|_0 \leq \delta \quad and \quad \boldsymbol{w} \succeq 0\}, \end{aligned} \tag{3}$$

where $\delta \in \mathbb{N}$ is the sparsity. The sparsity level of the sparsest solution is then given by

$$\delta^* \;=\; \min\{\operatorname*{argmin}_{\delta} f(\delta)\}. \tag{4}$$

The sparsest solution $\boldsymbol{w}^*$ is the one corresponding to δ^*.

3 Proposed Algorithms

In this section, we first propose a SEC technique based on convex optimization. Then, we derive a computationally less expensive algorithm by applying several approximations.

3.1 WTM via convex relaxation

The optimization problem defined by (3) and (4) can be rewritten as

$$\begin{aligned} &minimize \quad \|\boldsymbol{w}\|_0 \\ &subject\ to \quad \|\boldsymbol{z}_i - \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{z}_j\|_2^2 = \gamma \quad and \quad \boldsymbol{w} \succeq 0, \end{aligned} \tag{5}$$

where γ is the squared ℓ_2-norm of the residual error from (2). The minimization over the ℓ_0-"norm" usually requires exhaustive search and is therefore computationally expensive. Using convex relaxation [14], (5) can be written in terms of the ℓ_1-norm

$$\begin{aligned} &minimize \quad \|\boldsymbol{w}\|_1 \\ &subject\ to \quad \|\boldsymbol{z}_i - \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{z}_j\|_2^2 = \gamma \quad and \quad \boldsymbol{w} \succeq 0. \end{aligned} \tag{6}$$

The residual energy γ is usually not known in advance, so we rewrite (6) as a joint minimization problem

$$\begin{aligned} f(\delta) = &\quad \min_{\boldsymbol{w} \in \mathcal{W}(\delta)} \|\boldsymbol{z}_i - \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{z}_j\|_2^2 \\ \mathcal{W}(\delta) = &\quad \{\boldsymbol{w} |\ \|\boldsymbol{w}\|_1 \leq \delta \quad and \quad \boldsymbol{w} \succeq 0\}, \end{aligned} \tag{7}$$

$$\delta^* \quad = \quad \min\{\operatorname*{argmin}_{\delta} f(\delta)\}. \tag{8}$$

Eqs. (7) and (8), however, cannot be applied directly as $\boldsymbol{z}_i$ is unknown, since it constitutes the lost region. Let us consider, without loss of generality, $\boldsymbol{z}_i$ to be the group of pixels as shown in Fig. 1(a). Note that $\boldsymbol{z}_i$ can be split into two subsets: $\boldsymbol{x}_i$, which contains only the missing pixels and $\boldsymbol{y}_i$, which is formed only by received and correctly decoded pixels and can be seen as the spatial context of $\boldsymbol{x}_i$. All $\boldsymbol{z}_j \in \mathcal{Z}$ are split in a similar way, as shown in Fig. 1(a). Since $\boldsymbol{z}_i$ is (locally) stationary and $\boldsymbol{y}_i \subset \boldsymbol{z}_i$ then the weights obtained in (7) and (8) will be identical to the weights obtained by

$$\begin{aligned} g(\delta) = &\quad \min_{\boldsymbol{w}} \|\boldsymbol{y}_i - \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{y}_j\|_2^2 \\ \mathcal{W}(\delta) = &\quad \{\boldsymbol{w} |\ \|\boldsymbol{w}\|_1 \leq \delta \quad and \quad \boldsymbol{w} \succeq 0\}. \end{aligned} \tag{9}$$

$$\delta^* \quad = \quad \min\{\operatorname*{argmin}_{\delta} g(\delta)\} \quad = \quad \min\{\operatorname*{argmin}_{\delta} f(\delta)\}. \tag{10}$$

Finally, according to (2) the concealed group of pixels, $\hat{\boldsymbol{x}}_i$, can be approximated by a linear combination of blocks within its stationary neighbourhood

$$\hat{\boldsymbol{x}}_i \quad = \quad \sum_{j=1}^{|\mathcal{Z}|} w_j^* \boldsymbol{x}_j. \tag{11}$$

Computing the sparsity level δ^* of $\boldsymbol{w}^*$ (10) for every $\boldsymbol{x}_i$ is computationally expensive. Instead, we estimate the sparsity by assuming smoothness in the visual features of an image.

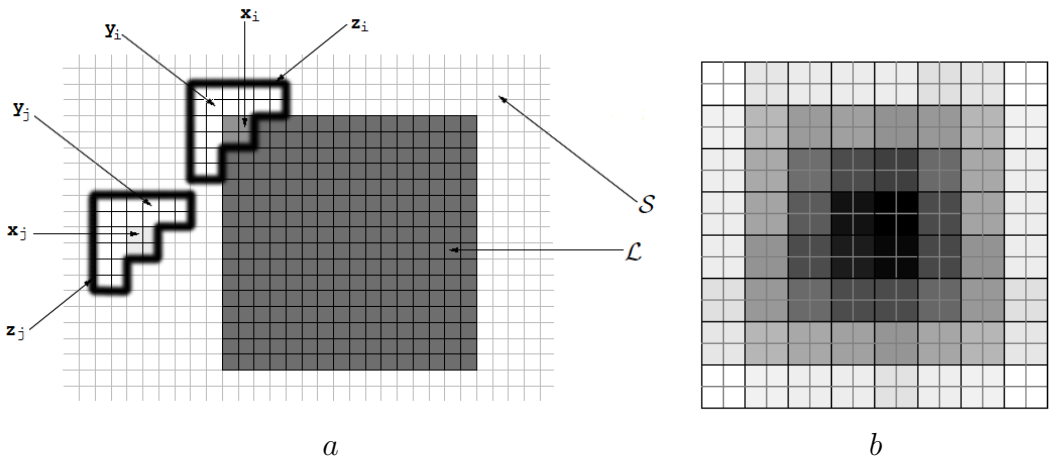


Figure 1: *(a) Example of configuration for the vectors $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$. $\mathcal{S}$ denotes the set of known pixels and $\mathcal{L}$ denotes the set of lost pixels. (b) Filling order for sequential reconstruction with 2×2 patches ($p = 2$). The regions illustrated by brighter level are recovered first.*

In the particular case of luma, it means that a reconstructed pixel could not be brighter (darker) than the brightest (darkest) pixel in $\mathcal{S}$. This requires that (11) must be a convex combination and it implies that $\delta = 1$. Finally, (9) and (10) can be reduced to

$$\begin{aligned} \underset{\boldsymbol{w}}{minimize} \quad & \|\boldsymbol{y}_i - \sum_{j=1}^{|\mathcal{Z}|} w_j \boldsymbol{y}_j\|_2^2 \\ subject\ to \quad & \|\boldsymbol{w}\|_1 \leq 1 \quad and \quad \boldsymbol{w} \succeq 0. \end{aligned} \tag{12}$$

The H.264/AVC coder packetizes the stream by slices so a loss of one packet implies a loss of, at least, one 16×16 macroblock. Applying (11) for $\boldsymbol{x}_i \in \Psi^{16\times16}$ would lead to significant imprecisions as well as blurring since it is often not possible to find a combination of $\boldsymbol{x}_j$'s that matches $\boldsymbol{x}_i$ well enough in the entire $\Psi^{16\times16}$ space. It implies that the residual error from (2) carries significant energy. In order to manage with this problem, we introduce sequential recovery. Thus, the macroblock is recovered using a set of square patches $\hat{\boldsymbol{x}}_i \in \Psi^{p\times p}$ with $1 \leq p \leq 16$. Pixel-wise reconstructions, as in [10], may introduce considerable blurring when high frequencies are involved (Fig. 4b). Using larger templates, the correlation within a template is better preserved and so is the texture (Fig. 4c). Let us consider, without loss of generality, $p = 2$ and let $\boldsymbol{y}_i$ include all the received or already recovered pixels within the 6×6 pixel neighbourhood of $\boldsymbol{x}_i$, as shown in Fig. 1(a). The macroblock is recovered sequentially by filling it with $\hat{\boldsymbol{x}}_i$ obtained by applying (12) and (11). The filling order is crucial for a high quality recovery and we base it on the reliability of $\boldsymbol{y}_i$. We define the reliability of the context $\boldsymbol{y}_i$, ρ_i, as the sum of reliabilities of its pixels. The reliability of a pixel is set to 1 if it has been correctly received and decoded. Missing pixels have zero reliability. When a pixel $x \in \boldsymbol{x}_i$ is concealed, its reliability is set to $\alpha\rho_i/m$, where $0 < \alpha < 1$ and m is the number of pixels contained in $\boldsymbol{y}_i$. We use $\alpha = 0.9$ in our simulations. The lost region $\hat{\boldsymbol{x}}_i$, whose context $\boldsymbol{y}_i$ produces the highest reliability, is recovered first. The reliability is non-increasing and the reconstruction evolves from the outer layer towards the centre of the corrupt macroblock. Fig. 1(b) shows the filling order of a 16×16 macroblock using 2×2 templates. Note that the first squares to be concealed

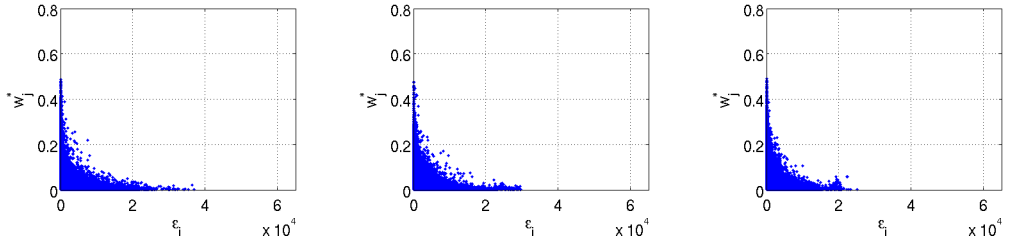
are the corners as their contexts are the largest, providing thus more reliable information which leads to a more accurate estimate of the weights.

3.2 WTM with exponentially distributed weights

Although there are efficient algorithms for solving convex optimization problems, the processing time remains high. In this section we develop a fast approximation for solving the minimization problem in (12). Specifically, we show that the weights $\boldsymbol{w}^*$ can be well modelled by an exponential distribution.

According to (12), every context $\boldsymbol{y}_j$ provides a weight w_j^*. Due to the high spatial correlation of an image, it is likely that contexts that produce smaller square error, ϵ_j, would generate larger weights, where

$$\epsilon_j \quad = \quad \frac{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2}{m}. \tag{13}$$



a *b* *c*

Figure 2: *Distribution of weights as a function of ϵ for (a) "Peppers", (b) "Cameraman" and (c) "Barbara".*

The weights w_j^*, obtained by (12), for three different concealed images are shown in Fig. 2 as a function of ϵ_j. The error pattern applied is the one shown in Fig. 3(b). Note that the weights appear to be exponentially decaying. There is, however, a cluster of relatively low weights associated with small quadratic errors. The mild sparsity constraint in (12) implies that given two candidate templates with similar quadratic error, the optimization algorithm picks one and suppresses the other instead of using them both. Since the contribution of templates with very small weights to the final reconstruction is negligible, we will approximate the distribution of the weights by an exponential distribution, i.e.,

$$\hat{w}_j \quad = \quad \exp(-\frac{1}{2}\frac{\epsilon_j}{\sigma^2}), \tag{14}$$

where $\hat{w}_j$ is the approximated w_j^* and σ^2 is the variance of the distribution and can be estimated for each macroblock (or each patch). For the sake of computational simplicity a fixed value of σ^2 is used. In order to find the appropriate σ^2, we minimize the following penalty function:

$$\pi(\sigma^2) \quad = \quad \sum_{j=1}^{|\mathcal{Z}|}(\hat{w}_j - w_j^*)^2 w_j^* \quad = \quad \sum_{j=1}^{|\mathcal{Z}|}(\exp(-\frac{1}{2}\frac{\epsilon_j}{\sigma^2}) - w_j^*)^2 w_j^*. \tag{15}$$

Note that each term in the sum is scaled by w_j^* in order to reduce the influence of the small weights. Given an image, we obtain a set of estimated variances, one per patch, by minimizing the penalty function in (15). Table 1 shows the mean and the median values for the majority of the tested images. For natural images, values around 10 lead to visually good results (Fig. 5b). Larger values of σ^2 lead to oversmoothing (Fig. 5c) while small values can lead to numerical instability and should be avoided (Fig. 5a) (unless there are extremely good candidate templates).

Finally, the weights are normalized and the missing area $\boldsymbol{x}_i$ is estimated as

$$\hat{\boldsymbol{x}}_i \;=\; \frac{1}{\sum_{j=1}^{|\mathcal{Z}|}\hat{w}_j}\sum_{j=1}^{|\mathcal{Z}|}\hat{w}_j\boldsymbol{x}_j. \tag{16}$$

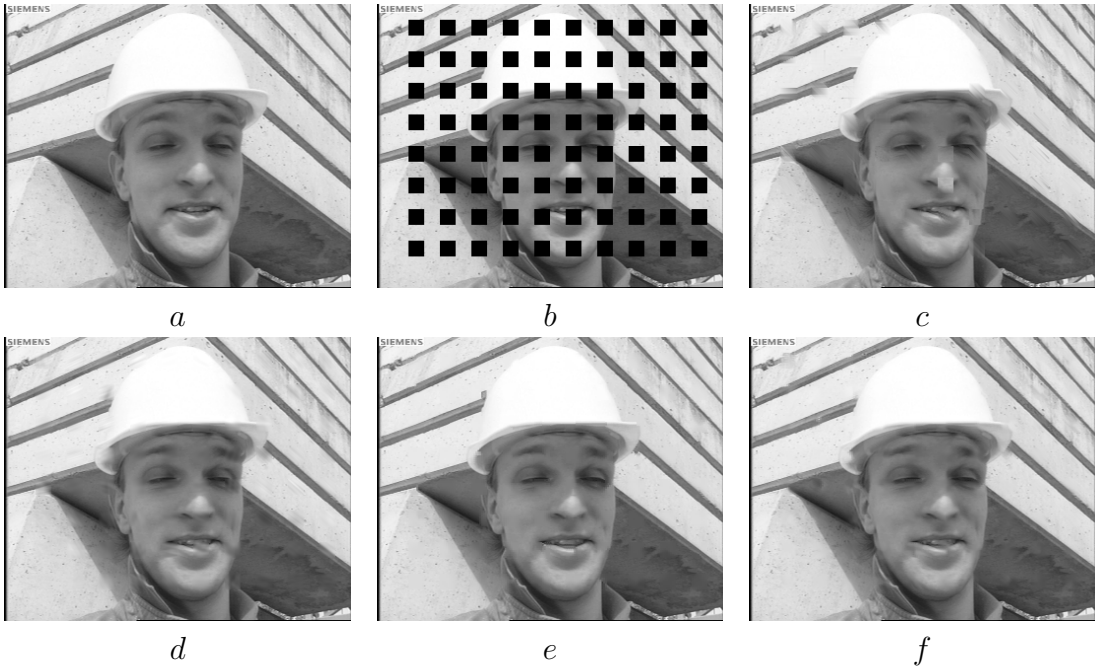


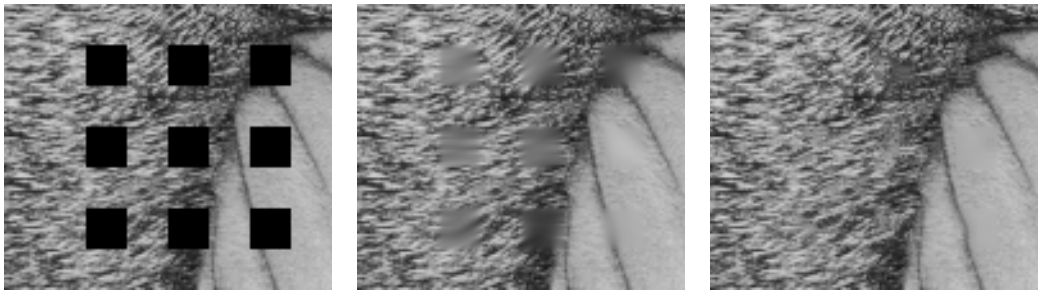
Figure 3: *SEC for the image of "Foreman" (a) Original image, (b) Received data, (c) Reconstruction using CAD (PSNR = 31.46dB, MS-SSIM = 97.54), (d) FSE (PSNR = 34.17dB, MS-SSIM = 98.03), (e) WTE (PSNR = 35.46dB, MS-SSIM = 98.73), (f) WTC (PSNR = 35.48dB, MS-SSIM = 98.68).*

σ^2	Barbara	Lena	Office	Peppers	Foreman	Cameraman	Average
mean	13.59	10.53	12.13	5.52	6.80	12.59	10.19
median	5.00	5.00	3.00	1.00	3.00	2.00	4.00

Table 1: *Estimated variance for tested images*

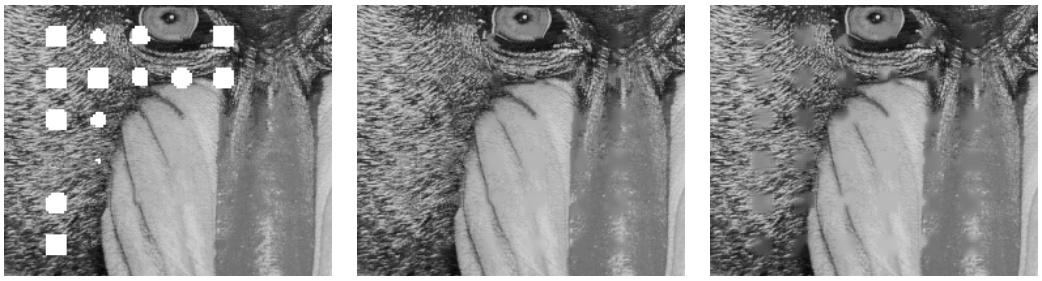
4 Simulation Results

In order to better take into account the perceptual quality, the multi scale structural similarity (MS-SSIM) index [15] is used for comparison along with the PSNR measure. In the



a b c

Figure 4: *(a) Received image, (b) reconstructed by OAI (PSNR = 27,22, MS-SSIM = 92,47) (c) reconstructed using (14) and (16) with $p = 2$ (PSNR = 25,56, MS-SSIM = 94,76).*



a b c

Figure 5: *EC for different variances σ^2: (a) σ^2 = 0.5, numerically unstable reconstructions are represented with white level, (b) σ^2 = 10, (c) σ^2 = 50.*

	BIL	EXT	SHT	CAD	AVC	MRF	INP	FSE	OAI	WTE	WTC
Average (PSNR)	26.75	27.41	27.01	28.09	27.77	29.37	28.94	30.48	30.28	**30.71**	31.18
Average (MS-SSIM)	93.93	95.03	94.29	94.84	95.14	96.31	95.42	96.69	96.28	**97.32**	96.86
Lena (PSNR)	30.00	29.39	30.47	30.44	30.42	31.89	30.88	32.72	**32.83**	32.55	32.85
Lena (MS-SSIM)	96.66	96.47	96.97	96.59	96.57	97.68	96.60	97.79	97.63	**97.94**	97.74
Peppers (PSNR)	29.47	30.24	29.68	31.26	32.13	32.76	32.16	33.48	**34.53**	33.95	34.35
Peppers (MS-SSIM)	94.90	96.86	95.26	96.84	97.26	97.86	97.29	97.99	98.39	**98.42**	98.23
Foreman (PSNR)	27.12	29.26	28.34	31.46	29.11	33.87	33.87	34.17	34.87	**35.46**	35.48
Foreman (MS-SSIM)	95.24	97.17	95.77	97.54	96.96	98.36	98.25	98.03	98.66	**98.73**	98.68
Barbara (PSNR)	26.19	26.85	25.85	26.78	26.40	27.74	28.04	**30.84**	29.37	30.79	31.91
Barbara (MS-SSIM)	95.26	94.88	95.57	95.77	95.10	96.29	95.81	97.72	96.81	**97.95**	98.19
Office (PSNR)	27.54	30.00	27.32	29.43	27.55	29.76	29.64	31.32	30.39	**31.34**	32.06
Office (MS-SSIM)	94.00	94.94	94.00	95.73	95.96	96.64	96.38	96.98	96.12	**97.43**	97.41
Baboon (PSNR)	24.15	24.72	24.14	24.92	25.42	25.17	25.06	26.02	**26.15**	26.02	26.21
Baboon (MS-SSIM)	88.84	91.35	88.69	91.80	91.92	91.94	90.90	93.27	92.69	**93.33**	93.39
Cameraman (PSNR)	25.96	25.05	26.16	25.96	26.14	26.41	25.96	**27.44**	27.00	27.24	27.26
Cameraman (MS-SSIM)	93.91	94.47	94.09	94.84	94.04	95.10	93.18	94.84	94.79	**96.92**	93.76
Tire (PSNR)	23.59	23.82	24.10	24.47	24.99	27.37	25.93	27.87	27.06	**28.33**	29.32
Tire (MS-SSIM)	92.63	94.09	93.97	89.59	93.28	96.60	94.92	96.91	95.11	**97.80**	97.46

Table 2: *PSNR values (in dB) and MS-SSIM indices (scaled by 100) for test images reconstructed by several algorithms for block dimensions 16 × 16. The best performances for each image are put in bold face (excluding the results for the more computationally expensive algorithm WTC).*

former case, the image is sequentially low-pass filtered and subsampled so a set of images is obtained, including the original resolution. Then, the SSIM index is applied for every subimage within the set. The SSIM index aims at approximating the human visual system

(HVS) response looking for similarities in luminance, contrast, and structure [15]. This index can be seen as a convolution of a fixed-sized mask with the residual error between the reference image and the concealed image [16]. A unique mask size is used for each of the images within the set so fine as well as coarse textures and objects are taken into account.

As shown in Fig. 4, the PSNR does not respond to perceptual visual quality as well as the MS-SSIM index does. In spite of that, the weights $\boldsymbol{w}^*$ are obtained according to the squared error (14) since the SSIM index tends to marginalize the influence of changes in intensity [15]. This is a desirable behaviour when measuring the perceptual image quality but not when finding candidate templates. Thus, the squared error is used when computing the weights while the MS-SSIM index is preferred for an overall quality measure. [2]

The performance of the proposed algorithm is tested on the images of "Lena" (512×512), "Barbara" (512×512), "Baboon" (512×512), Matlab built-in images "Peppers" (384×512), "Office" (592×896), "Cameraman" (256×256), "Tire" (192×224) and the first frame of "Foreman" (288×352) sequence. The test is carried out for macroblock dimensions of 16×16 and the rate of block loss is approximately 25%, corresponding to a single packet loss of a frame with dispersed slicing structure. We compare the performance with other SEC methods such as bilinear interpolation (BIL) [2], directional extrapolation (EXT) [3], a Hough transform based SEC (SHT) [4], content adaptive technique (CAD) [7], non-normative SEC for H.264 (AVC) [17], Markov random fields approach (MRF) [5], inpainting (INP) [9], frequency selective extrapolation (FSE) [12] and orientation adaptive interpolation (OAI) [10]. Both WTM via convex relaxation (WTC) and WTM with exponentially distributed weights (WTE) are tested. In the simulations, σ^2 is set to 10 and grey level images are used. Note that a pixel reconstructed by any of the aforementioned algorithms is usually real-valued and does not necessarily belong to Ψ. Thus, for comparison purposes, reconstructed pixels are rounded to the closest member of Ψ. Subjective comparison of different algorithms is shown in Fig. 3. As can be seen in Table 2, the proposed technique outperforms the others for all the tested images in terms of MS-SSIM. Moreover, the average MS-SSIM and PSNR are superior to those of state-of-the-art algorithms.

5 Conclusions

We have developed a weighted template matching algorithm, which recovers lost regions in images by filling them sequentially with a weighted combination of templates that are extracted from the available neighbourhood. The weights are obtained by solving a convex optimization problem that arises from a spatial image model. Alternatively, we show that the weights can be approximated by an exponential distribution. Our proposals achieve better PSNR and perceptual reconstruction quality than other state-of-the-art techniques. WTC is optimized for squared error so it achieves better PSNR than the approximated method. Simulations reveal, however, that WTE provides better MS-SSIM. Finally, by applying the approximated algorithm the processing time is reduced in a factor of 100.

Ongoing research is devoted to the extension of our algorithm into error concealment problems in the temporal domain.

[2] Note that the MS-SSIM index lies between [-1; 1]. In this section, we have scaled the index by 100 in order to better illustrate the differences.

References

[1] ITU-T, "ITU-T Recommendation H.264," International Telecommunication Union, 2005.

[2] P. Salama, N. Shroff, E. Coyle, and E. Delp, "Error concealment techniques for encoded video streams," in *Proceedings of ICIP*, pp. 9–12, 1995.

[3] Y. Zhao, H. Chen, X. Chi, and J. Jin, "Spatial error concealment using directional extrapolation," in *Proceedings of DICTA*, pp. 278–283, 2005.

[4] H. Gharavi and S. Gao, "Spatial interpolation algorithm for error concealment," in *Proceedings of ICASSP*, pp. 1153–1156, April 2008.

[5] S. Shirani, F. Kossentini, and R. Ward, "An adaptive Markov random field based error concealment method for video communication in error prone environment," in *Proceedings of ICIP*, vol. 6, pp. 3117–3120, 1999.

[6] W. Kung, C. Kim, and C. Kuo, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 789–802, July 2006.

[7] Z. Rongfu, Z. Yuanhua, and H. Xiaodong, "Content-adaptive spatial error concealment for video communication," *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 335–341, February 2004.

[8] P. Harrison, "Texture synthesis, texture transfer and plausible restoration," *PhD. Thesis, Monash University*, 2005.

[9] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, pp. 1200–1212, September 2004.

[10] X. Li and M. Orchard, "Novel sequential error-concealment techniques using orientation adaptive interpolation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 857–864, October 2002.

[11] G. Zhai, X. Yang, W. Lin, and W. Zhang, "Bayesian error concealment with DCT pyramid for images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, pp. 1224–1232, September 2010.

[12] J. Seiler and A. Kaup, "Fast orthogonality deficiency compensation for improved frequency selective image extrapolation," in *Proceedings of ICASSP*, pp. 781–784, March 2008.

[13] N. Jayant and P. Noll, "Digital coding of waveforms," Prentice Hall, 1984.

[14] J. Romberg, "Imaging via compressive sensing," *IEEE Signal Processing Magazine*, vol. 25, March 2008.

[15] Z. Wang, E. Simoncelli, and A. Bovik, "Multi-scale structural similarity for image quality assessment," *IEEE Signals, Systems and Computers*, vol. 2, pp. 1398–1402, November 2003.

[16] J. Østergaard, M. Derpich, and S. Channappayya, "The high-resolution rate-distortion function under the structural similarity index," *EURASIP Journal on Advances in Signal Processing*, 2011.

[17] V. Varsa and M. Hannuksela, "Non-normative error concealment algorithms," *ITU-T SG16, VCEG-N62*, vol. 50, September 2001.