PRELIB: PDE-BASED RESTORATION OF LOST IMAGE BLOCKS

Hossein Ajorloo
Multimedia Research Group, IT Faculty, ITRC, Tehran, Iran

ABSTRACT
In this paper, restoration of damaged image blocks with large sizes is considered and a Discrete Fourier Transform (DFT)-based approach is developed for this purpose, which requires interventions of both source encoder and decoder. Use of lossy codecs causes a processing error leading to noisy images after recovery. A Partial Differential Equation (PDE)-based image enhancement approach is proposed to reduce the visual effect of this processing error. Experimental results show that our method can restore large size lost blocks of images, which overcomes previously proposed approaches.

Index Terms— Image enhancement, Partial differential equations, Discrete Fourier transforms

1. INTRODUCTION
Considering errors and losses in the networks is a necessary challenge to have an effective communication of images and videos. There are many efforts in the literature to restore damaged blocks of images. For example, see [1–3] for recently proposed solutions. All of these algorithms try to restore blocks of size 8 × 8 in corrupted images. But, how can the decoder deal with the lost blocks of large sizes, about 64 × 64 or even more. We proposed in this paper to manually make correlation between the pixels of images to enable the decoder to recover the large lost regions of the images. As expected, this artificial correlation distorts the image. By choosing appropriate parameters, one can control this distortion to lie in acceptable ranges. Using this correlation, any lost block in the image that has a size smaller than a certain maximum value, introduced by the algorithm, can be retained exactly when using lossless codecs.

However, when the transmission system uses lossy codecs, they destroy the correlation made in the image and hence the recovery process encounters a processing error which appears as a perceptible noise in the recovered image. Here, we have proposed a PDE-based approach to enhance the resulting noisy image which preserves the edges of the image and avoids the blurring effects. The overall procedure is called PReLIB (PDE-based Restoration of Lost Image Blocks).

The remainder of this paper is organized as follows. In Section 2 the proposed recovery method is described. In Section 3 our PDE-based image enhancement is presented and then in Section 4, the experimental results are given. Finally, Section 5 concludes the paper.

2. RECOVERY OF LOST IMAGE BLOCKS
At the transmitter, the only required task is to convert $L_1 \times L_2$ Discrete Fourier Transform (DFT) frequency bins to zero. i.e.,

\[ X(k_1, k_2) = 0, \quad k_i \in L_i, \quad i = 1, 2, \]

where $L_1$ and $L_2$ are the odd numbers which should set to dimensions of any lost block (such as a macro-block or a slice) that we desire to restore at the receiver in our algorithm; $k_1$ and $k_2$ are the frequency bins indices; $L_1 = \{(N_1 - L_1 + 1)/2, \cdots, (N_1 + L_1 - 1)/2\}$; and $N_1$ and $N_2$ are the image dimensions. Note that the Hermitian symmetry is preserved in this procedure. Since, we have created a low-pass signal by this conversion, an artificial correlation is made between the pixels of the image.

If we represent the received image (decoded image with lost blocks) by $d$, and the error image by $e$, then these two images satisfy the following relation:

\[ x(n_1, n_2) = \epsilon(n_1, n_2) + d(n_1, n_2). \]

where $n_1$ and $n_2$ are the pixel indices. Note that $\epsilon$ equals to zero in positions that image is damaged or lost and to $x$ otherwise. On the other hand, $\epsilon$ equals to $x$ in the lost sample positions and to zero, otherwise. Thus, if we can find $\epsilon$, then $x$ can be obtained from (2). In the DFT domain, we can write:

\[ X(k_1, k_2) = E(k_1, k_2) + D(k_1, k_2), \]

where $k_1$ and $k_2$ are the DFT indices. But then, because of (1) we have:

\[ E(k_1, k_2) = -D(k_1, k_2), \quad k_i \in L_i. \]

For making the algorithm stable, one approach is to use the scrambled kernel of the DFT as $w_N = \exp(-j2\pi/qN)$ instead of $w_N = \exp(-j2\pi/N)$, where $q$ is a natural number prime with respect to $N$. All tasks of our technique can be done with this kernel and transformation.

We now define the error locator polynomials as

\[ H_1(k) = \prod_{m=1}^{L} (w_N^k - w_{N}^{i_{1m}}), \quad H_2(k) = \prod_{m=1}^{L} (w_N^k - w_{N}^{i_{2n}}), \]

where $i_{1m}$ and $i_{2n}$ are the lost pixel coordinates. For the sake of simplicity, and without any loss of generality, in (5) we assumed that $L_1 = L_2 = L$ and $N_1 = N_2 = N$. By expanding the right side of (5), we obtain,

\[ H_1(k) = \sum_{l=0}^{L} h_1(l) w_{N}^{il}, \quad H_2(k) = \sum_{l=0}^{L} h_2(l) w_{N}^{il}. \]

We thus have two 1-D DFTs

\[ h_1(l) = \text{IDFT}(H_1(k)), \quad h_2(l) = \text{IDFT}(H_2(k)). \]

Substituting $i_{1m}$ and $i_{2n}$ in (6), we obtain

\[ \sum_{l=0}^{L} h_1(l) w_{N}^{i_{1m}l} = 0, \quad \sum_{l=0}^{L} h_2(l) w_{N}^{i_{2n}l} = 0. \]
Multiplying both sides of Eqs. (8) by \( e(i_{1m}, i_{2n}) w_N^{r_1 i_{1m} + r_2 i_{2n}} \) and taking its summation with respect to \( n \) and \( n \), respectively, we get

\[
\sum_{n=1}^{N} h_1(l) \sum_{m=1}^{N} e(i_{1m}, i_{2n}) w_N^{r_1 i_{1m} + r_2 i_{2n}} = 0,
\]

\[
\sum_{n=1}^{N} h_2(l) \sum_{m=1}^{N} e(i_{1m}, i_{2n}) w_N^{r_1 i_{1m} + (r_2 + 1) i_{2n}} = 0.
\]  

(9)

On the other hand, the following relation exists between signals \( e(n_1, n_2) \) and \( E(k_1, k_2) \), since they are 2D-DFT pair:

\[
E(k_1, k_2) = \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} e(n_1, n_2) u_N^{k_1 n_1 + k_2 n_2}.
\]  

(10)

Noting that the 2-D signal \( e(n_1, n_2) \) is nonzero only in the positions of lost pixels, \( i.e. \), for \( i_{1m} \) and \( i_{2n} \), and is zero otherwise, (10) becomes

\[
E(k_1, k_2) = \sum_{m=1}^{L} \sum_{n=1}^{L} e(m, n) u_N^{k_1 n_1 + k_2 n_2}.
\]  

(11)

Now, from (9) and (11), the following recursive formulas are obtained:

\[
\sum_{n=1}^{N} h_1(l) E(r_1 + l, r_2) = 0, \quad \sum_{n=1}^{N} h_2(l) E(r_1, r_2 + l) = 0.
\]  

(12)

Thus, the recursion will yield all the unknowns of \( E(k_1, k_2) \), because of (3) where \( X(k_1, k_2) \) can be obtained, and by taking its inverse 2-D DFT, the original image is restored.

In the following, we have computed the average PSNR resulting from the artificial correlation we made in (1). The power of noise is computed from

\[
P_n = E\left\{ \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} [d(n_1, n_2) - d(n_1, n_2)]^2 \right\}.
\]  

(13)

Using the Parseval’s theorem, we can write

\[
P_n = \frac{1}{N^2 E\left\{ \sum_{k_1=1}^{L} \sum_{k_2=1}^{L} |X(k_1, k_2)|^2 \right\}}.
\]  

(14)

The average PSNR resulted from the correlation made in our algorithm can be computed by some manipulations as

\[
\text{PSNR}_{\text{avg}} = 10 \log \left( \frac{N^2 \times 255^2}{L^2 \sigma_x^2} \right),
\]  

(15)

where \( \sigma_x^2 \) denotes the variance of the original image. For example, assuming \( \sigma_x = 24 \) (a typical value for variance in the images), the PSNR_{avg} for making \( 65 \times 65 \) zeros (\( L = 65 \)) in the DFT domain of a \( 256 \times 256 \) image (\( N = 256 \)) is about 32 dB.

3. PDE-BASED IMAGE ENHANCEMENT

When we use lossy image codecs, the decoded pixels differ from the original ones where our recovery algorithm encounters a problem and the recovered image will contain a processing error. In this paper, we have used PDE-based restoration model \textit{Motion by Mean Curvature} (MMC) for the removal of the mentioned processing error. A complete review of the PDE-based algorithms can be found in [4].

Let us denote the erroneous part of the image by \( z \). We assumed the following additive error model

\[
z = x + \epsilon,
\]  

(16)

where \( x \) is the correlated clean image (the same appeared in (2)) in the lost block we want to find and \( \epsilon \) denotes the processing error. A good restoration technique is to minimize the sum of the squared error and the area under the amplitude of the gradient of the clean signal as follows [5]:

\[
\hat{x} = \arg\min_{x} \left( \int_{D} |\nabla x| + \frac{k}{2} \|z - x\|^2 \right).
\]  

(17)

where \( k \) is a constraint parameter, \( \|\cdot\| \) denotes the \( L^2 \)-norm, and \( D \) is the image domain which is an open subset in \( \mathbb{R}^2 \).

One can transform the minimization problem (17) into the Euler-Lagrange equation. Applying calculus of variations to (17), and parameterizing the descent direction by an artificial time \( t \), the Euler-Lagrange equation can be formulated as,

\[
\frac{\partial x}{\partial t} - \nabla \left( \frac{\nabla x}{|\nabla x|} \right) = k(z - x).
\]  

(18)

The diffusion contributed with diffusion coefficient \( 1/|\nabla x| \) in (18) can be larger when \( |\nabla x| \approx 0 \). This implies that (17) can enforce a strong diffusion in flat regions and therefore some pixels will become locally constant and hence the mosaicking phenomenon occurs. To solve this problem, we use the following model in this paper [4],

\[
\frac{\partial x}{\partial t} - |\nabla x| \nabla \left( \frac{\nabla x}{|\nabla x|} \right) = k(z - x),
\]  

(19)

where \( k \) is an empirical parameter. Note that the steady states of both (18) and (19) models are analytically the same.

An incomplete linearized Crank-Nicolson method for (19) can be formulated as follows [4],

\[
\frac{x^{n+1} - x^{(n-1)}}{\Delta t} + \frac{1}{2} (A_1^{(n-1)} + A_2^{(n-1)}) (x^{n} + x^{(n-1)}) = \beta \tilde{z}^{n-1},
\]  

(20)

where \( n \) is the iteration superscript and the operators \( A_l^{(n-1)} \), \( l = 1, 2 \) are defined as

\[
A_l^{(n-1)} x^{(m)} = - |\nabla x^{(n-1)}| \frac{\partial}{\partial n} \left( \nabla \left( \frac{\nabla x^{(n-1)}}{|\nabla x^{(n-1)}|} \right) \right) + \frac{\beta}{2} x^{(m)},
\]  

(21)

in which \((i_1, i_2)\) denote the pixel coordinates and \( l = 1, 2 \). For an efficient simulation, an incomplete \textit{Alternating Direction Implicit} (ADI) method is adopted for the time-stepping procedure as follows

\[
\begin{align*}
(1 + \frac{\Delta t}{2} A_1^{(n-1)}) x_l = &
(1 - \frac{\Delta t}{2} A_1^{(n-1)} - \Delta t A_2^{(n-1)}) x^{(n-1)} + \Delta t \beta \tilde{z}, \\
(1 + \frac{\Delta t}{2} A_2^{(n-1)}) x^{(n)} = &
x_l + \frac{\Delta t}{2} A_2^{(n-1)} x^{(n-1)},
\end{align*}
\]  

(22)

where \( x_l \) is an intermediate solution.
where $g_{ij}^{(n-1)}$, $l=1,2$ are defined by

$$
g_{1,ij}^{(n-1)} = \left\{ \frac{(x_{ij}^{(n-1)} - x_{i,j+1}^{(n-1)})^2}{4} + \frac{(x_{i,j+1}^{(n-1)} - x_{i,j+1}^{(n-1)})^2}{4} + \frac{(x_{i,j+1}^{(n-1)} - x_{i,j-1}^{(n-1)})^2}{4} \right\}^{1/2},$$

$$
g_{2,ij}^{(n-1)} = \left\{ \frac{(x_{i,j}^{(n-1)} - x_{i,j}^{(n-1)})^2}{4} + \frac{(x_{i,j}^{(n-1)} - x_{i-1,j}^{(n-1)})^2}{4} + \frac{(x_{i,j}^{(n-1)} - x_{i-1,j}^{(n-1)})^2}{4} \right\}^{1/2},$$

and $g_{1,ij}^{(n-1)} = g_{1,ij+1}^{(n-1)} = g_{2,ij}^{(n-1)} = g_{2,ij+1}^{(n-1)} = g_{2,ij}^{(n-1)} = g_{2,ij+1}^{(n-1)}$.

The value of $|\nabla x_{ij}^{(n-1)}|$ can be approximated by

$$|\nabla x_{ij}^{(n-1)}| \approx 2 \frac{g_{2,ij}^{(n-1)} - g_{2,ij}^{(n-1)}}{g_{2,ij}^{(n-1)} + g_{2,ij}^{(n-1)}}, \quad l=1,2.$$  

Then, it follows from (23) and (25) that

$$A_1^{(n-1)} = -\frac{P_{1,ij}^{(n-1)}}{g_{2,ij}^{(n-1)}} + \frac{P_{1,ij}^{(n-1)}}{g_{2,ij}^{(n-1)}} x_{ij}^{(n-1)} - \frac{P_{1,ij}^{(n-1)}}{g_{2,ij}^{(n-1)}} x_{i,j+1}^{(n-1)},$$

$$A_2^{(n-1)} = -\frac{P_{2,ij}^{(n-1)}}{g_{2,ij}^{(n-1)}} x_{i,j-1}^{(n-1)} + \frac{P_{2,ij}^{(n-1)}}{g_{2,ij}^{(n-1)}} + \frac{P_{2,ij}^{(n-1)}}{g_{2,ij}^{(n-1)}} x_{i,j+1}^{(n-1)},$$

where

$$P_{1,ij}^{(n-1)} = 2 \frac{g_{2,ij}^{(n-1)} - g_{2,ij}^{(n-1)}}{g_{2,ij}^{(n-1)} + g_{2,ij}^{(n-1)}}, \quad P_{2,ij}^{(n-1)} = 2 \frac{g_{2,ij}^{(n-1)} - g_{2,ij}^{(n-1)}}{g_{2,ij}^{(n-1)} + g_{2,ij}^{(n-1)}},$$

in which $l=1,2$. Substituting (26) into (22), we can obtain the following matrix equations by some manipulations:

$$P_{1,ij}^{(n-1)} X_{1,ij} = C_{1,ij}^{(n-1)},$$

$$P_{2,ij}^{(n-1)} X_{2,ij} = D_{ij}^{(n-1)},$$

where $P_{1,ij}^{(n-1)}$, $l=1,2$ are the tri-diagonal matrices which its elements for $l=1$ are defined as

$$P_{1,ij}^{(n-1)}(x,y) = \begin{cases} -\frac{\Delta x^2}{\Delta x^2} P_{1,ij}^{(n-1)}(x-1,y), & x = y - 1 \\ \frac{\Delta x^2}{\Delta x^2} P_{1,ij}^{(n-1)}(x,y), & x = y \\ \frac{\Delta x^2}{\Delta x^2} P_{1,ij}^{(n-1)}(x+1,y), & x = y + 1 \\ 0, & \text{otherwise} \end{cases}$$

in which the lost block size is assumed to be $L \times L$ and $P_{F0} = 2/\Delta x^2 + 2 + \beta/2$. For $l=2$, the matrix elements can be obtained similarly. In (28), $X_{1,ij}$ is a column vector containing the intermediate solutions corresponding to the $j$-th column of the erroneous part of the image defined as follows

$$X_{1,ij} = \left[ x_{1,ij}^{(n-1)}, x_{1,i+1,j}^{(n-1)}, \ldots, x_{1,i+L-1,j}^{(n-1)} \right]^T,$$

and $X_{2,ij}^{(n)}$ is a column vector containing the $n$th iteration solutions corresponding to the $i$-th row of the erroneous part of the image defined as follows

$$X_{2,ij}^{(n)} = \left[ x_{2,ij}^{(n)}, x_{2,i+1,j}^{(n)}, \ldots, x_{2,i+L-1,j}^{(n)} \right]^T,$$

and $C_{ij}^{(n-1)}$ is a column vector which each of its elements $C_{ij}^{(n-1)}$ equals to

$$C_{ij}^{(n-1)} = \Delta x^2 \frac{1}{2} P_{2,ij}^{(n-1)} x_{ij}^{(n-1)} - \frac{3}{4} \frac{\Delta x^2}{\Delta x^2} x_{ij}^{(n-1)} \right\}^{1/2}.$$
Correspondingly, $D^{(n-1)}_{ij}$ is a columnar vector which each of its elements $D^{2(n-1)}_{ij}$ equals,

$$D^{(n-1)}_{ij} = x_{i,j} + \Delta t \left\{ \frac{1}{2} p_{i,j}^{(n-1)} x_{i,j}^{(n-1)} + \left( 1 + \frac{1}{2} \right) x_{i,j}^{(n-1)} - \frac{1}{2} p_{i,j}^{(n-1)} x_{i,j}^{(n-1)} \right\}.$$ (33)

4. EXPERIMENTAL RESULTS

Fig. 1 shows the result of running our algorithm on the standard test image of baboon. The original $256 \times 256$ image is shown in Fig. 1-(a) and this image is correlated using Eq. (1) with $L_1 = L_2 = 65$ and $q_1 = q_2 = 63$, and then compressed with the compression ratio of $\rho = 0.32$ using the JPEG codec. The resultant image has the PSNR of 26.21 dB (Fig. 1-(b)). Fig. 1-(c) shows the same image in Fig. 1-(b) but with a lost $65 \times 65$ block. The recovered image is shown in Fig. 1-(d) which has the PSNR of 16.39 dB. Figs. 1-(e) and (f) show the denoised images resulted from applying the 2-D Wiener filter and our PDE-based image enhancement method, respectively. The PSNRs of the enhanced images in the restored block are 19.72 and 17.96 dB, respectively. All of the PSNRs are computed with respect to Fig. 1-(a). As can be seen, although Wiener filtering results in higher PSNR, the subjective quality of the image resulted from the proposed PDE-based method is better, since it preserved the edges of the image in the recovered region and it does not blur the image which results in the continuity of the image in the restored block with the surrounding regions.

Note that, since we could not find any other approach in the literature that can recover a large size lost block (e.g., a lost $64 \times 64$ block), comparing our algorithm with other existing ones is unfair. Therefore, we did not compare our experimental results with other existing recovery approaches.

In block based image and video codecs when an error occurs, some consecutive blocks constituting a slice are destroyed. To adjust our algorithm to deal with such errors, it suffices to choose an appropriate size for the number of zeros in the frequency domain. For example, one can make zero a rectangle of size $8 \times N_2$ in the DFT domain. At the receiver, we can reconstruct a corrupted slice of maximum size $8 \times N_2$.

Fig. 2 shows the PSNR of the noisy recovered block in three test images of baboon, lena, and boat and the PSNR of the enhanced blocks versus the compression ratio of JPEG codec. This parameter is directly related to the required quality we look for. In this figure, the enhanced PSNRs are shown by dashed lines. As can be seen, in the typical compression ratios, the PSNR of the enhanced image lies in the acceptable range of 20-30 dB.

5. CONCLUSION

In this paper, we have proposed a novel approach for restoration of lost blocks in the images which is efficient and fast enough to be implemented in the real-time applications. In the proposed method, we first artificially make correlation between pixels of the original image that distorts it. By adjusting appropriate parameters, one can control this distortion to lie in acceptable ranges. Then, using this correlation, we recovered the lost blocks of images without error in the case of lossless codecs.

We have also considered the error encountered by lossy codecs and have developed a PDE-based image enhancement approach to enhance this noisy image which preserves the edges of images and avoids blurring effects. Comparing our enhancement method with 2-D Wiener filter, we observed that although our algorithm leads to lower PSNRs, its subjective quality is better in the sense of preserving the edges. Moreover, our enhancement method preserves continuity between the recovered block and the surrounding regions, whereas this feature is not observed in 2-D Wiener filtering.

Our experiments show that the algorithm can restore large lost blocks of $65 \times 65$ pixels (or even more) in $256 \times 256$ images; the ability that other previously reported approaches have not achieved. Since we could not find any other recovery approach in the literature that can recover a large size lost block (e.g., a lost $64 \times 64$ block), comparing our algorithm with other existing ones will not be fair. Therefore, we did not compare our experimental results with other existing recovery approaches.

6. REFERENCES


