Lab 3: Spatial Error Concealment Techniques

Kyeong Soo (Joseph) Kim
Department of Electrical and Electronic Engineering
Xi'an Jiaotong-Liverpool University
April/20/2018

I. Introduction

In this Lab, you are to carry out spatial error concealment techniques for intra-coded frame. For simplicity, we do not consider quantization and block coding but directly work in the image domain. You are provided two grayscale 480x480 images — i.e., an original ('mandrill') and a damaged ('mandrill_damaged') — as 2x2 uint8 arrays stored in the MAT file "mandrills.mat"; the impact of packet losses are modeled as several 8x8 macroblocks whose values are set to zeros in the damaged image. A MATLAB script ("macroblocks.m") is also provided to demonstrate the conversion between an image array and a corresponding cell array of macroblocks and processing of those macroblocks to model packet loss effect.

II. Review of Spatial Error Concealment Techniques

Here we briefly review two simple spatial error concealment techniques proposed for MPEG-2 video codec [1], which are illustrated in Fig. 1 for 4x4 blocks within an 8x8 macroblock.

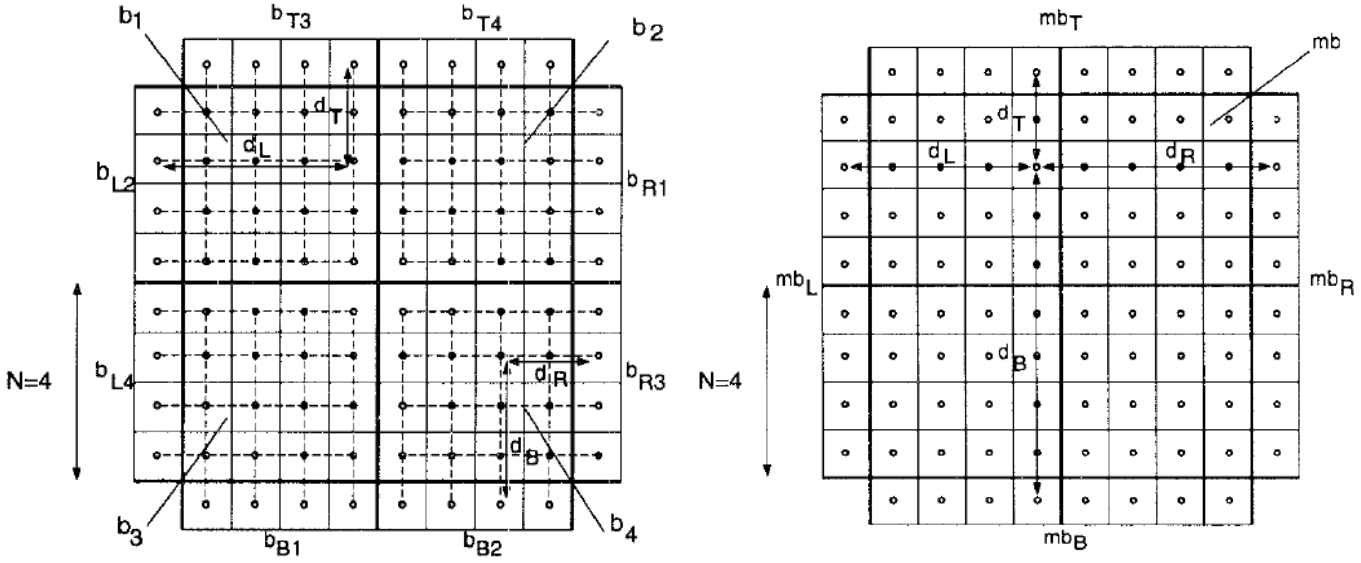

Fig. 1. Spatial interpolation techniques for error concealment: (a) Block based. (b) Macroblock based (from [1]).

Of the two techniques, the first one interpolates each single $N \times N$ block in one macroblock as shown in Fig. 1 (a). This interpolation can be described as follows: For $i,\ k = 1, \ldots, N$,

$$\begin{aligned} b_1(i,\ k) &= \frac{d_T b_{L2}(i,\ N) + d_L b_{T3}(N,\ k)}{d_L + d_T} \\ b_2(i,\ k) &= \frac{d_T b_{R1}(i,\ 1) + d_R b_{T4}(N,\ k)}{d_R + d_T} \\ b_3(i,\ k) &= \frac{d_B b_{L4}(i,\ N) + d_L b_{B1}(1,\ k)}{d_L + d_B} \\ b_4(i,\ k) &= \frac{d_B b_{R3}(i,\ 1) + d_R b_{B2}(1,\ k)}{d_R + d_B} \end{aligned} \tag{1}$$

where $b_l,\ l = 1, \ldots, 4$ is the l^{th} block of the current macroblock, $b_{Xl},\ l = 1, \ldots, 4$ with $X = L,\ R,\ T,\ B$ is the l^{th} block of the neighboured macroblock (Left, Right, Top, Bottom) and d_x with $X = L,\ R,\ T,\ B$ is the distance from the respective pixel of the block b_{Xl} to the current pixel $b_l(i,\ k)$.

The second technique interpolates each pixel of the whole macroblock with the adjacent pixels of the four neighbouring macroblocks. Fig. 1 (b) shows the macroblock with the boundary pixels of the neighbouring macroblocks. Each pixel of the current macroblock with the size $2N \times 2N$ will be concealed by simple interpolation of the four pixels of the surrounding macroblocks, i.e., for $i,\ k = 1, \ldots, 2N$,

$$\begin{aligned} mb(i,\ k) = \frac{1}{d_L + d_R + d_T + d_B} \Big(& d_R mb_L(i, 2N) + d_L mb_R(i, 1) + \\ & d_B mb_T(2N, k) + d_T mb_B(1, k) \Big) \end{aligned} \tag{2}$$

where mb is the current macroblock, mb_X with $X = L,\ R,\ T,\ B$ is the respective neighboured macroblock (Left, Right, Top, Bottom) and d_X with $X = L,\ R,\ T,\ B$ is the distance from the respective pixel of the macroblock mb_X to the current pixel $mb(i,\ k)$. This technique works better if the surrounding macroblocks exist. If some of the macroblocks do not exist for interpolation (e.g., if one whole stripe of macroblocks is damaged), the corresponding distance will be set to zero (for instance if mb_L do not exist d_R will be set to zero).

With only two available macroblocks mb_T and mb_B the equation (2) reduces to: For $i,\ k = 1, \ldots, 2N$,

$$mb(i,\ k) = \frac{d_B mb_T(2N, k) + d_T mb_B(1, k)}{d_T + d_B} \tag{3}$$

III. Task: Macroblock-Based Spatial Interpolation

For this task, you need to submit a Lab report with MATLAB script(s) for the following activities:

1) (5 points) Macroblock-based spatial interpolation.
   - Conceal the lost macroblocks of "mandrill_damaged" using the macroblock-based spatial interpolation technique described in Section II.
   - Display both the original ("mandrill") and the error-concealed images from step (1) for comparison (refer to the "macroblocks.m" in this regard).
   - Calculate PSNR of the error-concealed image with respect to the original image (again, refer to the "macroblocks.m" in this regard).
2) (5 points) Repeat 1) but this time using the more advanced *directional interpolation* technique described in [2, Section III].

The Lab report and MATLAB script(s) should be submitted in person and via email, respectively, by Friday, 25 May 2018.

APPENDIX
MACROBLOCKS.M: MATLAB SCRIPT FOR PROCESSING IMAGE USING MACROBLOCKS

```
% Convert a grayscale image into 8x8 macroblocks and process them to model
% packet loss impact.

mbsize = 16;          % size of macroblock (MB)

load('mandrills');  % load mandrill image (480x480 uint8 matrix)
figure(1);          % show original image
imshow(mandrill);

[imgx, imgy] = size(mandrill);  % image dimensions
mbx = imgx / mbsize;            % number of MBs in X
mby = imgy / mbsize;            % number of MBs in Y

mbs = cell(mbx, mby);           % cell array storing MBs
for i=1:mbx
    for j=1:mby
        mbs{i,j} = mandrill(mbsize*(i-1)+1:mbsize*i, mbsize*(j-1)+1:mbsize*j);
    end
end

% random discard of MBs to model packet loss impact
nd = 128;     % number of MBs to discard
xs=unidrnd(mbx, 1, nd);         % X index of MBs to discard
ys=unidrnd(mby, 1, nd);         % Y index of MBs to discard
damaged_mbs = mbs;
for i=1:nd
    damaged_mbs{xs(i),ys(i)}=zeros(mbsize,mbsize);     % set values of discarded MBs to 0
end

% convert damaged MBs back to an image
Y = zeros(imgx, imgy);
for i=1:mbx
    for j=1:mby
        Y(mbsize*(i-1)+1:mbsize*i,mbsize*(j-1)+1:mbsize*j)=damaged_mbs{i,j};
    end
end

mandrill_damaged = uint8(Y);     % convert to uint8 as a grayscale image
figure(2);                       % show damaged image
imshow(mandrill_damaged);

fprintf('\nPSNR = %0.4f\n', psnr(mandrill_damaged, mandrill));    % PSNR of damaged image
```

REFERENCES

[1] S. Aign and K. Fazel, “Temporal & spatial error concealment techniques for hierarchical MPEG-2 video codedc,” in Proc. 1995 IEEE ICC, Seattle, WA, USA, Jun. 1995, pp. 1778–1783.

[2] W.-Y. Kung, C.-S. Kim, and C.-C. J. Kuo, “Spatial and temporal error concealment techniques for video transmission over noisy channels,” IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 7, pp. 789–802, Jul. 2006.