TERM PAPER REPORT

EE-603

Advance Topics in Digital Filtering

PRACTICAL LOW BIT RATE PREDICTIVE IMAGE CODER USING MULTI-RATE PROCESSING

Under the guidance of

Prof. Sumana Gupta

Department of Electrical Engineering

IIT Kanpur

Submitted by:

Neha Tiwari (Y4225)

Sankalp Gulati (Y4383)

PRACTICAL LOW BIT RATE PREDICTIVE IMAGE CODER USING MULTI-RATE PROCESSING

EE 603, Advanced Digital Filtering

I. INTRODUCTION

Lossy compression in time domain is achieved by using Differential Pulse Code Modulation (DPCM). The encoder consists of a scalar quantizer and a feedback prediction loop with a predictive filter. The differencebetween the discrete time input sequence and its prediction, is quantized and coded for transmission. At the same time, the quantized prediction error is combined with previously predicted samples to form the input for the predictor.

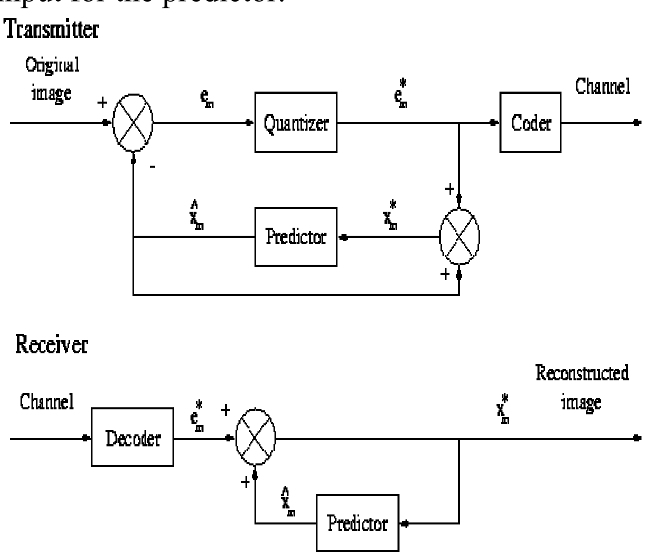


Figure 1: Signal Flow Chart of Predictor

Here,

$$\begin{aligned} e(n) &= x(n) - \hat{x}(n) \\ e^*(n) &= Q(e(n)) \\ x^*(n) &= \hat{x}(n) + e^*(n) \\ \hat{x}(n) &= P(x^*(n)) \end{aligned} \quad (1)$$

where Q is the quantizer and P is the prediction filter.

From the above equations it is clear that quantization error (e(n)-e*(n)) depends on the number of bits used in the quantizer. If the DPCM has high bit rate i.e. the number of bits assigned in the quantizer is more, the performance will be better. But that contradicts with the goal of compression. DPCM with low bit rates have poor performance.

So a modified DPCM structure is used where the input bit rate is reduced by downsampling the input image while the DPCM uses 4 to 5 bits for quantization. To avoid aliasing the image has to be passed through a low pass filter before downsampling. This can cause severe loss of data and cannot be implemented in images where there are large high frequency components. A more formal explanation can be found in the rate distortion theory approach which we will discuss next.

II. RATE DISTORTION THEORY

In a signal to be transmitted, there are certain frequencies where the power, as seen from the power spectral density curve, is equal to or less than the power of noise in the channel (assuming it is white). Now the signal at these frequencies will be heavily distorted by the channel noise and so if these frequencies are not transmitted the reconstructed signal will not be affected much.

In case of DPCM coding with uniform quantization, the channel noise corresponds to quantization error in the DPCM. If the step size is Δ, the quantization noise is given by $\Delta^2/12$. This noise is assumed to be uncorrelated (white).

Now if the input image has a decreasing Fourier transform, the frequencies with significant power content can be extracted by low pass filtering. The resulting signal can be downsampled accordingly (i.e. aliasing avoided). Thus the input bit rate can be reduced for such images without severe distortion.

III. BASIC TECHNIQUE

The input image is first passed through a low pass filter and then downsampled. The downsampled image is then passed through DPCM where the difference between this image and the predictor output is quantized.

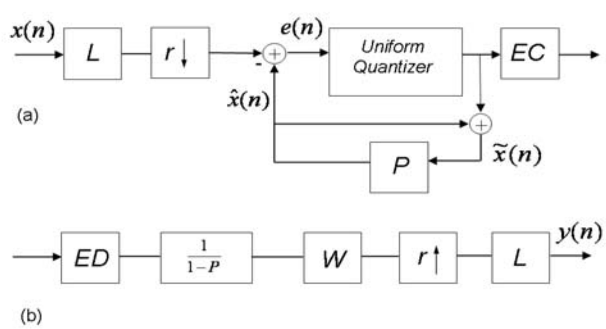


Figure 2: Block Diagram of Image Coder: (a) Encoder (b) Decoder. EC: Entropy Coder, ED: Entropy Decoder, L Lowpass Filter, P: Prediction Filter, W: Wiener Filter, r: Decimation/Interpolation Rate

As e(m,n) is difference of predicted value and original value, most of the values will be small and can be represented by 4-5 bits using a uniform quantizer. Non uniform quantizer causes error propagation in predictor and gives bad results.

The output of the 2D predictor used is given by:

$$\hat{x}(i,j) = \rho_1 \tilde{x}(i-1,j) + \rho_2 \tilde{x}(i,j-1) - \rho_1\rho_2 \tilde{x}(i-1,j-1)$$
$$|\rho_1|<1,\ |\rho_2|<1 \qquad (2)$$

Wiener filter is added to give an optimal trade off between suppression of additive quantization noise and linear distortion due to filtering.

Because of downsampling and then DPCM, two step compression is achieved. If initially each pixel was represented by 8 bits then the percentage compression can be calculated as

$$C = 100(1 - n/(8r^2))$$

where,

n = no. of bits used to represent error

r = downsampling factor

So for 80% compression an image with 8 bits/pixel representation can be almost recovered from 1.6 bits/pixel data. If entropy coding is used, the compression will be further increased. So the bits per pixel value should reduce further.

IV. SIMULATIONS

Simulation of the block diagram in Fig 1 was done in MATLAB, trying out different values of decimation factor, quantization and DPCM coefficients. Description about individual blocks and their codes is given below.

Low Pass Filtering: Low pass filtering before decimation was done using an order 50 filter having frequency response as follows.

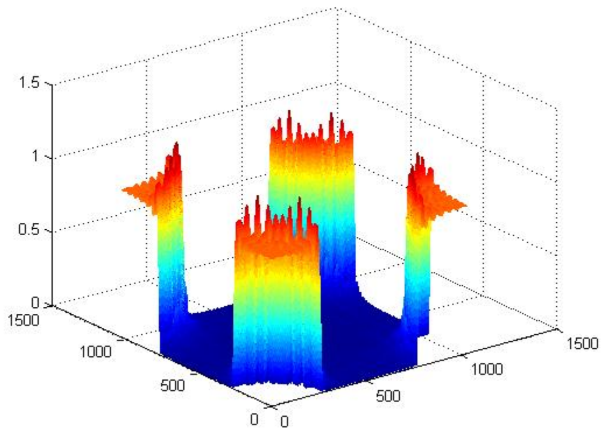


Figure 3: Frequency Response of the order 50 low pass filter.

Code:

```
function filter_matrix = lp(order, cutoff);
%Create desired frequency response
[f1,f2] = freqspace(order,'meshgrid');
d = find(f1.^2+f2.^2 < cutoff^2);
Hd = zeros(order);
Hd(d) = ones(size(d));
% Design the filter's impulse response
filter_matrix = fsamp2(Hd);
```

Decimation: Different values of decimation factors were tried out. Decimation of the image was done using MATLAB's inbuilt command.
imresize(image,factor);

DPCM: As shown in EQ.1 the prediction for pixel (i, j) id done using three neighboring pixels (i,j-1), (i-1,j), (i-1,j-1). A wide range of coefficients were tried for the prediction. Quantization of the error signal is done in the DPCM module itself by using a uniform quantizer in the range |a| with clipped values if they are above |a|.

Code:

```
function [r,xtilde]=dpcm2d(x,a,clip)
% Usage: [r,xtilde]=dpcm2d(x,a)
% Differential Pulse Coded Modulation
% x: input source vector to be encoded
% a: prediction filter
% r: residue vector quantized to integer
%
%    r(t) = Q[x(t)-xhat(t)] = Q[x(t)- sum
a(i)xtilde(t-i)]
%    xtilde(t) = xhat(t) + r(t)
x=int16(x(:,:,1));
[m,n]=size(x);
xhat=zeros(m,n);
r=zeros(m,n);
xtilde=zeros(m,n);

r(1,1:n)=x(1,1:n);
r(2:m,1)=x(2:m,1);

xtilde(1,1:n)=r(1,1:n);
xtilde(2:m,1)=r(2:m,1);

for (i=2:1:m)
for (j=2:1:n)
 xhat(i,j)=(a(1)*xtilde(i-1,j)+ a(1)*xtilde(i,j-1)-
a(1)*a(1)*xtilde(i-1,j-1));
r(i,j)=round((x(i,j)-xhat(i,j)));
r(i,j)=quant(r(i,j),clip);

xtilde(i,j)=(xhat(i,j)+r(i,j));
end
end
```

Quantization:

```
function value=quant(a,c)
```

```
if (a>c)
    a=c;
end
if (a<-c)
    a=-c;
end
value=round(a);
```

Dpcm Decoding code:

```
function [xtilde]=dpcm2d_dec(r,a)
[m,n]=size(r);
xtilde=zeros(m,n);
xhat=zeros(m,n);

xtilde(1,1:n)=r(1,1:n);
xtilde(2:m,1)=r(2:m,1);

for (i=2:1:m)
for(j=2:1:n)

xhat(i,j)=a(1)*xtilde(i-1,j)+a(1)*xtilde(i,j-1)-
a(1)*a(1)*xtilde(i-1,j-1);
xtilde(i,j)=xhat(i,j)+r(i,j);
if(xtilde(i,j)<0)
xtilde(i,j)=0;
end
if(xtilde(i,j)>1)
xtilde(i,j)=1;
end
end
end
```

Wiener filtering: Wiener lowpass-filters an intensity image that has been degraded by constant power additive noise. Wiener uses a pixelwise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.

$$I_{out}(i,j) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(I_{in}(i,j) - \mu)$$

Where, μ is the local mean of the image, σ is the local variance of the image, ν is the variance of the noise.

Bilinear Interpolation: Decimated image is converted to its original sized image by doing Bilinear interpolation, which can be accomplished using an inbuilt command in MATLAB.

Code:

```
imresize (image,factor,'bilinear');
```

PSNR Calaulation: Peak-Peak Signal to Noise Ratio of the reconstructed image was calculated to judge the image quality.

Code:

```
function result=PSNR(A,B)

if A == B
  error('Images are identical: PSNR has infinite
value')
end

max2_A = max(max(A));
max2_B = max(max(B));
min2_A = min(min(A));
min2_B = min(min(B));

if max2_A > 1 | max2_B > 1 | min2_A < 0 |
min2_B < 0
  error('input matrices must have values in the
interval [0,1]')
end
 error1 = A - B;
result =
20*log10(1/(sqrt(mean(mean(error1.^2)))));
```

V. RESULTS

PSNR of the reconstructed images was calculated for different values of DPCM coefficients, decimation factors, and bits per sample of error signal. Observed results are as follows:

1. DPCM coefficients' value giving best PSNR was found to be 0.84 for all other parameters. (Fig below)

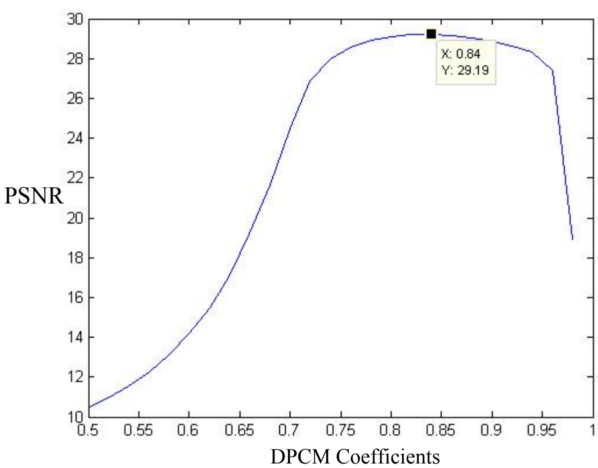


Figure 3: PSNR Vs DPCM coefficients

2. Without Wiener Filtering **(**For a=0.84)

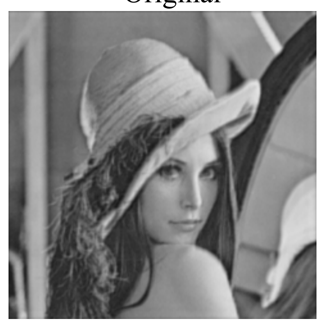
Comparison Table

Serial No.	Decimation factor	Bits/error sample	SNR dB	Compression (%)
1	1.5	5	29.19	72.22
2	1.5	4	25.74	77.78
3	2	5	28.18	84.37
4	2	4	24.98	87.5
5	3	5	24.85	93.05
6	3	4	21.89	94.44

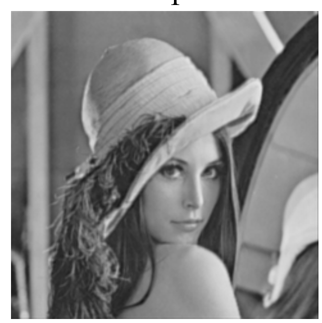
Images corresponding to the parameters set above are:

Original

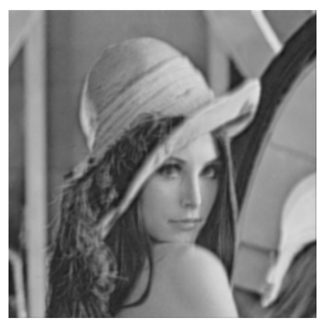
1



2



3



4

5

6

3. With Wiener Filtering (For a=0.84)

   It was found that after applying wiener filter the PSNR value was lower than the values obtained without wiener filtering as shown in the below given table

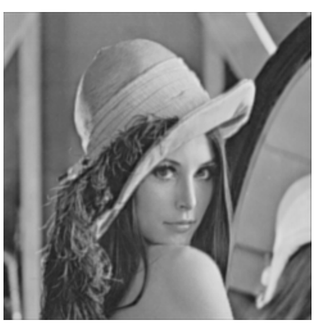
Comparison Table

Serial No.	Decimation factor	Bits/error sample	SNR dB	Compression (%)
1	1.5	5	28.75	72.22
2	1.5	4	25.52	77.78
3	2	5	27.75	84.37
4	2	4	24.75	87.5
5	3	5	24.45	93.05
6	3	4	21.66	94.44

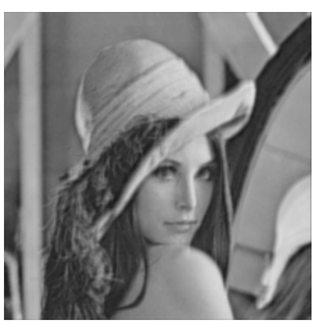
Images corresponding to above parameters are:

1

2



3



4

5

6

4. Wiener filtering gives a good result when the image is grainy, but in our case the predicted values were close enough to the original ones to avoid this types of noise. Hence wiener filtering in our case just caused blurring of the image, reducing the PSNR value.

VI. CONCLUSION

It is clear from the above result that the number of bits used in quantizer have a greater effect on PSNR ad image quality as compared to decimation factor (entries 1,2,3 of the comparison table). Hence it was a prudent technique to keep the DPCM operating at a high rate while decreasing te number of input bits by decimation.

VII. REFERNCE

1. *Anna N. Kim and Tor A. Ramstad*, “Practical Low Bit Rate Predictive Image Coder Using Multi-Rate Processing And Adaptive Entropy Coding”

2. *A.N. Kim and T.A. Ramstad*, “Improving the rate distortion performance of dpcm,” in Proceedings of Int. Symposium on Signal Proc. and its Applications (ISSPA), 2003.