**Image and Video Processing**

**Programming Assignment**

**Week – 11 Video Coding**

clear all

% Block based hybrid coder for p-frame

% Read video etc

vidObj = VideoReader('Radiohead.mp4');

nFrames = vidObj.NumberOfFrames;

vidH = vidObj.Height;

vidW = vidObj.Width;

mov(1:nFrames) = struct('cdata',zeros(vidH,vidW,3,'uint8'),'colormap',[]);

% movDiff(1:nFrames) = struct('cdata',zeros(vidH,vidW,3,'uint8'),'colormap',[]);

disp('Reading...');

for k = 2100:2300

mov(k).cdata = read(vidObj,k);

end

% Uni-directional prediction with integer or half pel EBMA

f1 = double(mov(2200).cdata);

f1 = (1/3)\*(f1(:,120:479,1)+f1(:,120:479,2)+f1(:,120:479,3));

% f1 = (1/3)\*(f1(:,:,1)+f1(:,:,2)+f1(:,:,3));

f2 = double(mov(2203).cdata);

f2 = (1/3)\*(f2(:,120:479,1)+f2(:,120:479,2)+f2(:,120:479,3));

% f2 = (1/3)\*(f2(:,:,1)+f2(:,:,2)+f2(:,:,3));

dimOne = size(f1);

% Integer Pel EBMA

fp = zeros(size(f1));

N = 24;

R = 24;

% vidH = dimOne(1);

vidWid = vidH;

%For every block

for i = R+1:N:vidH-N-R

for j = R+1:N:vidWid-N-R

MAD\_min = 256\*N\*N;

mvx = 0; mvy = 0;

%For every search range either side of a pixel

for k = -R:1:R

for l = -R:1:R

MAD = sum(sum(abs(f1(i:i+N-1,j:j+N-1)-f2(i+k:i+k+N-1,j+l:j+l+N-1))));

%Look for minimum error and substitute it everytime a lower

%error is found

if MAD < MAD\_min

MAD\_min = MAD;

dy = k;

dx = l;

end

end

end

%Use these MV values to move pixels in the frame

fp(i:i+N-1,j:j+N-1) = f2(i+dy:i+dy+N-1,j+dx:j+dx+N-1);

iblk = floor((i-1)/N+1);

jblk = floor((j-1)/N+1);

% disp(dx);disp(dy);

mvx(iblk,jblk) = dx; mvy(iblk,jblk) = dy;

end

end

fpInter = fp;

% Range = +-24

eInter = f2 - fpInter;

varInter = var(var(eInter));

% Find best intra prediction mode and the corresponding error block and its

% variance

[eIntra, varIntra, fpIntra] = intraP(f2);

% Choose prediction block whose error has smaller variance

if (eIntra < eInter)

eB = eIntra;

varB = varIntra;

fPredict = fpIntra;

else

varB = varInter;

eB = eInter;

fPredict = fpInter;

end

% Quantize all DCT coefficients with same step size Q

q = 1:11;

coeff = dct2(eB);

dimCoeff = size(coeff);

% meanV = mean(mean(coeff));

for iter = 1:length(q);

for i = 1:dimCoeff(1)

for j = dimCoeff(2)

% Check if first coefficient, in which case use mean = 128

quantC(i,j) = floor((coeff(i,j))/q(iter))\*q(iter);

end

end

% Count non-zero coefficients

nonZero = 0;

for i = 1:dimCoeff(1)

for j = dimCoeff(2)

% Check if first coefficient, in which case use mean = 128

if quantC(i,j) ~= 0

nonZero = nonZero+1;

else

continue;

end

end

end

% Reconstruct error block with quantized coeffs

eRecon = idct2(quantC);

% Reconstruct block by adding quantized error to prediction

fReckon = eRecon + fPredict;

% non-zero DCT coeffs - to measure bitrate

% Apply to 2 frames a for varous values of q and plot PSNR

PSNR(iter) = psnr(f2,fReckon);

% disp(PSNR);

end

plot(q,PSNR);

xlabel('Q'),ylabel('PSNR');

