Channel Coding Project

Group Members:

Abdelrahman Amr	34-1762	T-4
Zeyad Rehab	34-2285	T-4
Rana Ashraf	34-1512	T-5
Kareem Hesham	34-2877	T-4
Seif Ahmed	34-9798	T-5

Channel Coding Code:

```
clc
clear all
% Read Highway Video
obj = VideoReader('highway.avi');
a= read(obj);
frames=get(obj,'NumberOfFrames');

% extracting Frames
for i=1:frames
     I(i).cdata=a(:,:,:,i);
end

s=size(I(1).cdata);
mov(1:frames) =struct('cdata', zeros(s(1),s(2), 3,
'uint8'),'colormap', []);
% Trellis Generate rate half
t = poly2trellis(7,[171 133]);

% Probability vector with 6 elements
probvector=[0.0001:0.03998:0.2];
errorBitswithinc=zeros(1,length(probvector));
throughput=zeros(1,length(probvector));

% puncturing matrix
punGeneral=[1 1 1 0 1 0 1 0 0 1 1 0 1 0 1 0;1 1 1 0 1 0 1 0 1 1 1 0 1
0 1 0;1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0;1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
0;1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];

for idx = 1:length(probvector)

       errornumber=0;
       sentBits=0;

for Frame=1:frames
%Red Components of the Frame
R=I(Frame).cdata(:,:,1);
%Green Components of the Frame
G=I(Frame).cdata(:,:,2);
%Blue Components of the Frame
B=I(Frame).cdata(:,:,3);

% Reshaping Red Bits
[sz1,sz2]=size(R);
redBits=reshape(R,1,sz1*sz2);
% Reshaping Green Bits
[sz1,sz2]=size(G);
greenBits=reshape(G,1,sz1*sz2);
% Reshaping Blue Bits
[sz1,sz2]=size(B);
blueBits=reshape(B,1,sz1*sz2);

redBits=double(redBits);
redBits=de2bi(redBits);
redBits=reshape(redBits,1,[]);

greenBits=double(greenBits);
```

```
greenBits=de2bi(greenBits);
greenBits=reshape(greenBits,1,[]);

blueBits=double(blueBits);
blueBits=de2bi(blueBits);
blueBits=reshape(blueBits,1,[]);

% Concatinating bit final stream
totalBits=horzcat(redBits,greenBits,blueBits);

length(totalBits);
length(totalBits)/1024;
%we knownow that the total number of pkts is 594 , use this number
again

% Packets to be encoded
pkts=reshape(totalBits,594,1024);
% matrix to recieve decoded data
decoded=zeros(594,1024);


% 2 loops to apply puncturing if the recieved packet is not equal to
the
% data before encoding we increment the puncturing rate
for i=1:594
    j=1;
   while j<=5
        code = convenc(pkts(i,:),t,punGeneral(j,:));
        p = probvector(idx);
        recieved = bsc(code,p);
        decoded(i,:) =
vitdec(recieved,t,35,'trunc','hard',punGeneral(j,:));
        if(decoded(i,:)==pkts(i,:))
            %number of sent bits
           sentBits=sentBits+length(code);
            j=10;
        else
            j=j+1;
        end
   end
    if(j==6)
        %number of sent bits
       sentBits=sentBits+length(code);
       %error bits
       locs = pkts(i,:)~=decoded(i,:);
       errornumber = errornumber+sum(locs);
    end
end
% Reshaping decoded data into red, green, blue and writing video
% we know the limits of the red green and blue by the dimensions
%144*176*8=202752
%increment once again to get green then blue
%we know now that the total number of bits is 608256

totalBitsRecived=reshape(decoded,1,[]);
redBitsRecived=totalBitsRecived(1,1:202752);
greenBitsRecived=totalBitsRecived(1,202753:405504);
blueBitsRecived=totalBitsRecived(1,405505:608256);


redBitsRecived=reshape(redBitsRecived,25344,8);
```

```
redBitsRecived=bi2de(redBitsRecived);
redBitsRecived=uint8(redBitsRecived);
redBitsRecived=reshape(redBitsRecived,144,176);

greenBitsRecived=reshape(greenBitsRecived,25344,8);
greenBitsRecived=bi2de(greenBitsRecived);
greenBitsRecived=uint8(greenBitsRecived);
greenBitsRecived=reshape(greenBitsRecived,144,176);

blueBitsRecived=reshape(blueBitsRecived,25344,8);
blueBitsRecived=bi2de(blueBitsRecived);
blueBitsRecived=uint8(blueBitsRecived);
blueBitsRecived=reshape(blueBitsRecived,144,176);

mov(1,Frame).cdata(:,:,1) = redBitsRecived;
mov(1,Frame).cdata(:,:,2) = greenBitsRecived;
mov(1,Frame).cdata(:,:,3) = blueBitsRecived;
    end
  errorBitswithinc(idx)=errornumber;
  throughput(idx)=(608256*30)/sentBits;
end

errorBitswithinc = errorBitswithinc./(608256*30);
figure(1)
plot(probvector,errorBitswithinc)
title('Bit error Rate Using Incremental redundancy')
figure(2)
plot(probvector,throughput)
title('throughput Using Incremental redundancy')

writer = VideoWriter('Videos.avi','Uncompressed AVI');
writer.FrameRate=obj.FrameRate;
open(writer);
writeVideo(writer,mov);
close(writer);
```

Channel Coding With no Incremental Code :

```
clc
clear all
% Read Highway Video
obj = VideoReader('highway.avi');
a= read(obj);
frames=get(obj,'NumberOfFrames');

% extracting Frames
for i=1:frames
     I(i).cdata=a(:,:,:,i);
end

s=size(I(1).cdata);
mov(1:frames) =struct('cdata', zeros(s(1),s(2), 3,
'uint8'),'colormap', []);
% Trellis Generate rate half
t = poly2trellis(7,[171 133]);

% Probability vector with 6 elements
probvector=[0.0001:0.03998:0.2];
errorBitswithinc=zeros(1,length(probvector));

for idx = 1:length(probvector)

      errornumber=0;

    for Frame=1:frames
% Red Components of the Frame
R=I(Frame).cdata(:,:,1);
% Green Components of the Frame
G=I(Frame).cdata(:,:,2);
% Blue Components of the Frame
B=I(Frame).cdata(:,:,3);

% Reshaping Red Bits
[sz1,sz2]=size(R);
redBits=reshape(R,1,sz1*sz2);
% Reshaping Green Bits
[sz1,sz2]=size(G);
greenBits=reshape(G,1,sz1*sz2);
% Reshaping Blue Bits
[sz1,sz2]=size(B);
blueBits=reshape(B,1,sz1*sz2);

redBits=double(redBits);
redBits=de2bi(redBits);
redBits=reshape(redBits,1,[]);

greenBits=double(greenBits);
greenBits=de2bi(greenBits);
greenBits=reshape(greenBits,1,[]);

blueBits=double(blueBits);
blueBits=de2bi(blueBits);
blueBits=reshape(blueBits,1,[]);
```

```
% Concatinating bit final stream
totalBits=horzcat(redBits,greenBits,blueBits);

length(totalBits);
length(totalBits)/1024;
% Packets to be encoded
pkts=reshape(totalBits,594,1024);
% matrix to recieve decoded data
decoded=zeros(594,1024);

% loop on all packets and recieve the packet even with error
for i=1:594
        code = convenc(pkts(i,:),t);
        p = probvector(idx);
        recieved = bsc(code,p);
        decoded(i,:) = vitdec(recieved,t,35,'trunc','hard');
        locs = pkts(i,:)~=decoded(i,:);
        errornumber = errornumber+sum(locs);
end

% Reshaping decoded data into red, green, blue and writing video
% we know the limits of the red green and blue by the dimensions
%144*176*8=202752
%increment once again to get green then blue
totalBitsRecived=reshape(decoded,1,[]);
redBitsRecived=totalBitsRecived(1,1:202752);
greenBitsRecived=totalBitsRecived(1,202753:405504);
blueBitsRecived=totalBitsRecived(1,405505:608256);

redBitsRecived=reshape(redBitsRecived,25344,8);
redBitsRecived=bi2de(redBitsRecived);
redBitsRecived=uint8(redBitsRecived);
redBitsRecived=reshape(redBitsRecived,144,176);

greenBitsRecived=reshape(greenBitsRecived,25344,8);
greenBitsRecived=bi2de(greenBitsRecived);
greenBitsRecived=uint8(greenBitsRecived);
greenBitsRecived=reshape(greenBitsRecived,144,176);

blueBitsRecived=reshape(blueBitsRecived,25344,8);
blueBitsRecived=bi2de(blueBitsRecived);
blueBitsRecived=uint8(blueBitsRecived);
blueBitsRecived=reshape(blueBitsRecived,144,176);

mov(1,Frame).cdata(:,:,1) = redBitsRecived;
mov(1,Frame).cdata(:,:,2) = greenBitsRecived;
mov(1,Frame).cdata(:,:,3) = blueBitsRecived;
    end
  errorBitswithinc(idx)=errornumber;
end
figure(1)
errorBitswithinc = errorBitswithinc./(608256*30);
plot(probvector,errorBitswithinc)
title('Bit error Rate without Using Incremental redundancy')

% writer = VideoWriter('Video.avi','Uncompressed AVI');
% writer.FrameRate=obj.FrameRate;
% open(writer);
% writeVideo(writer,mov);
% close(writer);
```

No Channel Coding Code:

```
clc
clear all
% Read Highway Video
obj = VideoReader('highway.avi');
a= read(obj);
frames=get(obj,'NumberOfFrames');

%extracting Frames
for i=1:frames
     I(i).cdata=a(:,:,:,i);
end

s=size(I(1).cdata);
mov(1:frames) =struct('cdata', zeros(s(1),s(2), 3,
'uint8'),'colormap', []);
% Trellis Generate rate half

t = poly2trellis(7,[171 133]);


for Frame=1:frames
%Red Components of the Frame
R=I(Frame).cdata(:,:,1);
%Green Components of the Frame
G=I(Frame).cdata(:,:,2);
%Blue Components of the Frame
B=I(Frame).cdata(:,:,3);

% Reshaping Bits

[sz1,sz2]=size(R);
redBits=reshape(R,1,sz1*sz2);

[sz1,sz2]=size(G);
greenBits=reshape(G,1,sz1*sz2);

[sz1,sz2]=size(B);
blueBits=reshape(B,1,sz1*sz2);

redBits=double(redBits);
redBits=de2bi(redBits);
redBits=reshape(redBits,1,[]);

greenBits=double(greenBits);
greenBits=de2bi(greenBits);
greenBits=reshape(greenBits,1,[]);

blueBits=double(blueBits);
blueBits=de2bi(blueBits);
blueBits=reshape(blueBits,1,[]);
% Concatinating bit final stream
totalBits=horzcat(redBits,greenBits,blueBits);
```

```
length(totalBits);
length(totalBits)/1024;
% Packets to be encoded
pkts=reshape(totalBits,594,1024);
% matrix to recieve decoded data
decoded=zeros(594,1024);


for i=1:594
        p=0.1;
        recieved = bsc(pkts(i,:),p);
        decoded(i,:) = recieved;
end

totalBitsRecived=reshape(decoded,1,[]);
redBitsRecived=totalBitsRecived(1,1:202752);
greenBitsRecived=totalBitsRecived(1,202753:405504);
blueBitsRecived=totalBitsRecived(1,405505:608256);

redBitsRecived=reshape(redBitsRecived,25344,8);
redBitsRecived=bi2de(redBitsRecived);
redBitsRecived=uint8(redBitsRecived);
redBitsRecived=reshape(redBitsRecived,144,176);

greenBitsRecived=reshape(greenBitsRecived,25344,8);
greenBitsRecived=bi2de(greenBitsRecived);
greenBitsRecived=uint8(greenBitsRecived);
greenBitsRecived=reshape(greenBitsRecived,144,176);

blueBitsRecived=reshape(blueBitsRecived,25344,8);
blueBitsRecived=bi2de(blueBitsRecived);
blueBitsRecived=uint8(blueBitsRecived);
blueBitsRecived=reshape(blueBitsRecived,144,176);

mov(1,Frame).cdata(:,:,1) = redBitsRecived;
mov(1,Frame).cdata(:,:,2) = greenBitsRecived;
mov(1,Frame).cdata(:,:,3) = blueBitsRecived;
    end

%write the movie
writer = VideoWriter('Video.avi','Uncompressed AVI');
writer.FrameRate=obj.FrameRate;
open(writer);
writeVideo(writer,mov);
close(writer);
```