

主函数

```
1  clc;clear
2  [S,fs]=audioread('C:\Users\29066\Desktop\11.wav',[20000,200000]);
3  sound(S,fs);    %播放原视频
4  S=decimate(S,10);%采样
5  fs=fs/10
6
7
8  biggest=max(S);
9  %% 接收信号误码率随信噪比变化的曲线
10 snr=25:5:100
11 for i=1:length(snr)
12     encode_A13 = pcmCode_13(S);          %A律13折线编码
13     noisy_A13=addNoise(encode_A13 ,snr(i)); %信道传输加噪声
14     decode_A13 =pcmDecode_13(noisy_A13,biggest); %解码
15     error13(i)=errorRate(encode_A13,noisy_A13); %误码率
16     reality13(i)=distorRate(S,decode_A13);    %失真度
17     fsave(13,snr(i),decode_A13,fs)    %保存并播放音乐
18
19     encode_A11 = pcmCode_11(S);
20     noisy_A11=addNoise(encode_A11,snr(i));
21     decode_A11=pcmDecode_11(noisy_A11);
22     error11(i)=errorRate(encode_A11,noisy_A11);
23     reality11(i)=distorRate(S,decode_A11);
24     fsave(11,snr(i),decode_A11,fs)    %保存并播放音乐
25 end
26
27 figure('name','误码率和信噪比的关系');
28 plot(snr,error13,snr,error11);
29 xlabel('信噪比');ylabel('误码率');title('误码率伴随信噪比曲线');
30 legend('A律13折','11位均匀量化');
31
32 figure('name','失真度和信噪比的关系');
33 plot(snr,reality13,snr,reality11);
34 xlabel('信噪比');ylabel('失真度');title('失真度伴随信噪比曲线');
35 legend('A律13折','11位均匀量化');
36
```

添加噪声

```

1 function x= addNoise(S,snr)
2 %snr是信噪比
3 x=awgn(S,snr);
4 for i=1:length(x)
5     if x(i)>0.65
6         x(i)=1;
7     else x(i)=0;
8     end
9 end
10 end
11

```

失真度

```

1 function sum=distorRate(x,y)
2     %返回失真度
3     %x是原来的信号，y是经过传输解码后得到的]
4     sum=0;
5     len=length(y);
6     for i=1:len
7         sum=sum+(x(i)-y(i))^2/len;
8     end
9 end

```

保存音乐并且播放

```

1 function fsave(kind,snr,data,fs)
2     %存储声音
3     name=['UN',num2str(kind),'_-',num2str(snr),'.wav'];
4     audiowrite(name,data,fs);%转化为音频文件
5     file=audioread(name);
6     sound(file,fs);
7     pause(3);
8 end

```

11位均匀编码

```

1 function x=pcmCode_11(S)
2 %11位均匀量化编码
3     z=sign(S);
4     MaxS=max(abs(S));           %求s的最大值
5     S=abs(S/MaxS);              %归一化
6     Q=floor(1024*S);            %量化
7     L=length(Q);
8     x=zeros(1,11*L);
9
10    for i=1:L
11        t=i*11-10;

```

```

12         x(t)=z(i)>0;
13
14         Q(i)=Q(i)-1;
15         if Q(i)<0
16             Q(i)=0;
17         end
18
19         x(t+1:t+10)=de2bi(Q(i),10);
20     end
21 end

```

11位均匀解码

```

1 function y=pcmDecode_11(x)
2 % 11位均匀解码
3 L=length(x);
4 y=zeros(1,L/11);
5 now=1
6 for i=1:L/11
7     t=11*i-10;
8
9     for d=t+10:-1:t+1
10         if x(d)>0
11             y(i)=y(i)+now;
12         end
13         now=now*2;
14     end
15
16     if x(t)<1
17         y(i)=y(i)*(-1);
18     end
19 end
20 end

```

13位编码

```

1 function code=pcmCode_13(S)
2     z=sign(S); %判断S的正负
3     MaxS=max(abs(S)); %求S的最大值
4     S=abs(S/MaxS); %归一化
5     Q=2048*S; %量化
6     code=zeros(length(S),8); %代码存储矩阵（全零）
7
8     for i=1:length(S)
9         code(i,1)=z(i)>0;
10         loc=0; %用来确定段内位置
11         %确定段落码
12         if Q(i)>=1024
13             code(i,2:4)=[1,1,1];
14             loc=ceil((Q(i)-1024)/64);
15         elseif Q(i)>=512
16             code(i,2:4)=[1,1,0];
17             loc=ceil((Q(i)-512)/32);

```

```

18         elseif Q(i)>=256
19             code(i,2:4)=[1,0,1];
20             loc=ceil((Q(i)-256)/16);
21         elseif Q(i)>=128
22             code(i,2:4)=[1,0,0];
23             loc=ceil((Q(i)-128)/8);
24         elseif Q(i)>=64
25             code(i,2:4)=[0,1,1];
26             loc=ceil((Q(i)-64)/4);
27         elseif Q(i)>=32
28             code(i,2:4)=[0,1,0];
29             loc=ceil((Q(i)-32)/2);
30         elseif Q(i)>=16
31             code(i,2:4)=[0,0,1];
32             loc=ceil((Q(i)-16)/1);
33         else
34             code(i,2:4)=[0,0,0];
35             loc=ceil((Q(i)-0)/1);
36         end
37
38         if loc==0
39             code(i,(5:8))=[0,0,0,0];           %如果输入为零则输出"0"
40         else
41             k=num2str(dec2bin(loc-1,4));         %编码段内码为二进制
42             code(i,5)=str2num(k(1));
43             code(i,6)=str2num(k(2));
44             code(i,7)=str2num(k(3));
45             code(i,8)=str2num(k(4));
46         end
47     end
48
49     code = reshape(code', 1, []);
50 end
51

```

13位解码

```

1 function s=pcmDecode_13(encode, max)
2
3     encode=(reshape(encode',8,length(encode)/8))';
4     len=size(encode);
5     a=[0,16,32,64,128,256,512,1024];
6     b=[1 1 2 4 8 16 32 64];
7     c=[0 1.5:15.5];
8     for i=1:len
9         x=encode(i,1);
10        T=bin2dec(num2str(encode(i,(2:4))))+1;
11        Y=bin2dec(num2str(encode(i,(5:8)))));
12        if Y==0
13            s(i)=a(T)/2048;
14        else
15            s(i)=(a(T)+b(T)*c(Y))/2048;
16        end
17
18        if x==0

```

```
19         s(i)=s(i)*(-1);  
20     end  
21 end  
22 s = max*s;  
23 end
```