数字信号处理

专题实践　钢琴音频识别(2)

华东理工大学 信息科学与工程学院 万永菁

实验介绍

1、实验目的

通过钢琴乐音识别技术的研究，掌握满足**复杂工程问题需求的离散时间系统的基本设计方法与分析技术**，了解影响设计目标和技术方案的各种因素，并得出有效结论。

2、实验内容

钢琴乐音识别技术对钢琴乐音信号进行基频估计，然后根据基频大小来区分音高，从而实现对乐曲的识别。

课题针对钢琴音频信号进行乐音识别技术的研究。在对音频信号其进行**分帧、分音程检测**后，对各**音程段信号进行离散傅里叶变换**，识别频谱中所蕴含的**音符信息**，并**与乐谱进行比对，并得出结论**。

请查阅提供的文献资料，并结合自己查阅的资料，完成以下实验内容：

(1) 基础要求（必做）：实现不同组别的钢琴音阶识别；

(2) 进阶要求（选做）：实现钢琴乐曲《小星星》的整曲识别。

(3) 拓展要求（选做）：对钢琴乐曲《小星星》的整曲节奏进行分析评价。

钢琴音频识别—— 实验步骤（回顾）

Step1. 读入音频文件

Step2. 使用帧峰检测法提取音频的包络信号

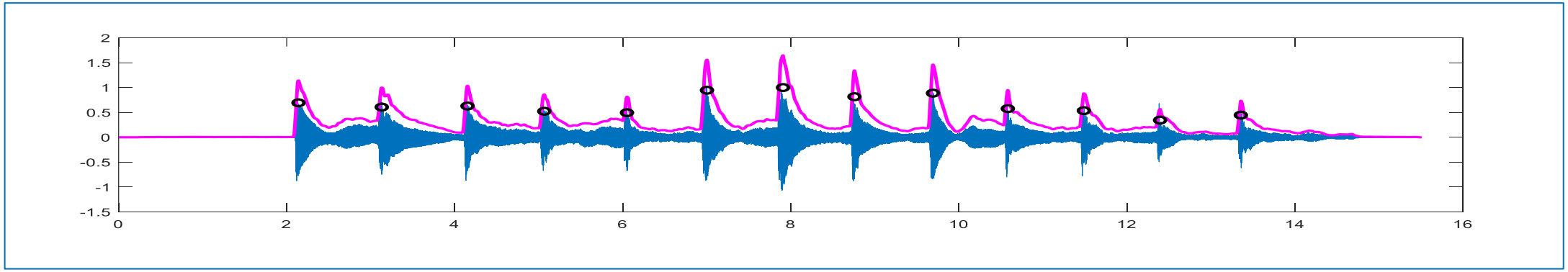
Step3. 尝试使用滤波算法，对包络信号进行滤波

Step4. 根据包络信号寻找音符的起始点

Step4-1. 寻找包络信号的峰值，可以用findpeaks()函数

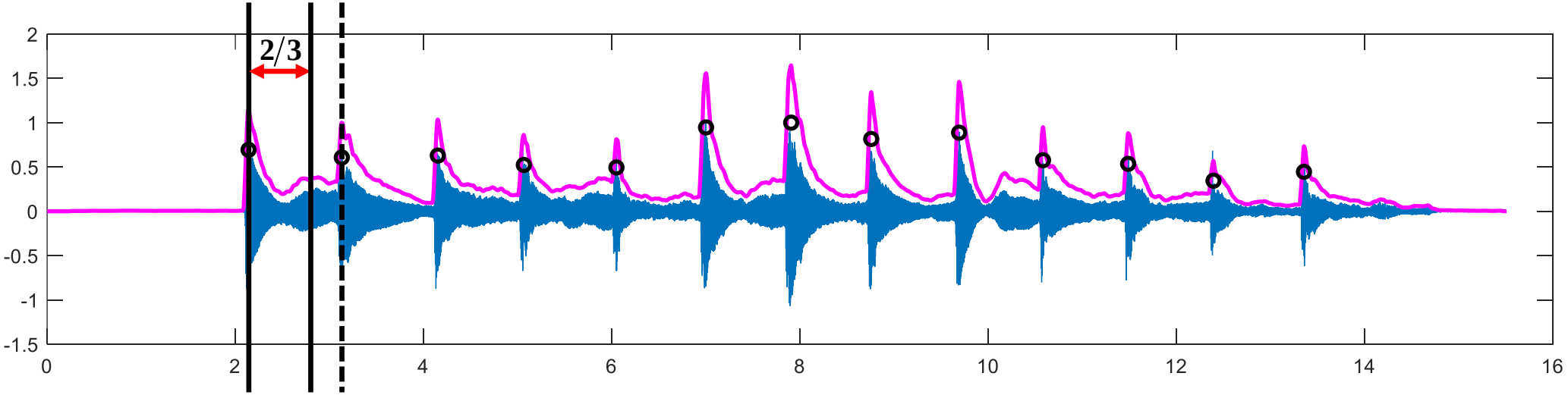
Step4-2. 可以设定一个阈值，去除较小的伪峰点（伪音符起始点）

Step4-3. 去冗余峰值点



Step5. 音程段提取与谱分析

Step5-1. 音程段提取策略



(1) 找到每一段的起始位置和终止位置，从原始音频信号中提取待识别的音频段，放入临时数组；

(2) 音频段信号乘窗函数后（加窗可缓解频谱泄漏），再做FFT；

(3) FFT的长度注意选择好（添零可缓解栅栏效应），幅频值存在数组Amp_MS中。

Amp_MS 的Size
(FFT点数 × 音程段数)

Step5-2. 音程段的谱分析

(1) 使用不同的窗函数，对比和分析频谱泄漏现象，形成结论；

(2) 用不同组别的音频做实验，观察结果，你发现了什么？

A) 对于不同组别的音频，当帧峰检测法的帧长和滤波策略不一样时，音程段切分的准确性不一样，**请选择策略并调试参数**，对所有音阶信号都能实现准确的音程切分；

B) 不同组别的音程段，在识别的时候，得到的谱峰的规律是不一样的，**分析并总结规律，探索识别策略**。

钢琴音频识别——音程提取与谱分析

- 1. 取**实际长度**：取每一音程段**前2/3长度**的点，**实际长度越长，实际能分辨的频谱间隔越小**
- 2. 乘**窗函数**：在实际取到的音程段上乘以相同点数的窗函数，以缓解**频谱泄漏**现象
- 3. 做**FFT谱分析**：按照观察的谱间隔要求**选取FFT点数**，一般会添零，以缓解**栅栏效应**

```
Amp_MS = [];
F0 = 0.1; % 假设观察的谱间隔需求定为0.1Hz（也可以换）
N = Fs/F0; %N为FFT点数，441000点

for i=1:length(Loc3)
    从music中取出音程段信号存入X_temp暂存变量中
    X_temp乘窗函数，注意矩阵的size;
    对X_temp做N点FFT，结果存入Amp_temp中
    Amp_MS = [ Amp_MS  abs(Amp_temp) ];
end
```

利用Loc3数组和Frame_Num

Size为(N，音程段数)

钢琴音频识别——音程提取与谱分析

```
Amp_MS = [];  F0 = 0.1;  %观察的谱间隔暂定为0.1Hz
N = Fs/F0;   %FFT点数

% 1. 取音程段
for i=1:length(Loc3)
    if i<length(Loc3)
        %若不是最后一段则取当前段与后一段的前2/3的信号
        X_temp = music( Loc3(i)*Frame_Num: floor(Loc3(i)+(Loc3(i+1)-Loc3(i))*2/3)*Frame_Num );
    else
        %若是最后一段，则按照前一段的长度取（因为最后一段后面没有下一段的标记了）
        X_temp = music( Loc3(i)*Frame_Num: floor(Loc3(i)+(Loc3(i)-Loc3(i-1))*2/3)*Frame_Num );
    end
    X_temp = X_temp.*hanning(length(X_temp)); % 2. 乘以窗函数（暂定汉宁窗）
    Amp_temp = fft(X_temp,N); % 3. 做FFT（N比X_temp的长度要长，相当于添零了）
    Amp_MS = [ Amp_MS  abs(Amp_temp) ];  %保存本段音程的谱分析结果
end
```

音程端点位置之前已经记录在Loc3数组中

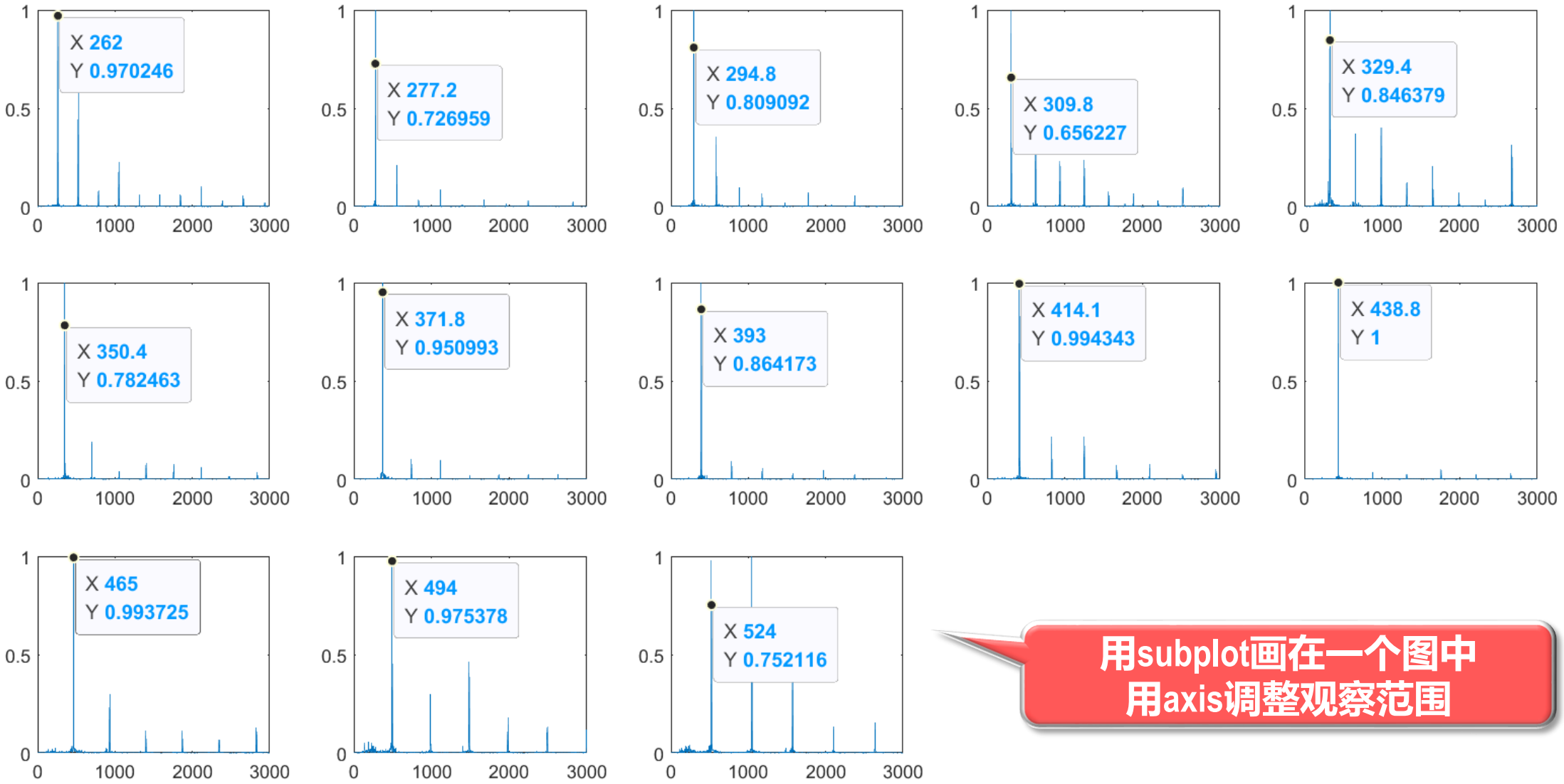
Step5-3. 音频信号谱图的可视化

（1）逐个音程画图

```
figure(4);
k=0:N-1;
for i=1:3
    for j=1:5
      if( (i-1)*5+j <= length(Loc3) ) %小于现有音程的段数
        subplot(3,5,(i-1)*5+j);
        plot(k*Fs/N, Amp_MS(:,(i-1)*5+j)/max(Amp_MS(:,(i-1)*5+j)) );
        axis([0 2000 0 1]);
      end
    end
end
```

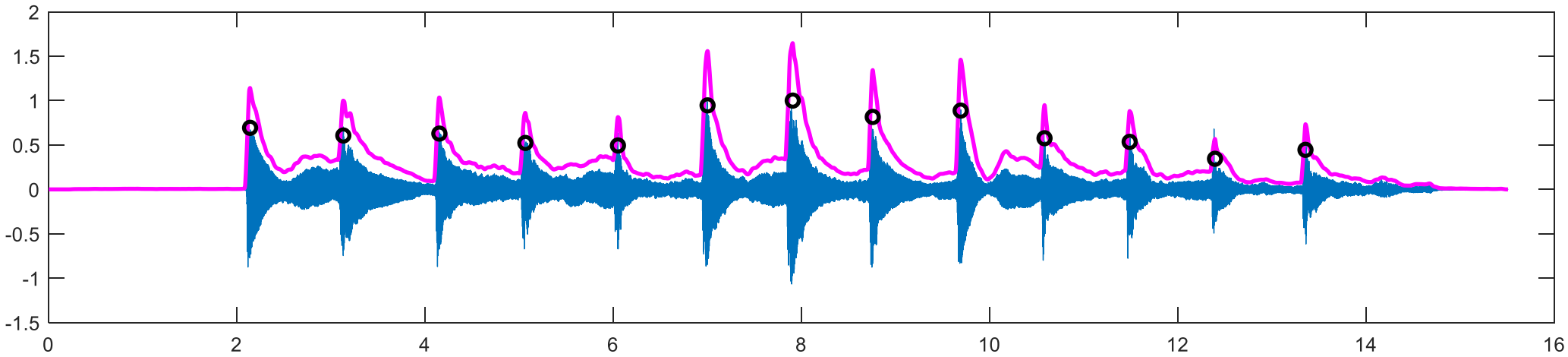
钢琴音频识别——音程提取与谱分析

M1_i1.wav的13个音程的幅频响应



用subplot画在一个图中
用axis调整观察范围

(2) 画整个音频的短时傅里叶谱图



① 对原始音频信号进行分帧，对每一帧信号进行傅里叶变换；

② 得到一个矩阵frame_A，其中记录所有帧的幅频值；

frame_A 的 size
（每帧fft点数×帧数）

③ 用imagesc函数，画frame_A矩阵，即画出短时傅里叶谱图。

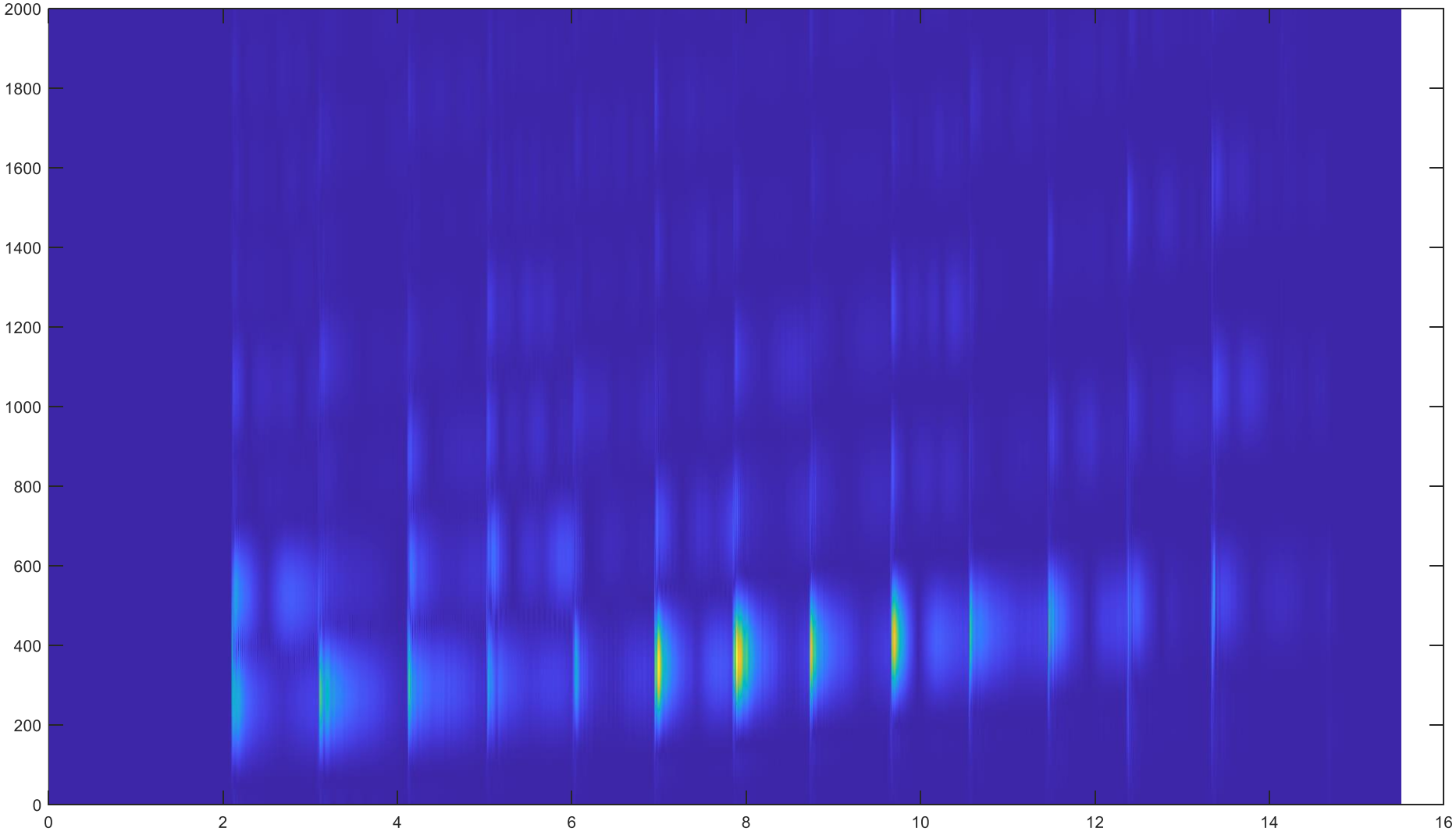
➢ 画短时傅里叶谱图

```
%% 画谱图信息（按照固定帧画）
test_Num = 400;  %画谱图的临时帧长（非前面计算包络的帧长Frame_Num） F0=1; %观察谱间隔暂定1 Hz
frame_N = floor(length(music)/test_Num); %帧数
frame_t = [0:frame_N-1]*test_Num/Fs; %谱图横坐标单位调为时间(s)
fft_N = Fs/F0; %每一帧FFT的点数，暂设为44100点
frame_f = [0:floor((fft_N-1))]*F0; %谱图纵坐标单位为Hz
frame_A = zeros(fft_N,frame_N); % frame_A记录某一帧的某一频率处的值，后面用颜色显示
for i=1:frame_N-1
    %每一帧400点，均乘400点哈明窗后，做44100点FFT
    frame_A(:,i)= abs( fft( music( (i-1)*test_Num+1 : i*test_Num ).*hamming(test_Num), fft_N) );
end
figure(5); subplot(211);
imagesc(frame_t,frame_f,frame_A);
axis xy; % axis xy ： 笛卡尔轴模式，原点在左下角，y轴是竖直的，由底至顶标数，x轴是水平的，从左往右标数
axis([0 M_T 0 2000]);  % M_T是音乐总的时间，M_T = ceil(length(音频点数)/44100);
```

钢琴音频识别——音程提取与谱分析

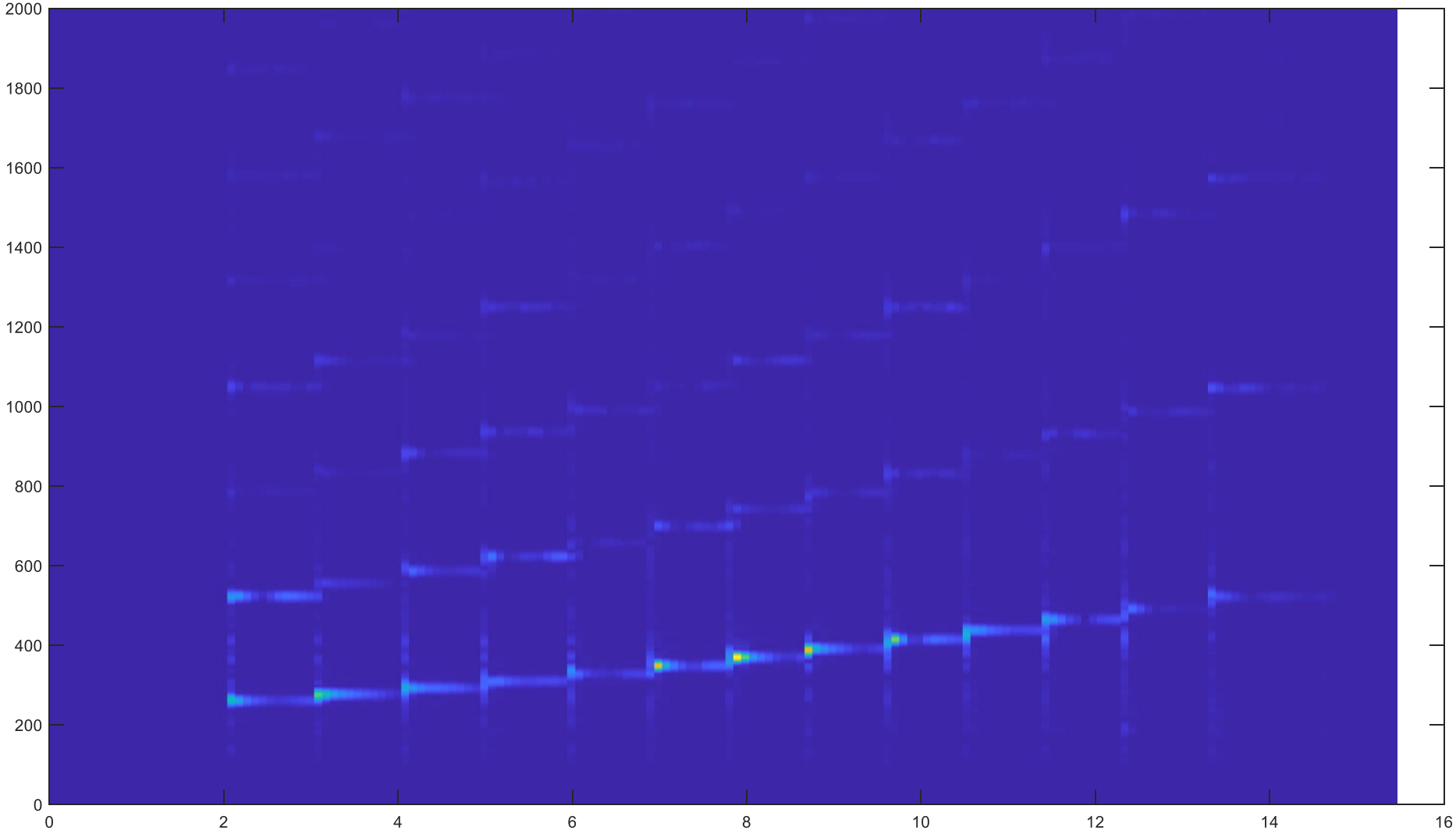
华東理工大學

- **画短时傅里叶谱图：** test_Num = 400;



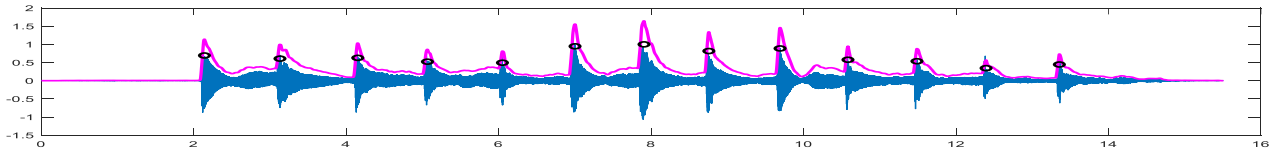
钢琴音频识别——音程提取与谱分析

➢ **画短时傅里叶谱图：** test_Num = 4000;



钢琴音频识别——音程提取与谱分析

➢ 不同帧长下的谱泄漏现象对比



```
figure(6);
starti=Loc3(1)*Frame_Num;
test_N = 400;
i=starti:starti+test_N;
test_music=music(i);
subplot(331)
plot(test_music);title('实际长度： 400点');
test_music_FFT=fft(test_music,44100);
subplot(334)
stem(abs(test_music_FFT)/max(abs(test_music_FFT)));
title('矩形窗谱图， 44100点FFT');axis([0 1000 0 1]);
test_music_FFT=…
    fft(test_music.*hamming(length(test_music)),44100);
subplot(337)
stem(abs(test_music_FFT)/max(abs(test_music_FFT)));
title('哈明窗谱图， 44100点FFT');axis([0 1000 0 1]);
```

实际取400点，
矩形窗，做44100点FFT
哈明窗，做44100点FFT

```
test_N = 4000;
i=starti:starti+test_N;
test_music=music(i);
subplot(332)
plot(test_music);title('实际长度： 4000点');
test_music_FFT=fft(test_music,44100);
subplot(335)
stem(abs(test_music_FFT)/max(abs(test_music_FFT)));
title('矩形窗谱图， 44100点FFT');axis([0 1000 0 1]);
test_music_FFT=…
    fft(test_music.*hamming(length(test_music)),44100);
subplot(338)
stem(abs(test_music_FFT)/max(abs(test_music_FFT)));
title('哈明窗谱图， 44100点FFT');axis([0 1000 0 1]);
```

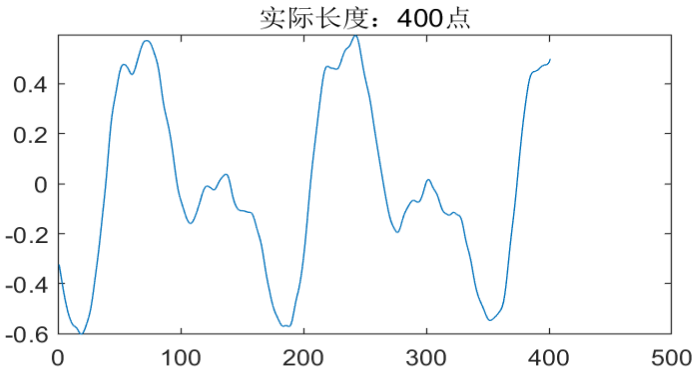
实际取4000点，
矩形窗，做44100点FFT
哈明窗，做44100点FFT

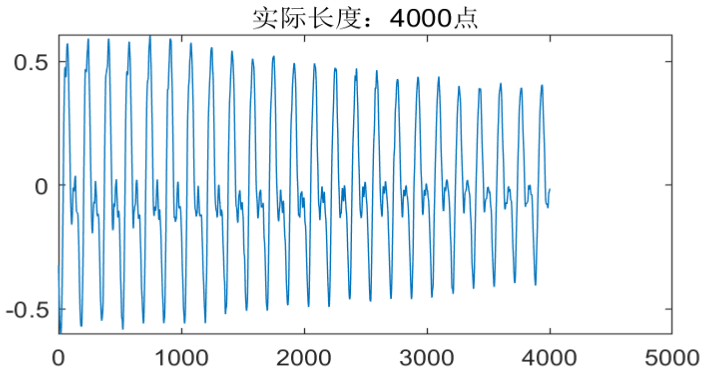
不同帧长下的谱泄漏现象对比

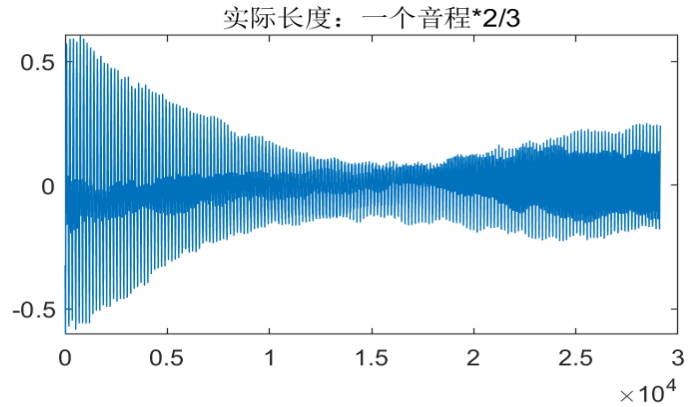
```
test_N = round((Loc3(2)*Frame_Num-Loc3(1)*Frame_Num+1)*2/3);
i=starti:starti+test_N;
test_music=music(i);
subplot(333)
plot(test_music);title('实际长度：一个音程*2/3');
test_music_FFT=fft(test_music,44100);
subplot(336)
stem(abs(test_music_FFT)/max(abs(test_music_FFT)));
title('矩形窗谱图，44100点FFT');axis([0 1000 0 1]);
test_music_FFT=fft(test_music.*hamming(length(test_music)),44100);
subplot(339)
stem(abs(test_music_FFT)/max(abs(test_music_FFT)));
title('哈明窗谱图，44100点FFT');axis([0 1000 0 1]);
```

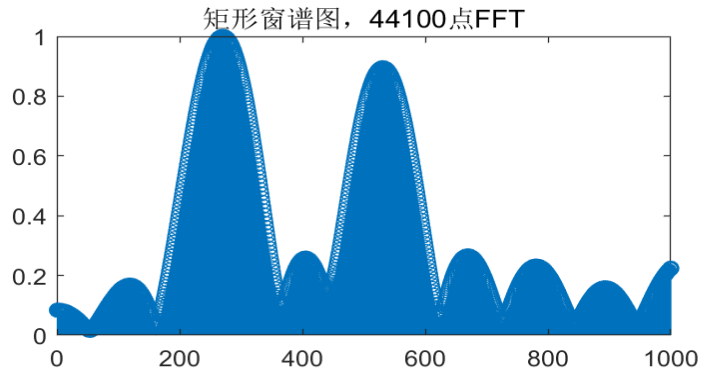
实际取一整个音程的前2/3，≈29314点
矩形窗，做44100点FFT
哈明窗，做44100点FFT

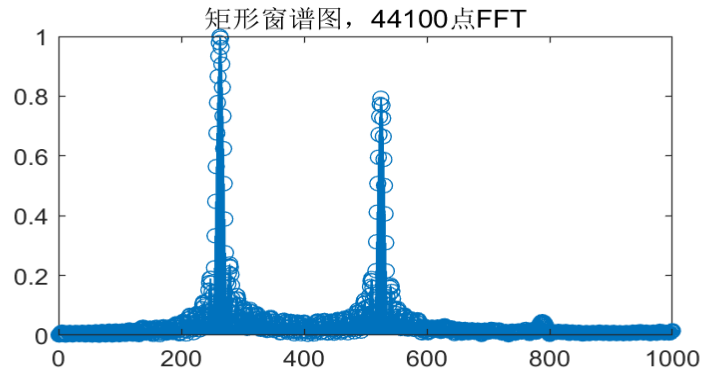
钢琴音频识别——音程提取与谱分析

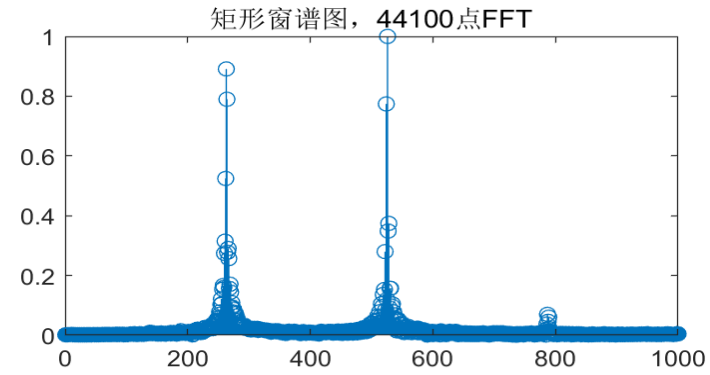


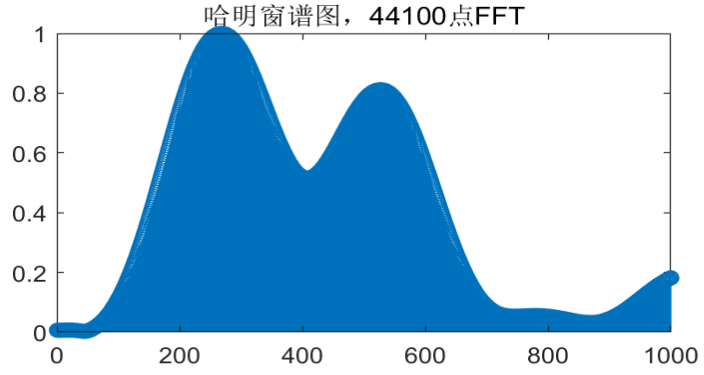


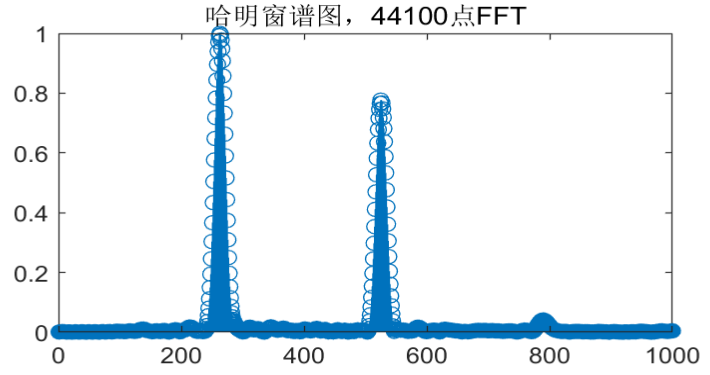


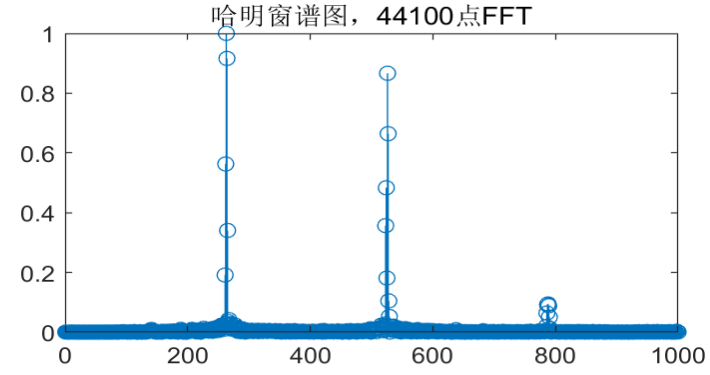












钢琴音频识别——音程提取与谱分析

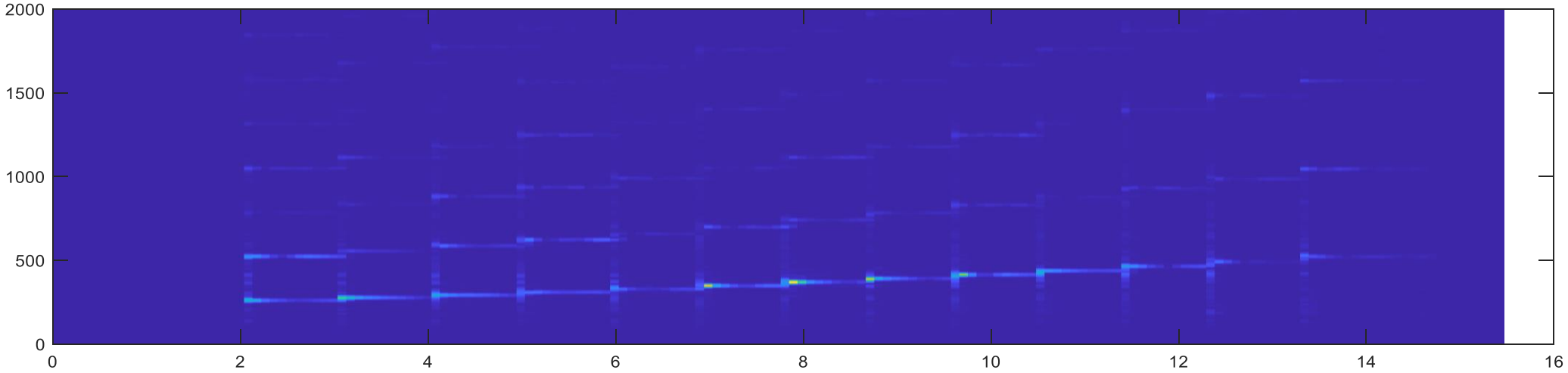
华東理工大學

➢ 画短时傅里叶谱图（按照音程画）

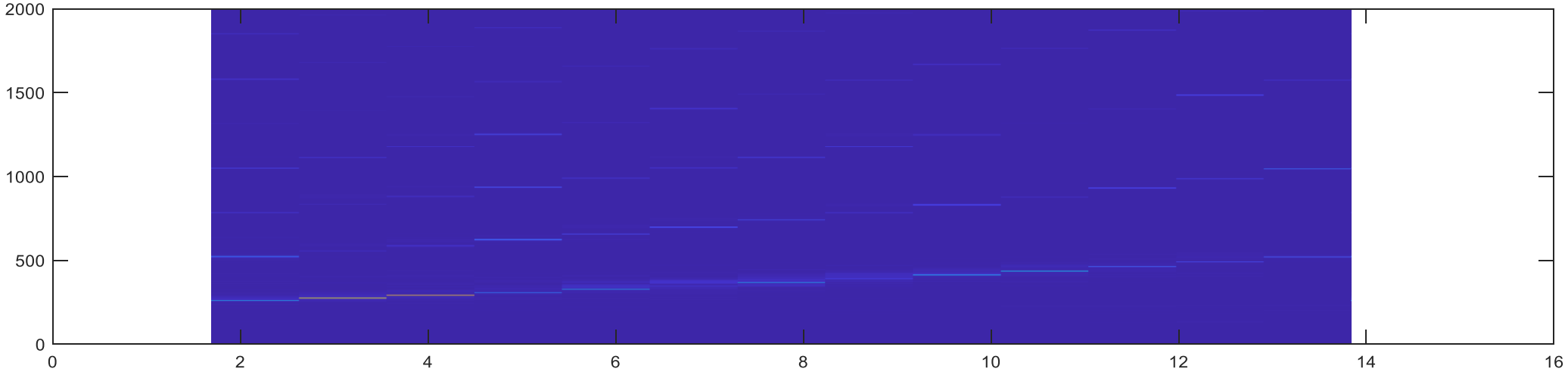
```
frame_t_ms = [Loc3*Frame_Num/Fs]; %谱图横坐标
frame_N = length(Loc3);
fft_N = Fs/F0;
frame_f_ms = [0:floor((fft_N-1))]*F0;
frame_A_ms = zeros(fft_N,frame_N);
for i=2:frame_N
    test_ms = music( Loc3(i-1)*Frame_Num+1 : Loc3(i)*Frame_Num );
    frame_A_ms(:,i-1)= abs( fft( test_ms.*hamming(length(test_ms)), fft_N) );
end
%最后一个音程的处理
test_ms = music( Loc3(i)*Frame_Num+1 : Loc3(i)*Frame_Num+length(test_ms) );
frame_A_ms(:,i)= abs( fft( test_ms.*hamming(length(test_ms)), fft_N) );

figure(5);subplot(212)
imagesc(frame_t_ms,frame_f_ms,frame_A_ms);axis xy; axis([0 M_T 0 2000]);
```

钢琴音频识别——音程提取与谱分析

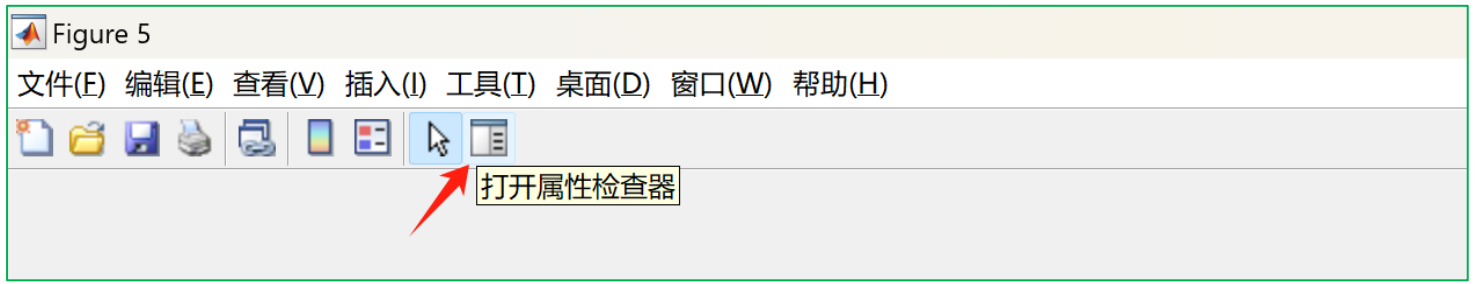
- 画短时傅里叶谱图： test_Num = 4000;

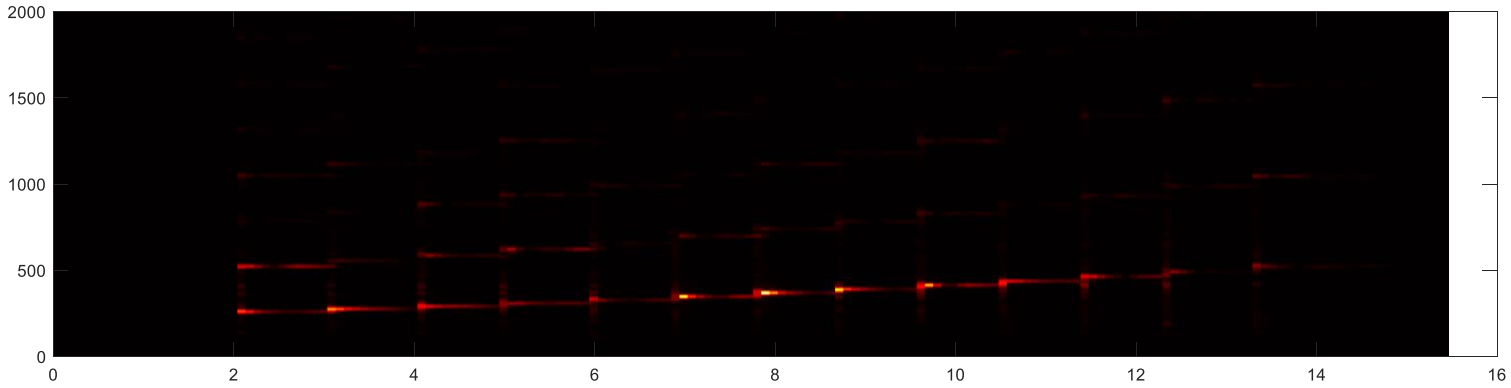


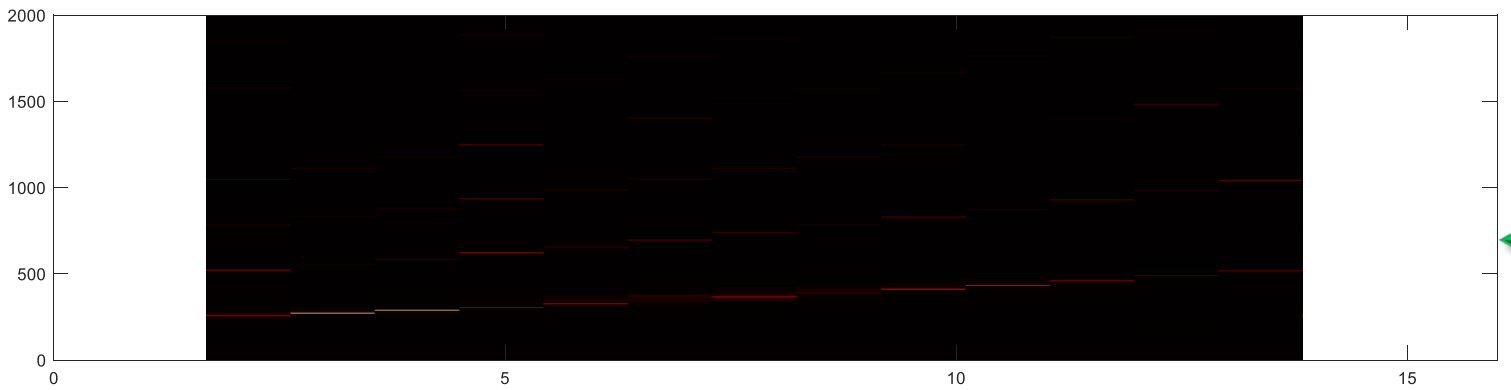
- 画短时傅里叶谱图（按照音程画）

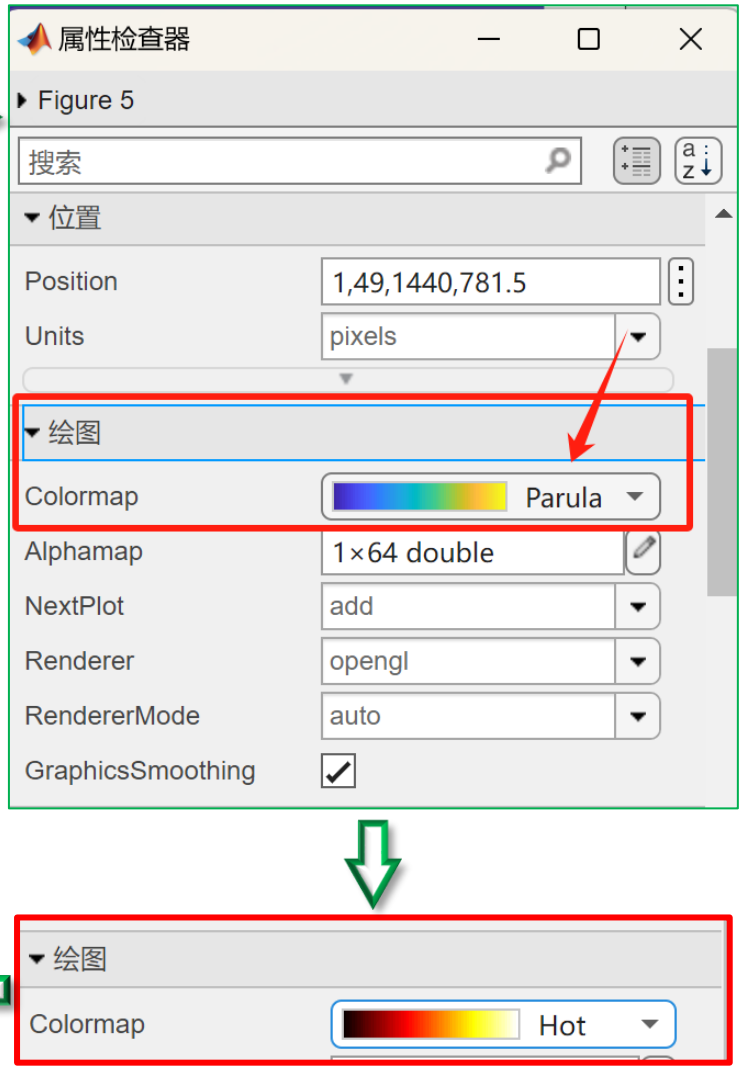


钢琴音频识别——音程提取与谱分析







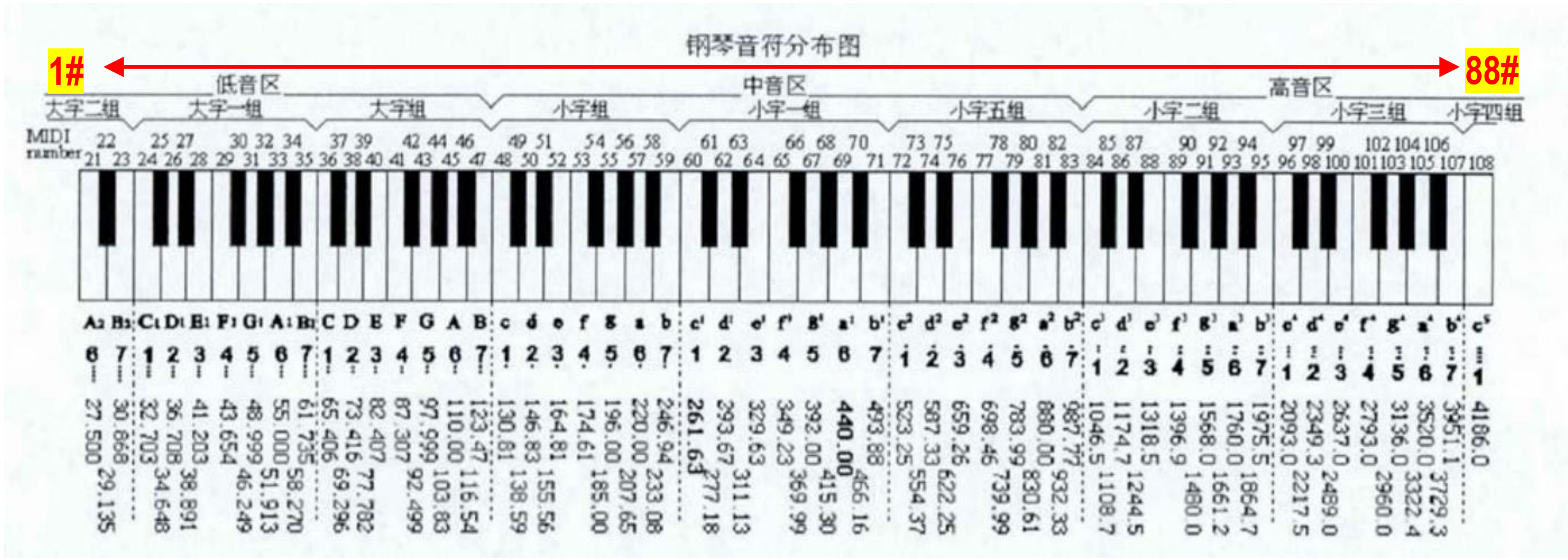


Stpe6. 钢琴音频整曲识别

用xlsread函数读入，存入Piano_F变量中

(1) 读入钢琴按键与频率的对应表

Piano_Key_F.xlsx



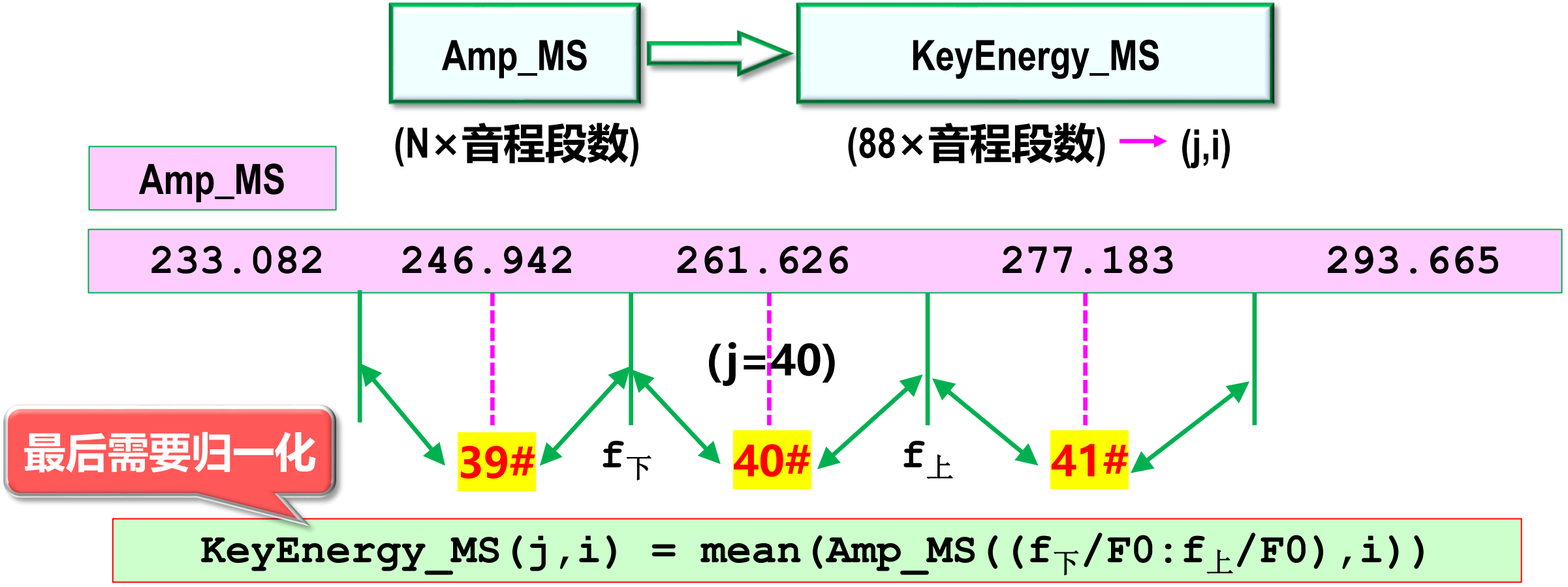
其中 k' 表示为对应音符的 MIDI 号。$f_{k'}^{m}$ 指 MIDI 号为 k' 的第 m 个谐波分量。这里我们认为每个音符主要由前 10 个谐波分量构成，当 $m>10$ 时，取其谐波能量为 0。各个音符的基频之间存在着如下关系：

$$
\begin{aligned}
&2f_{k'}^{0}/f_{k'+12}^{0}=1.0000, \quad 3f_{k'}^{0}/f_{k'+19}^{0}=0.9989\\
&4f_{k'}^{0}/f_{k'+24}^{0}=1.0000, \quad 5f_{k'}^{0}/f_{k'+28}^{0}=1.0079\\
&6f_{k'}^{0}/f_{k'+31}^{0}=0.9989, \quad 7f_{k'}^{0}/f_{k'+34}^{0}=1.0180\\
&8f_{k'}^{0}/f_{k'+36}^{0}=1.0000, \quad 9f_{k'}^{0}/f_{k'+38}^{0}=0.9977\\
&10f_{k'}^{0}/f_{k'+40}^{0}=1.0079
\end{aligned}
\tag{4-11}
$$

从上面可以看出，音符 k' 的二、四、八次谐波分量分别与比它高 12、24、36 个半音的音符基频重合，并且音符 k' 的三、五、六、七次谐波频率值分别与比它高 19、28、31、34 个半音的基频相等。由此可以看出经常是在比基频高 {12,19,24,28,31...}个半音处发生频谱重叠事件。

6. 钢琴音频整曲识别

(2) 将音频幅度矩阵映射到按键矩阵中



Step6. 钢琴音频整曲识别

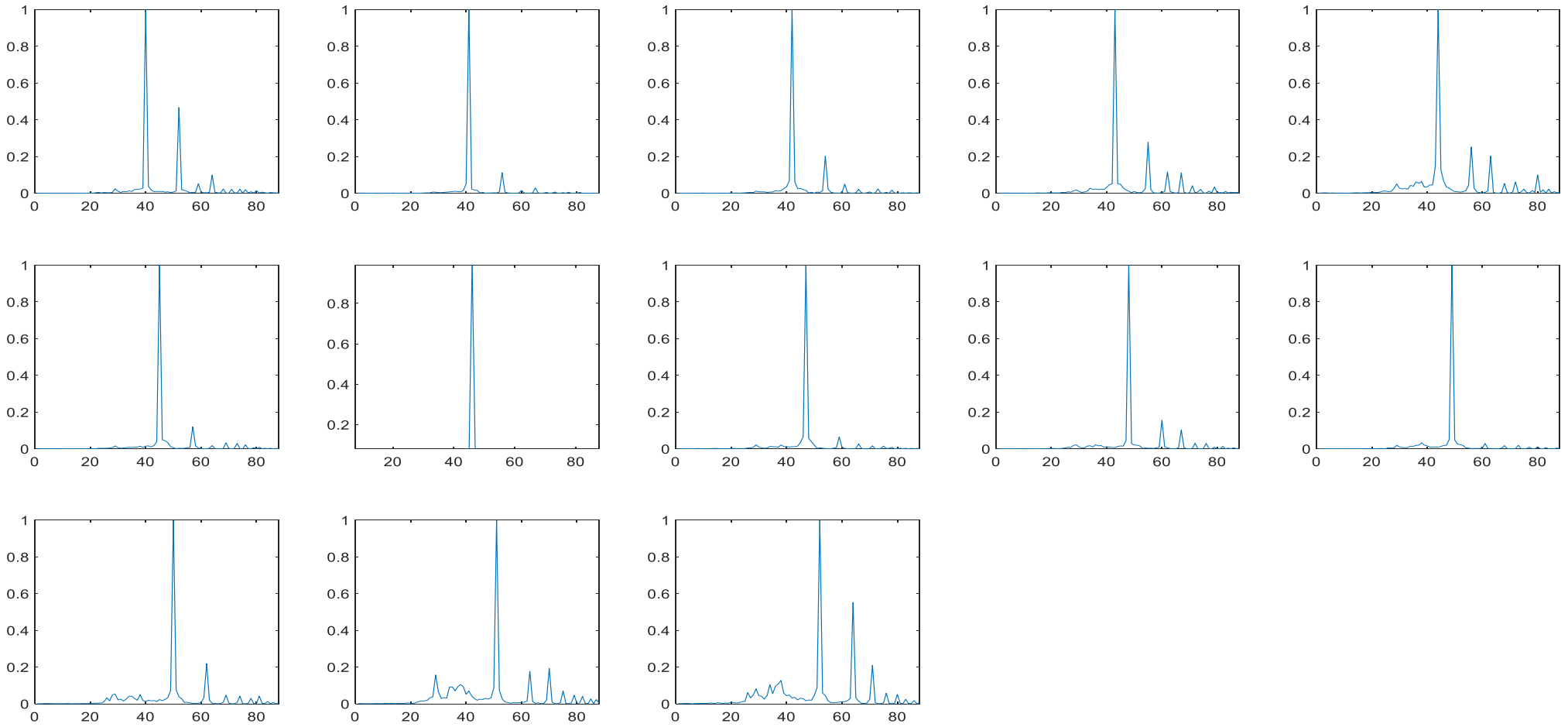
(3) 画出键号对应的图

```
key=1:88;
for i=1:3
  for j=1:5
      if( (i-1)*5+j <= length(Loc3) )   %有13个音程
          subplot(3,5,(i-1)*5+j);
          plot(key,KeyEnergy_MS(:,(i-1)*5+j));
          axis([0 88 0 1]);
      end
  end
end
```

Step6. 钢琴音频整曲识别

(3) 画出键号对应的图

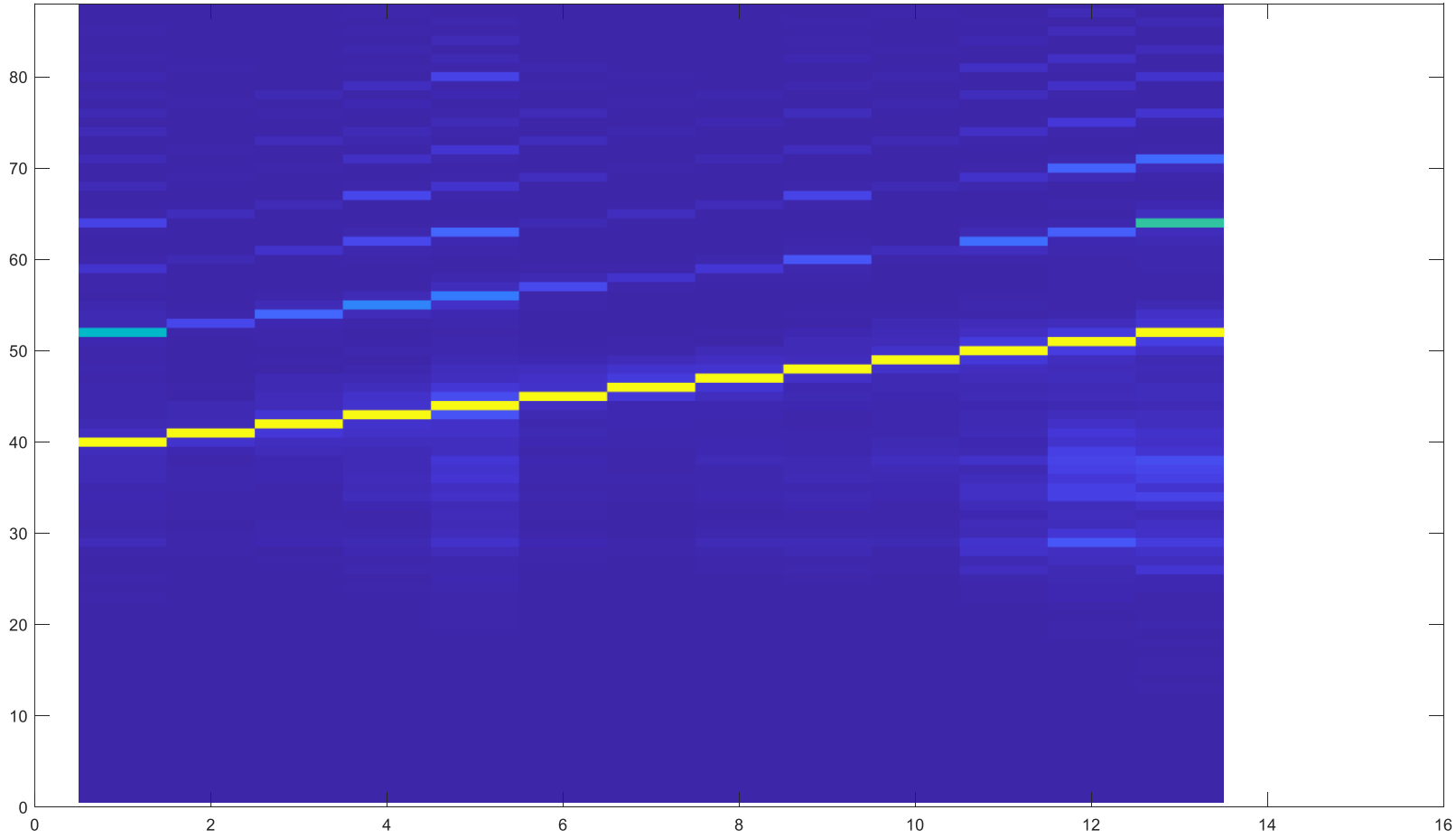
识别前，此图要仔细研究，才好建模



Step6. 钢琴音频整曲识别

(4) 也可以画出键号的瀑布图（类似谱图）

```
imagesc(1:length(Loc3),1:88,KeyEnergy_MS);
axis xy;  axis([0 M_T 0 88]);
```



Step6. 钢琴音频整曲识别

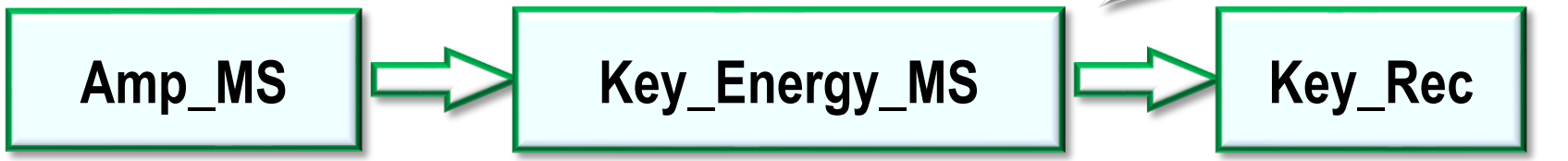
此处也可以用音阶音频来观察大字组和小字组的规律不一样

(5) 识别策略

观察KeyEnergy_MS矩阵，与标准曲谱比较，观察规律并建模。

(6) 显示识别结果

按照识别策略Key_Rec中的值非0即1



(N×音程段数)　(88×音程段数)　(88×音程段数)

Step6. 钢琴音频整曲识别

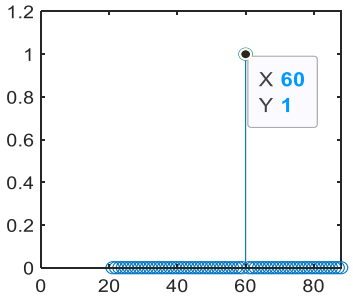
(6) 显示识别结果（显示每个音程的键号）

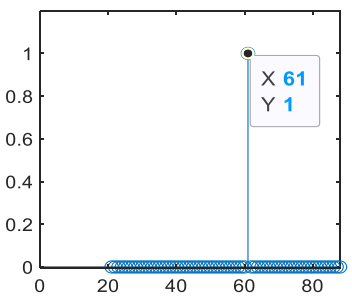
```
key=1:88;
for i=1:3
    for j=1:5
        if( (i-1)*5+j <= length(Loc3) )
            subplot(3,5,(i-1)*5+j);
            stem(key+20,Key_Rec(:,(i-1)*5+j));
            axis([0 88 0 1.2]);
        end
    end
end
```

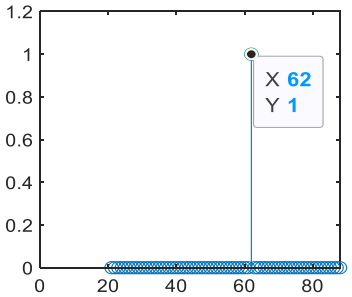
Step6. 钢琴音频整曲识别

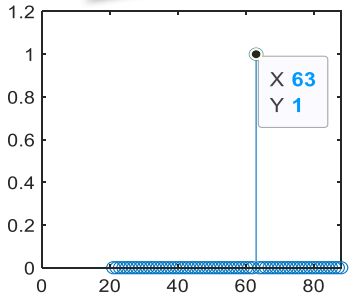
策略不完美的时候，有错误
需要调整参数和识别策略，分析正确率

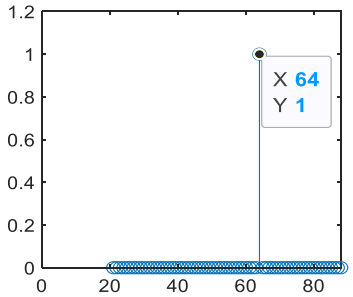
(6) 显示识别结果（显示每个音程的键号）

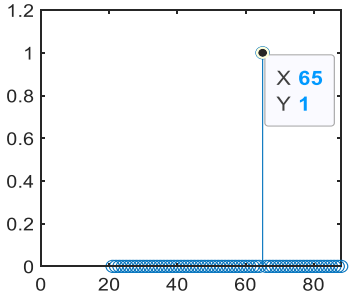


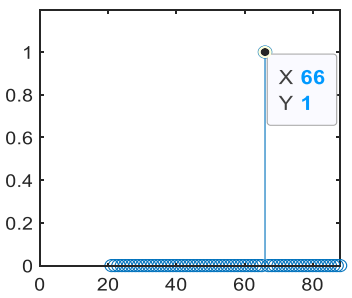


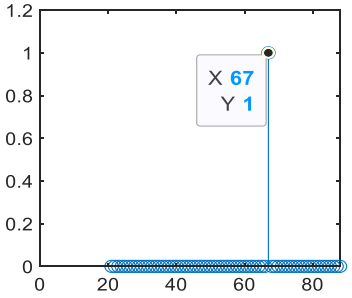


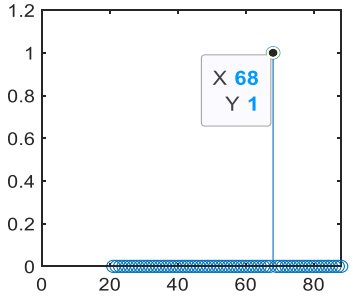


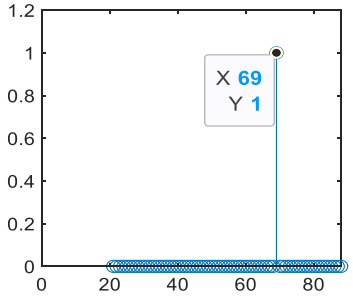


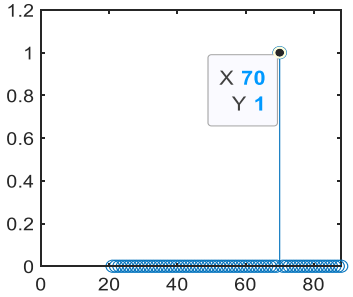


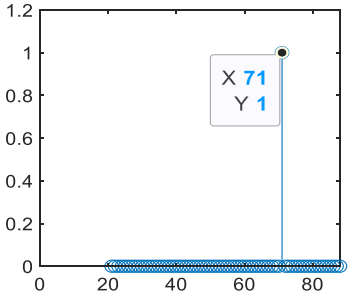


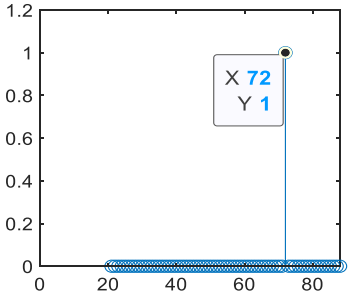








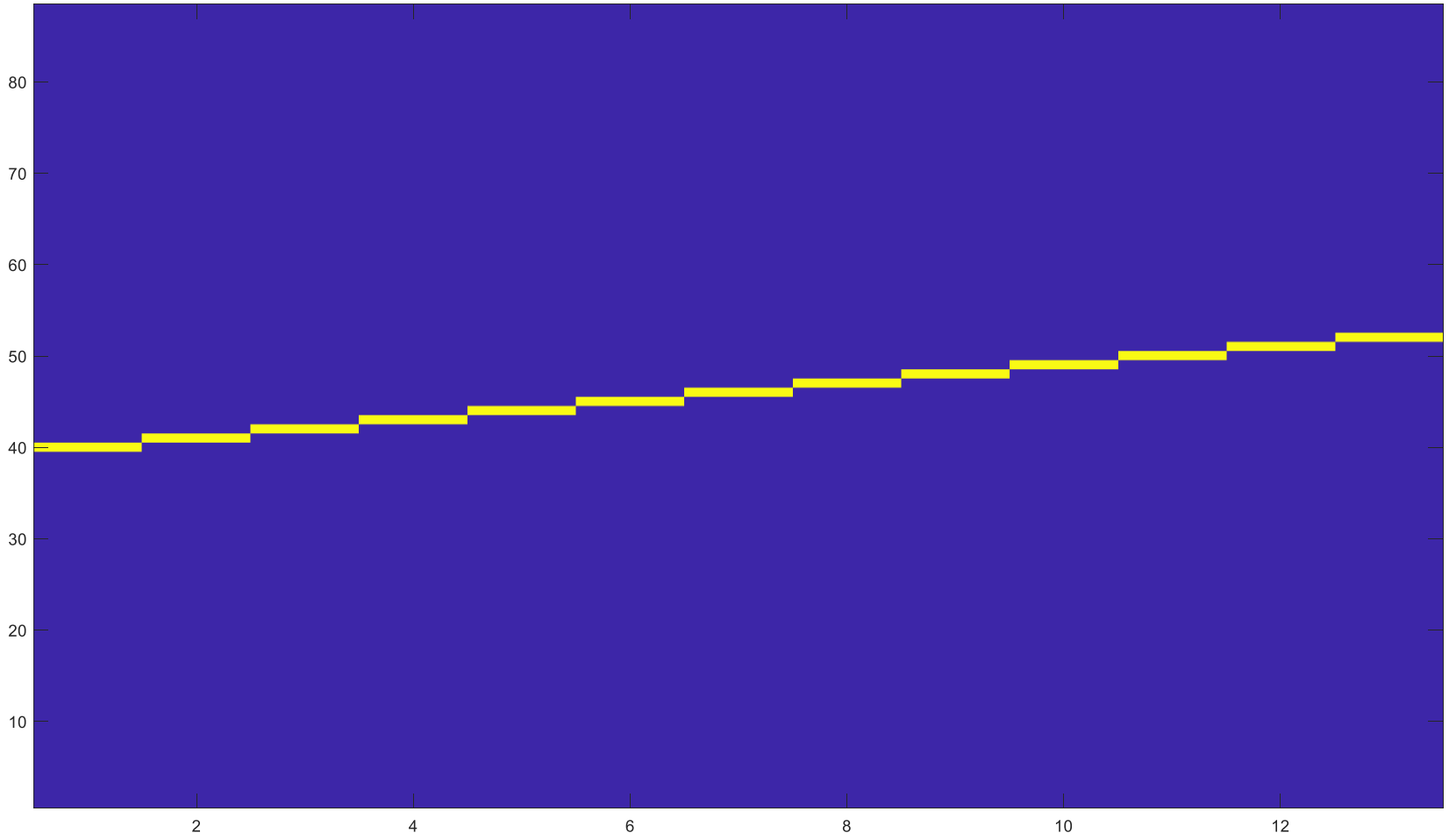




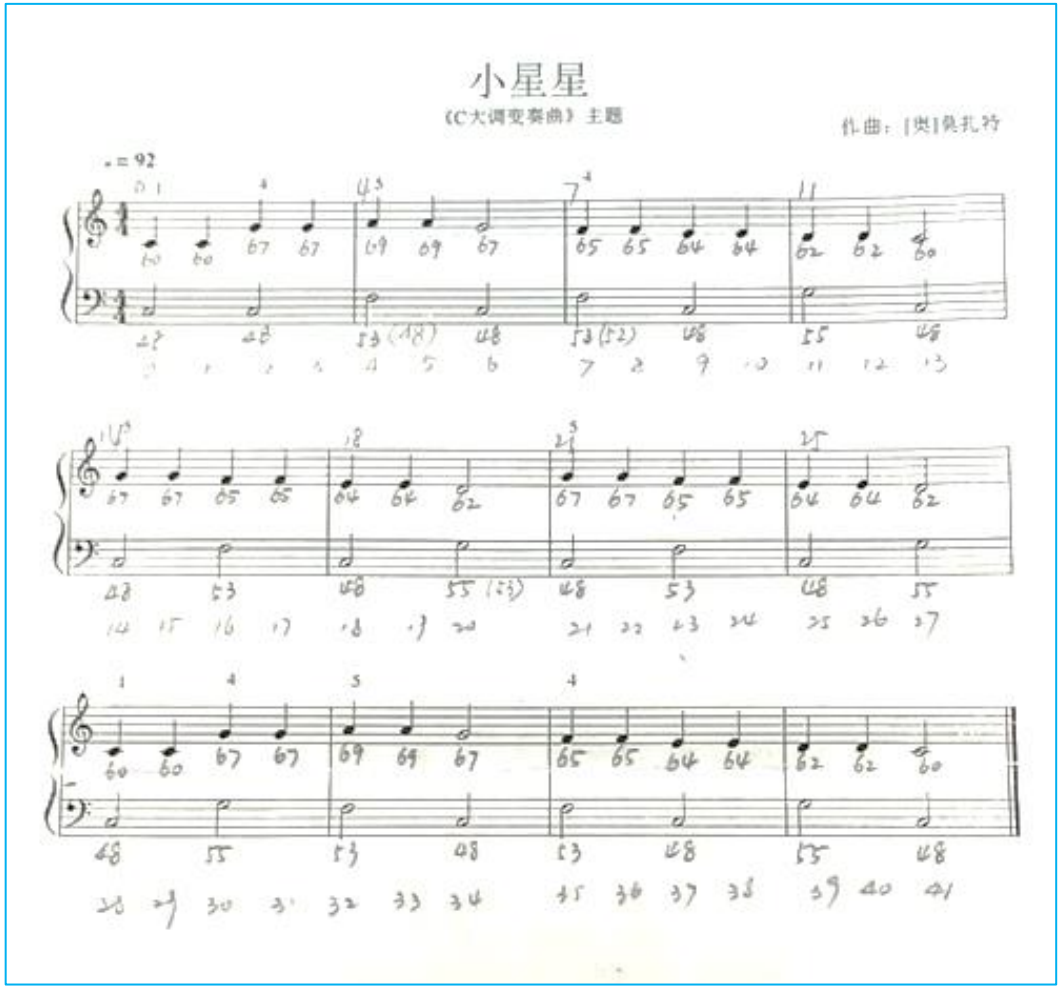
Step6. 钢琴音频整曲识别

(6) 显示识别结果(看瀑布图)

```
imagesc(1:length(Loc3),1:88,Key_Rec);
axis xy;
```



Step7. 钢琴音频节奏评价（用小星星曲谱举例）




```
% ideal_Beat_T为标准节拍
ideal_Beat_T = [1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 1
1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1];

% Real_Beat_T保存实际节拍
Real_Beat_T = zeros(1,length(Loc3)-1);

% Err_Beat_T保存节拍误差值
Err_Beat_T = zeros(1,length(Loc3)-1);
```

Step7. 钢琴音频节奏评价（用小星星曲谱举例）

假设♩=69，即1分钟(60s)是69拍 ⇨ $\text{Real\_Beat\_T} = \text{音程段时间} \times \frac{69}{60}$

Real_Beat_T的物理含义：

(1) 当Real_Beat_T<1，说明节奏快了，时长其实不到1拍

(2) 当Real_Beat_T>1，说明节奏慢了，时长其实超过1拍

```
subplot(211);
plot(1:length(Loc3)-1,Real_Beat_T,'bo');hold on;
plot(1:length(Loc3)-1,ideal_Beat_T,'r*');grid on;
axis([1 length(Loc3)-1 0 2.2])
subplot(212);
plot(1:length(Loc3)-1,Err_Beat_T,'bx-');grid on;
axis([1 length(Loc3)-1 -0.5 0.5])
```