

计算机语音技术附加实验

计算机 1505 班 王天舒 15281132

2018.05

一、实验目的

通过 Matlab 编程：

- 掌握动态规划 DP 算法
- 加深对语音基频参数特性的理解

二、实验内容

- 基于自相关系数 AC 的局部极大值构成基音周期候选点集，通过预设基频范围进行初步筛选；
- 采用动态规划 DP 算法，通过代价函数 $CostFunction()$ - 涉及目标、转移代价，利用帧同步搜索代价最小路径，检测出基频
- 结合孤立音节录音'tang1'的基频检测曲线，介绍代码及参数调试过程，对检测结果进行分析
- 自行录制一段连续语音发音，对检测结果进行分析，针对观察到的问题优化算法

三、实验原理

浊音信号的周期称为基音周期，它是声带振动频率的倒数。由于基音周期只具有准周期性，所以只能采用短时平均方法估计该周期，这个过程被称为基音检测（pitchdetection）。基音的提取和估计是语音信号处理中一个十分重要的问题，尤其是对汉语更是如此；因为汉

语是一种有调语言，基音的变化模式称为声调，它携带着非常重要的具有辨意作用的信息，有区别意义的功能。

本次实验主要先通过自相关系数的局部最大值构成基音周期候选点集，再通过动态规划算法求出基频，最后通过中值平滑进行得到基音频率。

1. 自相关系数

对于离散的语音信号 $x(n)$ ，它的自相关函数定义为：

$$\begin{aligned} R_n(k) &= \sum_{m=n-N+1}^{n-k} x(m)w(n-m)x(m+k)w(n-m-k) \\ &= [x(n)x(n-k)] * h_k(n) \end{aligned}$$

如果信号 $x(n)$ 具有周期性，那么它的自相关函数也具有周期性，而且周期与信号 $x(n)$ 的周期性相同，故自相关函数提供了一种获取周期信号周期的方法。在周期信号周期的整数倍上，它的自相关函数可以达到最大值，因此可以不考虑起始时间，而从自相关函数的第一个局部极大值的位置估计出信号的基音周期，这使自相关函数成为信号基音周期估计的一种工具。

但是，在计算短时自相关时，窗选语音段为有限长度 N ，而求和上限为 $N-1-k$ ，因此当 k 增加时可用于计算的数据就越来越少了，从而导致 k 增加时自相关函数的幅度减小。为了解决这个问题，提出了语音修正的短时自相关。修正的短时自相关函数，其定义如下：

$$\begin{aligned} \hat{R}_n(k) &= \sum_{m=-\infty}^{\infty} x(m)w_1(n-m)x(m+k)w_2(n-m-k) \\ &= \sum_{m=-\infty}^{\infty} x(n+m)w'_1(m)x(n+m+k)w'_2(m+k) \\ &= \sum_{m=0}^{N-1} x(n+m)x(n+m+k) \quad (0 \leq k \leq \bar{K}) \end{aligned}$$

注意窗口长度 N 需大于两倍基音周期。

2. 最优路径搜索（动态规划）

在通过短时自相关函数获得每帧的 pitch 候选值后，设置代价函数。可以通过选择某候选的自身代价及路径累计代价，找到代价函数最小化的一条路径（每帧只保留 1 个候选值）从而得到整段语音的基频曲线（pitch contour）。

- 代价函数 CostFunction()

转移代价： $CostT(n, i, j) = Diff(F0(Can_n(i)), F0(Can_{n+1}(j)))$

目标代价： $CostG(n, i) = Dist(ac(Can_n(i)))$

累计代价： $CostF(n+1, j) = \min_i(CostF(n, i) + CostT(n, i, j)) + CostG(n+1, j)$

$Can_n(i)$ 表示第n帧的第i个候选

$Diff()$ 表示两个变量间的差异， $F0()$ 表示基频

$Dist()$ 表示距离测度， $ac()$ 表示自相关函数

- 动态搜索: $F0[1], F0[2], \dots, F0[N]$

- Step 1 初始化第 1 帧的代价函数

$$CostF(1, i) = Dist(ac(Can_1(i))) \quad (i = 1, 2, \dots, L)$$

- Step 2 计算每帧的代价函数

$$CostF(n+1, j) = \min_i(CostF(n, i) + CostT(n, i, j)) + CostG(n+1, j)$$

$$Path(n+1, j) = \arg \min_i(CostF(n, i) + CostT(n, i, j)) \quad (i, j = 1, 2, \dots, L)$$

- Step 3 确定最优路径

$$\hat{I}_N = \arg \min_j(CostF(N, j))$$

- Step 4 路径回溯

$$\hat{I}_n = Path(n+1, \hat{I}_{n+1})$$

$$F0[n] = F0(Can_n(\hat{I}_n))$$

四、实验步骤及程序

0. 语音读入

```
clc;clear all;close all;
warning('off')
%[x,fs,nbits]=wavread('tang1.wav');           %打开录好的语音信号文件。
global fs;
[x,fs] = audioread('tang1.wav');           %打开录好的语音信号文件。
%info = audioinfo('tang1.wav');
%nbits = info.BitsPerSample;

x = x / max(abs(x));                       %幅度归一化到[-1,1]1. 基于自相
关系数 AC 的局部极大值构成基音周期候选点集，通过预设基频范围进行初步筛选；

%参数设置
kl = round(1 / 500 * fs);                 %500Hz
kr = round(1 / 80 * fs);                 %80Hz
N = 3 * kr;                               %帧长

inc = round(fs / 100);                   %帧移步长 10ms
```

其中 k_l 、 k_r 分别对基频分析的上下限频率（500Hz 和 80Hz），窗口长度等于这个下限频率的周期的 3 倍，步长 Time step 设置为 10 毫秒。

候选点集合，其中用到了 matlab 自相关函数 autocorr 和查找峰值函数 findpeaks 函数，需要注意的是，voicebox 自带一个 findpeaks 函数，故需要将 voicebox 的 path 调到最低。

```
A = enframe(x, N, inc);
R(n, :) = autocorr(A(n, :), N - 1);

[acCan,Can] = findpeaks(R(n, kl:kr), 'MinPeakHeight', R(n, 1) * 0.25,
'MinPeakProminence', 0.1);
```

2. 采用动态规划 DP 算法，通过代价函数 CostFunction() - 涉及目标、转移代价，利用帧同步搜索代价最小路径，检测出基频。

```
A = enframe(x, N, inc);
R = zeros(size(A));
l = zeros(1, size(R, 1));  l : 最优ac位置序列
f = zeros(1, size(R, 1));  f : 最优基频序列
Can = zeros(1, 10);
```

```

acCan = zeros(1, 10);
CostF = zeros(1, 10);
it = 0;
for n = 1:size(R, 1)
    R(n, :) = autocorr(A(n, :), N - 1);
    Can_ = Can;
    CostF_ = CostF;
    [acCan, Can] = findpeaks(R(n, kl:kr), 'MinPeakHeight', R(n, 1) *
0.25, 'MinPeakProminence', 0.1);
    Can = Can + kl - 1;
    sz = size(Can, 2);
    if sz ~= 0
        it = it + 1;
        if it == 1
            CostF = dist(acCan);
        else
            CostF = zeros(1, sz);
            Path = zeros(1, sz);
            CostT = diff(Can, Can_);
            for j = 1:sz
                [CostF(j), Path(j)] = min(CostF_ + CostT(j, :));
                CostF = CostF + dist(acCan);
            end
        end
        [~, l(n)] = min(CostF);
        ff = f0(Can);
        plot(n, ff, '.');
        hold on;
        f(n) = ff(l(n));
    else
        it = 0;
    end
end
end

```

得到基频向量 f，其中 f0、dist、diff 函数具体实现见附加。

3. 结合孤立音节录音 'tang1' 的基频检测曲线，介绍代码及参数调试过程，对检测结果进行分析。

除上述主要代码，还有绘图部分。

```

%语音波形图
subplot(3, 1, 1);
plot(x);
axis([1 length(x) -1 1])%限制x轴与y轴的范围。
xlabel('帧数');

```

```

ylabel('Speech');
legend('FrameLen = 552');

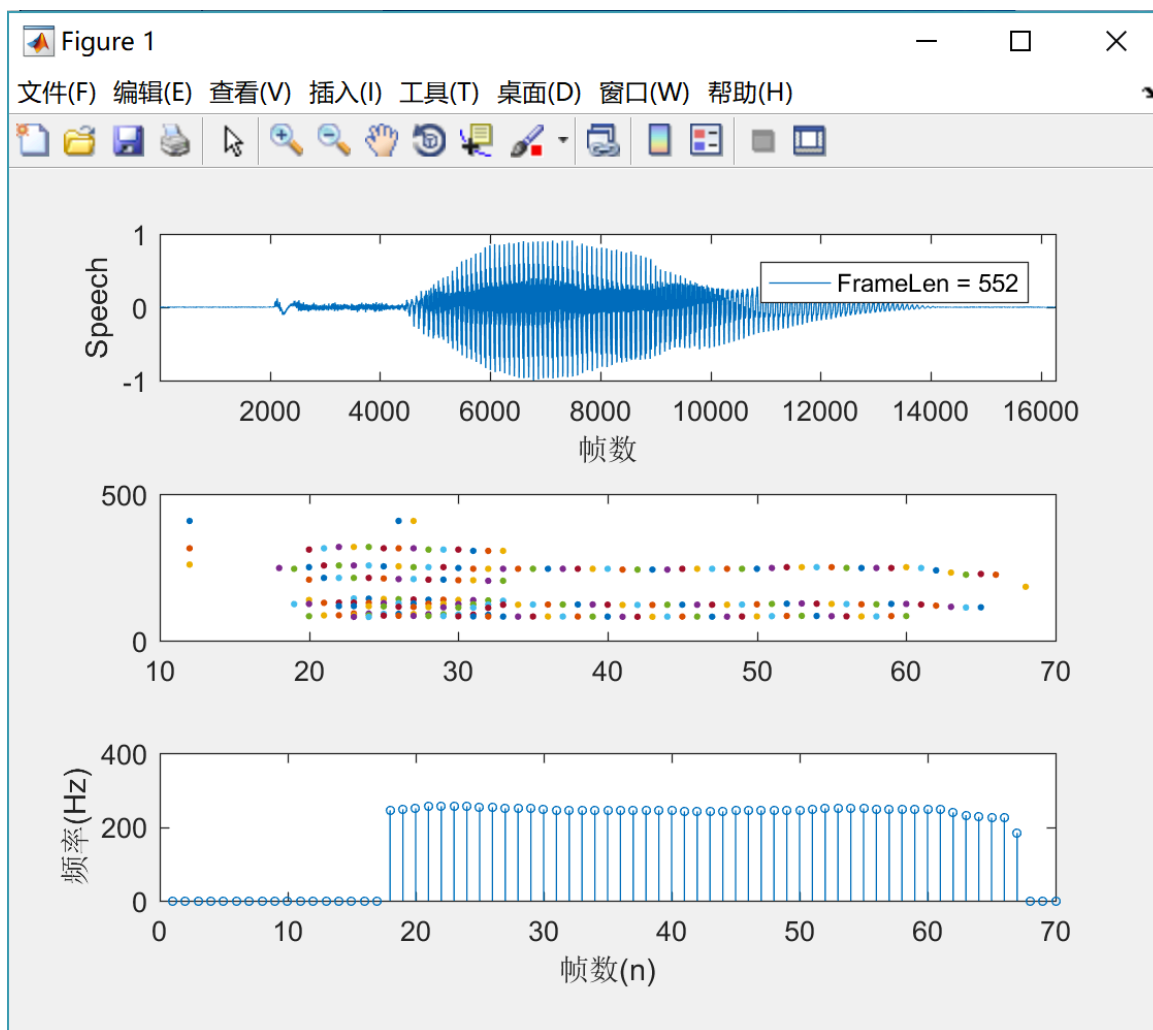
subplot(3, 1, 3);
f = medfilt1(f, 5);
stem(f, 'MarkerSize',3);
%axis([0 72 0 400]);
xlabel('帧数(n)');

ylabel('频率(Hz)');

```

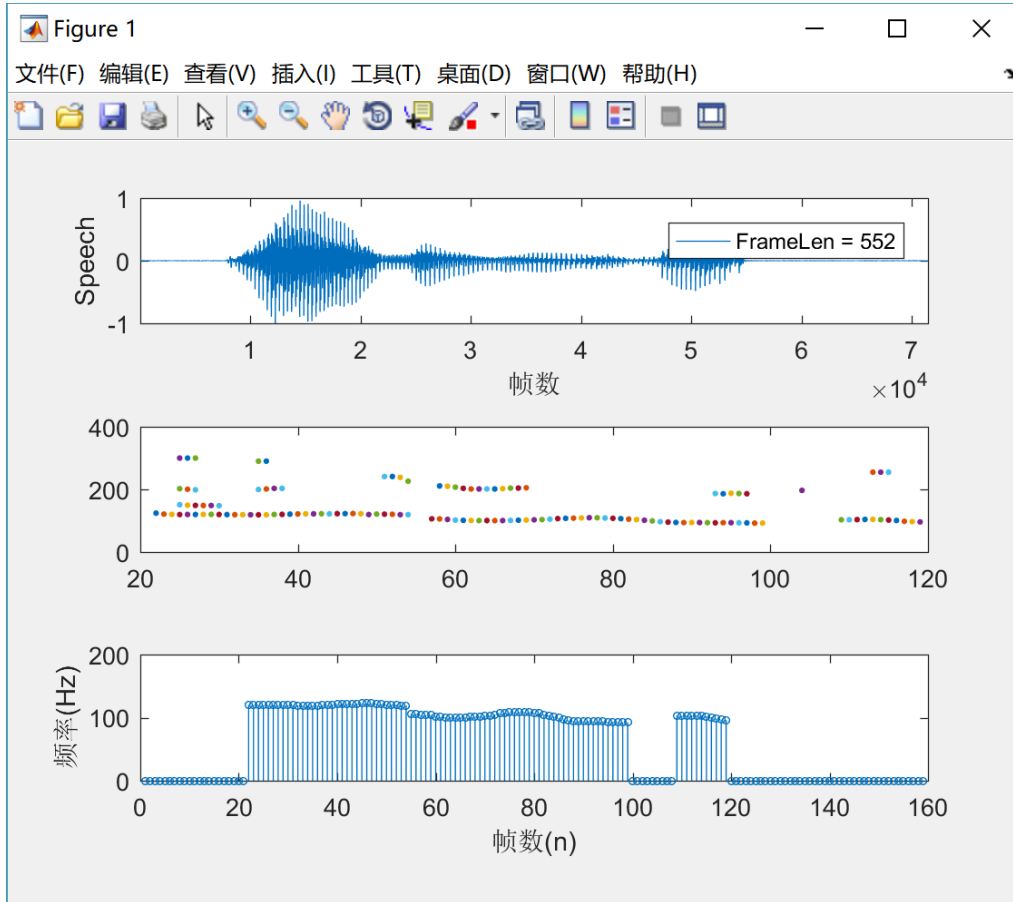
主要只需要调整寻找峰值'MinPeakHeight'和'MinPeakProminence'参数 `findpeaks(R(n, kl:kr), 'MinPeakHeight', R(n, 1) * 0.25, 'MinPeakProminence', 0.1);`

下图中二是每帧候选点，通过动态规划得到基频曲线见图三。

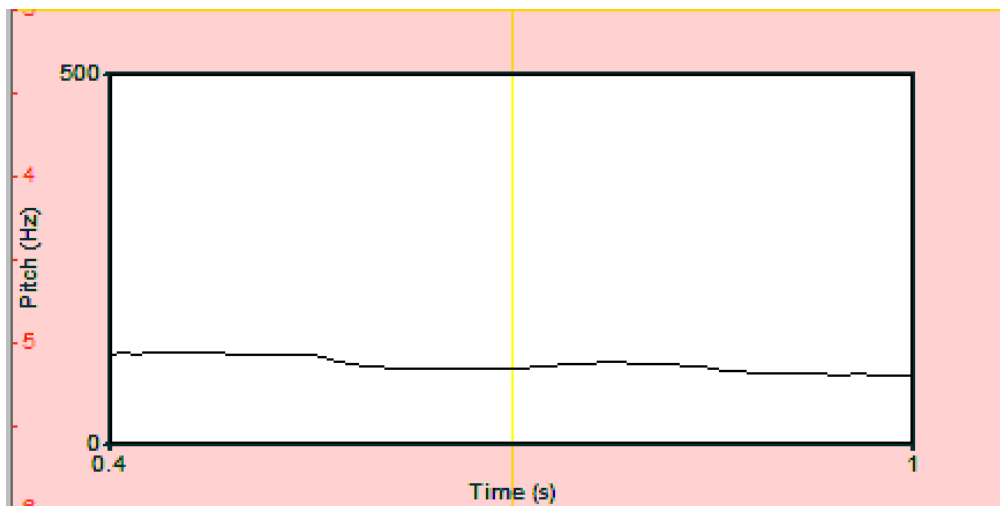


4. 自行录制一段连续语音发音，对检测结果进行分析，针对观察到的问题优化算法

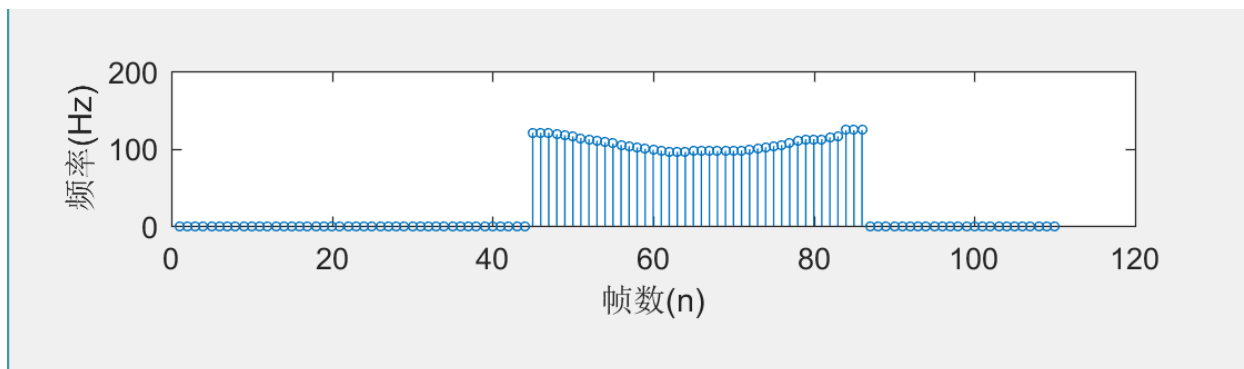
录制 1056，并调整寻找峰值'MinPeakHeight' 和 'MinPeakProminence'参数，



与 Praat 生成基频曲线对比基本类似。



另上面的连续发音不太能看出基频与音调关系，又单独录制了一个五，调整参数得到下图



五、实验体会

通过这次实验，首先是对 matlab 的使用更加熟悉，可以根据文档选择使用函数，以及掌握了部分通过 matlab 矩阵简化代码的技巧，然后对于短时自相关和基频曲线估计以及动态规划算法有了更深的了解。