第 38 卷 第 9 期
Vol.38 No.9
计 算 机 工 程
Computer Engineering
2012 年 5 月
May 2012
• 博士论文 •
文章编号：1000—3428(2012)09—0004—04 文献标识码：A 中图分类号：TN911.72

# 哼唱检索中联合音高与能量的音符切分算法

王恩成[1]，苏腾芳[1]，袁开国[2]，伍淳华[2]，王玉庆[2]
(1. 北方工业大学信息工程学院，北京 100144；2. 北京邮电大学计算机学院，北京 100876)

**摘 要：**为改善哼唱检索系统中利用旋律轮廓和节奏进行匹配的性能，提出一种新的联合音高与能量的音符切分算法。该算法改进基于自相关的基音提取算法，对提取的基音频率曲线进行后处理，并在切分过程中保持能量的分割信息，利用半音曲线的突变做切分，以提高音符切分的准确度。实验结果表明，在安静实验室环境下，该算法能获得 88.75%的分割准确度。

**关键词：**基音提取；自相关函数；旋律轮廓；节奏；哼唱检索；音符分割

# Note Segmentation Algorithm Combining Pitch and Energy in Query by Humming

WANG En-cheng[1], SU Teng-fang[1], YUAN Kai-guo[2], WU Chun-hua[2], WANG Yu-qing[2]
(1. School of Information Engineering, North China University of Technology, Beijing 100144, China;
2. School of Computer, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**【Abstract】**In order to improve the retrieval performance of using melody contour and rhythm in Query By Humming(QBH), this paper proposes a new algorithm using pitch and energy for note segmentation. Autocorrelation pitch extraction algorithm is improved. Post processing is used to smooth the pitch frequency curves, and at the same time, energy segmentation information is maintained in the process of note segmentation. Semitone curve is used for segmentation. This algorithm improves the accuracy of the note segmentation. Experimental results indicate that the accuracy rate of the algorithm is 88.75% in quiet laboratory environment which is of great significance in QBH.

**【Key words】**pitch extraction; auto correlation function; melody contour; rhythm; Query By Humming(QBH); note segmentation
**DOI:** 10.3969/j.issn.1000-3428.2012.09.002

## 1 概述

哼唱音乐检索(Query By Humming, QBH)通过哼唱歌曲的某个片段来找到想要查找的歌曲，是一种基于内容的音乐信息检索方式。

Ghias 等人[1]是最早的基于哼唱来检索乐曲的研究者，他们提出了哼唱检索系统的框架，并给出音高轮廓(pitch contour)的概念。在不考虑节奏特征情况下，把旋律的音高起伏表示成(U,D,S)的符号序列，匹配的时候使用了经典的字符串快速近似匹配法。

在哼唱检索系统中，主要的任务就是将用户的哼唱旋律准确地描述出来，但目前还没有很好的算法对音符序列进行准确切分，自动音符切分还是一个技术难题。音符切分主要是检测一个音符的起始点和终止点等信息，从而将一段音乐分割成单个音符，而其中的关键和核心是音符的起始点检测技术。音符起始点检测方法是寻找音乐信号中的起始点瞬时突变区域[2]。其中，瞬时突变区域有很多定义，如能量突变点、信号短时频谱变化等。

文献[3]比较了现有各类音符起始点检测算法，发现加权高频成分算法和自适应统计模型算法对于有调击弦打击类乐器(如钢琴)、有调弓弦非打击类乐器(如提琴)、无调打击类乐器(如鼓)及复杂混合乐音等具有更为优良的总体检测性能。文献[4]结合人耳听觉模型，提出一种基于差分全相位 MFCC 的音符起点检测算法，通过全相位预处理减小频谱泄露引起的频谱模糊，取得了较好的检测效果。

虽然在时域和频域上有许多适用于语音与音乐信号的音符切分方法，但是并不一定都适用于哼唱检索系统。其中，幅值跟踪检测算法在噪声信号下会失效，而加权高频成分算法和自适应统计模型算法中选取峰值检测函数是一个难点。针对这些问题，本文联合基频曲线与时域能量特征，提出一种计算量低、准确度高、无需峰值检测函数的音符切分算法。

## 2 算法流程与基音提取

本文算法的流程如图 1 所示。

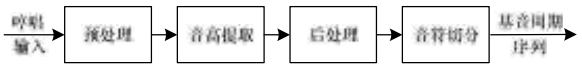



**图 1 本文算法流程**

### 2.1 预处理

假定哼唱纯净信号 $x(n)$ 被加性高斯白噪声 $v(n)$ 污染，则含噪哼唱信号 $y(n)$ 可表示为：

$$y(n)=x(n)+v(n) \tag{1}$$

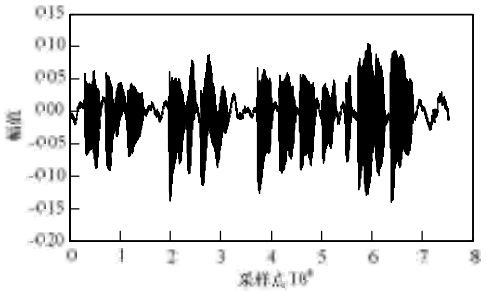
输入信号 $y(n)$ 经过一个截止频率为 60 Hz、阻带衰减为 30 dB 的 4 阶 ChebyshevII 型高通滤波器以抑制 50 Hz 的电源干扰，然后利用快速傅里叶变换(FFT)将时域信号转换为频域

**基金项目：**国家自然科学基金资助项目(60821001)；中央高校基本科研业务费专项基金资助项目(2011RC0210)；北京市教委面上基金资助项目(KM201010009005)

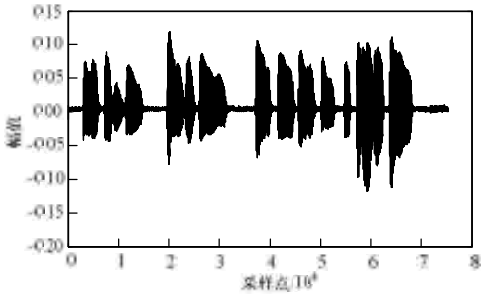
**作者简介：**王恩成(1976－)，男，讲师、博士，主研方向：信号处理，信息处理；苏腾芳，硕士研究生；袁开国、伍淳华，讲师、博士；王玉庆，硕士研究生

**收稿日期：**2011-12-16 **E-mail：**sutengfang@sina.com

信号，将高于 900 Hz 的 FFT 系数置 0，保留一二次谐波，然后进行 FFT 逆变换得到预处理后的哼唱信号 $y_{inv}(n)$[5]，图 2 给出含噪信号及其预处理后的信号。




(a)含噪的哼唱信号




(b)预处理后的哼唱信号

**图 2　预处理前后的哼唱信号**

### 2.2　自相关函数基音提取算法

语音信号的基音周期变化很大，从儿童到老年人，其基音频率 $f$ 的范围基本在 50 Hz~500 Hz 之间，自相关函数是对信号进行短时相关分析时最常用的特征函数。预处理后的哼唱信号 $y_{inv}(n)$ 经窗长为 $N$ 的窗函数 $w(n)$ 进行加窗，得到加窗信号 $S_n(m)$，定义每帧的自相关函数 $R_n(k)$ 为：

$$R_n(k)=\sum_{m=0}^{N-k+1}S_n(m)S_n(m+k) \tag{2}$$

由于自相关函数的性质，在基音周期的整数倍位置上会出现峰值，因此可通过峰值检测来提取基音周期值。自相关算法有很好的准确度，但也容易出现基音周期的倍数检测错误，影响音符的切分，下文将提出改进的自相关算法。

### 2.3　改进的自相关函数基音提取算法

为了更好地提取出正确的基音周期，对自相关算法进行改进，主要有 3 个方面。

(1)由于计算自相关函数的运算量很大，为了减少自相关算法的计算量，采用了三电平中心滤波法修正的互相关法，设中心削波的输出信号为 $Z_n(m)$，则：

$$Z_n(m)=\begin{cases}1 & S_n(m)>T\\ 0 & |S_n(m)|\quad T\\ -1 & S_n(m)<-T\end{cases} \tag{3}$$

一般情况下，削波电平 $T$ 取该帧语音最大幅度的 60%~70%。本文削波电平 $T$ 的选取方法如下：

$$T=0.65\times\min(\max(S_n(1:N/2-1)),\max(S_n(N/2:N))) \tag{4}$$

由此可得互相关计算法如下：

$$R'_n(k)=\sum_{m=0}^{N-k+1}S_n(m)Z_n(m+k) \tag{5}$$

(2)利用能量信息粗判清浊信号，能量信号在高信噪比时具有良好的音符切分性能，利用这个性质，能提高音符切分的准确性。设输入信号的短时能量用 $E_n$ 表示，则：

$$E_n(m)=\sum_{m=1}^{N}S_n^2(m) \tag{6}$$

由于短时能量对高电平非常敏感，因此采用短时平均幅度函数 $M_n$ 代替 $E_n$，定义为：

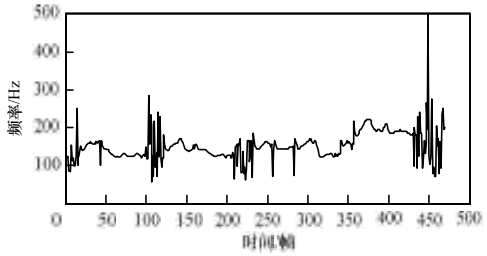
$$M_n(m)=\sum_{m=1}^{N}|S_n(m)| \tag{7}$$

当一帧的短时平均幅度函数的均值小于整段语音信号的平均幅度函数均值的 30%时，则认为此帧信号为无声帧或清音，将此帧的基音周期与基音频率设为 0。

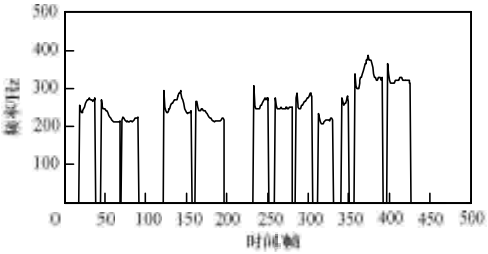
(3)基音周期的倍数检查主要是防止倍音被错误的当成基音周期，此过程需要当前帧的基音周期 $pitch$ 和阈值 $D_{th}$，设：

$$D_{th}=\begin{cases}0.5 & pitch>100\\ 0.75 & \text{其他}\end{cases} \tag{8}$$

找到满足条件 $R'_n(pitch/T_k)>D_{th}R'_n(pitch)$ 最大的 $T_k$，其中，$pitch/T_k\geqslant 20$，$T_k=8,7,\cdots,2$。如果找到满足条件的 $T_k$，则将 $pitch/T_k$ 前后各 5 点共 11 点提取峰值，将其中峰值最大的点作为当前点的基音周期，如果没有峰值，则保持原来的基音周期。为了保证在基音周期较小时基音周期提取的准确性，当 $T=pitch/T_k<30$ 时，倍数确认过程输出的是 $T$ 点附近最大峰值、$2T$ 点附近最大峰值中峰值较小的值。如果没有峰值则保持原来的基音周期，图 3 给出改进前后的基频曲线。




(a)自相关函数法提取的基频曲线




(b)互相关函数法提取的基频曲线

**图 3　改进前后的基频曲线**

## 3　后处理

哼唱信号通过改进的基音提取算法后，已经能够获得较为准确的基音频率序列，但该序列仍然存在由于环境噪声、气流声等引起的孤立点和倍音错误。为进一步降低错误率，一般采用后处理对基音周期初步估计结果进行平滑处理。

### 3.1　后处理流程

后处理部分的流程如图 4 所示。

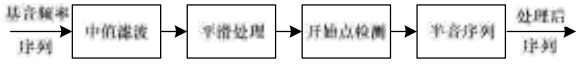



**图 4　后处理流程**

### 3.2　中值滤波

中值滤波是基于排序统计理论的一种能有效抑制噪声的非线性信号处理技术。中值滤波的主要内容是将数字序列中一点的值用该点的一个邻域中各点值的中值代替，让周围的

值接近真实值，从而消除孤立的噪声点。

### 3.3 平滑处理

中值滤波后，一个音符内可能存在多个连续的变化点，这些变化点可能会直接导致对一个音符进行多次切分，因此中值滤波后的基频曲线并不适合直接用于切分音符，为此增加了平滑处理过程。通过平滑处理，少量的突变点被清除，使一个音符的音高估计值更加精确。下面给出伪代码：

```
1: OUTSEQ=SMOTHPROCESS(input,T1,T2)
2: len=length(input);
3: for i=2 to T2{
4:       for j=1 to len-i{
5:          if abs(input(j+i)- input(j))<T1
6:                 &&abs(input(j+i)- input(j))>0{
7:                 for k=1 to i-1{
8:                    if input (j+k)!=0{
9:                    input(j+k)= (input(j)+ input(j+i))*0.5;
10:                          }
11:                      }
12:                  }
13:           }
14:    }
15：OUTSEQ= input;
```

在上文伪代码中，有 2 个阈值 $T1$ 与 $T2$，$T2$ 与帧移有关，$T1$ 与需要移除的跳变幅度有关。在本文中，窗长为 40 ms，帧移为 20 ms，因此，$T2$ 取 7，$T1$ 取 6。$T1$ 取值过大可能会导致正常的音高跳变点被平滑，$T2$ 转换成时长为 4 个窗长，合 160 ms，而一个音符的持续时长大都大于此值，因此，该阈值具有通用性而不影响音符的正常切分，图 5 给出平滑处理后的基频曲线。

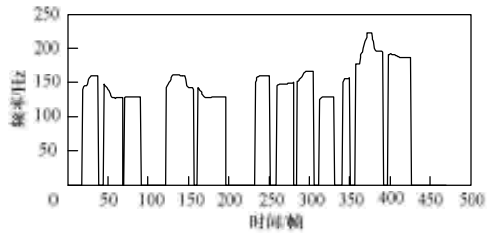



**图 5 平滑处理后的基频曲线**

### 3.4 半音序列

为了符合旋律信息的表达方式，按下式将基频曲线转换为半音曲线：

$$P = 69 + 12 \times \mathrm{lb}(f / 440) \tag{9}$$

具体的转换过程分为 3 步：

(1)取阈值 $T_s$ 为基频曲线中移除零点的均值的 60%。

(2)将第一个大于阈值 $T_s$ 的点记为开始点，开始点以前的点认为无声段或噪音段，被排除掉。

(3)将大于 0 的基音频率值转换为半音，图 6 给出处理后的半音序列。

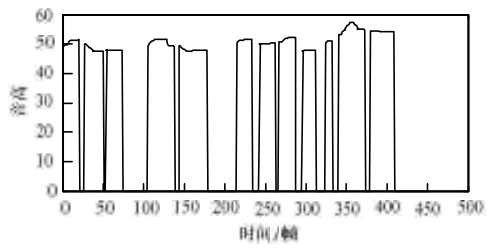



**图 6 处理后的半音序列**

## 4 音符切分

在音乐旋律中，大部分音符的跳变都超过一个半音，因此，利用一个半音作为阈值将有很好的效果，但在实际应用中，人们哼唱的并不如想象的那样准确，连音与跳变低于一个半音的情况随处可见，为了进一步提升哼唱检索的性能，采用了一些处理来减少这样的切分误差，提升音符切分的准确度。

### 4.1 音符粗切分

通过半音序列的生成，得到了半音曲线，因此，首先以 0.7 个半音为阈值将单个跳变点平滑成与它最近的半音值，然后检测跳跃超过 0.7 个半音的点，将 2 个跳跃点之间的值设为它们的均值，得到了平滑的半音序列，图 7 给出粗切分的半音序列。

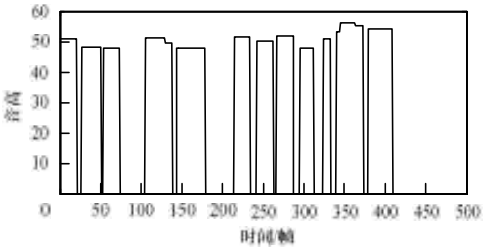



**图 7 粗切分半音序列**

### 4.2 音符合并

因为将阈值设为 0.7 个半音而不是 1 个半音，这样可能会产生错误的分割信息，那么需要对错误的分割进行合并而又不能影响到正确的分割信息。为了实现这个过程，将其分为 4 步：

(1)由于拥有了平滑的半音序列，因此可以提取出音高序列及其对应的持续音长序列。最后一个未跳变的音符由于不知它是否是一个完整的音符而被抛弃。

(2)找出音高大于 0 并且持续音长大于 7 的值组成新的持续音长序列，并对其进行排序，设排序后的持续音长序列为 $N_{\mathrm{seq}}$，由此设合并阈值 $T_{\mathrm{comb}}$ 在 $N_{\mathrm{seq}}$ 的长度大于等于 3 时，阈值为 $N_{\mathrm{seq}}$ 的 1/3~2/3 处均值的一半与 7 的最小值，在 $N_{\mathrm{seq}}$ 的长度小于 3 时，阈值为 $N_{\mathrm{seq}}$ 均值的一半与 7 的最小值。

(3)考虑不应合并的音符：1)音符持续时长大于 $T_{\mathrm{comb}}$ 或音符对应的音高为 0 的音符；2)持续时长大于 4 但小于等于 $T_{\mathrm{comb}}$ 并且相邻的音符音高均为 0 的音符。

(4)对其他的情况进行合并，将需要合并的音符并入与其音高最近的音符。如果并入下一个音符，则下一个音符的持续时长将加上当前音符的持续时长。继续进行下一个音符的处理，直到处理结束。图 8 给出音符合并后的半音曲线。

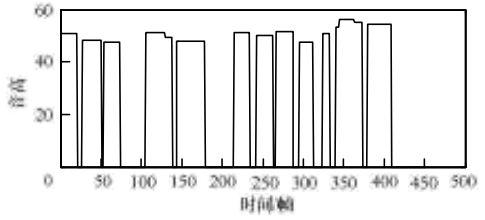
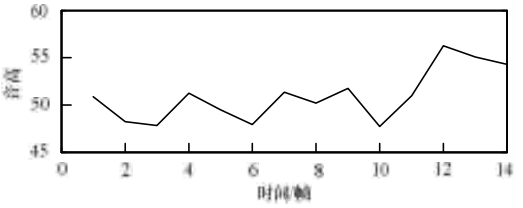



**图 8 音符合并后的半音曲线**

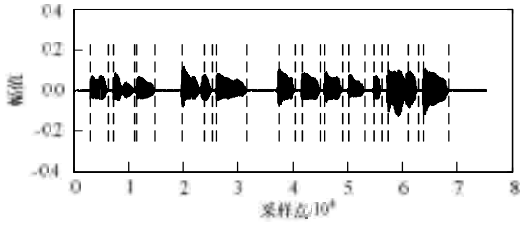
### 4.3 旋律轮廓曲线

通过音符合并，得到了最后的平滑半音曲线。利用此半音序列，再次提取出音高序列及其对应的持续音长序列，再将音长信息转换为时间信息，就能切分出原始音符。对音高

序列进行去零操作，得到了哼唱信号的旋律轮廓曲线。图 9 给出了音符旋律轮廓曲线与预处理后的哼唱信号的切分效果图。




(a)音符旋律轮廓曲线




(b)预处理后哼唱信号的切分效果

**图 9　音符旋律轮廓曲线与音符切分效果**

## 5　实验结果与分析

为了测试音符切分算法的准确性，实验采用了实验室安静环境下录取的约 10 s 的 25 个哼唱音频片段，一共 400 个音符，采样率为 8 000 Hz，16 bit 量化。为了确保对比音符的正确性，对哼唱音频片段的切分进行了人工干预，得到正确的切分信息。错误分析采用粗音符切分错误分析(Gross Note Error, GNE)。定义 GNE 错误偏差为 20%，即当音符切分点与标准音符切分点偏差小于当前音符持续时长的 20%时，则认为此音符切分正确，否则认为是切分错误。

算法性能主要用以下 4 个指标来衡量：总音符错误率，插入音符错误率，删除音符错误率及切分音符错误率。其中定义：总音符错误率=插入音符错误率+删除音符错误率+切分音符错误率。

系统的性能与基音提取算法也有密切的关系，因此，先测试改进的自相关算法的性能，实验过程中基音提取算法采用的对比算法有改进前的自相关算法(ACF)、CAMDF 和 YIN[6]算法，错误率分析采用粗基音错误率(Gross Pitch Error, GPE)：

$$E(n)=(|E_c(n)-E_p(n)|/E_p(n))\times 100\% \tag{10}$$

其中，$n$ 表示第 $n$ 帧；$E_c(n)$ 为各种基音提取算法得到的结果；$E_p(n)$ 为人工干预的基准值，当 $E(n)$>20%时则认为基音提取出错，是无效的检测帧。GPE 定义为无效检测帧数占总的帧数的百分比，表 1 显示了不同算法在 GPE 下的错误率。

**表 1　不同算法在不同信噪比下的粗基音错误率**　(%)

| 算法 | −10 dB | 10 dB | 30 dB | 50 dB | ∞ |
|---|---|---|---|---|---|
| ACF | 76.7 | 73.0 | 6.3 | 2.6 | 2.4 |
| CCF | 75.9 | 71.7 | 5.9 | 2.1 | 1.9 |
| CAMDF | 88.0 | 78.3 | 26.7 | 18.7 | 18.2 |
| YIN | 82.9 | 78.9 | 53.7 | 5.9 | 4.8 |

由于本文改进的自相关算法基音提取准确率在 GPE 准则下已经达到了 98.1%，因此音符切分算法不考虑基音提取算法对音符切分的影响。通过对实验数据的分析，总音符错误率为 11.25%，插入音符错误率为 1.5%，删除音符错误率为 8.5%，切分音符错误率为 1.25%，图 10 给出了各种错误类型的错误率。

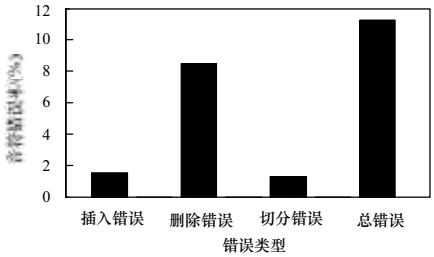



**图 10　音符错误率**

从图 10 可见，除总错误率外，删除错误率最高，这主要是由于哼唱时出现音高变化很细微的连音，或哼唱不正确产生的颤音。图 2 给出一段实验室自然安静环境下录制的哼唱信号，它的第 2 个、第 12 个音符由于哼唱不准确产生了颤音(类似嗯)。从图 9 的切分效果图可以看出，这个哼唱片段出现了 2 个错误，第 4 个音符切分点与标准音符切分点的偏差超过音长的 20%，因此出现了一个切分错误。第 12 个音符由于被分割成 2 个音符，因此出现了一个插入错误。从时域波形上看，第 2 个、第 12 个音符由于颤音现象都很像 2 个音符，当颤音较大时仔细听起来确实也很像一个音符，但对哼唱检索系统而言，旋律信息应该排除这种颤音。如果将颤音也当作音符进行切割，那么图 9 应该是第 2 个音符为删除错误。虽然总错误数目一样，但是实际的错误类型并不一样。本文以歌谱的旋律曲线为准，采用前一种类型的分配方式。

## 6　结束语

本文介绍了联合能量与音高的音符切分算法，在自然安静的实验室环境下音符识别总错误率仅为 11.25%，该算法不仅适用于用户自然哼唱，也可用于语音信号分割与带歌词的哼唱。虽然现在哼唱检索是一个热门的研究领域，但大部分的哼唱检索系统没有关注于自然哼唱语音的音符切分，基本都要求用户以一定的方式哼唱(如 DaDaDa 或 LaLaLa)，让每个音符之间的间隔比较明显，便于音符的准确切分。本文通过对哼唱乐音信号的分析，联合能量与音高特征，对哼唱信号进行音符切分，取得了良好的效果，为自然哼唱信号的音符切分提供了一个参考。

## 参考文献

[1] Ghias A, Logan J, Chamberlin D, et al. Query By Humming Musical Information Retrieval in an Audio Database[C]//Proc. of the 3rd ACM International Conference on Multimedia. San Francisco, USA: [s. n.], 1995.

[2] 李国辉, 李恒峰. 基于内容的音频检索: 概念和方法[J]. 小型微型计算机系统, 2000, 21(11): 1173-1177.

[3] Bello J P, Daudet L, Abdallah S, et al. A Tutorial on Onset Detection in Music Signals[J]. Speech and Audio Processing, 2005, 13(5): 1035-1047.

[4] 关　欣, 李　锵, 田洪伟. 基于差分全相位 MFCC 的音符起点自动检测[J]. 计算机工程, 2010, 36(11): 25-26.

[5] Shahnaz C, Zhu W P, Ahmad M O. A Pitch Detection Method for Speech Signals with Low Signal-to-Noise Ratio[C]//Proc. of International Symposium on Signals, Systems and Electronics. Montreal, Canada: [s. n.], 2007.

[6] Cheveigne A D, Kawahara H. YIN, a Fundamental Frequency Estimator for Speech and Music[J]. Speech Processing and Communi- cation Systems, 2002, 111(4): 1917-1930.

编辑　任吉慧