Multi-Pitch Streaming: Based on Clustering Algorithms

Yi Gao
University of Rochester, New York, United States
Department of Electrical and Computer Engineering
Ygao20@z.rochester.edu

ABSTRACT

Multi-pitch analysis is an important subjects in audio signal processing, while at the same time it is also a very challenging issue. In this paper, we are mainly focusing on the multi-pitch streaming part, while in order to perform the streaming process, we need first estimate the pitch values in individual time frame. Thus a multi-pitch estimation process is also provided here. Recently both multi-pitch estimation and multi-pitch streaming are receiving many research interests and several advances in these two areas were made.

In this paper we will describe two categories of multi-pitch estimation or MPE methods. And then we are going to introduce the basic concept of clustering as well as several clustering algorithms based on this concept. For each clustering algorithm presented, a recently proposed multi-pitch streaming method using this clustering algorithm will be detailed discussed. Experimental results and some defects of these methods will also be provided.

1. INTRODUCTION

When we are sitting in a noisy bar and talking to some of our friends, we can usually keep track of our conversation even though the friends' voices may partially overlap the voices of other speakers. This is called cocktail party problem [1]. A similar problem is when we listen to a symphony or other music with overlapping harmonic components emitted by different instruments, we can easily notice and track the sounds from different instruments even without musical training.

In [1], Simon Haykin and Zhe Chen categorized the solution of the problem into three underlying neural processes: Analysis, Recognition and Synthesis. The analysis part, the fundamental process for the entire solution, involves segmentation of incoming auditory signal to individual channels or streams. And a major cue for doing so is their pitch [2] despite other cues such as harmonic relationships among spectral components (harmonicity), temporal onsets/offsets, timbre, and patterns of amplitude modulation [3][4]. Thus, to understand how music is perceived, we need to understand how the pitch of a sound is determined. The estimation of pitch values of all sources at each individual time frame is a process known as the multi-pitch estimation or MPE. Although the detection of pitches from a single source is a solved problem, multi-pitch estimation are still open problems.

Once the pitches of every time frame are detected, we need to stream estimated pitches into a single pitch trajectory. To perform this, timbre information and some clustering algorithms are required. And this process is called multi-pitch streaming by [5].

In this paper, we will focus on the multi-pitch streaming process. This paper will be divided into 2 major sections: section 2 will describe the current methods of multi-pitch estimation and section 3 will describe the concept of clustering and some clustering algorithms can be used in multi-pitch streaming.

2. MULTI-PITCH ESTIMATION

Before the introduction of multi-pitch estimation, I would like to go through the well addressed issue: mono-pitch estimation. There are many approaches for mono-pitch estimation. For example, L. Rabiner proposed a method of performing autocorrelation analysis to detect the pitch [6], A. De Cheveigné and H. Kawahara proposed a well-known YIN algorithm for pitch detection [7]. Both these methods try to detect the pitch by retrieving a periodic pattern in a waveform. Other approaches beside these two utilizing the time domain properties are [8][9][10], they managed to find the pitch by utilizing the property that harmonics are at integer multiples of pitch. [11] and [12] also proposed some methods by combining both spectral and temporal cues. Another kind of approach is the one proposed by D. G. Childers who uses cepstrum information for pitch detection [13].

Although we do have so many approaches in mono-pitch estimation, the multi-pitch estimation still remains an

open area for further study. The results of current multi-pitch estimation is not as satisfying as that of mono-pitch estimation. However, the multi-pitch scenario occurs regularly in music signals, perhaps even more frequently than the single-pitch case, and often also in speech processing.

2.1 Autocorrelation Function (ACF)

Unlike the mono-pitch estimation using ACF, the multi-pitch estimation using ACF faces several problems. For example, one or two musical instruments or voice sources may have lower signal energy than others, when we use the autocorrelation analysis for the mono-pitch estimation here, the fundamental frequency of the source with lower energy may not be found and the harmonics of sources with higher energy may be detected as a fundamental frequency.

To solve this problem, Ray Meddis and Michael J. Hewitt proposed a method [14]. The stimulus first pass a bandpass digital-filter system, then autocorrelation analysis is performed on each channel. After that cross-channel summation of the ACFs is performed to form a pooled ACF and we can have a pitch at period p_1 by finding the peak in the pooled ACF. Then all the channels which do not have a peak at p_1 in their own ACF form a new (reduced) pooled ACF and we can find a new pitch in this new pooled ACF. Repeat this process until we will have multiple pitches.

Instead of finding multiple peaks from a single pooled ACF, R. Meddis and M.J. Hewitt's method tries to find the single peak from multiple pooled ACF and each new formed pooled ACF eliminates the frequency channel that contributes a pitch in the previous step. By doing so, even the pitch from sources which have lower signal energy can be detected. However, this method may still have a problem: when two pitch are really close in frequency domain, one of them may not be detected.

2.2 Mathematic Model

M. Davy and S. Godsill in 2006 first proposed such a mathematic model in their paper [15]. And in 2009 M.G. Christensen proposed a slightly modified mathematic model [16]. Here, in this paper, we will introduce the model provided by M.G. Christensen.

Consider a signal consisting of several, say K, sets of harmonics (hereafter referred to as sources) with fundamental frequencies ω_k, for k = 1, . . . , K, that is corrupted by an additive white complex circularly symmetric Gaussian noise, ω (n), having variance σ^2, for n = 0, . . . , N − 1, i.e.,

$$x(n) = \sum_{k=1}^{K} \sum_{l=1}^{L} a_{k,l} e^{j\omega_k ln} + \omega(n) \qquad (1)$$

where $a_{k,l} = A_{k,l} e^{j\phi_{k,l}}$, with $A_{k,l} > 0$ and $\phi_{k,l}$ being the amplitude and the phase of the *l*'th harmonic of the *k*'th source, respectively. The problem is then to estimate the fundamental frequencies $\{\omega_k\}$, or the pitches, from a set of *N* measured samples, $x(n)$. It seems impossible to obtain the fundamental frequencies $\{\omega_k\}$ and all other parameters only from the observed $x(n)$. While, if we have some well-founded principles from statistical signal processing and use some estimators proposed by [16], we can find the estimates of fundamental frequencies $\{\omega_k\}$. And once given the fundamental frequencies $\{\omega_k\}$, the amplitudes and phases can easily be found using one of the estimators proposed in [17].

In paper [16], M.G. Christensen presented an approximate nonlinear least-squares (NLS) method, a MUltiple SIgnal Classification (MUSIC) based method as well as a Capon-based method. According to the author, these three methods have the same simple form which is list below:

$$\{\hat{\omega}_k\} = arg \max_{\{\omega_k\}} \sum_{k=1}^{K} J(\omega_k) \qquad (2)$$

where the function $J(\cdot)$ depends only on the source k. This means that an estimate of the set of fundamental frequencies can be obtained by evaluating a cost function $J(\omega_k)$ for a coarse grid of values and then picking the *K* highest peaks. The difference between these three methods is the different definition of the cost function $J(\cdot)$. For more on these detailed definition and the results comparison of these three methods, I hereby refer the interested reader to [16] and the references therein.

3. MULTI-PITCH STREAMING

Once we know the pitches of each time frame, we need to stream these pitches into a single pitch trajectory for each source. Pitches in this trajectory share "similar" timbre feature, in some angle a stream is in fact a cluster so stream segregation can be modelled as a clustering problem.

This streaming part may be the most difficult and the most important part of the source separation. It is important since without this process, the pitch itself is useless. Only with these estimated pitches, it is clearly impossible to perform the recognition and synthesis part proposed by [1]; And it is difficult since recent method usually do not have a satisfying streaming results. We really need some well-designed methods to perform the multi-pitch streaming, and maybe some clustering algorithm ideas will help us doing so.

3.1 Concept of Clustering

Everitt [19] documents some of the following definitions of a cluster:

1. "A cluster is a set of entities which are alike, and entities from different clusters are not alike."
2. "A cluster is an aggregation of points in the test space such that the distance between any two points in the cluster is less than the distance between any point in the cluster and any point not in it."
3. "Clusters may be described as connected regions of a multi-dimensional space containing a relatively high density of points, separated from other such regions by a region containing a relatively low density of points."

Clustering is sometimes referred to as unsupervised classification [18] and it can be further described as follow:

If:

$(1) U = \{p1, p2, \dots, pn\}$

$(2) C_t \subseteq U, t = 1,2, \dots, k, C_t = \{p_{t_1}, p_{t_2}, \dots, p_{t_w}\}$

$(3) proximity\ (p_{m_s}, p_{t_r})$

Then:

$(1) \bigcup_{t=1}^{k} C_t = U.$

$(2) \forall\ C_m, C_r \subseteq U, if\ C_m \neq C_r, then\ C_m \cap\ C_r = \emptyset\ (for\ exculusive\ clustering\ only)$

$(3) MIN_{\forall p_{m_u} \in C_m, \forall p_{r_v} \in C_r, \forall C_m, C_r \subseteq U \& C_m \neq C_r} (proximity(p_{m_u}, p_{r_v})) > MAX_{\forall p_{m_x}, p_{m_y} \in C_m, \forall C_m \subseteq U} (proximity(p_{m_x}, p_{m_y}))$

U is the set of all the data points, pi is the p_{th} point and i={1,2,…,n}; Then *Ct* describes the t_{th} cluster and there are overall k clusters, p_{t_w}means the w_{th} point in the t_{th} cluster; Proximity depends on the algorithm used here. In most cases, it is a kind of distance between $p_{m_s}$and p_{i_r}. The conclusion part reveals three major things. First, the union of all the clusters is the set U, which means every point in set U is clustered. Second, if two clusters are not the same, the intersection of these two clusters is an empty set. In other words, each point in the set U is only bet clustered to one cluster. Then, the last one says that the proximity of any two points in any two different clusters must be larger than proximity of any two points in the same cluster. This can also be explained as: the minimum proximity of two points belong to two different clusters is larger than the maximum proximity of two points belong to the same cluster.

3.2 Hierarchical Clustering Algorithm

A hierarchical clustering is often displayed graphically using a tree-like diagram called a dendrogram, which displays both the cluster-subcluster relationships and the order in which the clusters were merged (agglomerative view) or split (divisive view). Based on whether the cluster is merged or split, we can category the hierarchical clustering algorithm into two major part: agglomerative hierarchical clustering technique and divisive hierarchical clustering technique.

Agglomerative: *Start with the points as individual clusters and, at each step, merge the closest pair of clusters. This requires defining a notion of cluster proximity.*

Divisive: *Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide which cluster to split at each step and how to do the splitting.*

According to [20], hierarchical clustering algorithm has four advantanges: 1) does not require the number of clusters to be known in advance, 2) computes a complete hierarchy of clusters, 3) good result visualizations are integrated into the methods, 4) a "flat" partition can be derived afterwards (e.g. via a cut through the dendrogram). Some examples of hierarchical clustering algorithms are: Balanced Iterative Reducing and Clustering using Hierarchies-BIRCH [21], Clustering Using REpresentatives- CURE [22] and CHAMELEON [23]. For more information on these clustering algorithms, please go through paper [20] and the references therein.

In 2003, W.M. Szeto and M.H. Wong proposed an agglomerative hierarchical clustering method for multi-pitch streaming [24]. In their algorithm, they first assume the pitch, the starting time and the ending time of each musical note are obtained. And a vector combines these three are called an event. The starting time and the ending time are in the unit of seconds and the pitch is in the unit of MIDI number. Then they defined the inter-event distance (EDIST) as:

$$EDIST(e_1, e_2) = \begin{cases} \sqrt{(\alpha d)^2 + (\beta(p_1 - p_2))^2} & if\ e_1 \nparallel e_2 \\ \infty & if\ e_1 \parallel e_2 \end{cases} \quad (3)$$

where α is the time weighting factor and the β is the pitch weighting factor. $e_1 \nparallel e_2$ means event $e_1$and e_2 are not overlapping in time domain while $e_1 \parallel e_2$ means these two events are overlapping in time domain. And d is the ending time of event 1 minus the staring time of event 2. For example we can have the distance matrix in table 1 of all the five event examples shown in figure 1. Their algorithm is listed on the next page.

In algorithm 1, C_i is the i_{th} cluster, a cluster can have one or more events and is also a vector with three parameters: staring time, ending time and a set of clustered events. Unlike the event, the starting time of cluster is defined as the first starting time of all its events and the ending time is

Algorithm 1 Adapted single-link clustering algorithm

1: Choose $\mathfrak{R}_0 \leftarrow \{C_i = \{e_i\}, i = 1, 2, \dots, N\}$ as the initial clustering.
2: $t \leftarrow 0$
3: repeat
4: $t \leftarrow t + 1$
5: Among all possible pairs of clusters (C_r, C_s) in $\mathfrak{R}_{t-1}$, find C_i, C_j such that CDIST(C_i, C_j)=$min_{r \neq s} CDIST(C_r, C_s)$
6: if CDIST(C_i, C_j)=∞ then
7: break
8: Merge C_i, C_j into a single cluster called $C_q$and form $\mathfrak{R}_t \leftarrow (\mathfrak{R}_{t-1} - \{C_i, C_j\}) \cup C_q$
9: until $\mathfrak{R}_{N-1}$ clustering is formed, that is, all events lie in the same cluster.

the last ending time of all its events. By doing so, we can easily judge whether two clusters are overlapping together. The inter-cluster distance CDIST here is defined as:

$$CDIST(C_1, C_2) = \begin{cases} min_{e_i \in C_1, e_j \in C_2}(EDIST(e_1, e_2)) if C_1 \nparallel C_2 \\ \infty \qquad\qquad\qquad\qquad if\ C_1 \parallel C_2 \end{cases} \tag{4}$$

The idea of this algorithm is to assign each event to a cluster for the first step, then find the cluster pairs have the minimum distance. Then these two clusters are merged into a new cluster and a new cluster pair with minimum distance within the new clusters set will be found. Repeat this process we will end up with a hierarchy structure. Once given the number of output clusters, the clustering result will be obtained.

Although the author claims the experiment results tested on Johann Sebastian Bach's two-part Invention No. 1 collected from [25] and Chopin' s Prelude No. 4 in E minor have very low error rates (0 and 0.025 respectively), this method suffers from one crucial problem: very large number of operations. The complexity of this algorithm is $O(N^3)$, this is something intolerable.

3.3 Model Based Algorithm

There's another way to deal with clustering problems: a model-based approach, which consists in using certain models for clusters and attempting to optimize the fit between the data and the model. Mentioned in [26], each cluster can be mathematically represented by a parametric distribution, like a Gaussian (continuous) or a Poisson (discrete). The entire data set is therefore modelled by a mixture of these distributions. Such an individual distribution used to model a specific cluster is often referred to as a component distribution.

A mixture model with high likelihood tends to have the two traits: 1) component distributions have high "peaks"

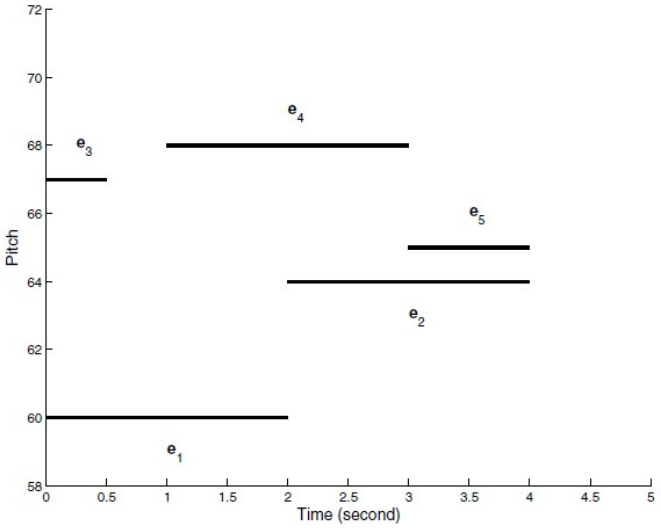

Figure 1. Five events example [24]

	e_1	e_2	e_3	e_4	e_5
e_1	-	4.00	∞	∞	5.10
e_2	4.00	-	3.35	∞	∞
e_3	∞	3.35	-	1.12	3.20
e_4	∞	∞	1.12	-	3.00
e_5	5.10	∞	3.20	3.00	-

Table 1. The inter-event distances of events in figure 1 (α=1; β=1)[24]

(data in one cluster are tight), 2) the mixture model "covers" the data well (dominant patterns in the data are captured by component distributions). Advantages of model-based clustering are: 1) well-studied statistical inference techniques available, 2) flexibility in choosing the component distribution, 3) for each cluster a density estimation is obtained, 4) a "soft" classification is available.

In 2005, H. Kameoka, T. Nishimoto, and S. Sagayama proposed a Gaussian model algorithm in multi-pitch estimation [27]. In their algorithm, they use Gaussian kernel model to represent the harmonic structure of a certain fundamental frequency and the power envelop over time. Assuming each frequency component distribution of an estimated fundamental log-frequency μ_k can be approximated by a Gaussian, a single harmonic structure can be modeled with a weighted sum of Gaussian kernels described as:

$$h_k(x) = \sum_{n=1}^{N} \frac{r_n^k}{\sqrt{2\pi\sigma_k^2}} exp\left[-\frac{\{x - (\mu_k + \log n)\}^2}{2\sigma_k^2}\right] \tag{5}$$

Where x is log-frequency, n and N are the n_{th} harmonic and total number of harmonics, k is the index of cluster and each weight parameter $r_n^k (\sum_{n=1}^{N} r_n^k = 1)$ is exactly related

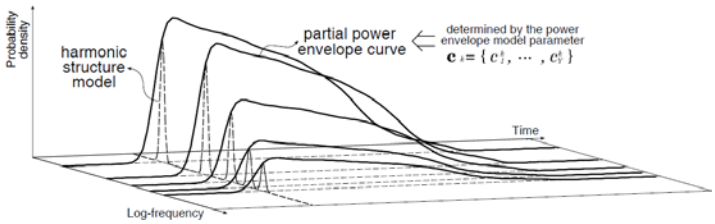

Figure 2. Parametric audio stream model

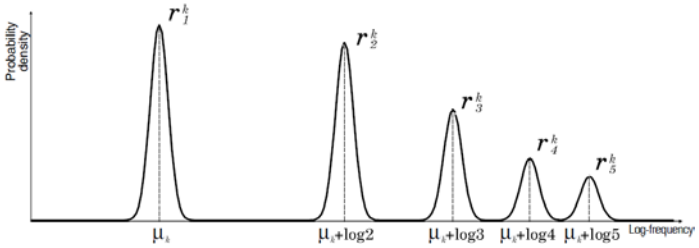

Figure 3. Gaussian kernel harmonic structure model

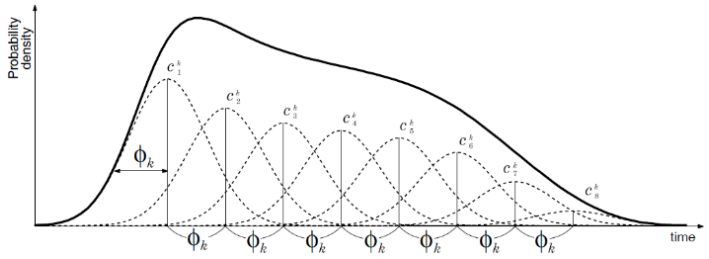

Figure 4. Gaussian kernel power envelope model

to the spectral components. In building the power envelop model, the author sets a specific feature: the standard deviation of each Gaussian and the interval of adjacent Gaussians are expressed with a same variable ϕ_k. Thus $g_k(t)$ is given as:

$$g_k(t) = \sum_{y=0}^{Y-1} \frac{C_y^k}{\sqrt{2\pi {\phi_k}^2}} \exp[-\frac{\{t-(O_k+y\phi_k)\}^2}{2{\phi_k}^2}] \tag{6}$$

where y and Y are the index and the number of the Gaussian kernels, O_k is the simply center of the forefront Gaussian and each of the Gaussian kernel is weighted with $c_y^k(\sum_{y=1}^{Y-1} c_y^k = 1)$. The harmonic structure model $h_k(x)$ of fundamental log-frequency μ_k of k_{th} audio stream is shown in figure 3, and the power envelop curve function $g_k(t)$ is shown in figure 4. Then, the k_{th} audio stream model $p(x,t|\Theta_k)$ is expressed as a multiplication of the 2 functions and power ω_k:

$$p(x,t|\Theta_k) = \omega_k h_k(x) g_k(t) \tag{7}$$

Where $\int_{\Omega_0}^{\Omega_1} h_k(x)dx = \int_{T_0}^{T_1} g_k(t)dt = 1$, different stream may have different power ω_k, while the summation of all the power of each stream is a constant F, thus $\sum_{k=1}^{K} \omega_k = F$. In equation (6), t and K are time (frame) and the total number of clusters, T_0, T_1 and Ω_0, Ω_1 are the lower and higher bounds of time (frame) and log-frequency ranges, respectively. The k_{th} audio stream model is shown in figure 2.

Then I would like to introduce the objective function which is well designed by the author. The objective function for this clustering is given as:

$$\sum_{k=1}^{K} \int_{T_2}^{T_1} \int_{\Omega_2}^{\Omega_1} (p(k|x,t,\Theta) f(x,t)) \times D(x,t|\Theta_k) dxdt \tag{8}$$

Where $f(x,t)$ is spectral density of wavelet transform spectrum. $p(k|x,t,\Theta$ is a membership probability of k_{th} cluster at the coordinates (*x*,*t*), depending on every model parameter Θ. Thus $p(k|x,t,\Theta)f(x,t)$ can be viewed as the spectral density of segregated audio stream. $D(x,t|\Theta_k)$ is related to $p(x,t|\Theta_k)$. I mentioned this objective function is well designed since the objective function will have the maximum value when $p(k|x,t,\Theta)f(x,t)$ and $p(x,t|\Theta_k)$ are close, which can be viewed as the observed distribution is close to the Gaussian kernel model distribution.

Therefore, if we have a prior knowledge or an expectation of how spectral and power envelops would shape like (then we can have the k_{th} audio stream model), we would obtain the clustering result by maximizing the objective function. This is basically the main idea of this algorithm. The author uses EM algorithm to update the parameter Θ of these models. Detailed process of EM algorithm and the use of prior distribution to prevent excessive deviation can be find in [27] and its references.

This algorithm obtained extremely high accuracy of 92.1% and 86.2% on two pieces of real music performance data excerpted from RWC music data. However, since the experiment was done with very limited test data and the pitches of the testing data are unlikely to change a lot in time domain, we cannot guarantee this method works well on all sort of musical data.

4. CONCLUSION

This paper has given two applications of clustering algorithm in multi-pitch streaming. Based on the results of these two methods, some future work including reducing the computational complexity for the hierarchical streaming algorithm and testing the Gaussian kernel model method on a greater set of data is required. In the future research of multi-pitch streaming, finding new clustering algorithm or applying other existing algorithms to the area of multi-pitch streaming might be reasonable approach in order to have better streaming results.

REFERENCES

[1] S. Haykin and Z. Chen, “The Cocktail Party Problem,” Neural Computation, vol. 17, pp. 1875–1902, Sep 2005.

[2] Bregman, A.S. *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press, 1990.

[3] Bee, M.A. and Micheyl, C. “The Cocktail Party Problem: What Is It? How Can It Be Solved? And Why Should Animal Behaviorists Study It?” J. Comp. Psychol. 122, 235–251, 2008

[4] Carlyon, R.P. “How the Brain Separates Sounds,” Trends Cogn. Sci. 8, 465–471, 2004

[5] Duan, Z., Han, J., and Pardo, B. “Multi-Pitch Streaming of Harmonic Sound Mixtures,” IEEE Trans. Audio Speech Language Processing, 2013

[6] L. Rabiner, “On the Use of Autocorrelation Analysis for Pitch Detection,” IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-25, no. 1, pp. 24–33, Feb. 1977.

[7] A. De Cheveigné and H. Kawahara, “YIN, A Fundamental Frequency Estimator for Speech and Music,” J. Acoust. Soc. Amer., vol. 111, no. 4, pp. 1917–1930, 2002.

[8] M. R. Schroeder, “Period Histogram and Product Spectrum: New Methods for Fundamental-Frequency Measurement,” J. Acoust. Soc. Amer., vol. 43, no. 4, pp. 829–834, 1968.

[9] J. C. Brown, “Musical Fundamental Frequency Tracking Using a Pattern Recognition Method,” J. Acoust. Soc. Amer., vol. 92, no. 3, pp.1394–1402, 1992.

[10] B. Doval and X. Rodet, “Fundamental Frequency Estimation and Tracking Using Maximum Likelihood Harmonic Matching and HMMs,” in Proc. Int. Conf. Audio Speech Signal Process (ICASSP), vol. 1, pp. 221–224, 1993.

[11] G. Peeters, “Music Pitch Representation by Periodicity Measures Based on Combined Temporal and Spectral Representations,” in Proc. Int. Conf. Audio, Speech, Signal Process. (ICASSP), Toulouse, France, pp. 53–56, 2006.

[12] V. Emiya, B. David, and R. Badeau, “A Parametric Method for Pitch Estimation of Piano Tones,” in Proc. Int. Conf. Audio, Speech, Signal Process. (ICASSP), Honolulu, HI, pp. 249–252, 2007.

[13] D. G. Childers, D.P. Skinner and R.C. Kemerait, “The Cepstrum: a Guide to Processing,” Proc. IEEE, vol. 65, no. 10, pp. 1428-1443, 1977.

[14] R. Meddis and M. Hewitt, “Modeling The Identification of Concurrent Vowels with Different Fundamental Frequencies,” J. Acoust. Soc. Am., vol. 91, pp. 233–245, 1992.

[15] M. Davy,S. Godsill, and J. Idier,“Bayesian Analysis of Western Tonal Music,” Journal of the Acoustical Society of America, vol. 119, no. 4, pp. 2498–2517, 2006.

[16] M. Christensen and A. Jakobsson, “Multi-Pitch Estimation”, ser. Synthesis lectures on speech and audio processing, B. Juang, Ed. San Rafael, CA: Morgan & Claypool, 2009.

[17] P. Stoica, H. Li, J. Li, “Amplitude Estimation of Sinusoidal Signals: Survey, New Results and an Application”, IEEE Trans. Signal Processing 48(2), pp. 338–352, 2000.

[18] P.N. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, chapter 8, Addison Wesley, 2005

[19] Jain AK, Dubes RC. *Algorithms for Clustering Data*. Prentice-Hall Advanced Reference Series, 1−334, 1988.

[20] S. Kotsiantis and P.E. Pintelas, “Recent Advances in Clustering: a Brief Survey,” WSEAS, Transactions on Information Science and Applications 1, 73–81, 2004.

[21] T. Zhang, R. Ramakrishnan and M. Linvy, “BIRCH: an Efficient Data Clustering Method for Very Large Data Sets,” Data Mining and Knowledge Discovery, 1(2): 141–182, 1997

[22] S. Guha, R. Rastogi and K. Shim, “CURE: an Efficient Clustering Algorithm for Large Data Sets”, in Proc. ACM SIGMOD Conference, 1998.

[23] G. Karypis, E.H. Han and V. Kumar, “CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling,” Computer 32(8): 68-75, 1999.

[24] Szeto, W.M. and Wong, M.H, “A Stream Segregation Algorithm for Polyphonic Music Databases,” In Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS), 2003.

[25] M. Project. Music listing: J. S. Bach’s inventions. http://www.mutopiaproject.org/cgibin/maketable.cgi?Composer=BachJS, 2001.

[26] C. Fraley, A.E. Raftery, “Model-Based Clustering, Discriminant Analysis and Density Estimation,” J Am Stat Assoc, 97:611-631, 2002

[27] H. Kameoka, T. Nishimoto, and S. Sagayama, “Audio Stream Segregation of Multi-Pitch Music Signal Based on Time-Space Clustering Using Gaussian Kernel 2-Dimensional Model,” in Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’05), vol. 3, pp. 5–8, Philadelphia, Pa, USA, 2005.