

# Noisy Node Detection in Wave File by using Iterative Silhouette Clustering and Vivaldi

Jagtap V.G.

PICT Pune-411043, Maharashtra, India

Pande S.S.

Assistant Professor

PICT Pune-411043, Maharashtra, India

## ABSTRACT

There are various approaches for the noise detection in the audio files. Human ear can detect the sound intensity change of 34 milliseconds. In this paper propose novel method for noisy node detection in the wave file. The wave file is divided in chunks depending on Vivaldi concept. Iterative  $k$ -means clusters are created. The silhouette noisy node is detected, by comparing silhouette nodes in all iteration for each chunk.

## Keywords

Wave File, Silhouette cluster, Vivaldi.

## 1. INTRODUCTION

The wave file is a type of audio file having the different sampling rate of data. The digital form of the wave file contains data of samples of the range -1 to 1. The wave file data selection depends on the sampling rate [1].

There is more noise can be added in mono wave file transferring over the network. Instead of using mono we can use stereo wave file can be used. Stereo AECs are used to give the realistic performance for user along with "missing-fundamental" phenomenon [2].

If there is noise is added if audio file have multiple channel. There is no use of low pass filter as that leads to remove noise at particular channel also the same removes the data at other channel also for same time samples. To overcome this problem parameterized multichannel non-causal Wiener filter is used [3].

Another scenario is, for finding noise in audio file, low pass filter is used generally, that can remove low frequency but important nodes in wave file. The method is used to overcome these problem adaptive filters and LMS-based algorithms are used [4].

In all above cases, casual filters are used, which are time-domain filters. But performance of the system is degraded as they are casual, to overcome this problem, non casual time-domain filters are used. This help to maximize SNR and reduce the distortion in audio [5].

Another approach is there for minimizing noise. Adaptive signal enhancing, without noise enhancing is done then adaptive noise cancelling is done [6].

All above method uses the filters in it. That may lose the important small data. In our proposed method, data nodes are selected for applying above methods on it, to improve performance.

One method to find the noise is search data in file per frame. In this method, the tradeoff between spectral and temporal is found. This method helps to find noise before reduce it [7]. The problem in this method is performance degradation as it checks each frame in the file.

There are different kinds of general clustering methods available like connectivity based, centroid based, distribution based, density based clustering. We preferred centroid based clustering which uses  $k$ -means clustering. To find correctness of node in the cluster, silhouette method is used [8].

Vivaldi is a simple, light-weight algorithm that assigns synthetic coordinates to hosts such that the distance between the coordinates of two hosts accurately predicts the communication latency between the hosts [9].

We proposed an efficient method to find noisy node with help of silhouette clustering. The method involves the creation of chunks based on Vivaldi concept. In silhouette clustering firstly, the  $k$ -means cluster is found in each chunk and silhouette gives how correctly the node is in that cluster as compare to other clusters.

From the experimental result we find the noisy node in each chunk. By setting the ranking in nodes we can apply the noise removal method on particular most noisy node to give better performance of the system.

## 2. PROCEDURE FOR NOISY NODE DETECTION

### 2.1 Chunk creation stage

Firstly the chunks are created using Vivaldi concept. The data of 34 milliseconds is selected with help of sample rate.

### 2.2 Noisy node detection state

After creating chunks, the  $k$ -means clusters are created in each chunks and silhouette clustering method is applied on each  $k$ -means cluster. Two to  $\sqrt{(n/2)}$  number of iteration are performed and the  $k$ -means clusters and silhouette data is calculated. By comparing all silhouette data, silhouette noisy node is detected.

### 2.3 Silhouette clustering method

Firstly create cluster with slandered clustering method like  $k$ -means, having  $k$  clusters. For each datum  $i$ , let  $p(i)$  be the average dissimilarity of  $i$  with all other data within the same cluster. Dissimilarity is measured by means of distance to centroid of cluster and node. It shows that why  $p(i)$  is dissimilar after creating cluster with slandered cluster creation. That shows that the node is not fit for this cluster rather than other clusters. Denote the lowest average dissimilarity to  $i$  of any such cluster by  $q(i)$ . The cluster with this lowest average dissimilarity is said to be the "neighboring cluster" of  $i$  as it is, aside from the cluster  $i$  is assigned, the cluster in which  $i$  fits best.

$$s(i) = \frac{q(i)-p(i)}{\max\{p(i),q(i)\}} \dots \dots \dots (1)$$

From the above definition it is clear that

$$1 \leq s(i) \leq 1 \dots \dots \dots (2)$$

For  $s(i)$  to be close to 1,  $p(i) \ll q(i)$ .

This method helps to find the dissimilar data set as it finds dissimilarity with other clusters data also [10].

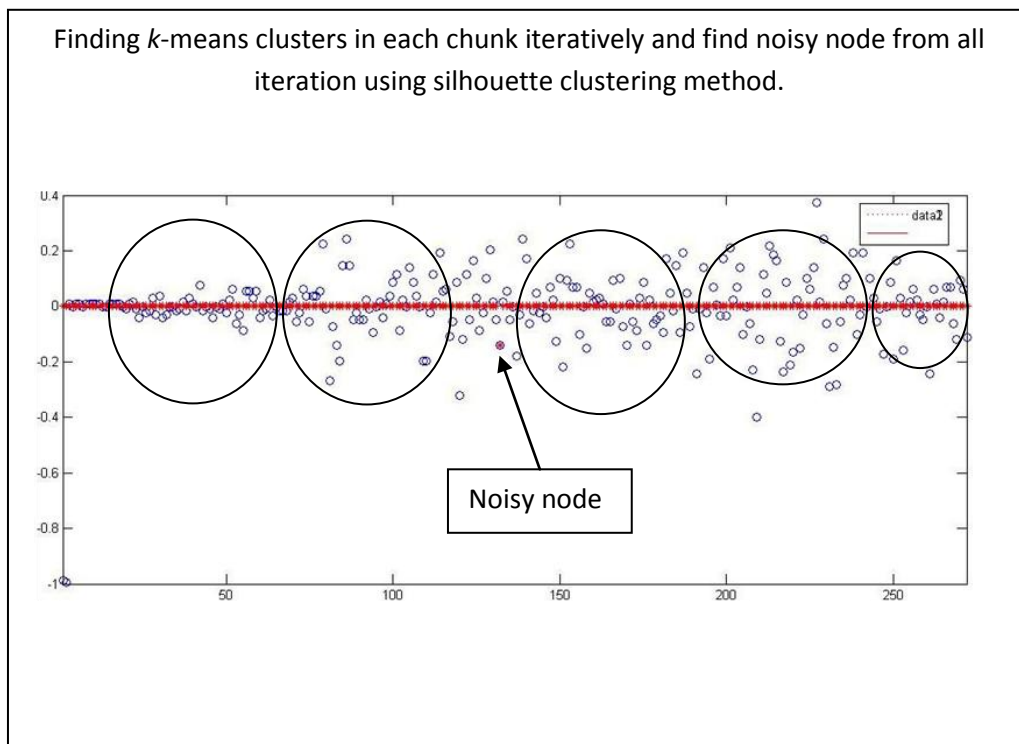
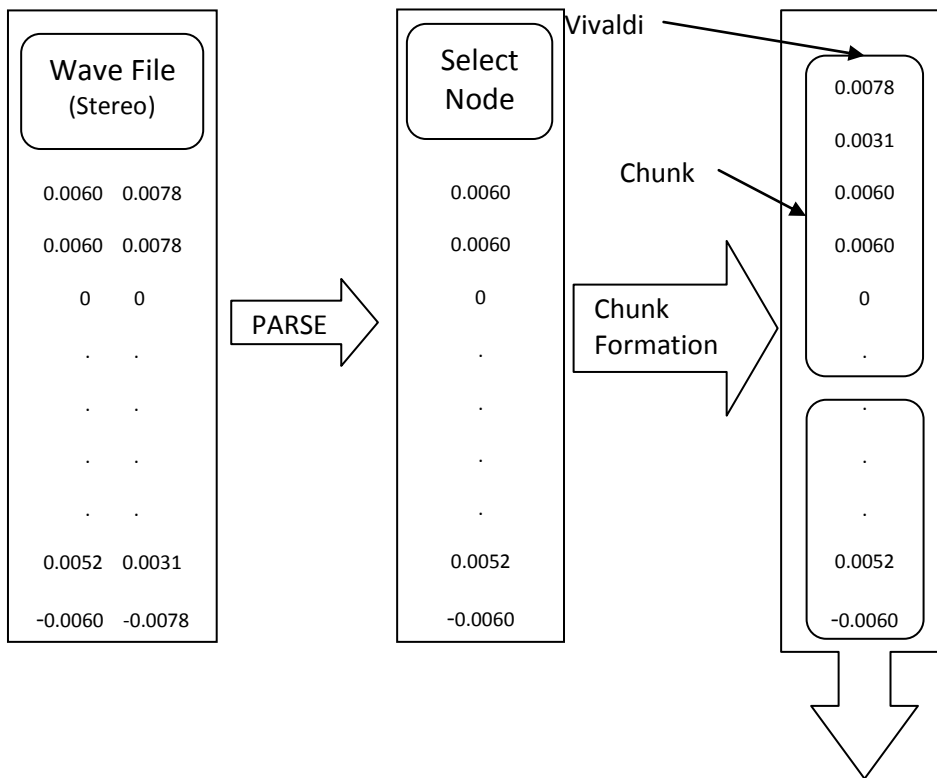


Fig 1: System overview

### 3. STEPS FOR THE NOISY NODE DETECTION

Let us consider a wave file of  $fs$  sampling rate and as we already know human ear can detect the intensity change of the 34 milliseconds. We can take the node selection less than 34 milliseconds, here we taken as 30. By using Vivaldi concept and  $fs$  we can calculate the no. of nodes in each chunk represented as  $n$  is given by,

$$n = fs \times \text{selection time} \dots\dots\dots (3)$$

Finding the  $k$ -means cluster in each chunks and then finding the silhouette for each cluster the steps used are as follow,

Step:1 Find the nodes in chunks for file by using

$$n = fs \times \text{selection time}$$

Step:2 For  $i=2:\sqrt{(n/2)}$ ;

$$(idx, c) = kmeans(n, i)\dots\dots\dots (4)$$

$$s(i)(n) = silhouette(n, c)\dots\dots (5)$$

Step:3 Comparing  $s(i)(n)$  the lowest value closer to -1 gives that node is distorted for that cluster.

### 4. EXPERIMENTAL SETUP

For this setup we use the Windows 7 home basic, Matlab R2011a. The out.wav file of size 113 KB is used for this experiment. For conducting this experiment, wave files of 16 bit stereo having sampling rate 8000, 22.1k, 44.1k FSB were considered. The data set consist of 50 wave files, having out.wav as one of them.

### 5. RESULT

#### 5.1 File

Fig. 2 shows that, the experiment performed on out.wav gives that the distorted node in file.

Out.wav is slandered 16bit mono wave file. The experiment performed on out.wav file is having 8000 sampling rate ( $fs$ ) the chunk size is 270 nodes. From all nodes, we find a node in chunk. The noisy node in file is shown by red dark dot where as other nodes are represented as blue.

For the data nodes in the wave file we firstly divide the file in chunks of equal length. The chunk size depends on human ear intensity change detection capacity. For more generous result we used  $k$ -means clustering but we can use fuzzy c-means clustering method.

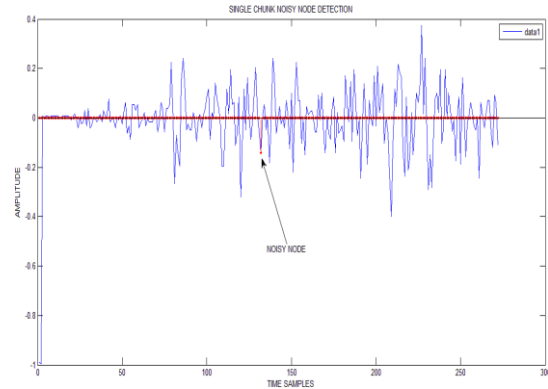


Fig. 3 noisy node in out.wav file

From all chunks and the noisy nodes, that we find we can calculate the ranking between that nodes as we know the distance between noisy node and its centroid. As the larger is the distance that means the node is having large noise in that file.

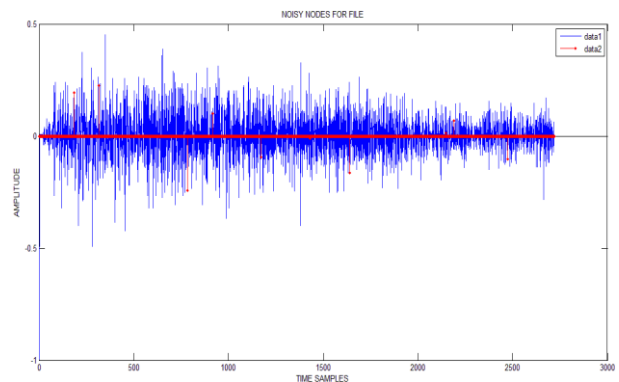


Fig 2: noisy node in out.wav file

#### 5.2 Single chunk

Fig. 3 shows that experimental result for a chunk in the file.

Table 1 show that data of some iteration on each chunk are shown below. For each chunk there are multiple iterations are taken but here we show six of them, from that we find the noisiest node in each chunk. Then, ranking has to be formed. For that distance is calculated. Then ranking is decided.

Chunk name	Iteration no for a chunk	1	2	3	4	5	6	Selection of node	Noisy node is
sample 1	node	129	214	54	36	161	189	129	
	distance	0.0941	0.0728	0.063	0.0823	0.0568	0.0449		
sample 2	node	102	132	102	239	168	250	132	132
	distance	0.1164	0.1198	0.0806	0.0795	0.0675	0.0557		
sample 3	node	4	52	55	13	96	111	4	
	distance	0.1123	0.0858	0.0658	0.0664	0.0561	0.0473		
sample 4	node	90	47	99	265	66	22	1	
	distance	0.0796	0.0682	0.0572	0.0481	0.047	0.0339		
sample 5	node	156	50	29	271	9	271	156	
	distance	0.0823	0.0533	0.0505	0.052	0.0545	0.052		

**Table 1 Ranking of noisy node in file out.wav**

(Where distance = Euclidian distance of noisy silhouette node from its centroid and node = Index of noisy silhouette node.)

## 6. CONCLUSION AND FUTURE SCOPE

For the compression and encryption there are different kind of types for wave data selection like 2<sup>nd</sup> bit is considered in generally. From the above approach we can select the random selection of the node with less noise occurrence after conducting experiment. This method can reduce the noise as it selects the mechanism for finding the noisy node. For further, this method can be used to reduce noise in steganography.

## 7. REFERENCES

- [1] Tag Vaughan, Multimedia – Making it work, 5th edition, TMGH, 2004.
- [2] Stefania Cecchi, Laura Romoli, Paolo Peretti, Francesco Piazza, "A Combined Psychoacoustic Approach for Stereo Acoustic Echo Cancellation", IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 19, NO. 6, AUGUST 2011.
- [3] Mehrez Souden, Jacob Benesty, Sofiene Affes, "On Optimal Frequency-Domain Multichannel Linear Filtering for Noise Reduction", IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 18, NO. 2, FEBRUARY 2010.
- [4] Martin Bouchard, Yu Feng, "Inverse Structure for Active Noise Control and Combined Active Noise Control/Sound Reproduction Systems", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 9, NO. 2, FEBRUARY 2001.
- [5] Jesper Rindom Jensen, Jacob Benesty, Mads Græsbøll Christensen, Søren Holdt Jensen "Non-Causal Time-Domain Filters for Single-Channel Noise Reduction", IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 20, NO. 5, JULY 2012.
- [6] Tetsuya Hoya, Toshihisa Tanaka, Andrzej Cichocki, Takahiro Murakami, Gen Hori, Jonathon A. Chambers, "Stereophonic Noise Reduction Using a Combined Sliding Subspace Projection and Adaptive Signal Enhancement", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 13, NO. 3, MAY 2005
- [7] Ismo Kauppinen, Kari Roth, "Improved Noise Reduction in Audio Signals Using Spectral Resolution Enhancement With Time-Domain Signal Extrapolation", IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 13, NO. 6, NOVEMBER 2005
- [8] R. Sibson (1973). "SLINK: an optimally efficient algorithm for the single-link cluster method". The Computer Journal (British Computer Society) 16 (1): 30–34.
- [9] COX, R., DABEK, F., KAASHOEK, F., LI, J., AND MORRIS, R. Practical, distributed network coordinates. In Proc. of the Second workshop on Hot Topics in Networks (HotNets-II) (Nov. 2003).
- [10] Peter J. Rousseeuw (1987). "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". Computational and Applied Mathematics 20: 53–65. doi:10.1016/0377-0427(87)90125-7