

Apparatus and Method For Detecting and Characterizing Signals in a Communication System

5

Background

1. Field of the Invention

The invention relates generally to communication systems, and more particularly to detecting and characterizing signals in a
10 communication system.

2. Discussion of Related Art

In today's information age, the number of personal computers used in homes, schools, and businesses continues to proliferate with
15 apparently no end in sight. This increasing use of personal computers has prompted the migration of many applications onto the personal computer. For example, in addition to providing standard computational and networking functionality, the personal computers of today often include such functionality as a modem for exchanging
20 data with other computers, a telephone (including speakerphone), a telephone answering system, a facsimile system, and teleconferencing/videoconferencing system. Thus, the personal computer can take the place of a multitude of otherwise separate devices, often saving cost, simplifying use, and providing additional
25 features as compared to the separate devices.

Whether used as separate devices or together in the personal computer, these communications applications typically have a number of common elements. Specifically, a processor is used for controlling the device, memory is used for storing information, a
30 signal processor is used for generating and processing the electrical

signals needed for communication, and interface components are used for interfacing with the communication system and for providing additional signal processing capabilities. When these communication applications are included in the personal computer, it is often
5 convenient to integrate two or more of the applications together so that the common elements do not have to be duplicated. This integration of applications further reduces the cost of providing such communication applications.

With the cost of personal computers falling and the competition
10 among vendors growing, computer manufacturers and third-party vendors are looking for a cost-effective way of providing the many communication applications. One solution is to implement predominantly all of the application functions in software (with the remaining functions implemented in specialized hardware) and to run
15 the software as a software application on the microprocessor in the personal computer. Implementing the often complex signal processing functions in software is feasible today due to the amount of processing resources provided by modern microprocessors. By eliminating most of the dedicated hardware components and utilizing
20 the processing and memory resources of the personal computer, the communication applications can be provided relatively inexpensively.

One issue with such an integrated software implementation is that the communication application software must share the processing resources of the personal computer with other application
25 software such as a word processor, spreadsheet program, or Internet browser. Thus, the software implementation consumes processing resources that otherwise would be available to the other application software. As a result, the performance of the other application software may be adversely affected when the communication

applications are running. Thus, it is important to implement the communication applications such that they use as little processing resources as possible, and also to distribute the processing demand so that the communication application software does not control the processing resources for an excessive amount of time.

One type of signal processing function that is utilized in many of the communication applications is the detection of, and distinction between, voice, tone, and noise signals. Uses include voice-activated automatic gain control (AGC) for teleconferencing and videoconferencing; voice detection for the telephone answering system; double-talk detection in the speakerphone application; DTMF tone detection for accessing special services such as retrieving messages from the telephone answering system, accessing voice mailboxes, and for other keypad-controlled services; and detection of special modem and facsimile tones such as dial tone, answer-back tone, call progress tones, and busy tone. These signal processing functions have typically been implemented separately. When running concurrently, these signal processing functions consume a significant amount of processing resources. Therefore, a need remains for an apparatus and method for providing efficient voice, tone, and noise detection which reduces the amount of processing resources required and also distributes the processing demand.

Brief Description of the Drawing

In the Drawing,

FIG. 1 is a high-level logic flow diagram of a detector;

FIG. 2 is a high-level logic flow diagram showing exemplary update interval logic;

FIG. 3 is a high-level logic flow diagram showing exemplary decision interval logic;

FIG. 4 is a high-level logic flow diagram showing exemplary hypothesis logic;

5 FIG. 5 shows a double buffer system used in an embodiment of the present invention; and

FIG. 6 shows two samples n and $n-K$ stored in the double buffer system.

10 ***Detailed Description***

As discussed above, the need remains for an apparatus and method for providing efficient voice, tone, and noise detection which reduces the amount of processing resources consumed and also distributes the processing demand over time. The present invention
15 provides for such efficient voice, tone, and noise detection by applying the Average Magnitude Difference Function (AMDF) over discrete time intervals to evaluate variations in pitch over time, allowing a hypothesis to be made as to whether a signal is a voice, tone, or noise signal.

20 AMDF is a well-known technique for pitch estimation which is described in M.J. Ross, H.L. Shaffer, A. Cohen, R. Freudberg, and H.J. Manley, "Average Magnitude Difference Function Pitch Extractor," IEEE Trans. Acoust., Speech and Signal Proc., Vol. ASSP-22, pp. 353-362, October 1974, incorporated herein by reference in its entirety.
25 Briefly, the fundamental concept of the AMDF technique is that, for a truly periodic signal, the difference between two signal samples $x(n)$ and $x(n-K)$ will be zero if K is equal to the pitch period. Because periodic signals may vary slightly due to noise, the difference between two signal samples $x(n)$ and $x(n-K)$ may not be zero but will

likely be close to zero at the pitch period K . Thus, the pitch of a signal can be estimated by finding the value K where the difference between the two signal samples $x(n)$ and $x(n-K)$ approaches zero.

The present invention applies the AMDF technique, not for estimating a pitch period K , but rather for evaluating variations in pitch over discrete sample periods to determine whether a signal is a voice signal, a tone signal, or a noise signal. The techniques of the present invention are based on the premise that a tone signal will maintain a relatively constant energy level at its fundamental pitch, a voice signal will have a varying energy level at its fundamental pitch, and a noise signal will have no distinguishable fundamental pitch. Thus, the received signal is analyzed over a predetermined range of pitch periods K , and a set of metrics are computed which characterize the signal as to pitch and variation in pitch. In the preferred embodiment, K is in the range 50 to 140, inclusive, which corresponds roughly to the range of human speech. The novel metrics allow a hypothesis to be made as to whether the signal consists of voice, tone, or noise.

One particular advantage of the preferred embodiments is that the signal analysis is done in the time domain rather than in the frequency domain. The frequency domain approach typically utilizes the Fast Fourier Transform (FFT), which is computationally intensive due to the number of multiplication operations required. The time domain approach of the present invention, on the other hand, utilizes predominantly addition and subtraction operations, and therefore the computational complexity is substantially reduced.

In a preferred embodiment, a detector implemented in software is used to evaluate the signal and to decide whether the signal consists of voice, tone, or noise. In a preferred embodiment, the

detector is invoked at 2 millisecond intervals and produces a decision every thirteenth interval based on calculations made during the previous 12 intervals as to whether a voice, tone, or noise signal was present. For convenience, the 13 intervals over which the decision is made is referred to as a "detection cycle," the first 12 intervals of the detection cycle are referred to as "update intervals," and the thirteenth interval of the detection cycle is referred to as the "decision interval." The interval duration as well as the number of intervals per detection cycle are preferred values that have been shown to work well during testing.

A high-level logic flow diagram of the detector is shown in FIG.

1. When the detector logic is invoked for an interval "m" during a detection cycle "i" in step 102, a determination is made in step 104 whether the detector is within the first 12 update intervals of the detection cycle (m less than or equal to 12) or is in the decision interval of the detection cycle (m equal to 13). If the detector is within the first 12 update intervals of the detection cycle, then the logic proceeds to execute the update interval logic in step 106, and then terminates processing for the interval in step 199. If the detector is in the decision interval of the detection cycle, then the logic proceeds to execute the decision interval logic in step 108, and then terminates processing for the interval in step 199.

When the detector is running, signal processing hardware continually samples and buffers the received signal. The input samples are sampled directly from the line (i.e., not AGC adjusted) and are signed 16-bit integers in the range +/- 32,767. In the preferred embodiment, a double buffer system as shown in FIG. 5 is employed for storing the input samples. The two buffers are contiguous, and each stores X input samples ($X > 140$). The two

buffers are initially filled with zeros. Each input sample S_n is stored at an equivalent slot in each buffer, so that the stored samples are X slots apart. Each buffer is treated as a circular buffer in that each slot is overwritten with a new sample every X samples.

5 During each update interval m , the update interval logic operates on the buffer of input samples. In the preferred embodiment, the interval m is 2 milliseconds and the sampling rate is 8 KHz, and therefore the update interval logic operates on 16 input samples per update interval m . The detector calculates a local AMDF

10 value over the interval m for each of the pitch periods K . The local AMDF value $AMDF_{16_m}(K)$ for each pitch period K is equal to:

$$15 \quad AMDF_{16_m}(K) = \sum_{n=1}^{16} |x(n) - x(n-K)|$$

where $x(n)$ is sample n from the buffer and $x(n-K)$ is a prior sample which precedes sample n by K samples. As shown in FIG. 6, the double buffer system (described above) stores a sufficient number of prior

20 samples so that $AMDF_{16_m}(K)$ can be calculated for all values of K .

For each value K , the detector maintains a global AMDF value $AMDF(K)$ which is a running sum of the local AMDF values over the 12 update intervals:

$$25 \quad AMDF(K) = AMDF(K) + AMDF_{16_m}(K)$$

30

The detector also determines the minimum local AMDF value MinAMDF16_m over all of the pitch periods K for the interval m :

$$\text{MinAMDF16}_m = \min [\text{AMDF16}_m(K)]$$

It is interesting to note that the value of K at which $\text{AMDF16}_m(K)$ is minimum represents the estimated pitch over the interval m for the prior art AMDF pitch estimation technique, although the particular value of K is irrelevant to the present invention.

Finally, the detector maintains an average difference of the minimum AMDF values AvgDiffAMDF which is a running sum of the differences between the minimum local AMDF value for the interval m and the minimum local AMDF value for the previous interval ($m-1$):

$$\text{AvgDiffAMDF} = \text{AvgDiffAMDF} + | \text{MinAMDF16}_m - \text{MinAMDF16}_{m-1} |$$

When computing AvgDiffAMDF for the first update interval in a detection cycle, the minimum local AMDF value from the last update interval of the previous detection cycle ($i-1$) is carried over and used as the value for MinAMDF16_{m-1} .

A high-level logic flow diagram showing exemplary update interval logic is shown in FIG. 2. When the logic is invoked in step 202, the logic updates the global AMDF value $\text{AMDF}(K)$ for each value K and the AvgDiffAMDF which are the running sums carried over from interval to interval. Thus, for each pitch period K beginning with pitch period K equal to 50 in step 204, the logic executes a loop which includes computing the local AMDF value $\text{AMDF16}_m(K)$ in step 206, updating the global AMDF value $\text{AMDF}(K)$ in step 208, checking

whether the local AMDF value $AMDF_{16_m}(K)$ is less than the current minimum local AMDF value $MinAMDF_{16_m}$ in step 212, and saving $AMDF_{16_m}(K)$ as the $MinAMDF_{16_m}$ in step 212 if $AMDF_{16_m}(K)$ is less than $MinAMDF_{16_m}$. The logic then increments K in step 214 and loops
 5 back to step 206 to execute the loop for the next value K if K is less than or equal to 140 (YES in step 216). When the execution loop has been completed for all pitch periods K (NO in step 216), the logic proceeds to update the running sum $AvgDiffAMDF$ in step 218. The interval m is then incremented for the next interval in step 220, and
 10 the update interval logic terminates in step 299.

When the detector logic is within the decision interval, the detector logic executes the decision interval logic. In the preferred embodiment, no processing is done on the 16 input samples for the decision interval. The decision interval logic uses the metrics
 15 computed during the update intervals, among other things, to form a hypothesis as to whether a voice, tone, or noise signal was present during the detection cycle i . After the 12 update intervals, the global AMDF for each value K is effectively equal to:

20

$$AMDF(K) = \sum_{m=1}^{12} AMDF_{16_m}(K)$$

The detector first finds the minimum of the global AMDF values
 25 $AMDF_{min}$ over all of the pitch periods K :

$$AMDF_{min} = \min [AMDF(K)]$$

30

The detector then computes a sum of the global AMDF values $AMDF_{sum}$ over all of the pitch periods K :

$$5 \quad AMDF_{sum} = \sum_{K=50}^{140} AMDF(K)$$

The detector computes a first metric $AMDF_{norm}$ which effectively compares the minimum of the AMDF over the pitch range to the average AMDF over the pitch range:

$$AMDF_{norm} = AMDF_{min} / AMDF_{sum}$$

The detector computes a second metric $AvgDiffAMDF_{norm}$ which measures the average variation of the minimum AMDF over the update intervals:

$$AvgDiffAMDF_{norm} = AvgDiffAMDF / AMDF_{sum}$$

It is important to note that by using the sum of the global AMDF values $AMDF_{sum}$ as the divisor rather than calculating an average of the global AMDF values, processing resources are conserved. It is also important to note that $AMDF_{norm}$ and $AvgDiffAMDF_{norm}$ are only computed if $AMDF_{sum}$ is non-zero in order to avoid a divide-by-zero error.

After computing the two metrics $AMDF_{norm}$ and $AvgDiffAMDF_{norm}$, the detector performs its hypothesis logic in order to decide whether a voice, tone, or noise signal was present during the detection cycle. The general principle applied by the hypothesis logic (although not the preferred embodiment, which is described in more detail below) is that a large value of $AMDF_{norm}$ is typical of a noise signal while a

small value of $AMDF_{norm}$ is typical of a non-noise (i.e., voice or tone) signal, although $AMDF_{norm}$ alone is insufficient to determine whether the non-noise signal is a voice signal or a tone signal. Therefore, if $AMDF_{norm}$ is small, $AvgDiffAMDF_{norm}$ is used to determine whether the non-noise signal is a voice signal or a tone signal. A large value of $AvgDiffAMDF_{norm}$ is typical of a voice signal while a small value of $AvgDiffAMDF_{norm}$ is typical of a tone signal.

A high-level logic flow diagram showing exemplary decision interval logic is shown in FIG. 3. When the logic is invoked in step 302, the logic proceeds to find $AMDF_{min}$ in step 304, and then computes $AMDF_{sum}$ in step 306. The logic then computes the $AMDF_{norm}$ metric in step 308 and the $AvgDiffAMDF_{norm}$ metric in step 310. Once the two metrics are computed, the logic executes the hypothesis logic in step 312 to determine whether a voice, tone, or noise signal was present during the detection cycle i . The interval m is then set back to one for the next detection cycle in step 314, and the decision interval logic terminates in step 399.

In practice, it has been found that the general hypothesis logic as described above can result in inaccurate decisions under certain circumstances. Specifically, because the two metrics represent averages over time, instantaneous changes from one type of signal to another may not be instantaneously reflected in the metrics. Thus, the hypothesis logic uses the metrics in combination with historic data (i.e., data from previous detection cycles) and appropriate threshold values to make its decision.

The hypothesis logic applies a set of rules which are based on observed characteristics of signals. A first observed characteristic is that once a noise or tone signal is detected, the metrics are likely to settle within particular ranges if the signal remains a noise or

tone signal, and therefore the criteria for detecting subsequent noise or tone signals can be made less stringent. A second observed characteristic is that, when transitioning from noise to tone, the AvgDiffAMDF_{norm} spikes to a high value and slowly decays back down toward levels more indicative of a tone. Therefore, to increase the speed of tone detection following a transition from noise, the tone detection threshold is raised after such a spike is detected. A third observed characteristic is that, when transitioning from tone to noise, the two metrics are slow to move to their respective noise levels and are consequently misinterpreted as voice. Therefore, the hypothesis logic is prevented from characterizing the signal as voice for two detection intervals following the end of a tone.

A high-level logic flow diagram showing exemplary hypothesis logic is shown in FIG. 4. When the logic is invoked in step 402, the logic proceeds to determine if the signal is a noise signal in step 404. In step 404, the signal is characterized as noise, and the logic proceeds to step 410, if any of a number of conditions is true. First, the signal is characterized as noise if the AMDF_{sum} is equal to zero. This case represents the detection of absolute silence. Second, the signal is characterized as noise if the AMDF_{norm} for the current detection cycle i is greater than a threshold N , representing a large value of AMDF_{norm}. Finally, the signal is characterized as noise if the signal detected in the previous detection cycle ($i-1$) was noise and the AMDF_{norm} is greater than a threshold N_{2N} which is less stringent than N . This condition applies the rule from the first observed characteristic described above, specifically that the threshold for detecting subsequent noise signals can be made less stringent.

If the signal is not characterized as noise in step 404, then the logic proceeds to determine if the signal is a tone signal in step 406.

In step 406, the signal is characterized as tone, and the logic proceeds to step 414, if any of a number of conditions is true. First, the signal is characterized as tone if the $\text{AvgDiffAMDF}_{\text{norm}}$ for the current detection cycle i is less than a threshold T . Threshold T is a relatively stringent threshold for initially detecting a tone signal. Second, the signal is characterized as tone if the signal detected in the previous detection cycle $(i-1)$ was tone and the $\text{AvgDiffAMDF}_{\text{norm}}$ for the current detection cycle i is less than a threshold $T2T$. This condition applies the rule from the first observed characteristic described above, specifically that the threshold for detecting subsequent tone signals can be made less stringent. Finally, the signal is characterized as tone if the signal detected in the previous detection cycle $(i-1)$ was noise and the $\text{AvgDiffAMDF}_{\text{norm}}$ for the previous detection cycle $(i-1)$ is greater than a threshold $H1$ (i.e., the spike referred to above) and the $\text{AvgDiffAMDF}_{\text{norm}}$ for the current detection cycle i is less than a threshold $N2T$. This condition applies the rule from the second observed characteristic described above.

If the signal is not characterized as tone in step 406, then the logic proceeds to step 408 to apply the rule from the third observed characteristic described above, specifically to prevent the hypothesis logic from characterizing the signal as voice for two detection intervals following the end of a tone. In step 408, the signal is characterized as noise, and the logic proceeds to step 410, if the signal detected in either of the previous two detection cycles $(i-1)$ and $(i-2)$ was tone; otherwise, the signal is characterized as voice, and the logic proceeds to step 412.

As discussed above, the metrics are average values, although the metrics are computed without normalizing over the number of elements over which the average is taken. Instead, the threshold

values are scaled appropriately to account for the number of elements over which the metrics were averaged. This scaling technique reduces the computational complexity of computing the metrics by avoiding division operations, thereby reducing the processing resources consumed by the detector.

Thresholds N and $N2N$ apply to $AMDF_{norm}$, which is averaged over the range K only. Therefore, thresholds N and $N2N$ are divided by the number of elements in the average. In the preferred embodiment, threshold N is equal to $0.65/90$ and threshold $N2N$ is equal to $0.5/90$.

Thresholds T , $T2T$, $N2T$, and HI apply to $AvgDiffAMDF_{norm}$, which is averaged over the range K as well as over the 12 intervals.

Therefore, thresholds T , $T2T$, $N2T$, and HI are multiplied by the number of intervals 12 and divided by the number of elements in the average. In the preferred embodiment, threshold T is equal to

$0.0015 \cdot 12/90$, threshold $T2T$ is equal to $0.003 \cdot 12/90$, threshold $N2T$ is equal to $0.009 \cdot 12/90$, and threshold HI is equal to $0.015 \cdot 12/90$.

It is worth noting that the threshold values are described above as though the metrics are averaged over 90 elements. In reality, the metrics are averaged over 91 elements (50 to 140, inclusive). This factoring error does not affect the outcome of the hypothesis logic, since it is the absolute values of the thresholds that determines the outcomes. The absolute threshold values were obtained through experimentation and are based on actual observations of signal characteristics.

While the preferred embodiment distributes the processing demand for each detection cycle over 13 intervals, it will be apparent to a skilled artisan that the input samples for each of the update intervals may be stored and that all calculations may be deferred until the decision interval. It will also be apparent to a skilled

artisan that some or all of the intermediate calculations made during each update interval may be deferred until the decision interval.

It will also be apparent to a skilled artisan that the detection cycle can be shortened to 12 intervals, with the decision interval logic for a detection cycle i computed during the first interval of the subsequent detection cycle $(i+1)$.

It will also be apparent to a skilled artisan how the update interval logic and the decision interval logic can be changed for different interval durations, sampling rates, and pitch frequency ranges.

□□ The present invention may be embodied in other specific forms without departing from the essence or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive.

What is claimed is:

1. A method for characterizing a signal over a detection cycle i , the detection cycle i having a number of intervals, each interval
5 having a predetermined number of input samples, the method comprising the steps of:

determining an Average Magnitude Difference Function (AMDF) value for each of a predetermined range of pitch frequencies K over the intervals;

- 10 determining an average difference AMDF value over the intervals equal to the sum of the difference between a first minimum AMDF value from each interval m and a second minimum AMDF value from each interval $(m-1)$;

determining a minimum AMDF value over the intervals;

- 15 determining a sum of the AMDF values over the intervals;

computing a first metric equal to the minimum AMDF value over the intervals divided by the sum of the AMDF values over the intervals;

- 20 computing a second metric equal to the average difference AMDF value over the intervals divided by the sum of the AMDF values over the intervals; and

utilizing said first metric and said second metric to determine whether the signal is one of a noise signal, a tone signal, and a voice signal.



2. A device for characterizing a signal over a detection cycle i , the detection cycle i having a number of intervals, each interval having a predetermined number of input samples, the device comprising:

5 logic for determining an Average Magnitude Difference Function (AMDF) value for each of a predetermined range of pitch frequencies K over the intervals;

10 logic for determining an average difference AMDF value over the intervals equal to the sum of the difference between a first minimum AMDF value from each interval m and a second minimum AMDF value from each interval $(m-1)$;

 logic for determining a minimum AMDF value over the intervals;

 logic for determining a sum of the AMDF values over the intervals;

15 logic for computing a first metric equal to the minimum AMDF value over the intervals divided by the sum of the AMDF values over the intervals;

 logic for computing a second metric equal to the average difference AMDF value over the intervals divided by the sum of the

20 AMDF values over the intervals; and

 logic for utilizing said first metric and said second metric to determine whether the signal is one of a noise signal, a tone signal, and a voice signal.



3. An apparatus comprising a computer usable medium having computer readable program code means embodied therein for characterizing a signal over a detection cycle i , the detection cycle i having a number of intervals, each interval having a predetermined number of input samples, the computer readable program code means comprising:

computer readable program code means for determining an Average Magnitude Difference Function (AMDF) value for each of a predetermined range of pitch frequencies K over the intervals;

computer readable program code means for determining an average difference AMDF value over the intervals equal to the sum of the difference between a first minimum AMDF value from each interval m and a second minimum AMDF value from each interval ($m-1$);

computer readable program code means for determining a minimum AMDF value over the intervals;

computer readable program code means for determining a sum of the AMDF values over the intervals;

computer readable program code means for computing a first metric equal to the minimum AMDF value over the intervals divided by the sum of the AMDF values over the intervals;

computer readable program code means for computing a second metric equal to the average difference AMDF value over the intervals divided by the sum of the AMDF values over the intervals; and

computer readable program code means for utilizing said first metric and said second metric to determine whether the signal is one of a noise signal, a tone signal, and a voice signal.

