Things That Talk

by

Vadim Gerasimov

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences
at the
Massachusetts Institute of Technology
June 1996

Signature of Author ____
Program in Media Arts and Sciences
February 2, 1996

Certified by ____
Walter Bender
Associate Director for Information Technology
MIT Media Laboratory
Thesis Supervisor

Accepted by ____
Stephen A. Benton
Chairperson
Departmental Committee on Graduate Students

Things That Talk

by

Vadim Gerasimov

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on February 5, 1996
in partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

The main purpose of this research is to explore features of sound as a computer-to-computer communication medium. There are already several well-developed technologies of wireless interdevice communication such as infra-red and radio, but sound can be a good alternative. This research has to answer the question of suitability of sound for computer communication. As devices are getting smarter, it is becoming clear that they must have an ability to share their knowledge without human help. The problem of wireless communication will appear in most of the Things That Think projects. Sound may become one of the best means of interdevice communication.

Thesis supervisor: Walter Bender

Title: Associate Director for Information Technology, MIT Media Laboratory

This work was supported in part by the News in the Future consortium

Things That Talk

by

Vadim Gerasimov

The following people served as readers for this thesis:

Reader: ______________________________

Mitchel Resnick
Assistant Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Reader: ______________________________

Brian Silverman
Director of Research
Logo Computer Systems, Inc.

Table of Contents

Overview

Each device knows something that it can share with others. Almost all modern electronic devices have internal clocks, but there is no way to automatically transfer time/date settings from one device to another. Many devices use audio or video signals to tell something to people. But those signals can carry information important to other devices. A phone ring can tell your computer who is calling. Or a wrist-watch signal can turn off a stove. Remote controls are a simple example of a talking device. But it is only one way communication, and one of the devices is passive. Wouldn't it be nice if a TV set would transfer its channel tuning to a VCR on your command? My research is focused on usage of sound for computer-to-computer communication. The two main areas of research in acoustics related to computers are speech synthesis and recognition. But both those tasks are far from ultimate solutions, and require too much computational resources to be used in small devices in the near future.

Basic Facts about Sound

Nature of Sound

Sound is longitudinal waves that consist of alternating compressions and rarefactions in a medium.

- Speed in air at 0°C (increases with temperature and pressure) 331 m/s
- Speed in water (increases with temperature, salinity, depth) 1,400-1,600 m/s
- Speed in lead 1,210 m/s
- Speed in rolled aluminium 5,000 m/s

Maximum possible frequency of sound in any medium is about $1.25 \cdot 10^{13} Hz$.

Sound is very slow comparing to any kind of electromagnetic waves. Therefore, it has much shorter wavelength at the same frequencies. For example, the wavelength of 80 kHz sound in air is only about 4 mm. A wavelength corresponds to the minimal size of objects which can reflect the wave. Sound has a set of propagation features quite different from that of light. That makes it quite useful as a substitution of light vision in situations where light does not work well. Slowness of sound makes the Doppler effect strong enough for motion detection of relatively slow objects around us.

Many animals use high-frequency sound to detect the position and speed of objects in their environment. That method of orientation is usually called echolocation. It is especially useful for animals like bats, which live in dark caves and need to hunt for insects at night, and whales, which spend their lives in turbid water.

Sound frequencies used for echolocation by animals.

Whales 50,000 - 200,000 Hz in water (7-32 mm).

Bats 30,000 - 80,000 Hz in air (4 -11 mm).

People have not overlooked those good properties of sound. In medicine, low-power ultrasonic waves (300kHz) are used in place of X rays to make an image of internal bodily structures. Focused ultrasound can cause localized heating and destroy bacteria or tissue without harmful effect on nearby tissue. Ultrasonic (100-16,000MHz) microscopy gives a better resolution than optical microscopy, and allows people to investigate a wider range of material properties in amazing detail. An important advantage of ultrasonic microscopes is their ability to distinguish parts of live cells by their viscosity. Also, because sound practically does not affect life function of cells, ultrasonic microscopy can study living cells. Ultrasonic sonars are widely used for underwater exploration.

Unfortunately, sound in computer technology has been limited to music, simple sound effects, speech synthesis and recognition. But, it may successfully be used for interdevice communication, and even might be a good alternative for optical computer vision. Why do computers feebly try to recognize people by pictures of their faces taken in visible light or fingerprints, if they can use sonar technology to measure a lot of unique body parameters, such as volumes, shapes, and locations of different organs?

Table 1 Frequency Range of Hearing

Species	Frequency (Hz)	
	low	high
Humans	20	20,000
Cats	100	32,000
Dogs	40	46,000
Elephants	16	12,000
Bats	1,000	150,000
Grasshoppers	100	50,000
Rodents	1,000	100,000
Whales and Dolphins	70	200,000

Sound in computer-to-computer communication

Sound has many positive features which can make it a good alternative to IR light or radio. People (and most of the other beings in nature) use sound to communicate. Therefore, sound can work wherever people live or work. The human environment is usually designed to localize and preserve sound. Sound is much more predictable for people than IR or radio. It does not have unexpected foes such as sunlight, rain, metal objects, etc. Sound can be hidden from people if its frequency is higher than 20,000 Hz. IR usually requires direct visibility, does not work in sunlight or bright halogen light, and has some unexpected features. For example, it propagates though materials, which are not transparent in visible light, and reflects from different objects. Radio suffers from interference problems, and can be blocked by metallic objects. Sound can solve those problems. It goes around a corner, but can still be localized within any room. Most of the devices we use already have a speaker or beeper to tell us what is going on. Some of the devices, such as telephones or faxes, have microphones. Piezoelectric beepers in wristwatches can also be used as microphones. Many electronic things around us are designed to speak and/or listen.

However, sound has a set of features that make it hard to analyze. Sound travels relatively slow (about 300 m/s in air), and that creates two serious problems. First, a signal reaches its target with a noticeable delay. Second, reflections of the signal from walls or other objects can be comparable in intensity to the original signal, and can reach the receiver with different delays, interfering with the main signal. All audio equipment have non-linear characteristics and tends to shift signal phase. Rooms can have very different and practically unpredictable acoustic characteristics. Because of the slow speed of sound, echo may create quite long disturbances of a sound signal. That is makes it necessary to use rather long coding packets to transfer information. The only reliable method of acoustic signal analysis in an unknown environment is spectral analysis. Relative amplitude changes of spectral components of transmitted signal can then be detected by receiver.

Audible sound and lower-frequency ultrasound have similar features. Most of the small speakers and microphones used in electronic devices perform reasonably well at frequencies up to 10,000 Hz, beyond human perception. The frequency limit of piezoelectric devices is much higher. It means that the same devices can use ultrasound whenever their communication is not relevant for people and might be disturbing.

Sound signals and environment

Sound signals have several convenient mathematical representations. Each representation splits the received sound signal to a function of the original signal plus noise.

$$f_{recv}(t) = f_{noise}(t) + \int_0^\infty f_{sent}(t-x)g(x)dx$$

Figure 1

A received signal can be represented as a sum of the convolution of the transmitted signal with an impulse response of the environment and noise [Figure 1]. The lower bound in the convolution integral can be set to 0 because the system impulse response must be equal to 0 for a negative argument, since the system does not predict the future. Unfortunately, whenever air or any objects in a given environment move, the parameters of the system change, and as a result, the impulse response may change in time. Therefore, g(x) depends on t, and cannot be calculated in the beginning and then used to analyze or predict the received signal. However, in any environment that can be used for acoustic communication, $g(x)$ has to decrease exponentially as $x \to \infty$. Also the integral of $g(x)$ over the whole axis must be less or equal to 1.

The noise component [Figure 1] represents all other sources of sound in the environment plus a deviation of the system behavior from the regular convolution formula due to non-linearities. Noise is non-controllable and unpredictable.

An impulse function gives the most complete description of an environment. Unfortunately, numerical methods that can find an impulse function and deconvolve a signal are not well developed, and modern computers are not yet fast enough to process signals on that level of abstraction in real time.

It is possible to use a simpler representation of the received signal. The original signal can reach the receiver either directly or after reflection from one or more surfaces. Therefore, the received signal can be represented as a sum of directly received signal, all reflections (echoes) of the original signal that reach the receiver, and noise [Figure 2].

$$f_{recv}(t) = f_{noise}(t) + \sum_i \alpha_i f_{sent}(t - \tau_i)$$

Figure 2 Where $0 < \alpha_i \le 1$ *is an energy reduction coefficient and* τ_i *is the corresponding delay of a reflected or directly received signal.*

The new formula has some useful features that help to explain why spectral analysis works in this kind of tasks. If the original signal is a sine wave, the sum of all reflections of the original signal is also a sine wave of the same frequency [Figure 3]. The amplitude of that wave is proportional to the amplitude of the original signal, and the phase is hardly predictable. In the worst case the resulting amplitude can be constant 0. It happens if the receiver (microphone) is exactly at a trough of the interference pattern produced by the wave in the environment. Although it is theoretically possible, the probability of that in a real environment is extremely low.

$$\sum_i \alpha_i A_{orig} \sin(\omega t - \tau_i) = A_{orig} \sum_i \alpha_i \sin(\omega t - \tau_i) = A_{orig} A_{sum} \sin(\omega t - \tau_{sum})$$

Figure 3

Any signal can be broken down to a sum of sine waves. The spectrum of the signal gives the amplitudes of those waves. And, according to the previous conclusions, all components of the spectrum of the received signal are proportional to corresponding components of the spectrum of the sent signal. Therefore, the spectrum of the original signal can be estimated using the spectrum of the received signal.

The fourier transform allows the system to calculate a spectrum of a signal. However, if the sender uses only a few frequencies to encode data, there is not much sense in calculating the whole spectrum. It is enough to calculate only the fourier coefficients that correspond to the coding frequencies.

Since sound reflects from different objects and propagates rather slowly, a sharp change of the original signal spectrum will cause several changes of the received signal spectrum. And receiver can correctly analyze the signal only after a certain settle time. Therefore, the system can work only if either coding frames are longer than the average settle time or if the system has a good echo toleration reserve. Although the detectable echo tail in a medium-size room can be a few seconds long, the strongest echo components are the first reflections from large surfaces including walls, floor, ceiling, furniture, etc. The strongest echo components in a room usually have delays up to several hundredths of a second.

Communication Protocols

Overview

Acoustic computer communication requires development of special protocols. The protocols used in modems, faxes, telephones, and remote controls are made for mediums different from sound. However, all those protocols can be modified to work reliably with acoustic signals.

Modem protocols

The obvious step in this research is to try to adopt existing fax/modem protocols for acoustic transmission. In fact, modems and faxes are pseudo-acoustic devices and cannot effectively communicate through air. They only work well with electric signals and require a wire to isolate the system from external disturbances. The fastest modem protocols use almost all properties of existing telephone lines, rely on phase information, and do not tolerate multiple echoes. The earlier modem protocols (up to 2400 bps) are less telephone line dependent and can be used for air communication after minor modifications.

In low speed modem protocols, the audio signal stream consists of a sequence of small coding frames. Transmitter generates several predefined frames, which must be recognized by the receiver. All coding frames have the same length. Different coding frames must have different spectral characteristics to be distinguished by the receiver. Coding frames can contain either a single frequency, or a mixture of several frequencies. The simplest technique used in modems, phase-shift-keying (FSK), uses single-frequency frames. The frequencies are chosen so that they have a whole number of waves per frame and can be easily distinguished by the receiver.

After traveling in air, the signal is blurred because of the echo. An easy solution of this problem is to increase the frame length beyond an average echo settle time. The echo toleration of this method can also be increased by changing coding frequencies used in adjacent frames.

Data-hiding Protocols

Data hiding research [Reference 2] gives several possible techniques of adding recoverable data, which cannot be heard by people, to an audible acoustic signal. The most useful technique in my case is echo encoding. Most of the other methods do not preserve hidden data after traveling by air, and are basically only useful for digital or high-quality analog transfers.

Impulse (Remote Control) Protocols

IR remote controls use a sequence of light bursts with different delays between them to send data. This method can be used with sound. An acoustic remote control can send short impulses with pauses between them. This protocol is very easy to implement and does not require serious signal processing, but it cannot give high data transfer rates. I used this protocol in the handy-board project [Page 24].

Touch Tones

Touch tones used to dial telephone numbers and send information among telephone nodes work fine in air, but are too slow and noisy. However, it is a ready-to-use method for computers to talk to telephones or faxes. Recognition and generation algorithms are very easy to implement.

Each touch tone code is a mixture of two tones out of a set of eight basic tones. Table 2 shows the standard telephone codes. The digits, *, and # are used to dial a number and A-D are special signals.

I have written a program that generates and recognizes touch tone codes on PC using a sound board. The program generates patterns for each touch tone code during initialization, and sends them to the sound driver when the user presses the corresponding buttons. The recognition algorithm uses 8 digital filters to detect basic tones. The filters are similar to ones used in Sonicom. The touch tones work well if they are at least 20 ms long. Thus, the maximum reliable data transmission speed which can be achieved using touch tones is about 200 baud.

The standard touch tones are 70 ms or longer, which is above an average echo settle time. Therefore, computers, telephones, and faxes can talk well using touch tones.

Table 2 Standard Touch Tone Codes

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	**1**	**2**	**3**	**A**
770 Hz	**4**	**5**	**6**	**B**
852 Hz	**7**	**8**	**9**	**C**
941 Hz	*	**0**	**#**	**D**

Research Results

Overview

My research was devoted to exploration of different data encoding schemes for acoustic data transmission. In order to verify robustness of the protocols, I have built several working prototypes of systems that use sound to talk to each other.

Data-hiding Protocols

My experiments with echo encoding indicated that this method can give positive results at speeds up to 100 baud, but it requires too much computational power to implement a robust real time communication protocol. Although a simplified echo detection algorithm is fast enough to work in real time on 8 MHz computers, its noise tolerance and signal independence leave something to be desired. This algorithm can reliably recover about 10 bits/sec. Since the coding frame size is longer than an expected echo tail in a room, this protocol does not suffer from echo in environment.

The advantage of this technique is that it adds information to any acoustic signal in audible spectrum (speech, music, noise), so that people do not clearly hear the communication between devices. One of the possible tricks is to mix a message for people with a message for devices. It can be used, for example, in my battleship demo, where computers can mix human voice commands with a hidden echo code.

The echo encoding has the following problems.

1. Silence has no echo, therefore the echo encoding technique will not work well with signals that contain pauses. It can be avoided by adding some noise or ultrasonic tone to the signal, but noise spoils the quality, and ultrasound requires special equipment.
2. Speech and music may have fragments which look like an echo. For example, if the original signal is sine-wave and the chosen echo delay is equal to an integer number of the signal wavelength, you will get an echo-like fragment.
3. People can hear even very weak echoes. The echo encoding cannot completely hide the additional signal, it just makes it less noticeable.

In my experiments I use a 1-bit echo modulation. An echo added to a signal corresponds to 1 and subtracted from the signal to 0. The formula for this kind of modulation looks like:

$$f_{out}(b,t) = f_{in}(t) + (-1)^b \alpha f_{in}(t - \Delta t)$$

Figure 4 Echo modulation. Where Δt is the echo delay; $(0 < \alpha < 1)$ is the echo/signal ratio; b is the encoded bit.

The echo should not change its sign abruptly, otherwise any change from 0 to 1 or from 1 to 0 will sound as a click. A trivial way to avoid that effect is to increase and decrease the echo component linearly at both sides of each frame.

There are several methods of detecting an echo with a given delay. The most robust of them require fourier transform of the signal and analysis of its spectrum, and need a lot of processing power. Since my programs use computer's main CPU to process signals, and are required to do that in real time, I use a simpler method, which require only one multiplication per sample [Figure 5].

$$F_{out}(t) = \sum_{i=0}^{N-1} F_{in}(t+i\tau) \cdot F_{in}(t-\Delta t+i\tau)$$

Figure 5 Echo detection. Where τ is the sampling period; Δt is the echo delay; N is the number of samples used by the filter.

When a modulated signal is fed to the filter, the filter value can be presented as a sum of two components [Figure 6]. The first component is a sum of squares of the original function, and the second is a sum of products of the original function in different points. The detection of echo is based on the fact that the absolute value of the first component is usually greater than the absolute value of the second component, and therefore the sign of the filter output is defined by the encoded bit.

$$\begin{aligned}
&F_{out}(t) = \sum_{i=0}^{N-1} f_{out}(x,t+i\tau) \cdot f_{out}(x,t-\Delta t+i\tau) = \\
&\sum_{i=0}^{N-1} [f_{in}(t+i\tau)+(-1)^b f_{in}(t+i\tau-\Delta t)] \cdot [f_{in}(t+i\tau-\Delta t)+(-1)^b f_{in}(t+i\tau-2\Delta t)] = \\
&(-1)^b \sum_{i=0}^{N-1} f_{in}^2(t+i\tau-\Delta t) + \\
&\sum_{i=0}^{N-1} [f_{in}(t+i\tau) \cdot f_{in}(t+i\tau-\Delta t)+(-1)^b f_{in}(t+i\tau) \cdot f_{in}(t+i\tau-2\Delta t) + f_{in}(t+i\tau-\Delta t) \cdot f_{in}(t+i\tau-2\Delta t)]
\end{aligned}$$

Figure 6 Filtering of a modulated signal.

The program does not analyze the original signal, and, therefore, may give bad results if the signal contains pauses or resonance frequencies. That makes it necessary to use error correction code and verify check sums to support reliable data transfers.

Sonicom

Overview

Sonicom is a communication program prototype, which uses regular sound boards to send and receive signals. It currently can work in two different modulation modes. The first one uses 2-frequency frames to transmit 4 bits at a time. It sends 64-byte packets as a synchronous sequence of frames with a special hail frame in the beginning. The receiver recognizes the hail signal and tries to synchronize with the incoming data. If this stage is successful, it attempts to receive the following 64 bytes of information. The system uses a partial fourier transform to check frequency pattern of the received signal. The second modulation mode uses an amplitude modulated 18.4 kHz sound signal. In this mode, the system uses the same hail signal synchronization protocol and sends data as a set of 1 bit frames. In both modes, the system can transfer data at speed up to 1.4 kbaud.

Transmitter

The program generates and keeps in memory samples for each coding frame.

Frames in 2-frequency mode encode 4 bits. Each packet is either 2 different tones mixed together or silence [Figure 8]. In order to define 16 different packets (4 bits), the system uses 6 different frequencies. There are $C_6^2 = 15$ frames made of a pair of frequencies. The 16th frame is silence.

The system uses a hail frame to designate the beginning of a data packet. The hail signal has a frequency that is different from ones used to encode data, so that the receiver may always correctly find the synchronization impulses.

All frames are multiplied by the Blackman function, which is usually used as a framing function in discrete fourier analysis. Apparently, the system works better when the Blackman function is used by the sender to form signal frames instead of the receiver to filter signal. That change also reduces the amount of calculations on the receiver side, which is important.

$$f_{Blackman}(x) = 0.42 - 0.5 \cdot \cos\frac{2\pi x}{N} + 0.08 \cdot \cos\frac{4\pi x}{N}$$

Figure 7 The Blackman function: N is the frame width.

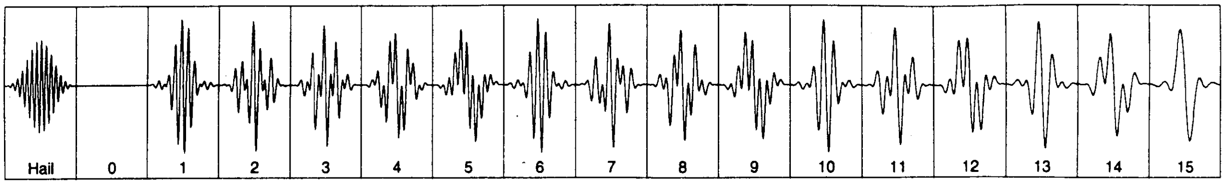

Figure 8 Waveforms of the hail signal and 16 information signals.

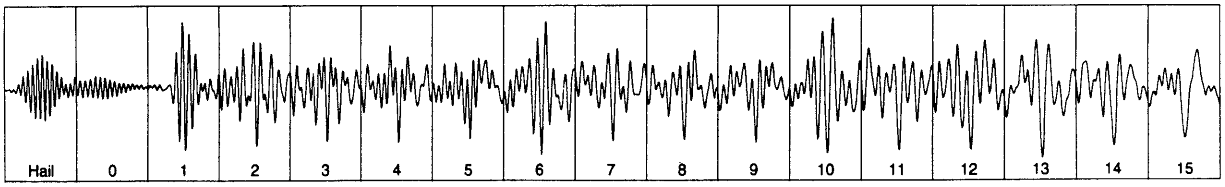

Figure 9 The same signal after traveling about 1 meter through air includes noise and echo added by the environment

In general, it is always better to use as few different frequencies to encode data as possible. Fewer frequencies require fewer filters, and, therefore, reduce the amount of computations necessary to decode a signal. The fewer frequencies are used - the better frequency separation and noise tolerance of the system.

There is a way to define 4 bits using only 5 frequencies. $C_5^2 = 10$ frequency pairs plus 5 single frequencies plus silence. Apparently, the receiver can not clearly recognize single frequency packets because of the echo from previous packets. The scheme with 6 frequencies appears to be much less sensitive to echo and noise.

The ultrasonic mode uses a high-frequency tone to encode data and the same kind of hail signal as in the 2-frequency mode. The data frames are only 30 samples wide, and each frame represents only one bit. A 18,375 Hz tone corresponds to 1 and no tone (silence) is a 0. Although the theoretical frequency limit for regular computer sound boards is about 22 kHz, they do not work well at this upper limit. I had to use a lower frequency. Usual computer microphones and speakers can handle frequencies up to about 30 kHz, but sound boards have maximum sampling rates of 44.1 or 48 kHz. The hail signal is audible. That is good for a system demonstration, but must be changed for truly ultrasonic protocol.

Receiver

The program conveys received sample data from the sound board driver to a set of digital filters. Then the system uses the filter outputs to detect and decode incoming information. The receiver program has an idle state when no incoming packets are detected. The system in that state uses only one filter that detects hailing frequency. When a hail signal is received, the filter output increases and then decreases, giving a one-frame-wide peak. The receiver estimates the middle of the peak to synchronize with incoming data, and then activates all frequency filters to decode the received packets.

The digital filters are written in assembler (Intel 80386 command set). They use fixed point arithmetic, which makes the program fast enough for Intel 80486DX 33MHz computers. The value of each filter is calculated by the formula:

$$F_{out} = \left(\sum_{k=0}^{N-1} x_k \sin \frac{2\pi i f_{filter}}{f_{sample}} \right)^2 + \left(\sum_{k=0}^{N-1} x_k \cos \frac{2\pi i f_{filter}}{f_{sample}} \right)^2$$

Figure 10 Filter function: N is the length of filter in samples; f_{filter} is the filtering frequency; f_{sample} is the sampling frequency; x_k is sample values. The value of each filter is a squared amplitude of the fourier coefficient corresponding to the filtered frequency.

Filters are the most computationally intensive part of the system. I optimized the filter block so that it performs only 3 multiplication operations per filter per sample. The program calculates the values of **sin** and **cos** during initialization and keep them in memory arrays for quick filter state updates. The system also keeps the sum arrays for quick sum calculation.

Technical Details

Software development system: Borland Delphi 1.0

Output Resolution: 8-bit 44,100 samples/second

Input Resolution: 16-bit 44,100 samples/second

Coding Packets 2-frequency mode: 120 samples/packet; 6 frequencies (735 Hz, 1470 Hz, 2205 Hz, 2940 Hz, 3675 Hz, 4410 Hz); hail frequency 5512.5 Hz; encoding 4 bits/packet (15 combinations of 2 out 6 frequencies plus silence); transmission speed: 1470 baud

Coding packets ultrasonic mode: 30 samples/packet; coding frequency 18375 Hz; hail frequency 5512.5 Hz; encoding 1 bit/packet (signal/silence); transmission speed: 1470 baud

Reliability

This method has a good echo toleration. It make it possible to use a short frame size. This protocol works well with distances between transmitter and receiver up to 2 m without obstacles. The reception improves if the receiver uses two microphones and sums the inputs.

Sonicom - data transfer scheme

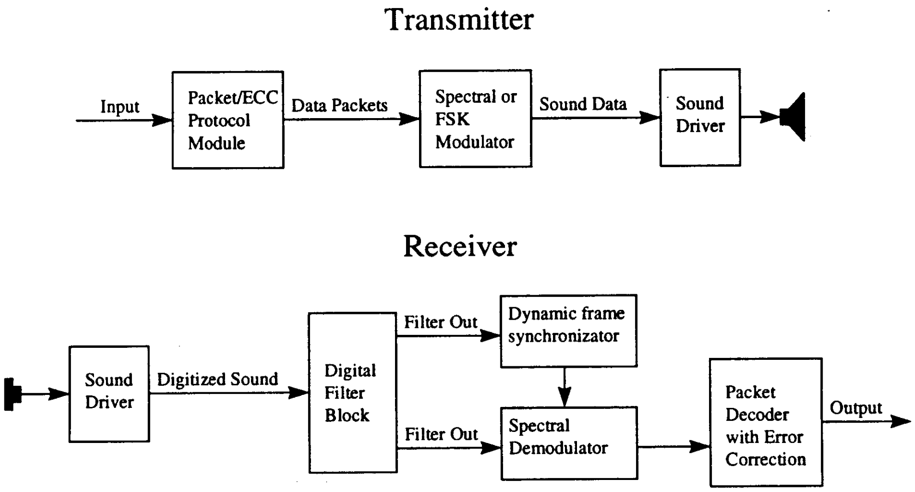

Figure 11

Battleship demo program.

The battleship demo was prepared for 10/10 Media Lab anniversary. Two PC computers played battleship with each other. The computers used sound to negotiate during the game. Two "talking balloons" from the physics lab were used as speakers in that demo.

The battleship program implements the traditional game rules. It displays a game progress and comments sent and received messages. The game algorithm is aware about basic tactical and strategical tricks of the battleship and is good enough to play with human partner.

The program uses modules from Sonicom to send/receive and modulate/demodulate acoustic signals. Because of bad acoustic properties of the balloons, I had to use two low-frequency tones for 1-bit data encoding and much wider frames. Although the balloons can work as microphones, I decided to use regular microphones to receive sound because balloons give really low quality input signal. I also could not get rid of a feedback effect when the same balloon was connected to a sound board as both a speaker and a microphone.

Handy-board clock & notebook.

Overview

A handy-board is a tiny computer developed in the Epistemology and Learning group of the Media Lab. I have programmed it as a clock with a notebook. It is now capable of sending the contents of the notebook records using sound to other computers. A handy-board has Motorola MC68HC11 CPU (8MHz), about 32K of static RAM, a 16x2 alpha-numeric LCD display, a tiny speaker, infrared receiver, an RS-232 serial port, several analog and digital input/output ports, several LEDs, two buttons, and a knob.

My project has three main parts: software development tools for MC68HC11 processor; a program for the handy-board to make it work as a clock / telephone book and send data using sound; and PC programs to decode sound signals sent by a handy-board and to edit the contents of the notebook records.

Handy-board software development tools.

In order to efficiently develop software for the board on Intel based PCs, I wrote an integrated cross-assembler/disassembler with software uploader. That program can compile programs for MC68HC11 processor, written in assembly language, and upload them into a handy-board though a serial interface. The assembler can use either original Motorola's command mnemonics, or modified ones, created by Brian Silverman. The latter mnemonics have simplified operand syntax and include addressing modes into command names, which makes it easier to find misprints during debugging. The system can save compiled code into a binary file for later use or upload it directly to a handy board if one is connected to computer's serial port. The disassembler can load a binary file and display its contents in MC68HC11 command mnemonics.

Clock-notebook Program.

The clock-notebook was programmed in MC68HC11 assembly language. I used Brian Silverman's logo interpreter code for handy board as the main source of useful subroutines and general information on handy-board programming. The program can switch among several pages of information. The first page has 2 lines that show current time and date in HH:MM:SS, MM-DD-YY format. All other pages display notebook records. The total size of notebook records is limited to 16 Kbytes. Although the records can have any number of lines and any number of characters in each line, the handy board program only displays the first 16 lines of a record, and the first 16 characters of a line. A user can switch pages consequently by pressing the right button, or return to the first page with time/date by pressing both buttons simultaneously. The handy board program sends out a notebook record using the speaker, when user pushes the left button.

The program can send and receive the notebook records using its serial interface. A user may run a special program on a host computer to update the contents of the notebook.

The program uses a protocol that is reminiscent of Morse code, but instead of using impulses of different length, it sends the same impulses with different pauses between them. The protocol encodes one bit at a time. The value of an encoded bit corresponds to the distance between two consequent sound impulses. A short pause means 0, and a long one means 1. The similar protocol is implemented in infra red remote controls. This protocol proves to be quite reliable. However, it suffers from echo, as every other protocol described here. The positive sides of the protocol include easy implementation, little computational overhead on both sender and receiver. But it is very noisy, if uses audible impulses, and slower than protocols that use multifrequency encoding.

The technical characteristics of the protocol are following:

Coding impulse frequency 4.405 kHz

Coding impulse width 5.7 ms (50 semi-waves)

Bit 0 pause 5.7 ms (50 semi-waves)

Bit 1 pause 17 ms (150 semi-waves)

Average expected trasmittion speed 59 baud (7-8 bytes/sec)

Receiver

The receiver is a PC program that uses one digital filter [Figure 10] to detect impulses sent by the handy-board. It dynamically adjusts its response level depending on the maximum filter output level from the last received impulse. The program measures the distance between raising edges of two consequent impulses. If the distance is reasonably close to the coding length of 0 or 1, the receiver adds decoded bit to a buffer. Otherwise, it resets the buffer.

Notebook editor

The notebook editor is a PC program that talks to the notebook program on a handy-board using serial interface. A user can download the contents of the notebook from a handy-board, edit it as a usual text and send it back. The program has an edit box, and two buttons to send and receive information.

Medical concerns.

Sound certainly has a psycho-physiological effect on people. There are actually two questions about audible sound and ultrasound. Audible sound can probably be used either in rather short data exchanges, when people may want to control the transmission (phone rings, acoustic remote controls, data exchange between wrist watches and computers), or for long data exchanges, when people are not there. Ultrasound in 20,000-500,000 Hz (and probably higher) range is harmless, if it is not very loud. If ultrasound is extremely intensive, it may cause heating and scattering of hard objects. Noise at frequencies lower than 100,000 Hz may be unpleasant for pets. There are many natural sources of ultrasound, such as bats, different kinds of insects, which may produce very loud ultrasonic noise. However, these do not seem to affect people or animals. Besides, almost all TV sets and computer monitors constantly squeak with line-scanning frequency, which is not recognized as a source of medical problems.

Evaluation of Acoustic Communication Methods.

I have conducted a short survey with 5 people from the Media Lab to find out what encoding techniques produce less irritating noise. I offered them to listen to ultrasonic, FSK, impulse (handy-board), and echo encoding. Since ultrasonic and echo encodings are practically inaudible, everybody found them the least irritating. Four people preferred the touch tones to other noisy techniques. They explained that either they were accustomed to touch tones or that the data transmission sounded nice and melodic to them. And one person from the group pointed out that since FSK is much faster than the touch tones, it might be better to listen to it for a short period of time, than to touch tones or impulse encoding for a longer period.

I think that devices should use ultrasound to communicate with each other and send audible signals only when they want to tell something to people. The FSK modulation can be used with ultrasound to achieve even higher data transfer rates. However, that will require special equipment, which can either work at higher than regular sound boards sampling rates or process signal on analog level.

In my opinion, the touch tones are rather attractive not only because we are used to them, but also the combinations of two frequencies produce beats (characteristic pulsations) which make sound alternately soft and loud. That reminds voices of musical instruments, and makes the tones quite pleasant for people. The touch tones are typically much longer than frames in FSK modulation. It explains why those two data modulations sound so different. Perhaps, the attractiveness of double-frequency tones can be used to make an encoding that reminds pleasant music.

The only advantage of the impulse (remote control) modulation, which is used in the handy-board project, is its simplicity. It requires minimal acoustic hardware quality and processing power, and can be used in small devices with limited resources.

Ideas for Future Research

Watch backchannel

A wristwatch can use its beeper to send data back to a computer. It may be a good solution for wristwatches that keep address/phone data.

Airport announcements

An airport announcement can contain data for wristwatches, PDAs, or laptops. A computer may have an easier time than people in recognizing the contents of those announcements.

Wall/table Ethernet

Several computers close to the same wall/floor/ceiling/table can use an acoustic transceiver to transfer data through a solid object. It would work as a wireless Ethernet. Sound would travel much faster than in air, and it can have higher frequencies.

Acoustic Position Tracker

Three acoustic responders placed on a ceiling can serve as reference points to track positions of other objects in the room. A thing that wants to know its position sends a ping to the responders, receives responses, measures time delays, and calculates the distances to each of the responders. This method may give relative coordinates of the thing with very high precision. The theoretical precision limit for a digital processing with a regular sound board (44,100 samples/sec) is about 15 mm, but analog ultrasonic devices can achieve much better results. The advantage of this method over electromagnetic sensors is that all coordinate calculations are linear.

References

1. Signal processing methods for audio, images and telecommunications / edited by Peter M. Clarkson, Henry Stark. 1995
2. Techniques for data hiding / Walter Bender, Daniel Gruhl, Norishige Morimoto, 1995
3. Digital signal processing in communication systems / Marvin E. Frerking. 1994
4. Signal coding and processing / Graham Wade. 1994
5. Pipelined adaptive digital filters / Naresh R. Shanbhag, 1994
6. Understanding Musical Sound with Forward Models and Physical Models Connection Science, / Casey, M. A. 1994.
7. The Handy Board. / Fred Martin 1995 http://lcs.www.media.mit.edu/groups/el/projects/handy-board/
8. Encyclopedia Britannica. http://www.eb.com
9. Справочник по математике. / И.Н.Бронштейн, К.А.Семендяев, 1974