

SonART: A framework for data sonification, visualization and networked multimedia applications

Woon Seung Yeo, Jonathan Berger *, Zune Lee
CCRMA, Stanford University
brg@ccrma.stanford.edu

Abstract

SonART is a flexible, multi-purpose multimedia environment that allows for networked collaborative interaction with applications for art, science and industry. In this paper we describe the integration of image and audio that SonART enables. An arbitrary number of layered canvases, each with independent control of opacity, RGB values, etc., can transmit or receive data using Open Sound Control. Data from images can be used for synthesis or audio signal processing and vice versa. SonART provides an open ended framework for integration of powerful image and audio processing methods with a flexible network communications protocol. Applications include multimedia art, collaborative and interactive art and design, and scientific and diagnostic exploration of data.

1 Introduction

SonART provides a framework for image and data sonification. In addition, tools for integrating image graphics, digital audio, and other data provide a robust platform for multimedia interaction. Originally created for sound driven data exploration and diagnostic purposes (O. Ben Tal, 2002), the software also provides a powerful tool for multimedia performance over a network and facilitates real-time interactive creation, manipulation and exploration of audio and images. The program is currently implemented as a Cocoa-based OS X application. Functionality includes the ability to load and draw multi-layered images, and to send/receive visual and control data to/from other multimedia programs that support Open Sound Control (M. Wright, 2001).

SonART's core features include:

- image display and processing capabilities
 - intuitive GUI features of audio mixers and of both audio and graphic editing software are preserved.
 - analogies between audio and visual processes are utilized.

- an arbitrary number of images can be simultaneously layered.
- user controlled parameters (i.e., opacity, compositing operations, visibility, location and size) for each layer can be controlled within the program interface or remotely by OSC messaging.
- a set of image processing filters and functions are available, and the program is designed to facilitate plug-ins for any type of image processing.
- Transmitting and receiving data for sound and image generation and processing
 - data such as pixel location on selected layers, opacity settings, RGB values on a per pixel and per layer basis, etc can be both transmitted and received to and from an arbitrary number of network connections using OSC.
 - data can be used for sonification using any OSC enabled synthesis or audio processing engine, or image processing on another host running a separate SonART application
 - scientific, medical, financial or other statistical data can be received from any networked source for directly mapped sonification and visualization purposes.
- Data sonification
 - SonART incorporates a parameter mapping framework (described separately in (W. Yeo, 2004)) for preprocessing data for auditory display.
 - sonification can be made directly from images or from data linked to image data. For example, the RGB values of a given layer may be used as sonification parameters, or an image pixel might be a reference to a vector of data in a given data set.

* supported by DARPA F41624-03-1-7000

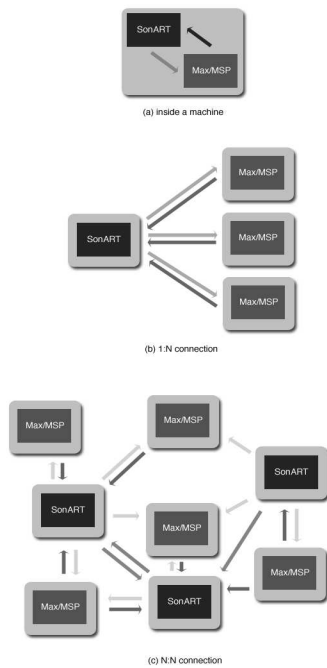


Figure 1: Flexibility of networked intercommunications between one or more instances of SonART and other OSC enabled audio or image processing programs

2 Network communications

Networked communications with SonART is managed with Open Sound Control (OSC) (M. Wright, 2001). OSC is a protocol for communication among various sound control and/or synthesis engines (i.e., computers, synthesizers, and other multimedia devices), offering flexibility and power. Using OSC, it is possible to transmit data between applications, both locally and through network. Moreover, implementation includes most widely-used sound synthesis softwares, including Csound, Max/MSP, Pd, and Supercollider, providing a wide range of synthesis and audio processing possibilities. At the core of OSC is a specification for the format of messages. Each OSC message contains a URL / path style target address and a variable number of parameter arguments. The message format is independent of transport layer, however in our case it is implemented using UDP packets over IP networks.

2.1 Flexibility of connections

The flexibility afforded by OSC provides for mappings between arbitrary combinations of parameters. Typical mappings of visual parameters in SonART are represented in 1. These include:

- mapping to and from sound synthesis applications. This

would be the most usual case for image sonification. The program can be run on the same machine as SonART, or any other place through network, and multiple applications can be simultaneously connected.

- internal communications within SonART. Parameters can be mapped to any other parameters within the same document, and/or between an arbitrary number of other documents.
- communications between multiple platforms running SonART.

2.2 Address space

The address space of OSC provides a hierarchical and manageable organizational method for communications to, within, and from SonART. For example, data coming into a document of SonART will have a URL-like address such as the following:

```
/toSonART/document_name.sonart/layer_name/parameter_name
```

Here, `/toSonART` identifies destination as SonART. If a document named `document_name.sonart` is open, it will receive the message and do corresponding processes according to the `layer_name` and `parameter_name`. For outgoing parameter data to sound applications, choice of address name depends on the implementation of their OSC routings. For consistency sake we address incoming data with an identifier beginning with `/fromSonART`.

2.3 OSC Implementation: WSOSC

Transparency of communication control is achieved with WSOSC, an Objective-C library of OSC objects and methods available as a Cocoa framework for Mac OS X. Compared to the previously available OSC libraries for Cocoa, WSOSC provides an integrated environment for handling OSC, providing easier ways to receive and parse OSC packets. Classes for client applications are currently being written.

WSOSC is a Cocoa framework, written in pure Objective-C, for parsing Open Sound Control data. Currently, it can parse the following information:

- packet validity: data length and content
- packet type: message/bundle.
- address patterns
- type tag string
- arguments: int32, float32, OSC-string

2.3.1 Class information

WSOSC has three major classes: WSOSCPacket, WSOSCBundle, and WSOSCMMessage. Methods of these classes are described below.

- WSOSCPacket

- (id)initWithDataFrom:(NSString *)data; Initiates and returns an instance of WSOSCPacket with the data parsed from (NSString *)data, which is the input from UDP socket.
 - * (id)packetParsedFrom:(NSString *)data; Similar to - (id)initWithDataFrom:(NSString *)data;. This returns an *autoreleased* instance.
- (void)parseFrom:(NSString *)data; Parses the data and stores information.
- (int)type; Type of the parsed packet. This can return four values:
 - * -1: Invalid packet (neither a message nor a bundle).
 - * 0: Invalid packet (size is not a multiple of 4).
 - * 1: Packet is a valid message.
 - * 2: Packet is a valid bundle.
- (id)content; Returns the *content* of parsed packet. Depending on the type of the packet, an instance of WSOSCMMessage or WSOSCBundle will be returned.

- WSOSCMMessage

- (id)initWithDataFrom:(NSString *)data; Initiates and returns an instance of WSOSCMMessage with the data parsed from (NSString *)data (mostly handed over from the instance of WSOSCPacket or WSOSCBundle).
 - * (id)messageParsedFrom:(NSString *)data; Similar to - (id)initWithDataFrom:(NSString *)data;. This returns an *autoreleased* instance.
- (void)parseFrom:(NSString *)data; Parses the data and stores information.
- (NSString *)addressString; Returns address patterns as one NSString.
- (NSArray *)addressPattern; Returns address patterns as an NSArray, which contains each address pattern as elements.
- (int)numberOfAddressPatterns; Returns the number of address patterns.
- (NSString *)addressPatternAtIndex:(int)index; Returns the address pattern at *index*.

- (BOOL)hasTypeTag; Checks if the message has a type tag; this is necessary to deal with some old implementations of OSC.
- (NSString *)typeTagString; Returns type tags as one NSString.
- (char)typeTagAtIndex:(int)index; Returns the type tag character at *index*.
- (int)numberOfArguments; Returns the number of arguments in message.
- (NSMutableArray *)arguments; Returns arguments as an NSMutableArray, which contains each argument as elements.
- (id)argumentAtIndex:(int)index; Returns the argument at *index*.

- WSOSCBundle

- (id)initWithDataFrom:(NSString *)data; Initiates and returns an instance of WSOSCBundle with the data parsed from (NSString *)data (mostly handed over from the instance of WSOSCPacket).
 - * (id)bundleParsedFrom:(NSString *)data; Similar to - (id)initWithDataFrom:(NSString *)data;. This returns an *autoreleased* instance.
- (void)parseFrom:(NSString *)data; Parses the data and stores information.
- (int)numberOfBundles; Returns the number of bundles in message.
- (NSNumber *)bundleTimeTag; Returns bundle time tag (not implemented yet).
- (NSMutableArray *)bundles; Returns bundles as an NSMutableArray, which contains each bundle as elements.
- (WSOSCMMessage *)bundleAtIndex:(int)index; Returns the bundle at *index*. Here, each bundle is equivalent to a message.

Detailed information about WSOSC can be found at <http://www-ccrma.stanford.edu/groups/soni/SonART/>

3 Research and creative applications

In this section we describe how SonART is currently used as a sonification framework, and offer examples of ways in which the application could be used in creating multimedia art and collaborative performances.

3.1 Sonification

The auditory system is an underused resource in data analysis and diagnostics. Sonification, or auditory display of data has a wide range of applications in scientific and medical research and in industry. Our research focuses on two aspects of sonification research:

1. Discovering and implementing intuitive, easily learned representations of complex multi-dimensional data. To this end we have been concentrating upon models of the human vocal tract (R. Cassidy, 2004). Human capacity to integrate dimensions into the percept of a particular vocalization (a vowel for example), as well as our ability to discriminate one or more vowels simultaneously, even in the presence of a considerable amount of other sound, makes this a highly promising approach. A generalization of this approach that we concurrently explore is the implementation of other physical models of acoustic sound sources for sonification. For example, imagine a virtual membrane surface simulating a drum skin of arbitrary size and shape. Meaningful sonification could be achieved by using data to adjust the parameters of the virtual instrument, producing a variety of natural-like sounds reflecting various states or conditions of the data being analyzed.
2. Exploration of meaningful cross modal representations of data such that images can be sonified, and conversely, sounds can find innovative graphical representations.

SonART provides a robust framework for integration of auditory and visual display to expose patterns or trends in data.

3.2 Compositional uses of sonification

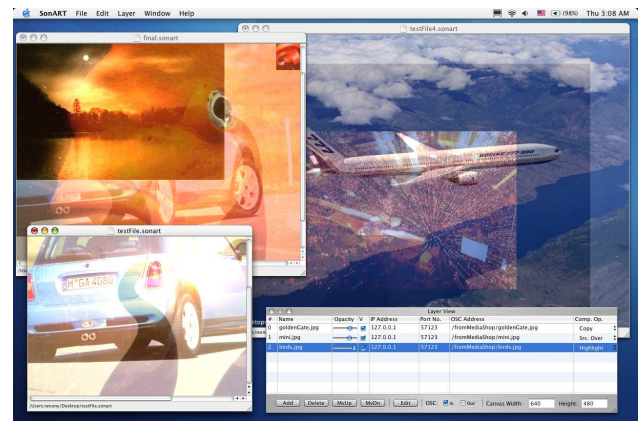
Numerous composers and artists have used sonification for artistic purposes (Berger and Tal, 2004). A creative example of real time auditory display for artistic purposes is Mansen and Rubin's *Listening Post* (Hansen and Rubin, 2002) which sonified internet chatroom activity. Of particular interest to us is the use of physical models as a paradigm for intuitive sonification. Recent work such as (Castagne and Cadoz, 2002) demonstrate the enormous potential of combined graphical and auditory display of physical models for research, composition and auditory display. Muth and Burton's *Sodaconductor* (Muth and Burton, 2003) generates musical data from dynamic physical simulations. Meijer's *vOICE* method (Jones, 2004) sonifies images both for

creative purposes and as an assistive technology for the sight impaired. Similarly, SonART provides both artistic and applicative outlets from a single sonification paradigm.

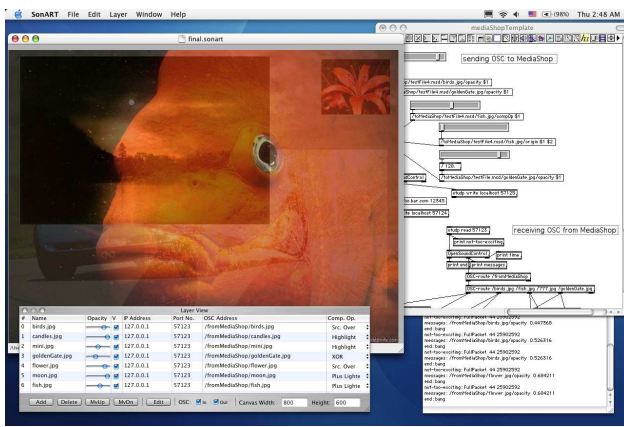
Potential industrial applications of SonART driven sonification include medical imaging diagnostics, financial and stock analysis, and homeland security and defense. We are currently working with hyperspectral medical imaging data of colon cells and have demonstrated the ability to distinguish between benign and malignant cell structures with auditory display. The SonART platform provides a powerful exploratory tool for diagnostic analysis of this highly dimensional data. The ability to receive any type of data from any type of remote source using OSC protocol over a network provides SonART with the capability of serving as a real time sonification system.

3.3 Collaborative art

The integration of visual and audio processing tools with a highly efficient hierarchically based network communication protocol brings a broad range of other applications for SonART. Integrating SonART with OSC-enabled synthesis engines such as MAX/MSP, PD or STK across a distributed network facilitates arbitrarily complex synthesis and signal processing architectures shared among multiple platforms with data sent from a single SonART host.



layered images with variable opacity controlled by audio parameters communicated through OSC



MSP control of image parameters

Image, sound and data can be processed and shared simultaneously by an arbitrary number of users. In addition to data sonification, the networked integrated multimedia framework of SonART provides a mechanism for artistic and creative work both by a single artist or in a collaborative environment. Examples include:

- Real-time playing by drawing.
By choosing different drawing or visual filtering tools, users can choose different timbres for musical sounds, associate brush width or paint mode with a particular musical texture, analogously apply visual blur filters with audio reverberation, localization, pitch or harmonic distortion or virtually any other type of audio processing using DSP or MIDI.
- Interactive sonification and visualization.
SonART's network capability allows for collaborative art and design. An example scenario might have a visual digital artist creating or processing an image. One or more aspects of her work, perhaps pixel by pixel RGB values, or the opacity ratios of one or more image layers, is sent in real-time to a musician who manipulates this data as musical parameters. The musician's data can be directed back to the visual artist and contribute to the image creation, or be transmitted to another platform in which the musical data is visualized by SonART. In fact, the interaction can incorporate remote sensing data, whether sensing the motions of an artist or actor, or receiving real-time data from any internet data transmission. An arbitrarily large community of artists can thus work in parallel collaborating on a single art project over the internet.

One of the challenges of audio-visual integration is a coherent and compelling cross modal sensory mapping. SonART's flexible mapping paradigm allows for a wide range of experimentation in this area. An example of integrated audio-visual processing is analogous mapping of filters to images and sound. A lowpass audio filter

might, for instance, be deemed analogous to an image blur filter, audio delay lines might be mapped analogously to Gaussian image blur filters, or spatialization mapped to image location.

4 Summary

SonART is a powerful creative tool for artistic creation, multimedia editing, interactive video games, and learning environments as well as a tool for managing auditory display of data.

5 References

*References

- Berger, J. and O. B. Tal (2004). De natura sonoris. *Leonardo Music Journal* 37(3).
- Castagne, M. and C. Cadoz (2002). Creating music by means of 'physical thinking': the musician oriented genesis environment. In *Proceedings of the fifth annual Conference on Digital Audio Effects, Hamburg, Germany*, pp. 999-999. Conference on Digital Audio Effects.
- Hansen, M. and B. Rubin (2002). Listening post: Giving voice to online communications. In *Proceedings of the International Conference on Auditory Display*, pp. 999-999. International Community for Auditory Display.
- Jones, W. (February 2004). Sight for sore eyes. *IEEE Spectrum - on line*.
- M. Wright, A. Freed, e. a. (2001). Managing complexity with explicit mapping of gestures to sound control with osc. In *Proceedings of the International Computer Music Conference*, pp. 999-999. International Computer Music Association.
- Muth, D. and E. Burton (2003). Sodaconductor. In *Proceedings of the Conference on New Interfaces for Musical Expression (NIME-03), Montreal Canada*, pp. 999-999.
- O. Ben Tal, J. Berger, e. a. (2002). Sonart : The sonification application research toolkit. *Proceedings of the International Conference on Auditory Display*.
- R. Cassidy, J. Berger, e. a. (2004). Auditory display of hyperspectral colon tissue images using vocla synthesis models. *Proceedings of the International Conference on Auditory Display*.
- W. Yeo, J. Berger, e. a. (2004). A flexible framework for real-time sonification with sonart. *Proceedings of the International Conference on Auditory Display*.