Audio Engineering Society

# Convention Paper

Presented at the 123rd Convention
2007 October 5–8 New York, NY



# Sharing Acoustic Spaces over Telepresence using Virtual Microphone Control

Jonas Braasch[1], Daniel L. Valente[1], Nils Peters[2]

[1] *Rensselaer Polytechnic Institute, Troy, NY, 12180, USA*

[2] *CIRMMT, Schulich School of Music, McGill University, Montreal*

Correspondence should be addressed to Jonas Braasch (braasj@rpi.edu)

**ABSTRACT**

This paper describes a system which is used to project musicians in two or more co-located venues into a shared virtual acoustic space. The sound of the musicians is captured using spot mics. Afterwards, it is projected at the remote end using spatialization software based on virtual microphone control (ViMiC) and an array of loudspeakers. In order to simulate the same virtual room at all co-located sites, the ViMiC systems communicate using the OpenSound Control protocol to exchange room parameters and the room coordinates of the musicians.

## 1. INTRODUCTION

Currently, the authors participate in a university-based music project, in which music is performed regularly over the internet with co-located musicians. At each site, the sound of the musicians at the other end are projected using a loudspeaker array of typically eight speakers. The general transmission scheme is shown in Fig. 1.

One of the major difficulties of two-way transmissions is to avoid feedback loops that occur when the microphones at the receiver end pick up the loudspeaker signals used for monitoring and return these signals to the sender site. Though tolerable with speech signals, commercial echo-cancellation systems induce coloration effects that are undesirable in music applications. The easiest way to avoid echoes is to place the microphones close to the instruments (spot-mic recording) and so achieve a high signal-to-noise ratio. Unfortunately, by doing so the spatial information is lost after the transmission.

Virtual auralization techniques have proven to be a successful tool to resynthesize or newly create the spatial information of spot-mic recordings. For this

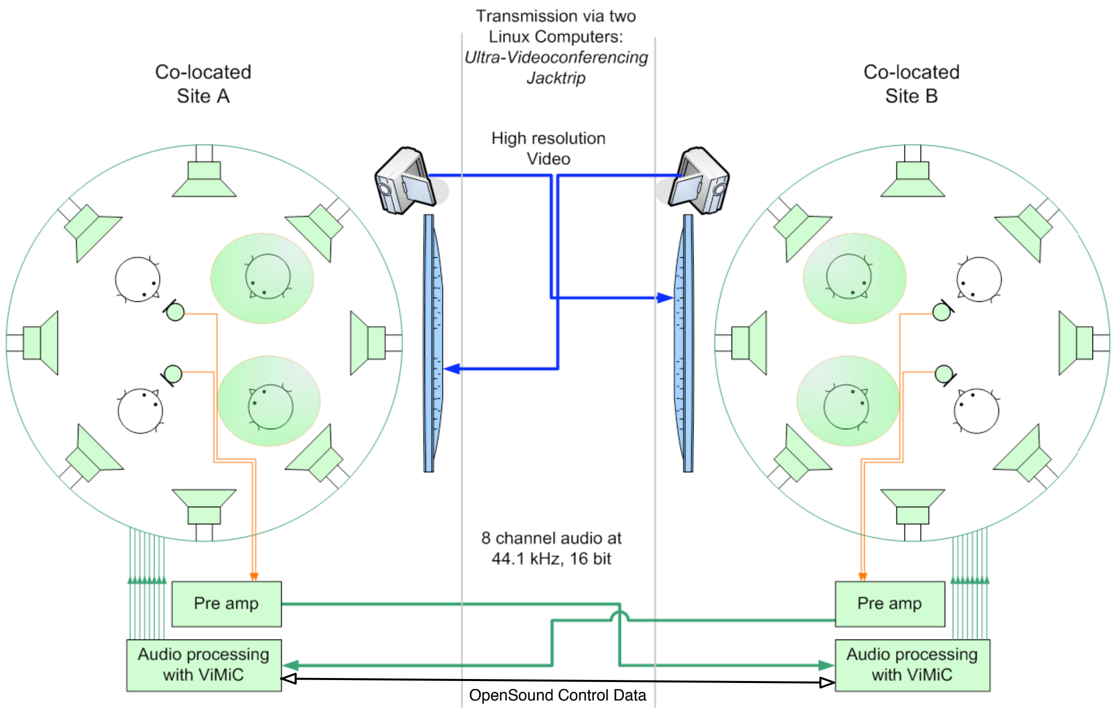


**Fig. 1:** General transmission scheme.

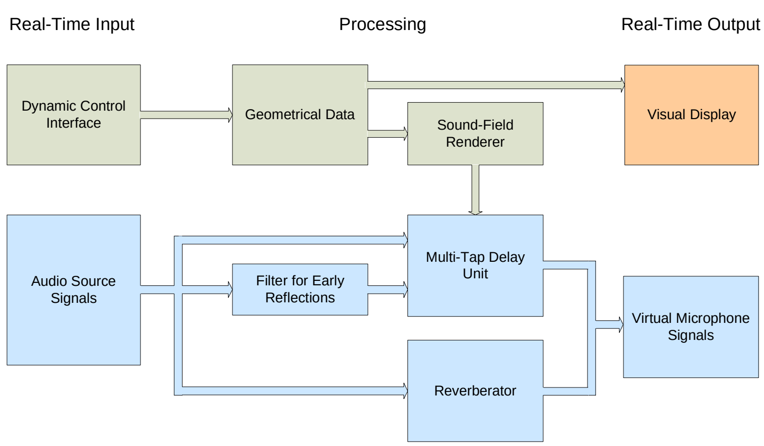


**Fig. 2:** Architecture of the auditory virtual environment based on Virtual Microphone Control (ViMiC).

purpose, several panning algorithms are presently available such as the spatialisateur [9], Vector-Based Amplitude Panning (VBAP) [12], and Virtual Microphone Control (ViMiC) which was used for the following study.

## 2. VIRTUAL MICROPHONE CONTROL

The ViMiC system, which was presented at an earlier AES convention [2], is based on an array of virtual microphones which is used instead of panning laws to address an array of loudspeakers. In order to allow the user to freely position a sound in space, an array of virtual microphones with simulated directivity patterns is created. The axial orientation of these patterns can be freely adjusted in 3D space, and the directivity patterns can be varied between the classic patterns that are found in real microphones: omnidirectional, cardioid, hyper-cardioid, sub-cardioid, or figure-eight characteristics. For example, the array could consist of five cardioid microphones that are arranged according to ITU standard with axis orientations of $\pm 110°$, $\pm 30°$, and $0°$ at a distance of 0.5 m from a common center position. The virtual microphone signals are then fed to an array of loudspeakers. For this study, we are using an eight-channel microphone set-up that goes along with the 8-channel loudspeaker set-up ($45°$ inter-loudspeaker spacing starting from $0°$).

After determining the spatial arrangement of microphones and sound sources, the delay and gain of a sound source is calculated for each microphone from the distance between both (the microphone and the sound source), and the axis orientation of the directivity pattern and the directivity pattern itself. Early reflections can be considered in the calculations as well.

## 3. VIMIC IMPLEMENTATION

The ViMiC system has been implemented in C++ using the Pure Data (PD) [11] and Max/MSP environments [7]. The basic architecture of the system is shown in Fig. 2. The Sound-Field Renderer Unit calculates the gain and delay between each sound source and virtual microphone, and thus determines the sound field at the microphone positions. Using the data provided by the Sound-Field Renderer, the

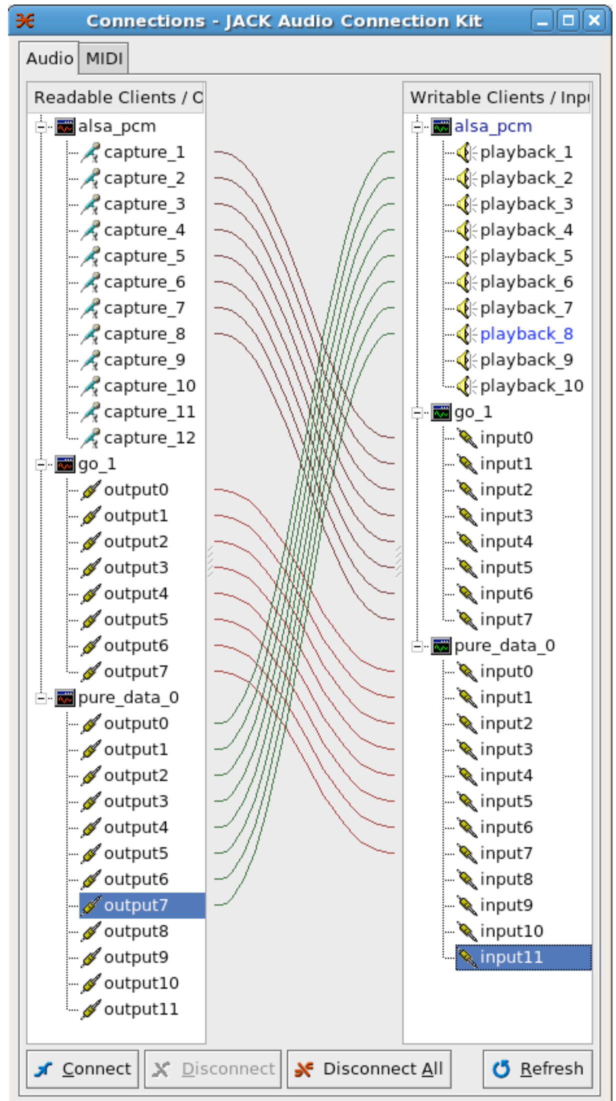


**Fig. 3:** Internal Audio Routing for the ViMiC system based on a transmission with Jacktrip using *qjackctl*.

dry sound is processed using a multi-tap delay network for spatialization. The gain and delay for each output tap is provided by the Sound-Field Renderer unit.

In addition to handling the direct sound sources, the ViMiC module calculates the gains and delays between all first-order reflections and the microphones. Second-order reflections can be rendered as well if needed. The coordinates of the reflections are calculated using the mirror-image technique [1]. By using this approach, the present algorithm is limited to rectangular rooms. Other techniques, e.g. ray tracing, could be implemented into the ViMiC system, should it become necessary to simulate more complex room shapes. The three room dimensions (width, length, and height) can be set freely. The absorption coefficients of the walls, ceiling, and floor are simulated using FIR filters. The simulation of diffusion has not been implemented yet. In the present implementation, the late reverberation is generated using a multi-channel reverberation algorithm based on feedback loops and a 16×16 Hadamard Matrix. Two equalizers are used for timbral balance, one is working in front of the feedback-delay network, the second one is integrated within the delay network to simulate the frequency-dependent absorption characteristics of acoustical wall materials.

## 4. INTEGRATION INTO TRANSMISSION SYSTEMS

The ViMiC software has been used with several transmission software environments including Ultra Video Conferencing developed at McGill University [10], [6], [14] and Jacktrip, which was designed at CCRMA, Stanford University [4]. The most compact installation has been achieved using the ViMiC environment in Pure Data with the Linux Distribution Fedora Core 6, which is also the standard environment for Jacktrip. The benefit is that both programs can be run simultaneously using the low-latency audio server application *Jack*. The video component of Ultra Video Conferencing can also be implemented onto the same system for a bi-directional A/V transmission. Figure 3 shows the internal audio routing which was set using jack graphical user interface *qjackctl*.

The captured audio is routed directly from the sound card to the remote site using jacktrip (The connection is labeled "go_1."). The transmitted audio signal is then spatialized using the Pure Data implementation of ViMiC (Fig. 4). For this purpose, the output of Jacktrip (also labeled "go_1") is routed through PD for spatial processing, before it is sent to the output of the audio interface (labeled "alsa_pcm"). The audio routing of the second computer at the remote site can be established the same way.

The underlying geometrical data is transmitted from the sending computer using the OpenSound Control protocol [15]. Currently, the data has to be adjusted manually, but in future we expect to integrate an acoustic tracking system to estimate the positions of the sound sources (talkers, musical instruments) in real-time. Apart from the positions several other acoustical parameters can be transmitted as well, as described in Section 5, including room acoustical parameters.

## 5. CONTROL INTERFACES

The communication between the ViMiC environment and its user interfaces is established through OpenSound Control (OSC). The OSC protocol allows to address the ViMiC environment from another computer through a network connection. A current list of controllable parameters for the ViMiC C-external is provided in Table 1. Other parameters such as reverberation time, the equlizer settings for early reflections and late reverberation can be set using OSC as well. Several graphical user interfaces exist for ViMiC. Figure 4 shows a graphical user interface for Linux, Fig. 5 a GUI, which has been designed in Max/MSP. With the latter, the ViMiC unit can be controlled from most commercially available Digital Audio Workstations (DAW). A plug-in was designed to run on VST, RTAS or Audio Units host applications. The control plug-in based on the Pluggo Runtime Environment for Max/MSP [7]. A separate plug-in unit can be loaded for each audio track to be spatialized with ViMiC. The DAW software automation can be used to control the values of all ViMiC parameter data. The ViMiC control plug-in communicates with the dedicated auditory rendering system through OSC control messages and a

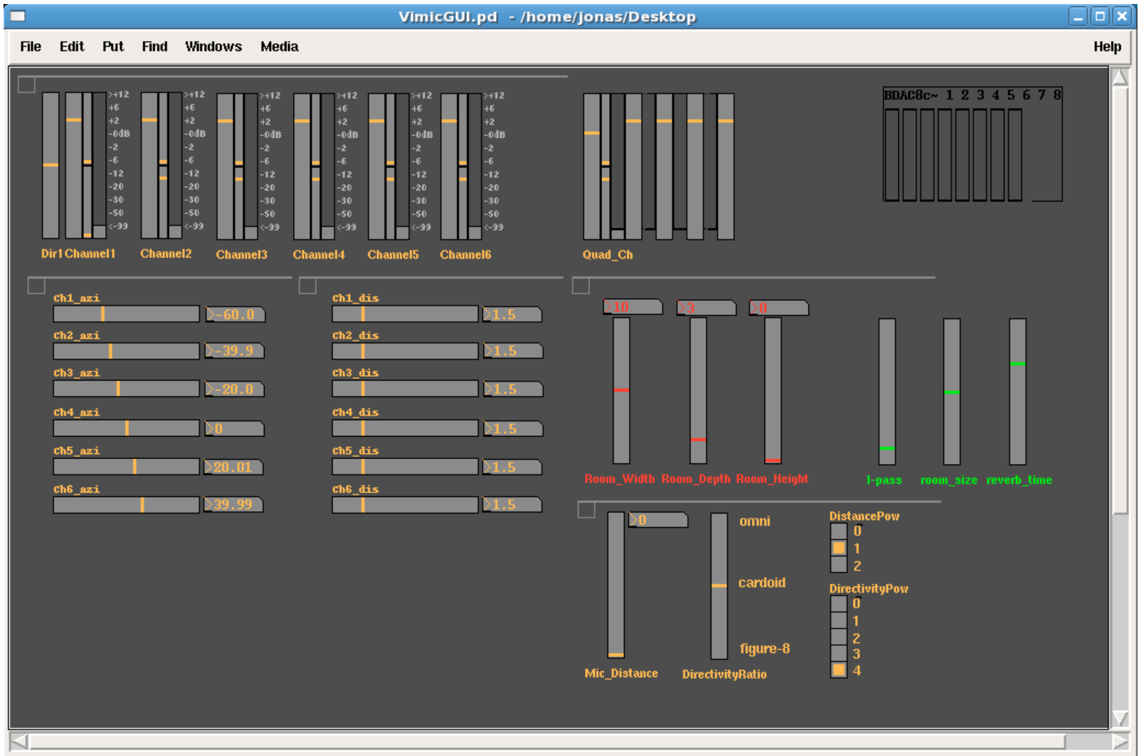


**Fig. 4:** Graphical User Interface for ViMiC in Pure Data (Fedora Core 6 implementation).

UDP network. The pre-recorded audio tracks can be streamed from the DAW to the ViMiC unit through a digital multichannel audio connection.

Both GUIs can be used to change the room acoustical parameters at the local and the remote site at the same time. Hence, both sites share the same acoustical space, which can be updated continuously. Currently, we are working on a system to automatically adjust the spatial locations of sound sources as described in the next section.

## 6. AUTOMATIC SOUND SOURCE TRACKING

The automatic tracking system, which is described in detail in [3], is only briefly introduced here. The localization process is based on time delay differences between various channels of a small-aperture pyramidal five-microphone array. While these type of systems work well with single sound sources, scenarios with multiple sound sources remain to be a problem. The performance of the system was improved significantly by analyzing the lavalier microphone signals time-frequency wise to calculate the signal-to-noise ratio (SNR) between each talker/musician and the concurrent talkers/musicians. An algorithm was designed to select time-frequency bins that showed a high SNR for robust localization of the various talkers/musicians and to identify the talkers/musicians of the localized sources. It was found that correlating the talker/musician-worn microphones with the microphone array allows for a greater accuracy and precision of localization than with only the microphone array. Currently, the algorithm is operated offline, but in future we expect to have a real-time version that can be used to track the current positions of talkers/musicians. These data will then be transmitted via OSC to the remote site for accurate spatial reproduction of the original positions of each talker/musician.

| Command | Parameter(s) | Description |
|---|---|---|
| /SourceXpos | index $m$, $x$ [m] | changes $x$ position for sound source with index $m$ |
| /SourceYpos | index $m$, $y$ [m] | changes $y$ position for sound source with index $m$ |
| /SourceZpos | index $m$, $z$ [m] | changes $z$ position for sound source with index $m$ |
| /SourcePos | index $m$, $x$ [m], $y$ [m], $z$ [m] | changes position ($x$,$y$,$z$) for sound source with index $m$ |
| /RoomSize | $x$ [m], $y$ [m], $z$ [m] | changes the room size of the virtual room ($x$,$y$,$z$) |
| /RoomWidth | $x$ [m] | changes width $x$ of virtual room |
| /RoomDepth | $y$ [m] | changes depth $y$ of virtual room |
| /RoomHeight | $z$ [m] | changes height $z$ of virtual room |
| /MicXpos | index $n$, $x$ [m] or<br>only $x$ [m] to address all mics | changes $x$ position of virtual microphone with index $n$ |
| /MicYpos | index $n$, $y$ [m] or<br>only $y$ [m] to address all mics | changes $y$ position of virtual microphone with index $n$ |
| /MicZpos | index $n$, $z$ [m] or<br>only $z$ [m] to address all mics | changes $z$ position of virtual microphone with index $n$ |
| /MicPos | index $n$, $x$ [m], $y$ [m], $z$ [m] | changes position ($x$,$y$,$z$) of virtual microphone with index $n$ |
| /MicCenterDistance | $d$ [m] | positions all microphones at distance $d$<br>from the center of the microphone array<br>at their current angles |
| /MicAzi | index $n$, $alpha$ [deg] or<br>only azi to address all mics | determines azimuth angle $alpha$<br>for the directivity pattern of the virtual mic |
| /MicAzi | index $n$, $theta$ [deg] or<br>only azi to address all mics | determines elevation angle $theta$<br>for the directivity pattern of the virtual mic |
| /MicAngle | index, $alpha$ [deg],<br>$theta$ [deg] | determines azimuth and elevation angle<br>of virtual mic $n$ |
| /Directivity | $\Gamma$ | determines the directivity pattern of a mic with<br>0= index $n$, figure-8, 0.5=cardioid, 1=omni |
| /DirPow | index $n$, $\delta$ | provides the directivity power for mic $n$<br>$\delta = 1$ 1st-order microphone |
| /DisPow | index $n$, $r$ | distance power, determines the amplitude decay<br>with distance, 1=1/$r$ law, 0=no amplitude<br>decay with distance |
| /ReportAll | bang | ViMiC will print out the following data:<br>number of channels, source positions, room size<br>($x$,$y$,$z$), Mic Array Center ($x$,$y$,$z$) and micro-<br>phone data including delay and sensitivity |
| /Report | 1 (on), 0 (off) | will print out every executed command<br>with variables if /Report is set to 1 |

**Table 1:** Implemented OpenSound Control commands for ViMiC.


**Fig. 5:** ViMiC control as a VST plug-in

## 7. OUTLOOK

The system described here was part of a music transmission project in which the authors participated. During an evening concert at the International Conference on Auditory Displays on June 26, 2006 in Montreal, two music ensembles – Tintinnabulate and Soundwire –, which are based at RPI and Stanford were featured. The revised version of the ViMiC system, which is described in this article, will be used for a second concert at the upcoming SIGGRAPH conference in San Diego.

As outlined in the previous section, the automated tracking of musician using a microphone array in combination with lavalier microphones tops our agenda. We are also planning to measure the co-located acoustic environments in real-time using dummy heads to adjust the desired acoustic settings adaptively. This way, people at two co-located sites will be able to share the same acoustic environment even if the physical enclosures of both spaces have different acoustic characteristics.

## 8. ACKNOWLEDGMENT

We would like to thank Pauline Oliveros, Chris Chafe, Juan-Pablo Caceres, and members of the Tintinnabulate Ensemble for the musical collaboration that inspired much of the work presented here. We would also like to thank Jeremy Cooperstock for creating a Fedora-Core-6 version of his Ultra-Video Conferencing software. We are also indebted to the help of network administrators at RPI, McGill, Stanford, and UCSD to create a flawless transmission. In particular, Nigel Westlake at RPI helped us to work out details to successfully troubleshoot problematic connections.

## 9. REFERENCES

[1] Allen, J.B., Berkley, D.A. (1979) *Image method*

*for efficiently simulating small-room acoustics*, J. Acoust. Soc. Am. **65**, 943–950.

[2] Braasch, J. (2005) *A loudspeaker-based 3D sound projection using Virtual Microphone Control (ViMiC)*, Convention of the Audio Eng. Soc. **118**, May 2005, Preprint 6430.

[3] Braasch, J., Tranby, N. (2007) *A sound-source tracking device to track multiple talkers from microphone array and lavalier microphone data*, $19^{th}$ International Congress on Acoustics, Madrid, Sept. 2–7.

[4] Juan-Pablo Caceres, *JackTrip – Multimachine jam sessions over the Internet2* SoundWIRE research group at CCRMA, Stanford University, http://ccrma.stanford.edu/groups/soundwire/

[5] Chafe, C. (2003), *Distributed Internet Reverberation for Audio Collaboration*, Proc. of the AES 24th Int. Conf., Banff, 2003.

[6] Cooperstock, J.R., Roston, J., Woszczyk, W. (2004), *Broadband Networked Audio: Entering the Era of Multisensory Data Distribution*, 18th International Congress on Acoustics, Kyoto, April 4–9.

[7] Cycling'74. *Max/MSP and Pluggo*, http://www.cycling74.com.

[8] ITU (1994) *Multichannel stereophonic sound system with and without accompanying picture*, Standard BS.775-1, International Telecommunication Union, 1992–1994.

[9] Jot, J.-M. (1992) *Étude et réalisation d'un spatialisateur de sons par modèls physiques et perceptifs*, Doctoral dissertation, Télécom Paris.

[10] McGill Ultra Videoconferencing Research Group, http://canarie.mcgill.ca/

[11] Puckette, M.: *Pure Data, a patchable environment for audio analysis, synthesis, and processing, with a rich set of multimedia capabilities*, http://www-crca.ucsd.edu/˜msp/software.html.

[12] Pulkki, V. (1997) *Virtual sound source positioning using vector base amplitude panning*, J. Audio Eng. Soc. **45**, 456–466.

[13] Settel, Z. and SAT Audio Group: *nSLAM audio suite*, http://tot.sat.qc.ca/eng/nslam.html.

[14] Woszczyk, W., Cooperstock, J., Roston, J., Martens, W. (2004) *Environment for immersive multi-sensory communication of music using broadband networks.* 23rd Tonmeistertagung VDT International Audio Convention, Leipzig, November 5–8.

[15] Wright, M., Freed A., and Momeni A. (2003). *OpenSound Control: State of the Art 2003*, Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03), Montral, Canada, 154–159.