COSM: A TOOLKIT FOR COMPOSING IMMERSIVE AUDIO-VISUAL WORLDS OF AGENCY AND AUTONOMY

Graham Wakefield, Wesley Smith

University of California Santa Barbara
Media Arts and Technology, USA
wakefield|whsmith@mat.ucsb.edu

ABSTRACT

Spatial music can create immersive experiences of alternate realities. To what degree can this be expanded to the composition of immersive, navigable, audio-visual worlds? We present an integrated collection of extensions to the Max/MSP/Jitter environment to assist the tightly integrated construction of such worlds, designed for use within audio-visual virtual environments and other projects using spatial simulation. The perspective taken is that spatial composition can go beyond the specification of locations and trajectories to the composition of autonomy, agency and interactions.

1. INTRODUCTION

The primary motivation for the *Cosm* toolkit is the creation of immersive, explorable three-dimensional worlds as a unified form of composition.[1] It intimately relates the spatialization of electronic music with virtual environments and computational simulations, and has been used for CAVE™-like systems (Figure 1), audio-visual performance, generative artworks, non-real time composition, speculative research and the prototyping of physical installations. Although it crosses many disciplines, our narrative in this paper is biased toward computer music: unraveling the motivation of *composing worlds* by extrapolating from spatio-musical composition to the stranger ontologies of generative worlds.

1.1. From composing spatial music...

Spatial electronic music was pioneered by composers such as Edgard Varèse, Karlheinz Stockhausen and Iannis Xenakis, and is now a core component of compositional practice with standardized techniques [3, 2]. Spatial composition depends on specifying spatial properties of sounds, such as placing sounds in individual loudspeakers or virtual point-like locations between them, moving them along trajectories over time, and spreading them into diffuse and enveloping sonic images. Spatial elements interact with other parameters of music (and other media when present), and together with expectations and memories construct a complex whole experience.

[1] Cosm originated in the development of inaugural content [24] for the AlloSphere at the University of California Santa Barbara [1].

Figure 1. *Cosm* used in an immersive CAVE™ environment: the Banff New Media Institute Visualization Lab.

Techniques may rely on the simulation of spatial sound-fields: (re)constructing artificial sound-worlds within the physical spaces in which we listen to them. The simulation of sound-fields may be based on acoustical properties of sound propagation (such as the speed of sound and reverberation) and psycho-physical properties of human audition (such as localization by inter-aural time differences).

While spatialization can contribute to a greater feeling of immersion within an artifical world, it is important to distinguish the illusion of spatiality in general from the illusion of spaces or places in particular [3]. This marks the beginning of the composition of the elements that make up a world.

Composed worlds need not be purely mimetic: composition frequently relinquishes accurate reproduction to explore more interesting and unusual sound-worlds; ‘breaking the physicality’ to create disjunct and contradictory images or spatial impressions that do not exist in nature. In Emmerson’s words: “From the direct imitation – the acoustics of a forest or mountain landscape – through vague impressions to more remote evocations of colour and texture. In addition we can create imaginary landscapes – the ‘mindscapes’ of expressionism. These ideas may overlap – sometimes reinforcing, sometimes contradicting each other – within the same transformation. Sound artists have just the same possibilities of surrealism, dreams and fantasies as any Dali or de Chirico.”[10]

1.2. ...to strange ontologies

It seems fair to consider therefore to what degree this can be extrapolated to the composition of immersive *audio-visual* worlds. Although integrated audio-visual composition may immediately suggest precedents such as Wagner's notion of the *Gesamtkunstwerk [18]*, the convergence of media and algorithm afforded by computation today can present us experiences of a different kind: spaces and worlds of autonomy and exploration.

The potential is clearly stated by Novak: "If music is a landscape then it is possible to extract as many types of conventional music as there are trajectories through that landscape... Navigable music is not an organization of sounds in time, it is the organization of a matrix of sonic, visual, behavioral, and other possibilities." [16] This of course encompasses navigable scores (such as the topophonies described in [8]). However the intention is something closer to a fusion of Xenakis' conception of composer as "a sort of pilot [who] supervises the controls of a cosmic vessel sailing in the space of sound, across sonic constellations and galaxies that he could formerly glimpse only as a distant dream" [29], with his concept of the *Polytope*, "a domain of spatial complexity that may be articulated by sound and light in movement." [11]

There has never been a better time to explore alternate modes of spatial, mobile, audio-visual complexity. Composers today can draw upon models of complex systems and use their nonlinear behaviors as generative devices to construct self-organizing, explorable worlds. The use of swarming and flocking to control spatial sound for example is well established [4]. Whitelaw offers the term 'strange ontologies' to describe works utilizing models such as multi-agent systems to create worlds that present alternative models of being-in-time; relationships and morphologies of individual and environment. "Especially in works using simulation and related techniques, abstract generative art performs *cosmogeny*: it brings forth a whole artificial world, saying, here is my world, and here's how it works." [26]

The goal of the Cosm toolkit is to assist the construction, composition and exploration of such worlds.

2. SYSTEM DESIGN

Our implementation integrates with and extends the Max/MSP/Jitter [30] development environment with features required for composing worlds; re-using existing components and protocols wherever possible to better leverage the vast range of existing synthesis & algorithmic capabilities as well as the mind-share of users. The toolkit provides individual elements to build a spatial world, rather than a single unified model: it thus remains open for those rules to be broken or exceptions to be easily made at many different layers of granularity.

Hendricks et. al. observe that creating a virtual world consists of two distinguishable steps – modelling the environment (populating the static components) and specifying dynamic interactions and behavior between objects and their surroundings [12]. In our view composing spatial worlds means attaching static elements and dynamic processes to local and global spatialities. Specifically, through *Cosm* objects, data-flows in Max can be given semantics that are global (operating on the scale of the world) or local (as situated points within the world). Beyond the powerful support for interactivity and process interaction already available in Max, the toolkit adds an *agent-driven architecture* [21] in which human observers/operators can be tightly linked into the behaviors of agents and environmental processes.

2.1. Environment, Observer, and Agents

In *cosm,* spatial objects are situated in a world of known dimensions and structure: a bounded Euclidian space which can be navigated by the observer. The properties of this world are shared between all *cosm* objects, providing coherency as well as neighbourhood services; many of these properties are accessible to the user via the *cosm.world* object.

The navigating observer is represented as a mobile oriented point (or local coordinate frame) constructed from a position-vector and orientation-quaternion[2], accessible via the *cosm.master* object. Both audio and graphical rendering is calculated relative to this coordinate frame. In addition *cosm.master* contains methods and attributes for specifying linear and angular velocities, to continuously translate and rotate itself through space.

Data-flows are made spatially local through the *cosm.nav* object, which use the same coordinate frame implementation and interface as *cosm.master*. However the output of a *cosm.nav* object can be directly attached to any 3D graphical *(jit.gl.*)* object to co-locate them in the world, and/or to spatialize any MSP audio signal via the *cosm.audio~* object.

For both *cosm.master* observer and *cosm.nav* agents, the specification of trajectories is relative to the moving object itself, rather than to the static environment. This is a departure from the specification of trajectories typical in computer music; it reflects a shift of emphasis from the subjective center toward an objective autonomy appropriate to the composition of worlds.

2.2. Agent interaction and communication

Motions become more complex and interesting if objects become sensitive to each other, the observer, and variations in the environment. All *cosm.nav* objects can be named, and their spatial properties can be queried and modified though the *cosm.query* object. The observer is also available under the reserved name "camera". This allows direct communication between agents. In addition *cosm.nav* objects can report the names and distances of their nearest neighbors, through a form of collision detection based on spherical intersections [6]. The basis vectors (e.g. right, up, forward) are directly accessible, permitting

[2]The toolkit provides helper objects to convert between quaternion, axis/angle, Euler and basis vector representations of orientations.

the composition of compound spatial objects and dataflows; and thus spatial interactions such as paired sensors and actuators (see Figure 2).

Agent-environment interactions are described in section 2.4.

2.3. Spatial audio pipeline

A *cosm.audio~* associated with a *cosm.nav* generates azimuth, elevation and distance values for the object, calculated according to the object and observer locations and orientations in the world. All relative positions and orientations are updated at frame-rate. The calculations of azimuth, elevation and distance (and thus localization filters) however are updated at audio block-rate, with smoothing filters to prevent audible discontinuities.

The *cosm.audio~* object also applies filtering to the audio signal for distance cues, following a pragmatic approach similar to [15]. The *cosm.world* has a set of defined attributes for affecting these distance cues, including *near-clip* and *far-clip* radii. Sounds within *near-clip* are unaffected, while sounds beyond *far-clip* are inaudible. Amplitude attenuation is approximated as $(1\text{-}d)^2$, where d is the normalized distance from zero (at or below *near-clip*) to one (at *far-clip*). Air absorption is simulated by a simple low-pass filter, whose cutoff frequency descends from the Nyquist frequency (at *near-clip*) to a fraction thereof (at *far-clip*) according to the absorption world attribute. Doppler shift is simulated by an interpolated delay-line; a global world attribute sets the maximum delay duration (the time it takes for a sound to travel over the *far-clip* distance), which can be derived as *farclip * speed_of_sound*.

This distance-filtered signal can then be fed to a spatial rendering technique to reproduce 3D orientation simulation. The *Cosm* toolkit supplies an implementation using higher-order Ambisonics based on prior work [25]. Encoded signals for each sounding object in the world are mixed into a shared multi-channel sound-field bus; these signals are passed to the world's decoder(s) to reproduce the sound-field in real-space, configured according to the actual speaker layout in use.

2.3.1. Open-ness

Modulating the global rendering parameters of the audio system allows some exploration of less realistic spaces. Furthermore, by separating out tasks to individual objects connected through messaging, there is plenty of scope to explore both alternative rendering methods, as well as wildly stranger designs. The raw/filtered signals and controls can be routed to alternative ambisonic implementations ([23, 20]) or different spatial rendering techniques (HRTF, VBAP, DBAP, Wavefield etc.; for overviews see [2, 1]).

Since the *cosm.nav* reports orientation implementations that also support directed radiation patterns or incidence filters can be correctly integrated. The distance parameter can be used to apply alternative distance-

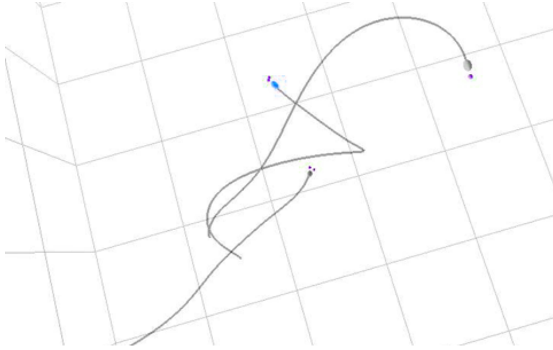

Figure 2. Braitenberg's Vehicles [5] translated to 3D.

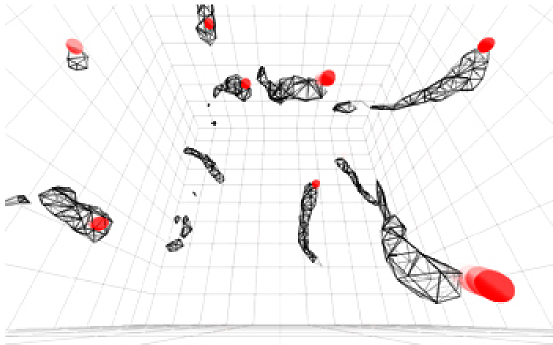

Figure 3. Isosurface rendering of agent pheromone trails.

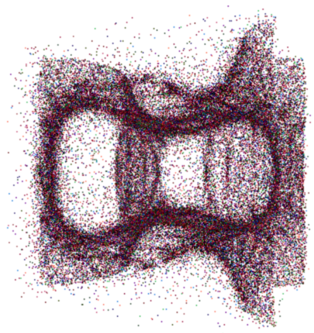

Figure 4. A three-tone Chladni pattern in zero-gravity.

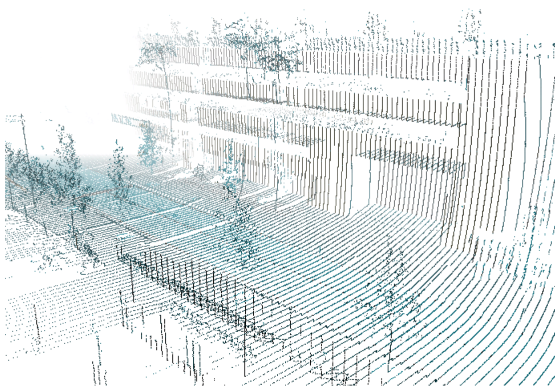

Figure 5. Importing external data: here LIDAR captured city-scape. Data courtesy of the House of Cards project shared under the Creative Commons CC BY-NC-SA 3.0 license, copyright 2008 Radiohead.

cue transformations to the signals, bypassing the built-in pipeline. Likewise, the simple reverberator provided can easily be replaced by more powerful and project-specific room-response simulations.

2.4. Fields

There are properties of worlds that can be modeled as quantities or intensities which vary across space and time, but have a momentary value at any point within this space, such as temperature, density, concentration etc. Such continuous fields can be approximated by discretization of the volume into grids of voxels, associating a scalar value with each. In Cosm we use 3D Jitter matrices to represent these fields[3]. Time-varying fields, such as the products of dynamical field equations, are simulated by operating on the 3D matrix in one or more feedback loops. By using the generic Jitter matrix data-structure, fields can be generated and processed with the vast set of existing built-in and third-party objects and device drivers.

Sonifying or visualizing 3D fields poses a challenge however: how to hear and see the variations of a value that are present in all points of the world? Visual volume rendering is often achieved in Jitter using isosurfaces[4] via the *jit.gl.isosurface* object. Fields can also be visualized, as well as influenced, by large quantities of homogenous point collections (partlcles represented by Jitter matrices) via the *cosm.field* object, with efficient 3D interpolated reading and writing (see Figure 4). We are currently investigating the sonification of fields through stochastic clouds of particles centered on the observer[5].

2.4.1. *Diffusion*

If we drop a block of sugar into a glass of water, initially the concentration will be high near the base of the glass, where the cube landed, but over time the concentration will diffuse until it is roughly the same value throughout the entire glass. Diffusion is a vital environmental process for many biologically inspired models such as dissipative structures, reaction-diffusion systems, and stigmergy. Diffusion cannot be efficiently with built-in Max objects, but can be achieved in 2D and 3D with *cosm.diffuse* (see Figure 6). We are also currently developing a *cosm.advect* object to model stable fluids.

2.4.2. *Agent-field interaction*

Through the *cosm.field.query* object, a world position (such as the output of *cosm.nav*) can be used to both read information from and write information to a named Jitter matrix[6]. 3D matrix values will be linearly interpolated; writing values into the matrix also distributes ('splats') to nearest neighbors using linear interpolation.

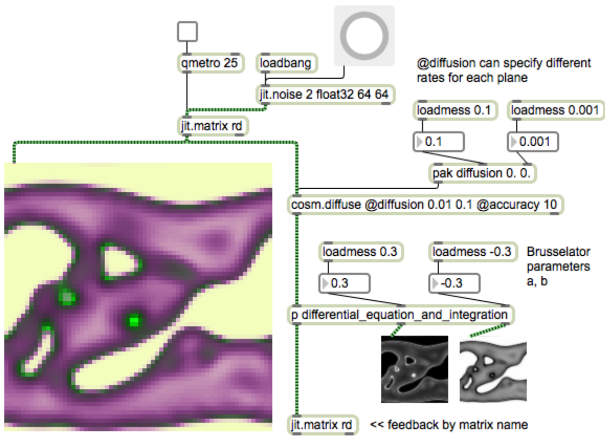

Figure 6. Brusselator [27] Reaction-Diffusion simulation (showing 2D case for legibility).

This allows the transfer of information between local and global dynamics; a 'mediated' form of interaction affording the construction of complex spatial behaviors including biologically inspired processes such as chemotaxis and stigmergy (see Figure 3), especially with larger populations. Global properties can be used to interactively perform the world: for example, increasing diffusion allows faster detection over a distance, while reducing diffusion promotes spatial variegation, increasing the ability of agents to leave 'signals' behind in space to be discovered.

Since the navigating observer is also an agent in the world, local field intensities can be used to modulate global rendering: for example, sensing camera-local field intensity to determine whether we are 'inside' or 'outside' a dynamic isosurface, and setting different reverberation parameters accordingly.

2.5. Stereographic visualization

Jitter is capable of both active and passive stereographic rendering, but managing the message sequencing to the various objects involved is tricky and error prone. *Cosm* simplifies this with the *cosm.render* object, which transparently handles the sequencing of messages to the visual rendering pipeline.

2.6. Remote rendering

For installations with large numbers of loudspeakers or screens, rendering can be distributed across multiple machines using a simple star archictecture: a central controller distributes updates to multiple render nodes. The audio rendering can be distributed by sharing the compact ambisonic domain signals with multiple decoders for different speaker configurations; the visual rendering can

[3] Multi-plane matrices can be use to store multiple values for each cell, such as modeling the concentration of several different chemicals. The dimensions of the field matrix do not need to match those of the world; objects such as *cosm.field.query* will scale coordinates automatically. Computing dynamics over 3D matrices can become rapidly CPU intensive, so tuning the dimensions of the field matrix can be important to maintain efficientcy.

[4] An isosurface is a 3D contour representing points of a constant value, which thus determines a surface.

[5] An interesting approach is detailed in [13]

[6] Interactions with multi-plane matrices will accept and return multiple values (lists).

be split between *cosm.master* and *cosm.render* with per-projector rotations applied to the observer viewpoint.

3. COSM IN USE

3.1. Spatial composition

Emmerson summarizes the qualities contributing to the total space created as “objects-motion-environment” [10], and there is nothing in this set of elements that cannot equally apply to the composition of immersive audio-visual worlds.

3.1.1. Objects

The tight integration of sonic, visual and spatial processing has led to some novel possibilities, correlating visual and spatial form. Perturbing geometry according to the spectra of source signals is a commonplace technique, but the sonic signal itself can be used to contruct visual form, such as lissajous and harmonograph figures, while variations of spatial form can be utilized as control data for sonic synthesis. Driving both sonic and visual qualities from a combined process can result in greater perceptual fusion, particularly if there are variations at sub-audio rates.

However whether a realistic, non-realistic or confusing spatial audio-visual perception is intended, composers should be aware of the different perceptual context that the spatial audio-visual world metaphor imposes. Varying an object’s visual size, or its sonic amplitude for example conflicts with perceptual cues of distance. Likewise, frequency changes can conflict with Doppler effect, and phase modulations with spatial positioning cues; as such the incorporation of delay-based effects in the source sound may significantly confuse the spatial illusion. In a similar way, visual brightness can conflict with distance cues when simulating fog effects On the other hand, sounds rich in harmonics and textural variation or regular transients give better perceptual positioning.

3.1.2. Motion

Beyond the textural qualities of sound and image there is great potential in the composition of motions and behaviors. Motion has often been suggested as a vital perceptual link in the perceptual fusion of sound and image (for example [9] and [14]). The integration of sonic envelope and spatial movement in particular can be striking: oscillators and triggers can drive accelerations and angular velocities to create complex and interesting trajectories. A simple example (included in the distribution) makes use of the amplitude of sound-objects, each pulsing at different divisions of a basic meter, to drive oscillating movements through the space. The result is a kind of navigable phase-music, in which the oscillating doppler shifts contribute an important component to the sonic result.

Combinations of sporadic, discrete changes of speed and orientation (brownian motions, pulsing, aperiodic bursts) with more continuous movements generates a remarkably convincing sense of autonomy and *animacy*. Behavior conveys yet more agency if agents reveal interrelationships. A case in point is the extension of Braitenberg’s famous Vehicles experiments [5] into 3D. In Figure 2 two vehicle-agents are strongly and competitively attracted to a third, which has a more exploratory tendency; the result is quite literally a swooping life-like dance. This qualitative autonomy can easily be integrated with interactions with the navigating observer.

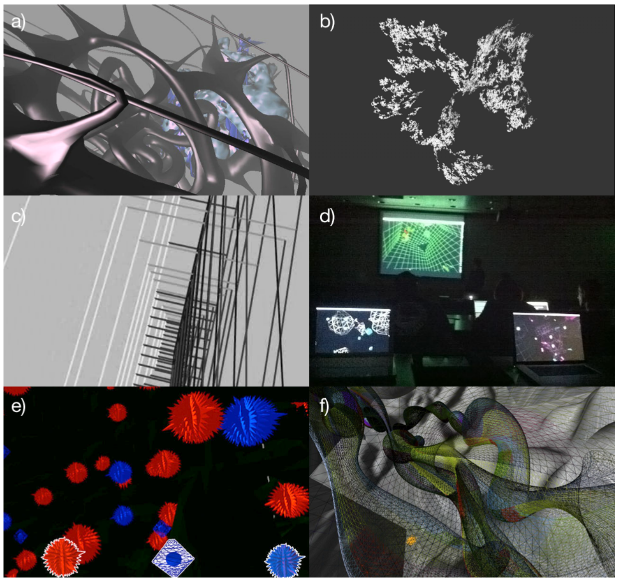

Figure 7. a) *Cosmoses07* (Frederico Fialho); Masters thesis project. b) *The Money Changing Machine* (Dennis Adderton); visualizing and sonifying currency exchange rates. c) *Enso* (Aaron McLeran); generative audio-visual composition based on a rhythmically broadcast stochastic event structure. d) Workshop in which participants construct and populate a world with bacteria whose swimming and tumbling behaviours follow sugar concentration gradients (chemotaxis). e) and f) student work from the Hyperbody graduate program at TU Delft, Holland.

3.1.3. Environment

Richer worlds derive from sense of ’place’ above ’space’. That is, different regions of the world exhibit distinct properties, forms and populations of behaviors, with attendent sound-fields, tapping into the important interplay between memory, anticipation and exploration. Importing external data sets or executing generative algorithms can be a simple way to create the image of a world (Figure 5), but integrating such data within the interactions of agents leads to more deeply engaging results.

Emmerson noted that the perception of spatial music “is determined both by our expectations of ‘real’ behaviour learnt over the considerable period of our evolution and by our personal experience.” [10] Similarly the composed integration of sound, image, space and time of a world creates a whole that transcends combination.

3.2. Education

The toolkit has been succesfully used as a significant component of gradate-level courses in both media arts and architecture programs including UC Santa Barbara [7], the Southern California Institute for Architecture[8] and TU Delft [9], as well as various individual workshops, with varying emphases of world-making, the composition of immersive environments and multi-dimensional mediascapes (see Figure 7).

3.3. Physical Installations

Cosm has also been used for a diversity of installations and performances outside of immersive facilities (see Figures 8 and 9). Cosm has been embedded as a crucial component within the Future Cinema Lab's CAVE™ and Augmented Reality framework (including integration with *Intersense* optical tracking systems), developed by Andrew Roth at the Augmented Reality Lab at York University, and used at the Banff New Media Institute (see Figure 1)[10]. It has also been connected to optical trackers such as the *OptiTrack* system through VRPN[11] (see Figure 10).

Figure 8. *Integration02* (Dieter Vandoren with Mark-David Hosale). The toolkit was used to simulate, prototype and then perform an audiovisual composition fusing choreographic live performance with an electronically controled light system and spatialized audio.

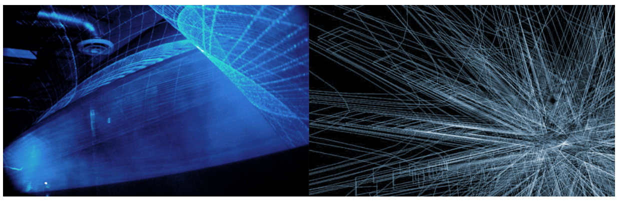

Figure 9. *Alternity* (F. Myles Sciotto). The toolkit was used along with EEG sensors to drive harmonic audiovisual patterns (right) spatialized and projected through a layered construction of mesh screens and colored haze (left), as an experiment in architecture's capacity to produce alternative forms of experience.

[7] http://translab.mat.ucsb.edu, http://mat.ucsb.edu/594P/

[8] www.sciarc.edu

[9] http://www.hyperbody.nl/

[10] http://futurecinema.ca/arlab/

[11] Virtual Reality Peripheral Network, an open standard used by many tracker systems.

Figure 10. *Turbulent Topologies* (Marcos Novak). Trackers are used by visitors as 3D cursors to reveal sounds from an invisible sculpture in the shared physical space of the exhibition.

4. RELATED WORK

4.1. Gaming and virtual reality engines

Most gaming engines provide facilities to play back sound files, usually with some (often unspecified) spatialization algorithms. There have been many cases of existing 3D virtual reality engines and game engines being connected to dedicated audio-processing software for more interesting synthesis and tighter control over spatialization. The interconnection is usually via some form of network messaging.

The Blender game engine for example has been connected to CLAM via OSC [17], the Ygdrasil engine has been connected to SuperCollider [19], Max/MSP[12], and the SGI Bergen Sound Server, while the Avango engine connected to Spat tools in Max/MSP via UDP [15]. The Uni-Verse VR engine extends Eskil Steenberg's Verse format (compatible with Blender, OpenSceneGraph etc) and has involved spatial audio engines developed in PD[13] and Max/MSP[14]. The SPIN framework[15] in use at SAT in Montreal is a C++ framework also based around OpenSceneGraph, which can communicate via OSC to sonic spatializers in Max/PD, SuperCollider, including the Audioscape project.

The myu project is notable: it provides a bridge between Max/MSP/Jitter and the respected Unity game engine [7]. Interaction between the two applications is via the network messaging, but in addition to basic control messages, textures generated in Jitter can be sent to Unity over TCP.

Pairing game engines and sound renderers is an effective strategy to achive richer spatial audio for virtual environments. However, the network-based coupling makes it difficult to share memory and resources, with limited bandwidth and latencies in the relationships between dif-

[12] http://www.alexshill.com/projects/spatialosc.htm

[13] http://wiki.blender.org/index.php/Extensions:Uni-Verse/General_overview_of_the_audio_system

[14] https://trac.v2.nl/wiki/btr/SpatialMixer

[15] http://spinframework.org/

ferent media and processes. Since our toolkit is built entirely within a single environment, it can support shared memory and tighter feedback loops between modalities, objects and processes, with higher bandwidth and lower latencies.

Our toolkit cannot provide all the powerful features of a full fledged virtual reality or game engine. The *Cosm* toolkit eschews the characteristic hierarchical scene-based approach of game engines for a more open-ended canvas of spatial data-flows. By using the same protocols and interfaces as the many other audio-visual media processing capabilties that the Max environment supports, we can relax the a priori design of the world, handing over much greater computational control to the composer.

4.2. Audioscape

The Audioscape [28] engine provides a novel approach to spatial audio, as a graph of sinks and sources; thus (unlike our toolkit) there can be multiple observers, and chains of sonic processing distributed across a world. The reproduction of spatial audio features is highly detailed, including cardiod patterns for the directivity of emission and reception of sound and various distance-based filters. It is implemented in Pure Data [22], and embeds Open Scene Graph for the 3D visualization of spatial objects.

Supporting non-mimetic aspects of composition ("rule-bending") is also considered vital: "Realistic audio propagation is only a starting point, and users can choose to depart from that as desired." However it is important to note that the primary motivations were the *visualization* of a sonic, spatial scenes "for audio engineering, musical creation, listening, and performance", rather than construction of a complex, integrated audio-visual world as such.

4.3. Other alternatives

The success of artificial life and complex systems has led to some general tools for the construction of simulations. Few of these tools are suitable for real-time multimedia, while those that are (such as StarLOGO TNG, Framsticks and Breve) usually have only primitive audio capabilities.

Fluxus is described as "A 3D game engine for live-coding worlds into existence."[16]; it is a powerful and very flexible run-time Scheme-based environment which includes many features well-suited to constructing algorithmic and visual worlds, however the audio capabilities are comparatively limited.

At the other extreme, composers can use purely off-line tools for developing worlds in compiled languages of C++ and Java, and which can be tightly integrated with existing spatialization libraries. The lower-level language and preclusion of live-editing however is higher barrier to experimentation.

[16] http://www.pawfal.org/fluxus/

5. CONCLUSIONS & FUTURE WORK

We have described a development environment in which the fields of computer music, virtual worlds, and complex simulations can be deeply and tightly interwoven. As described in Section 2 and shown in the examples in Section 3, the toolkit engenders an approach to spatial composition that emphasizes autonomy and agency. Composers can draw upon models and simulations of complex systems that were previously unavailable, present them immersively, both sonically and visually, and interact with them in real-time: navigating through self-organizing worlds. Since it integrates seamlessly with the Max environment and its existing message, signal, and data types, only a few primitive objects are needed to produce the wide range of examples previously described.

We have already mentioned several areas of current development, including sounds with distinct radiation patterns, sonification via stochastic particle clouds, and the simulation of vector field advection. A facility for multiple listening points is also being considered.

One constraint in the current version of *Cosm* is that the shape of the world is limited to a finite 3D Euclidean space with toroidal boundary conditions. We would prefer to support worlds extending infinitely in all directions; however the implementation would place additional cognitive burden on composers. In a similar vein, the emphasis on autonomy and agency in *Cosm* points toward the idea of the dynamic generation of audio-visual processes, which can only currently be realized in a limited manner within Max. Pursuing autonomous open-endedness pushes against the limits of what the Max environment can currently provide, however the cost of abandoning integration within a well-established and highly flexible media-processing environment (which was certainly an advantage for eductional use) is very high.

Getting Cosm

Cosm is in the public domain and available for Mac OSX and Windows from www.allosphere.ucsb.edu/cosm. The shared-memory requirement of the world model implies a more complex installation process, however automatic installers are provided.

Acknowledgements

This work would not have been possible without the inspirational drive of Professors Marcos Novak and JoAnn Kuchera-Morin. Research funding support was gratefully received from the AlloSphere Research Group. Parts of the audio processing derive from Lance Putnam's Gamma[17] library.

[17] http://mat.ucsb.edu/gamma/

6. REFERENCES

[1] X. Amatriain, J. Castellanos, T. Hollerer, J. Kuchera-Morin, S. Pope, G. Wakefield, and W. Wolcott, "Experiencing audio and music in a fully immersive environment," *Computer music modeling and retrieval: sense of sounds: 4th International Symposium, CMMR 2007, Copenhagen, Denmark, August 27-31, 2007: revised papers*, p. 380, 2008.

[2] M. A. Baalman, "Spatial composition techniques and sound spatialisation technologies," *Org. Sound*, vol. 15, no. 03, pp. 209–218, Dec 2010.

[3] N. Barrett, "Spatio-musical composition strategies," *Org. Sound*, vol. 7, no. 3, pp. 313–323, Dec 2002.

[4] T. M. Blackwell, "Swarm Music: improvised music with multi-swarms," in *Proceedings of the 2003 AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, 2003, pp. 41–49.

[5] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, Feb 1986.

[6] N. Brodu, "Query sphere indexing for neighborhood requests," *Journal of Graphics, GPU, & Game Tools*, vol. 13, no. 3, pp. 35–51, 2008.

[7] I. I. Bukvic and J.-S. Kim, "myu max-unity3d interoperability toolkit," pp. 1–4, Jan 2011.

[8] R. Cahen, "Sound design for navigation in topophonies," *Proceedings of the 14th International Conference on Auditiory Display*, 2008.

[9] M. Chion, *Audio-Vision*. Columbia University Press, April 1994.

[10] S. Emmerson, "Aural landscape: musical space," *Org. Sound*, Jan 1998.

[11] M. Harley, "Music of sound and light: Xenakis's polytopes," *Leonardo*, Jan 1998.

[12] Z. Hendricks, G. Marsden, and E. Blake, "A meta-authoring tool for specifying interactions in virtual reality environments," ... *on Computer graphics*, Jan 2003.

[13] E. Klein and O. Staadt, "Sonification of three-dimensional vector fields," *Proceedings of the SCS High Performance Computing Symposium*, Jan 2004.

[14] N. Moody, "Motion as the connection between audio and visuals," *niallmoody.com*.

[15] M. Naef, "Spatialized audio rendering for immersive virtual environments," pp. 1–8, May 2002.

[16] M. Novak, "Trans terra form," *arch.usyd.edu.au*. [Online]. Available: http://web.arch.usyd.edu.au/~petra/2006/DESC9181/Novak_TransTerraForm.pdf

[17] N. Olaiz, pau Arumi, T. Mateos, and D. Garcia, "3d-audio with clam and blender's game engine."

[18] R. Packer and K. Jordan, *Multimedia: from Wagner to Virtual Reality*. W. W. Norton and Company, 2001.

[19] D. Pape, J. Anstey, M. Dolinsky, and E. Dambik, "Ygdrasil—a framework for composing shared virtual worlds," *Future Generation Computer Systems*, Jan 2003.

[20] R. Penha, "Distance encoding in ambisonics using three angular coordinates," *SMC: Sound and Music Computing Conference*, 2008.

[21] D. Playne, A. Gerdelan, A. Leist, C. Scogings, and K. Hawick, "Simulation modelling and visualisation: Toolkits for building artificial worlds," *Research Letters in the Information and Mathematical Sciences*, vol. 12, pp. 25–50, 2008.

[22] M. Puckette, "Pure data: another integrated computer music environment," *Proceedings of the Second Intercollege Computer* ..., Jan 1996.

[23] J. Schacher and P. Kocher, "Ambisonics spatialization tools for max/msp," *Omni*, vol. 500, p. 1.

[24] G. Wakefield, D. Overholt, L. Putnam, and W. Smith, "The allobrain: an interactive stereographic, 3d audio immersive environment," *CHI Conference Workshop on Sonic Interaction Design*, Jan 2008.

[25] G. Wakefield, "Third-order ambisonic extensions for max/msp with musical applications," *Proceedings of the 2006 International Computer Music Conference. International Computer Music Association*, Mar 2006.

[26] M. Whitelaw, "System stories and model worlds: A critical approach to generative art," *Readme*, vol. 100, pp. 135–154, 2005.

[27] A. Winfree, "The prehistory of the belousov-zhabotinsky oscillator," *Journal of Chemical Education*, vol. 61, no. 8, p. 661, 1984.

[28] M. Wozniewski, "A framework for interactive three-dimensional sound and spatial audio processing in a virtual ...," *cim.mcgill.ca*, Jan 2006.

[29] I. Xenakis, *Formalized Music: Thought and Mathematics in Composition (Harmonologia Series, No 6)*, 2001.

[30] D. Zicarelli, "How i learned to love a program that does nothing," *Computer Music Journal*, Jan 2002.