About This Issue

The first article in this issue describes an award-winning new musical interface, the "Hyper Drumhead" by Victor Zappi, Andrew Allen, and Sidney Fels. The device's physical interface consists of a 42-inch transparent screen, onto which images are projected from below, and a MIDI foot controller. The screen is covered by a touch-sensitive film on which up to 100 simultaneous touches can be detected. Software allows the musician to divide the surface into arbitrarily shaped areas, each serving as an independent percussion instrument. Moreover, the player can make real-time changes, such as modifying the simulated material of which each virtual drumhead is composed. Sounds are generated via physical modeling—not by an approximation such as digital waveguide synthesis, but by direct real-time solution of the equation of two-dimensional wave propagation, using the finite-difference time-domain method. Although solving the wave equation can yield a variety of realistic as well as imaginative sounds, it is computationally expensive. The authors address this challenge by implementing the algorithm on parallel graphics processing units.

Whereas the Hyper Drumhead's sound synthesis uses physical modeling, the next article, by Janko Gravner and Kyle Johnson, presents a technique from the domain of abstract sound synthesis. In other words, rather than attempt to model the acoustics of a known or hypothetical instrument, the technique converts an arbitrary mathematical construct into sound. Within the vast universe of mathematical constructs, dynamical systems, particularly nonlinear ones, have proven to be especially alluring for computer musicians. See, for example, the *Computer Music Journal* articles by Pressing (*CMJ* 12:2), Dodge (12:3), Gogins (15:1), Bidlack (16:3), Monro (19:1), Slater (22:2), Rodet and Vergez (25:3), Röbel (25:2), Spasov (39:3), Stefanakis, Abel, and Bergner (39:3), and Rhys (40:3), among others. Gravner and Johnson's article in the present issue continues this tradition, introducing some sonic prospects of a category of dynamical systems known as coupled map lattices.

A variety of research projects these days involve computers that "listen" to real-time audio or MIDI input and respond to it using knowledge gained from training. The training may be done offline (i.e., using a separate corpus in batch mode ahead of the listening session) or online (i.e., using the real-time input to which the system is listening). The article by Stefano Fasciani and Lonce Wyse in our previous issue described one such project, whose goal was vocal control of real-time sound synthesis. The current issue's final two articles similarly discuss research that utilizes machine learning to react to musical input. In these articles, however, the goal is not for a human to control sound synthesis, but for the machine to improvise a musical performance. These two articles therefore reflect the theme annunciated on this issue's cover, "Machine Learning for Improvisation."

Michael Krzyzaniak presents his system that can discriminate between the timbres of several different drum strokes performed by a human on the djembe, a West African hand drum. His neural-net-based software uses online training to learn rhythmic patterns that incorporate these different strokes, and it produces musically plausible responses in the same style, performed by a robotic percussionist. The robot is not limited to literally mimicking previously heard patterns: It can adapt in real time to music that contains no repeated patterns. In other words, it improvises based on the current musical context. Krzyzaniak's article also elucidates how to handle the challenges of timing delays inherent in such a system—such as audio buffering, analysis time, and the robot's mechanical delay—as well as delays resulting from musical criteria such as the need to play on a beat or a stylistically plausible fraction thereof.

The article by Ken Déguernel, Emmanuel Vincent, and Gérard Assayag likewise examines machine improvisation. These authors are interested more generally in "multidimensional" improvisations, where "dimensions" can refer to musical features such as rhythm and timbre (as in Krzyzaniak's article), but also melody and harmony. In one probabilistic model that the authors describe, messages are passed between software agents, so that each can decide its next step in the improvisation based on its belief about the other agents' intentions. The goal is to achieve vertically coherent co-improvisations without the need for a priori conventions or hierarchy. The model can incorporate both offline training, prior to a performance, and online training, in which the system can "listen" to (for example) a human musician while it is improvising. (In

doi:10.1162/COMJ_e_00464

Front cover. Two diagrams from the article by Michael Krzyzaniak, illustrating the online training of a robot musician that improvises while "listening" to a human percussionist.

Back cover. A musician plays the Hyper Drumhead. (From the article by Victor Zappi, Andrew Allen, and Sidney Fels.)

the case of this research, the model was trained and tested with input in the form of symbolic musical data, rather than with audio input as in Krzyzaniak's work.) As a case study, the authors trained their system offline on a corpus of 40 improvisations by jazz saxophonist Charlie Parker. The system then generated duets consisting of a melody with a chordal accompaniment. Several experiments were run to investigate different capabilities, and three professional jazz musicians evaluated the software's musical output.