

BEAT DETECTION ANIMATING VIRTUAL ENVIRONMENT MODELS

Prabath Weerasinghe & Michael Cohen
 Spatial Media Group, University of Aizu
 Aizu-Wakamatsu, Fukushima-ken 965-8580
 email: {m5141110, mcohen}@u-aizu.ac.jp

ABSTRACT

Alice is an innovative 3D programming environment that makes it easy to create an animation. Many virtual environment (VE) models are available in the Alice 3D environment. We created VE scenes using the Alice 3D IDE (integrated development environment). We deploy a beat detector to detect the rhythm of a song, based on PD (Pure Data, a free dataflow programming environment similar to MAX/MSP). It can extract the beat of a song in realtime. Using our Alice-CVE (Collaborative Virtual Environment) Bridge and CVE-PD Bridge, we can create a communication link between the beat detector and Alice 3D environment. The CVE is a Java client-server protocol in which clients connect to session server host via channels, and when clients need to communicate with each other they subscribe to the same channel. The Alice-CVE Bridge allows any device that can connect to the CVE server to communicate with Alice without regard for architectural differences. When a song is played, its beat can be detected and data sent to animate objects and avatars. An avatar can dance (admittedly poorly) in the Alice 3D environment while receiving a realtime rhythm data stream from a CVE session server. Stage color and other attributes a can also be rhythmically changed according to the beat using the same communication link.

1. INTRODUCTION

In music, the beat is the basic unit of time. Beat can refer to a variety of related concepts such as tempo, meter, rhythm, and groove. In hip hop and R&B music, the term ‘beat’ commonly refers to the non-vocal layer of the song not including instrumentals. To allow unrestricted response to the rhythm of music, beat detection and phase-locking loop (PLL) are required. Alice 3D objects can be programmed using the NetBeans IDE. We can specify the event according to the rhythmic data. Alice 3D IDE connects to the CVE using the Alice-CVE Bridge, and the beat detector connects to the CVE using PD-CVE Bridge. After that, when we attach an audio source to the beat detector, the system achieves avatar animation in virtual environments.

2. IMPLEMENTATION AND DATA FLOW

Modern beat detection algorithms, while not perfect, do a fairly good job [1]. (The algorithm by Goto [2] seems

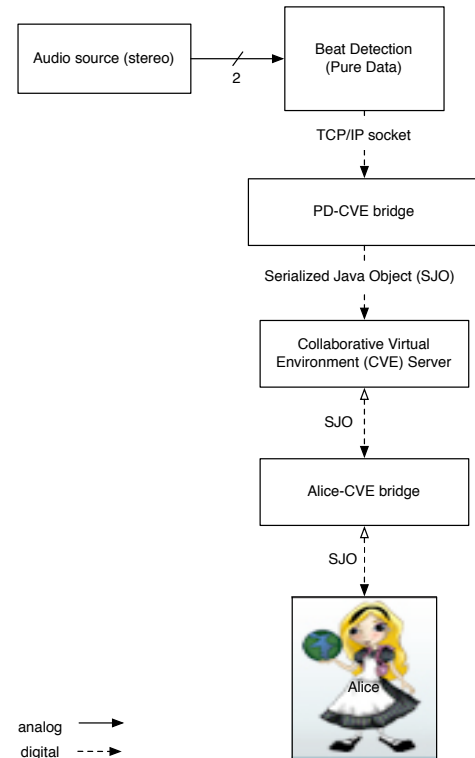


Figure 1. System schematic and data flow.

to be especially robust.) We programmed a beat detector using Pure Data (PD) [6], a dataflow development environment similar to Max/MSP. We use the aubiotempo [sic] patch to extract the beat, which is based on aubio [sic again], a C library for audio labelling created by Paul Brossier [7]. (The aubio misspelling is deliberate, intended to emphasize imperfection.) The library is fast enough for online deployment. Tone onsets are detected using a sophisticated phase vocoder mechanism. Detected onsets are used to calculate the tata (shortest time intervals between onsets), tacti (beat periods), and measures (time signatures of the music). A tatus is usually a multiple of a tatum, and in Brossier’s library is calculated with an autocorrelation function. The system is fast enough to track tempo variations in realtime: there is some jitter, and occasional missing beats, but it is robust enough to track most popular songs. As illustrated by Figure 1, any music player (such as an iPod or modern mobile phone) serves as an audio source. Such generalized stereo sig-



Figure 2. Virtual dancer: Beat Detection Animating Virtual Environment Models.

nals can be captured by an audio port. Alternatively, one can simply just sing into a live microphone (background noise, ambient hum, static, and the fact that an a cappella voice doesn't have a rhythm section with a distinct pulse, the metalingual accents being not as pronounced as an explicit drum and bass, notwithstanding). PD communicates as a network stream using internet sockets (via null network at a loopback address) to communicate with the Collaborative Virtual Environment bridge, a servant (hybrid server/client) which acts as middleware [5]. This servant forwards the messages to the CVE server (as "ExtraParameter"s, an extensibility feature built into the CVE protocol that types name/value pairs as string and float, respectively), whence the events are multicast to other CVE client applications in a session.

2.1. Alice-CVE Bridge

Alice [4] is an innovative 3D programming environment with an integrated development environment (IDE) that makes it easy to create rapid prototypes, "machinema" (machine cinema), and animations. In Alice's interactive interface, one can drag and drop 3D objects from a gallery to a scene and graphic textual tiles to a coding workbench to create a program. Since Alice (v. 3) compiles into Java, it is possible to modify a scenario with the help of a NetBeans plugin. We use such capabilities to establish connection with a CVE server and convey rhythm via Alice avatars, as shown Figure 2.

3. CONCLUSION

We have deployed simple animation according to the beat of the music. Using this idea it is easy to create virtual dancers in a virtual environment [9]. Many aspects of a virtual environment can be modulated by externally derived beat, including location (heave, surge, sway), orientation (roll, pitch, yaw), and position of objects (including avatars), size, opacity (visibility, translucency, and transparency), color, and pose. By dividing a beat counter by a small integer such as 2, 3, or 4, the remainder can be used to parameterize the phase of a simple dance. After study-

ing this kind of virtual dancing module we can implement dancing training software.

4. REFERENCES

- [1] Shiu, Y. and Kuo, C., 2007. *On-Line Musical Beat Tracking with Phase-Locked-Loop (PLL) Technique*, Proc. ICCE: Int. Conf. on Consumer Electronics, Las Vegas. ISBN: 1-4244-0763-X.
- [2] Goto, M., 2001. *An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds*, J. of New Music Research, 30(2):159–171.
- [3] Pausch, R., 1995. "Alice: Rapid prototyping for virtual reality", *IEEE Computer Graphics and Applications*, 15:8–11.
- [4] Dann, W. P., Cooper, S., and Pausch, R. 2008. *Learning to Program with Alice*, Prentice Hall, second edition.
- [5] Konno, T., Cohen, M., Nagashima, Y., and Hoshino, T., 2001. "Mobile Control of Multimodal Groupware in a Distributed Virtual Environment", In Proc. ICAT: Int. Conf. on Artificial Reality and Tele-Existence, Susumu Tachi, Michitaka Hirose, Ryohei Nakatsu, and Haruo Takemura, editors, pages 147-154, Tokyo. ISSN 1345-1278.
- [6] Pd-community, 2009. *The Pure Data Portal*, ([Online; accessed 9- Sept.-2009]).
- [7] Brossier, P., 2006. *Automatic Annotation of Musical Audio for Interactive Applications*, Ph.D. Thesis, U. of London.
- [8] Cohen, M., Villegas, J., Ishikawa, M., Inoue, A., Sato, H., Tsubakihara, H., and Yamadera, J., 2009. "VMP My Ride: Windshield Wipers That Swing", In Proc. ASIA-GRAPH, pages 126–130.
- [9] Shiratori, T., Nakazawa, A., and Ikeuchi, K., 2006. "Dancing-to-Music Character Animation", In Proc. EUROGRAPHICS, E. Gröler and L. Szirmay-Kalos, editors.