2012 International Conference on Computer and Software Modeling (ICCSM 2012)
IPCSIT vol. 54 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V54.04

Dhvani : An Open Source Multi-touch Modular Synthesizer

Denny George [1], Binita Desai [2] and Keyur Sorathia [1] [+]

[1] Embedded Interation Lab, IIT Guwahati

[2] Dhirubhai Ambani Institute of Information and Communication Technology

Abstract. This document describes the design and development of Dhvani – a tangible and multi-touch modular music synthesizer developed through open source softwares and frameworks. Physical objects associated with a musical function are used to interact with an interactive table to create different soundscapes. Sound is generated using Oscillators and Sample players and is manipulated using filters or controlling parameters like frequency and amplitude. Interactions are designed through variety of means such as adding objects on an interactive table, rotating those objects and touch based interaction of graphical user interface (GUI). This paper carefully describes the design components, hardware and open source software components, libraries and framework used for the development of Dhvani.

Keywords: Electronic Music, Fiducial Tracking, Interaction Design, Tabletop User Interface, Sound Synthesis

1. Introduction

The initial experiments in sound synthesis using computers began in 1957, when researchers at Bell Telephone Laboratories in New Jersey proved that a computer could be used to synthesize sounds with time-varying frequency. The idea of Unit Generators was a seminal concept developed by one of their researchers Max Matthews. A Unit Generator is analogous to functions in computer programs. The input and output of Unit Generators can be either audio data or control data. Audio data is the data that generates sound and control data is used to control parameters like frequency or amplitude.

In 1919, with the introduction of Lumigraph, a notion of "visual music" was introduced. Such instruments control music production through a visual interface. Music mouse made in 1985, Instant music developed by EA arts in 1986 and Jam-O-Drum made in 1998 were a result of ongoing efforts to make music production an audio-visual endeavour.

Since then many tabletop musical instruments have been developed. Tabletop interfaces make the mapping of input and output very direct and provides for direct interaction with the system. Dhvani is a tangible multi-touch user interface where tangible objects with fiducial markers attached to their bottom surface represent components of a sound synthesizer. These blocks are placed on top of a translucent tabletop with a camera underneath the table. It captures them and these images are then processed to determine the orientation, identity and the location of the blocks. This information is sent to the sound processing unit of the Dhvani software via a network socket using the TUIO protocol which is a protocol layered on top of Open Sound Control which in turn is a protocol layered on top of the UDP protocol. The sound generating unit upon receiving this information processes it to generate sound depending on these parameters and the resulting soundscape is visualized on the screen. This one to one mapping between the input and the output makes the whole process, what electronic musicians have long referred to as – feeling alive. The open source development platform provides varied possibilities of the kind of music that can be made using Dhvani.

[+] Denny George. Tel.: + 91-9085509406; E-mail address: denny.george90@gmail.com
Binita Desai. Tel.: + 91-9328724664; E-mail address: binita_desai@daiict.ac.in
Keyur Sorathia. Tel.: + 91-9328724664; E-mail address: keyur@iitg.ernet.in

2. Motivation

The project is an outcome of a belief that digitizing a physical object democratizes it by definition. The existing products are either not available to hobbyist musicians and developers to tinker around with or they are closed source and available commercially at a very high price. The techniques to make a multi-touch table are freely available on the internet and if the software for such platforms were to be made public, it would not only encourage amateur musicians to try out the instrument without much financial investment, it would also give an opportunity to software engineers with an interest in computer music synthesis and digital signal processing to fork off from the basic code framework and add to its functionality. A basic skeletal framework that takes care of communication between the various modules and component of Dhvani has been made. Some modules which generate sounds, send control data to sound objects, filter them and control global parameters like volume have been developed. This project provides an abstraction to musicians and developers to make modules suited to their needs. The aim is not to make a product to compete with the existing commercial ones but to provide learning opportunities to musicians and developers which is facilitated by the spirit of tinkering of open source community and the user friendly and intuitive nature of a multi-touch interface.

3. Concept

Rear Diffused Illumination technique is used to make a multi-touch surface. An open source computer vision framework called ReacTIVision enables Dhvani to do robust and optimised tracking of multiple fingers and specially designed markers. 2D tracking of objects is used to obtain the object's position and orientation. The ReacTIVision server sends these data to a client using TUIO protocol over a network layer. The client then sends these data to the graphic synthesizer to generate visualization and the sound synthesizer to generate sounds. Dhvani lays down the framework for all these communications. Figure 1 showcases the hardware and software structure used in Dhvani. Developers only need to focus on developing modules suited to their sound needs. Five basic modules have been developed as part of Dhvani's core code.

- To play pre-recorded sound files.
- To generate audible mathematical functions like Sine wave, square wave, triangle wave etc.
- A low frequency oscillator to create a vibrato effect.
- A Ring Modulator to create a steely voice effect.
- A global object to control the global volume of Dhvani.

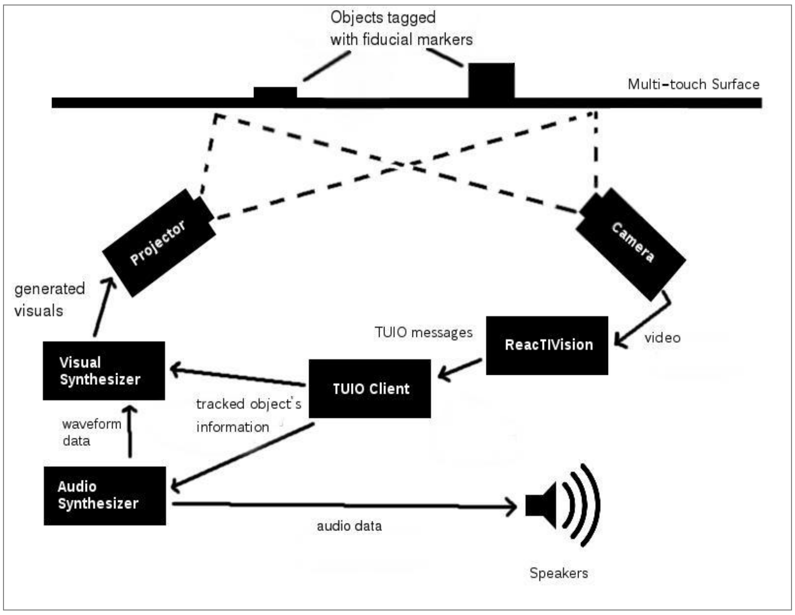

Fig. 1: Diagram showcasing hardware and software structure of Dhvani

4. Implementation

The project has the following components:

- Rear Diffused Illumination (DI) multi-touch table
- TUIO Client
- Audio Synthesizer
- Video Synthesizer
- ReacTIVision computer vision framework

All the components except the last one were built for the application.

4.1 ReacTIVision

ReacTIVision is an open source, cross-platform computer vision framework used for the fast and robust tracking for fiducial markers attached onto a physical object as well as for multi-touch finger tracking in a video stream. It was mainly designed for the rapid development of table based tangible user interface (TUI) and multi-touch interactive surfaces. ReacTIVision is a standalone application, which sends TUIO messages via UDP port 3333 to any TUIO enabled client application.

4.2 TUIO Client using TUIO C++ Developer API

A TUIO Client was made for Dhvani which would listen to messages from ReacTIVision and help coordinate the processing of audio and visuals for the project. The C++ API gives access to all the data that is being sent by ReacTIVision. Classes for Object and Cursor help access the parameters of objects and touch input. Functions are called by this the client whenever an object or cursor was added to, moved, rotated or removed from the table. All objects and cursors are identified with a unique session ID that is maintained over their lifetime. Each object also has a fiducial ID that corresponds to its attached fiducial marker number. Alternatively you can also poll current object and cursor state. These methods return individual or list of objects and cursor objects depending upon their presence on the table.

4.3 Audio Synthesizer

An open source C++ toolkit called openFrameworks was used to develop the application. OpenFrameworks has great support for playing audio and generating sound using the ofSoundStream class. The ofSoundStream class provides low level access to the sound card and hence time varying data can be sent to it and which can be heard as sound from an audio output device. The following classes were written as part of the project – Sample, WaveGenerator, RingModulator and LFO.

4.3.1 Sample

This class has functions to load, play, stop and pause a sound file of .wav format. It also generates a waveform corresponding to the amplitude values of the sound. This class is used at the back-end of the Sample Player object of the project. When it is placed on the table, a preloaded sound sample stored in the application's data folder is played in a loop. You have the option of changing the volume and the rate of speed at which the sound is played.

4.3.2 WaveGenerator

This class is used to generate waves of the following type – Sine, Sawtooth, Triangle, Square and white noise. Frequencies of discrete musical notes have been coded into the class and hence rotation of the Oscillator object runs through musical notes depending on the nature of rotation. Clockwise rotation increases the frequencies of the notes and Anti clockwise rotation decreases it. The volume of the wave and the type of the wave generated can be controlled by the user by either rotation the object or interacting with the UI on the table surface.

4.3.3 Low Frequency Oscillator

A Low Frequency Oscillation is an electronic signal which is generally a signal vibrating with a frequency less than 20Hz. It is used to create a rhythmic sweep effect. In the project it can be applied to a signal's amplitude to create a tremolo effect. Like the WaveGenerator class, the wave generated by a LFO can be set to any of the 4 types mentioned before.

4.3.4 Ring Modulator

This class implements the signal processing effect of Ring Modulation a computer music term for amplitude modulation. It involves multiplying two signals where one is typically a Sine wave. Our class has the option of setting a frequency between 80Hz to 2000Hz. Ring Modulation outputs a signal which has a sum and difference of the two signals. It is produced in electronic music to generate a rich sound with harmonics of the signal. The effect when applied to voice samples produces robotic sounding output.

4.3.5 Global Object

This module helps control the global parameters of Dhvani. As of now it only supports altering the global volume of Dhvani but it can be easily extended to features like tempo that are common to all objects in a song.

4.4 Graphics Synthesizer

OpenFrameworks uses OpenGL as its graphic library. OpenGL is a software interface to graphics hardware. It is designed as a streamlined, hardware-independent interface to be implemented on many hardware platforms. The synthesizer draws objects as filled circles at the position of the object and draws lines to indicate connections between compatible objects. Waveform or control data between objects is visualized as well.

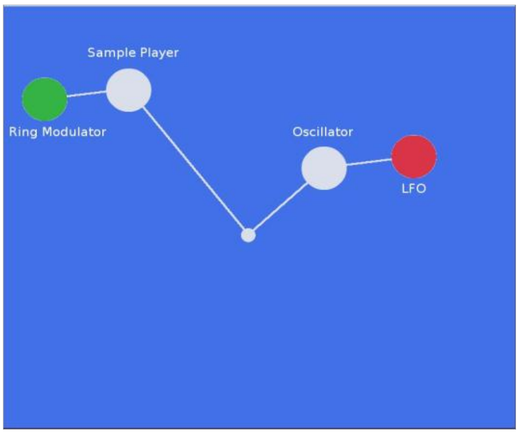

Fig. 2: User Inteface of Dhvani

4.5 Multi-touch Table

There are several ways to create a multi-touch surface – capacitive, resistive, and acoustic and many other non-traditional techniques but we chose to go with optical methods because of the ease and cost effectiveness considering an average person. Out of the four major Optical vision based techniques - Diffused Surface Illumination (DSI), Laser-Light Plane (LLP), Frustrated Internal Reflection (FTIR), and Diffused Illumination (DI), we chose Diffused Illumination. Even though each technique has its own pros and cons, DI fulfilled the requirements of the project of object and finger tracking.

The working of DI is explained below:

The table top is generally made of clear acrylic or glass. Some material like translucent paper is placed at the top or bottom of the glass to diffuse the incident light. Infrared light is shined on to the screen surface from the bottom and a IR camera is placed underneath to capture this image. A visible light filter is placed on the camera's lens to reduce confusion between the projected image and the object being tracked. When a finger is put on the surface it reflects more light than the rest of the material around it and hence is perceived by thecamera as round blobs. This way fiducial and finger blobs are captured by DI.

5. Challenges

Much of the challenge lies in building the hardware for Dhvani. IR LEDs suited for this application are difficult to obtain in India. A lot of hit and try methods are done to find the proper positioning and orientation of the IR LEDs to obtain an uniform illumination across the table surface. Since Dhvani takes care of various technical details behind the scene developers can move beyond them and get down to implementing different modules and features. A lot of optimized algorithms are freely available and all one needs to do is code them in C++ to incorporate them in Dhvani.

6. Conclusion

Connection between visual and audible performance can facilitate exploration learning of the abstract concept of music theory. Dhvani is intuitive for a novice users. The open nature of the various tools used to implement this project can facilitate development of collaborative musical tools on a multi-touch platform. Developers can add to the code and implement and check out new functionalities and hence give back to the open source community.

We would like to see this project being developed further by others and see how they mould it into musical instruments for different genres. Further development to facilitate input from MIDI controllers and musical instruments or microphones should be initiated.

7. Acknowledgement

We would like to thank all the members of the Natural User Interface and OpenFrameworks discussion forum for providing an active and rich source of information whenever we got stuck somewhere. We would also like to thank all those people in general who have shared their source code with the open source community as it helped a lot in the digital signal processing aspects of the project.

8. References

[1] Hochenbaum, J. and O. Vallis. Bricktable, “Musical Tangible Multi-Touch Interface,” in Proceedings of the Berlin Open. 2009. GermanyHochenbaum, J. and O. Vallis. Bricktable, “Musical Tangible Multi-Touch Interface,” in Proceedings of the Berlin Open. 2009. Germany

[2] Patten, J., Recht, B. and Ishii, H., Audiopad: A Tagbased Interface for Musical Performance. Proc. Conference on New Interface for Musical Expression(2002), 24-26.

[3] Sergi Jorda, Gunter geiger, Marcos Alonso, Martin Kaltenbrunner, “The Reactable : Exploring the synergy between live music performance and Tabletop Tangible Interface,” Proceedings of th e International Computer Music Conference (ICMC2005)

[4] C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan, “Direct-touch vs. mouse input for tabletop displays,”1997 , p. 647-656

[5] J. Schöning, P. Brandl, F. Daiber, F. Echtler, O. Hilliges, J. Hook, M. Löchtefeld, N. Motamedi, L. Muller, P. Olivier, T. Roth, and U. von Zadow, “Multi-Touch Surfaces: A Technical Guide,” Technical Report TUMI0833: Technical Reports of the Technical University of Munich, (2008).

[6] Davidson, P.L. and J.Y. Han, “Synthesis and control on large scale multi-touch sensing displays,” in Proceedings of the 2006 conference on New interfaces for musical expression. 2006, IRCAM Centre Pompidou: Paris, France.

[7] Kaltenbrunner, M. Bovermann, T. Bencina, R. Costanza, E, “TUIO - A Protocol for Table-Top Tangible User Interfaces,” Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation (GW 2005), Vannes (France)

[8] Joshua Noble, Programming Interactivity: A Designer's Guide to Processing, Arduino, and Openframeworks; O'Reilly, 2009

[9] Curtis Roads, The Computer Music Tutorial; The MIT Press, 1996

[10] TUIO C++ API - http://www.tuio.org/?cpp

[11] OpenFramworks documentation - http://www.openframeworks.cc/documentation/

[12] Natural User Interface Group - http://nuigroup.com