2-12-2014

The Penn Haptic Texture Toolkit for Modeling, Rendering, and Evaluating Haptic Virtual Textures

Heather Culbertson
University of Pennsylvania, hculb@seas.upenn.edu

Juan Jose Lopez Delgado
University of Pennsylvania, juanlop@sas.upenn.edu

Katherine J. Kuchenbecker
University of Pennsylvania, kuchenbe@seas.upenn.edu

Please fill in your information on this form to be contacted about updates or bugs.
Data files updated 12/17/2013.
Rendering files updated and support for Windows application added 2/12/2014.

The Penn Haptic Texture Toolkit for Modeling, Rendering, and Evaluating Haptic Virtual Textures

Abstract
The Penn Haptic Texture Toolkit (HaTT) is a collection of 100 haptic texture and friction models, the recorded data from which the models were made, images of the textures, and the code and methods necessary to render these textures using an impedance-type haptic device such as a SensAble Phantom Omni. This toolkit was developed to provide haptics researchers with a method by which to compare and validate their texture modeling and rendering methods. The included rendering code has the additional benefit of allowing others, both researchers and designers, to incorporate our textures into their virtual environments, which will lead to a richer experience for the user.

Keywords
haptic texture rendering, haptics, virtual reality

Disciplines
Applied Mechanics | Electro-Mechanical Systems | Engineering | Mechanical Engineering

Comments
Please fill in your information on this form to be contacted about updates or bugs.

Data files updated 12/17/2013.

Rendering files updated and support for Windows application added 2/12/2014.

The Penn Haptic Texture Toolkit for Modeling, Rendering, and Evaluating Haptic Virtual Textures

Heather Culbertson*, Juan José López Delgado†, and Katherine J. Kuchenbecker‡

Haptics Group, GRASP Laboratory
Mechanical Engineering and Applied Mechanics
University of Pennsylvania, USA

February 12, 2014

Abstract

The Penn Haptic Texture Toolkit (HaTT) is a collection of 100 haptic texture and friction models, the recorded data from which the models were made, images of the textures, and the code and methods necessary to render these textures using an impedance-type haptic device such as a SensAble Phantom Omni. This toolkit was developed to provide haptics researchers with a method by which to compare and validate their texture modeling and rendering methods. The included rendering code has the additional benefit of allowing others, both researchers and designers, to incorporate our textures into their virtual environments, which will lead to a richer experience for the user.

1 License

This tookit is made publicly available under copyright from the University of Pennsylvania. The toolkit may be used freely for non-commercial purposes such as research. You are free to alter, transform, or build upon the work included in this toolkit. However, if you use any of the included data, models, or rendering code, you must attribute them to the Penn Haptic Texture Toolkit [1]. Please see the attached license document for full copyright and permission information. Also see [1] for a full description of our recording, modeling, and rendering methods. The rendering code is distributed with permission from Geomagic, the maker of the Omni and OpenHaptics[1].

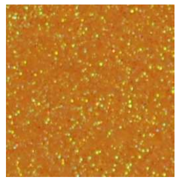
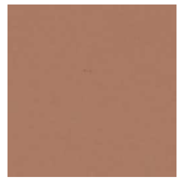
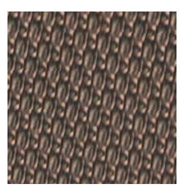
2 Texture Samples

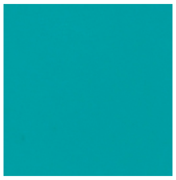
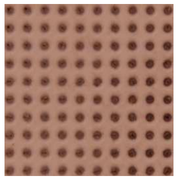
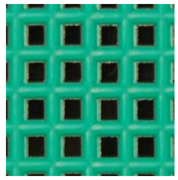
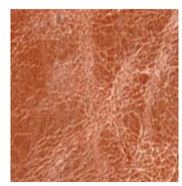
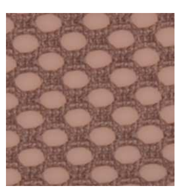
This toolkit includes information for 100 isotropic and homogeneous textures. As shown in Table 1, these textures are divided across ten material categories (paper, plastic, fabric, tile, carpet, foam, metal, stone, carbon fiber, and wood). With the exception of metal, all materials were mounted on acrylic using double-sided tape.

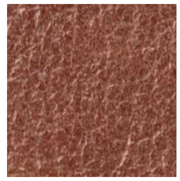
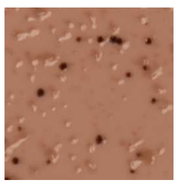
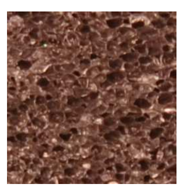
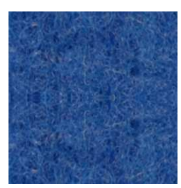
Images of all 100 textures are included in the toolkit. These images were taken with a Nikon D40 digital camera and are stored as square bitmaps with 1024 pixels on each edge. The physical scale of the images is 15 pixels/mm. The images shown in Table 1 are zoomed in from the images included in the toolkit to show detail.

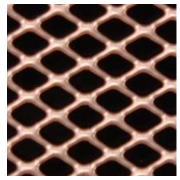
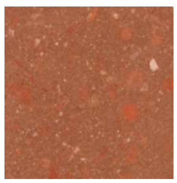
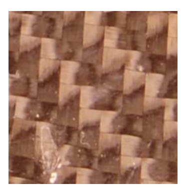
*e-mail: hculb@seas.upenn.edu
†e-mail: juanlop@sas.upenn.edu
‡e-mail: kuchenbe@seas.upenn.edu
[1] http://www.geomagic.com/en/products-landing-pages/haptic

Table 1: Texture Samples

Paper				
Book 	**Bubble Envelope** 	**Cardboard** 	**Coffee Filter** 	**Dot Paper** 
Folder 	**Gift Box** 	**Glitter Paper** 	**Greeting Card** 	**Masking Tape** 
Paper Bag 	**Paper Plate 1** 	**Paper Plate 2** 	**Playing Card** 	**Resume Paper** 
Sandpaper 100 	**Sandpaper 220** 	**Sandpaper 320** 	**Sandpaper Aluminum Oxide** 	**Textured Paper** 
	Tissue Paper 		**Wax Paper** 	
Plastic				
ABS Plastic 	**Binder** 	**Candle** 	**File Portfolio** 	**Frosted Acrylic** 

Nitrile Glove	**Plastic Mesh 1**	**Plastic Mesh 2**	**Tarp**	**Wavy Acrylic**
				
Fabric				
Athletic Shirt	**Blanket**	**CD Sleeve**	**Canvas 1**	**Canvas 2**
				
Canvas 3	**Cotton**	**Denim**	**Felt**	**Flannel**
				
Fleece	**Leather 1 Back**	**Leather 1 Front**	**Leather 2 Back**	**Leather 2 Front**
				
Microfiber Cloth	**Nylon Bag**	**Nylon Mesh**	**Pleather**	**Portfolio Cover**
				
Silk 1	**Silk 2**	**Textured Cloth**	**Towel**	**Velcro Hooks**
				

Velcro Loops 	**Velvet** 	**Vinyl 1** 	**Vinyl 2** 	**Whiteboard Eraser** 
		Tile		
Floortile 1 	**Floortile 2** 	**Floortile 3** 	**Floortile 4** 	**Floortile 5** 
	Floortile 6 		**Floortile 7** 	
		Carpet		
Artificial Grass 	**Carpet 1** 	**Carpet 2** 	**Carpet 3** 	**Carpet 4** 
		Foam		
EPDM Foam 	**Pink Foam** 	**Polyethylene Foam** 	**Scouring Pad** 	**Styrofoam** 
		Textured Rubber 		

		Metal		
Aluminum Foil 	**Aluminum** 	**Metal Mesh** 	**Metal Shelving** 	**Textured Metal** 
		Whiteboard 		
		Stone		
Brick 1 	**Brick 2** 	**Ceramic** 	**Painted Brick** 	**Stone Tile 1** 
	Stone Tile 2 		**Terra Cotta** 	
		Carbon Fiber		
	Carbon Fiber 		**Resin Carbon Fiber** 	
		Wood		
Cork 	**MDF** 	**Painted Wood** 	**Stained Wood** 	**Wood** 

3 Recorded Data

The toolkit includes two recorded data files for each texture. Three axes of acceleration, force, and position data were recorded while the experimenter explored each texture with a custom recording device using natural and unconstrained motions. The recording device was fit with a 3.175 mm diameter stainless steel hemispherical tooltip. Each data file is 10 seconds long and is stored at a sample rate of 10 kHz. Table 2 shows the information included in the recorded data files, which are stored in XML format. These data files were used to create the texture and friction models presented in this toolkit.

Table 2: Recorded Data Files

	Field	**Description**
Accel_*MATERIAL* Files	<AccelUnits>	Units of acceleration
	<SampleRate>	Sample rate in Hz
	<Accel>	Combined acceleration signal using DFT321 algorithm
	<Accel_x>	Acceleration in x-direction
	<Accel_y>	Acceleration in y-direction
	<Accel_z>	Acceleration in z-direction
Position_*MATERIAL* Files	<SpeedUnits>	Units of speed
	<PositionUnits>	Units of position
	<SampleRate>	Sample rate in Hz
	<Speed>	Absolute speed value
	<Position_x>	Position in x-direction
	<Position_y>	Position in y-direction
	<Position_z>	Position in z-direction
Force_*MATERIAL* Files	<ForceUnits>	Units of force
	<SampleRate>	Sample rate in Hz
	<ForceNormal>	Force normal to surface
	<ForceTangential>	Force tangential to direction of motion
	<Force_x>	Force in x-direction
	<Force_y>	Force in y-direction
	<Force_z>	Force in z-direction

4 Texture Models

Each texture's recorded acceleration signal is modeled as a piecewise autoregressive (AR) process. The models are stored in a Delaunay triangulation and are labeled with the normal force and scanning speed used when recording the data.

To increase the flexibility and utility of HaTT, the toolkit includes a method for resampling the texture models so they can be used to render textures at a sampling rate lower than the 10 kHz used when recording data. Table 3 provides a summary of the files included, which resample the models and write the XML files necessary for the rendering code.

Table 3: Model Resampling Files

	File	Description
Main function	CreateResampledModels.m	Main resampling function. Takes in new sampling rate in Hz as argument.
Helper functions	readValueXML.m	Reads in single value from XML file
	readArrayXML.m	Reads in array of values from XML file
	resampleModels.m	Takes in the parsed model information, the original sampling rate (10,000 Hz), and the desired sampling rate. Returns the model information of the downsampled models.
	modelXSL_createHTML.xsl	XSL stylesheet to create an HTML file for each new model XML file

The toolkit includes both the original models at 10 kHz and the resampled models at 1 kHz. Table 4 shows the information included in the model XML files. These model XML files are used in the rendering code included in the toolkit. HTML files are also included for visualization of the model sets.

5 Rendering Code

The code presented in the toolkit is based on OpenHaptics 3.0 and is for the implementation of our texture rendering methods with a SensAble Phantom Omni. The rendering code may be adapted to run on other hardware, but this has not yet been tested. The rendering methods presented in the toolkit are available for Linux and Windows computers. The Linux version of the rendering code was implemented on a computer running Ubuntu version 12.04 LTS with a GeForce FTX 570/PCIe/SSE2 graphics card. The Windows version of the rendering code was implemented in Visual Studio 2008 on a computer running 64-bit Windows 7 with an Intel HD Graphics 2000 graphics card.

Table 5 provides a description of the files included in the toolkit to run the sample rendering code. Third party code is explicitly labeled. In addition, the Boost Random Number Library[2] is needed to compile and run the code.

6 Acknowledgements

The code was developed from the original TexturePad haptic rendering system designed by Joseph M. Romano. This material is based upon work supported by the National Science Foundation under Grant No. 0845670. The first author was supported by a research fellowship from the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-0822.

[2]http://www.boost.org/

Table 4: Model Files

	File	Description
General Information	<Material>	Texture name
	<htmlPicture>	File path to image of texture for HTML
	<renderPicture>	File path to image of texture for rendering
	<mu>	Kinetic friction coefficient
	<numARCoeff>	Number of AR coefficients
	<numMACoeff>	Number of MA coefficients
	<numMod>	Number of models
	<maxSpeed>	Maximum modeled speed
	<maxForce>	Maximum modeled force
	<speedList>	Array of all modeled speeds
	<forceList>	Array of all modeled forces
Units	<AccelUnits>	Units of acceleration
	<SpeedUnits>	Units of speed
	<ForceUnits>	Units of force
	<SampleRate>	Model sample rate in Hz
Delaunay Triangulation	<numTri>	Number of triangles in Delaunay triangulation
	<DTpicture>	File path to image of Delaunay triangulation
	<tri>	Model vertices for one triangle
Model Information	<model>	Information for one ARMA model, including:
	<modNum>	Model number
	<ARcoeff>	Array of AR coefficients
	<ARlsf>	Array of AR line spectral frequencies
	<MAcoeff>	Array of MA coefficients
	<MAlsf>	Array of MA line spectral frequencies
	<var>	Model variance
	<gain>	Gain of MA coefficients
	<speedMod>	Model speed
	<forceMod>	Model force

References

[1] H. Culbertson, J. J. López Delgado, and K. J. Kuchenbecker. One hundred data-driven haptic texture models and open-source methods for rendering on 3d objects. In *Proc. IEEE Haptics Symposium*, February 2014.

Table 5: Rendering Files

	Field	Description
src folder	main.cpp	Runs graphics loop and haptics loop; calculates texture vibrations
	AccSynthHashMatrix.cpp	Constructs hash entries to store models; interpolates between models
	ContactModel.cpp⋆	Defines equation of the force fields
	helper.cpp⋆	Sets the graphics state
	pugixml.cpp§	XML parser
include folder	AccSynthHashMatrix.h	Hash table class for texture generation
	autoGenHashMatrix.h	Parses model parameters from xml file and stores models in memory
	shared.h	Shared memory file for texture generation
	sharedInit.h	Initializes variables in shared memory for texture generation
	ContactModel.h⋆	Defines constants for contact in simulation
	Constants.h⋆	Defines physical constants of simulation
	pugixml.hpp§	XML parser
	pugiconfig.hpp§	Configuration file for XML parser
	foreach.hpp§	Boost.Foreach support for pugixml classes
build folder	Models_*MATERIAL*.xml	Model XML files
images folder	*MATERIAL*_square.bmp	Texture image for display in rendering

⋆Adapted from OpenHaptics sample code, §Copyright 2006-2012 by Arseny Kapoulkine