

Hands-on Demonstration of Heterogeneous Haptic Texturing of Mesh Models Based on Image Textures

Arsen Abdulali, Waseem Hassan, Baek Seung Jin and Seokhee Jeon

Department of Computer Science and Engineering, Kyung Hee University,
Yongin-si 446-701, Republic of Korea
{abdulali, waseem.h, jksdgh78, jeon}@khu.ac.kr

Abstract. In this article, we introduce a framework for heterogeneous assignment of multiple haptic textures to mesh objects based on image textures. The framework consists of two applications, i.e., texture assignment and rendering programs. A user-friendly interface of the framework allows to assign and render textured mesh models in four steps. First, the user provides an image texture file to the algorithm that automatically selects perceptually closest haptic texture from the library. Then, the user is offered to texture object faces by stroking over an object surface with a virtual brush. Several haptic textures can be assigned to different object surfaces of a single mesh model. The haptic information for each face is embedded to mesh object and stored in a generic *.ply file. Finally, a textured mesh object is loaded by rendering application.

Keywords: Haptic Texture Assignment, Rendering

1 Introduction

Recent haptic texture rendering algorithms provide a highly realistic vibrotactile feedback. These algorithms could potentially drive the research in virtual reality to a new level by allowing users to interact with a virtual object and feel surface textures. However, a shape of objects in a virtual world is usually changed during simulation having several heterogeneously distributed textures. For example, a human avatar in a computer game usually has distinct textures for skin, cloth, and shoes. In such cases, it becomes challenging to assign a haptic texture model to a specific dynamically transformed surface of a three-dimensional mesh object.

Current approaches render a haptic texture in a two-dimensional canvas of a touch screen of a tablet[4, 2] or assign a single haptic texture to a complete mesh model [3]. Culbertson et al. rendered isotropic haptic textures on the tablet where the vibrotactile signal is interactively synthesized with respect to contact force and a movement velocity[4]. This idea was further extended in [2] to rendering anisotropic haptic texture by decomposing movement velocity into two-dimensional velocity vector. In [3], the authors developed a rendering framework with one hundred haptic textures, where an isotropic haptic texture model can be assigned to a single mesh object.

In this paper, we developed a framework that allows VR designers to manually texture a mesh object with multiple haptic texture models and render it in a virtual environment. In order to alleviate the need of modeling haptic textures, we applied an automatic haptic texture assignment algorithm [5], which selects perceptually closest haptic texture model from the library based on a given image texture. Our framework consists of two software applications, i.e., object texturing and rendering programs. The framework is user-friendly and requires only four steps, which are briefly summarized as follows:

- A user provides an image texture file and the algorithm selects perceptually closest sample from the library.
- A user applies selected haptic texture to faces of a mesh object using a computer mouse or touchscreen.
- The textured mesh model is saved into "*.ply" file where haptic texture models are assigned to object faces.
- A textured mesh object is loaded by the rendering software allowing a user to explore virtual textures.

2 Object Texturing Application

The application for the object texturing consists of texturing interface, i.e., a virtual scene with a mesh model and navigation panel for basic object manipulations, such as mesh model loading, object rotation, scaling and image texture loading (see Fig. 1). In order to texture a mesh model, the user should provide a path to the "*.ply" file and image texture by pressing corresponding "OPEN" buttons. Once a user selects an image texture, the automatic assignment algorithm is invoked.

Automatic assignment algorithm [5] computes a set of distinct features from the given image texture. The authors established a haptic texture library of 84 samples in the form of three-dimensional perceptual space. In order to retrieve a perceptually closest sample from the library, the perceptual space was subdivided into sixteen groups of textures using a clustering algorithm. Support Vector Machine (SVM) was used to map a feature vector of an image to one out of sixteen perceptual groups. Once a group with perceptually closest samples is determined, the algorithm selects the most similar sample within the group based on Binarized Statistical Image Features (BSIF) features using chi-squares distances. The detailed information is provided in [5].

The user can start manual texturing using a virtual brush when a mesh model is successfully loaded and perceptually closest haptic model is selected. In order to provide visual feedback, surfaces with assigned haptic models are textured by the sample selected from the library. In rendering, the original image texture can be used instead. When the texturing process is over, the user presses the "Save PLY" button and specifies a path to a resultant "*.ply" file.

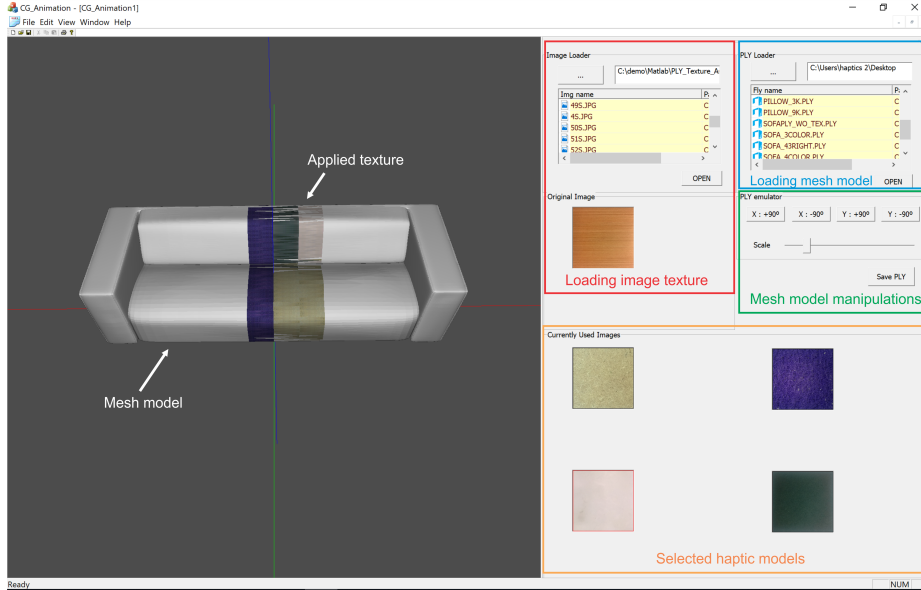


Fig. 1. Graphical user interface of the object texturing application

3 Rendering Application

In order to render the mesh model, the user should start the second application and provide a path to the textured `*.ply` file. The mesh model appears on the scene (see Fig. 2), and the user can touch the object using end-effector of a haptic device. Upon contact, a collision detection algorithm returns the id of the face and the id of the corresponding haptic texture model is retrieved.

In order to render haptic textures, we utilized a Radial Basis Functions Network (RBFN) based stochastic model, which we previously proposed in [1]. In this work, we focus on rendering isotropic haptic textures. Therefore RBFN models were trained for two-dimensional input, i.e., normal force and velocity magnitude. During rendering, a selected RBFN model along with two contact inputs (normal force and velocity magnitude) are fed into a runtime computing library, which we demonstrated in [2]. The computing library estimates a vibrotactile signal and applies to a vibrotactile actuator (Haptuator MM3C; Tactile Labs Inc., Montreal, Canada) via data acquisition device (NI USB-6251; National Instruments, Austin, Texas, USA).

4 Demonstration Protocol

The demonstration consists of two steps. First, a demonstration attendee sits in front of the tablet-PC running our texturing application. The user can select one or several image textures and apply corresponding haptic texture models

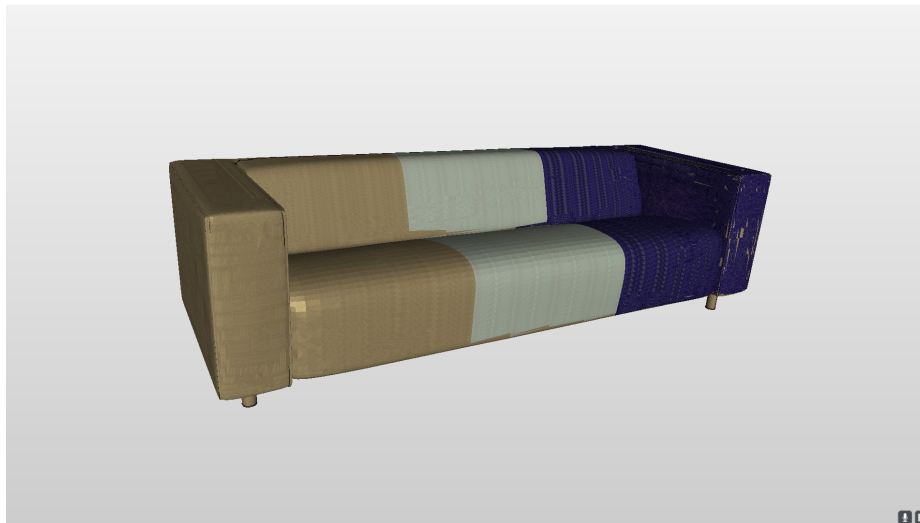


Fig. 2. Graphical user interface of the rendering application

to the mesh model. The image textures can either be in the form of a photo taken by the user or some texture image form online datasets. When the user finishes object texturing, the ".ply" file is saved and transferred to the rendering PC through the network. Second, the attendee moves to the rendering PC and explores a virtual mesh model. After the demonstration, attendees will be offered to volunteer in a short survey.

Note that perceptual characteristics of the data-driven model were evaluated in [1] and were felt realistic by Asia Haptics 2016 conference attendees [2]. On the other hand, the accuracy of haptic texture assignment algorithm was rated 71.4 % in [5]. In this work, we developed a manual haptic texturing concept for VR and computer games designers. Thus we aim to get feedback regarding user experience for proposed texturing and rendering interfaces.

5 Conclusion

In this article, we developed a framework to demonstrate the concept of haptic texturing. Our framework consists of two applications. The first program is a graphical user interface that allows a user to apply a haptic texture to a mesh model and save it into a generic "*.ply" file. The second application renders a haptically textured mesh object.

Acknowledgment

This work was supported by the NRF of Korea through the Global Frontier R&D Program (2012M3A6A3056074).

References

1. Abdulali, A., Jeon, S.: Data-driven modeling of anisotropic haptic textures: Data segmentation and interpolation. In: International Conference on Human Haptic Sensing and Touch Enabled Computer Applications. pp. 228–239. Springer (2016)
2. Abdulali, A., Jeon, S.: Data-driven rendering of anisotropic haptic textures. In: Proceedings of the Asia Haptics 2016 Conference (December 2016)
3. Culbertson, H., Delgado, J.J.L., Kuchenbecker, K.J.: One hundred data-driven haptic texture models and open-source methods for rendering on 3d objects. In: 2014 IEEE Haptics Symposium (HAPTICS). pp. 319–325. IEEE (2014)
4. Culbertson, H., Romano, J.M., Castillo, P., Mintz, M., Kuchenbecker, K.J.: Refined methods for creating realistic haptic virtual textures from tool-mediated contact acceleration data. In: Haptics Symposium (HAPTICS), 2012 IEEE. pp. 385–391. IEEE (2012)
5. Hassan, W., Abdulali, A., Abdullah, M., Ahn, S.C., Jeon, S.: Towards universal haptic library: Library-based haptic texture assignment using image texture and perceptual space. IEEE transactions on haptics (2017)