Spring 2014 Haptics Class Project Paper presented at the University of South Florida, April 30, 2014

A Movement Based Method for Haptic Interaction

Matthew Clevenger

Abstract— **An abundance of haptic rendering schemes have been developed to help further presence and immersion in virtual environments. Increasing realism is the main driving force behind this development. Schemes have progressed from simple one-dimensional springs to six degree of freedom particle models, with a wealth of surface and material properties. Because of their relative unimportance, haptic rendering schemes for video games have been relatively ignored. Video games generally do not need to have ultra-realistic haptic feedback. Less realistic, more entertaining feedback is often beneficial. In opposition to most haptic rendering schemes, which use position control, movement based haptic feedback functions under acceleration control, a common control scheme in video games. The presented method displays normal force, friction, virtual object weight, water drag, water inertia, and impact forces. Force direction is calculated using the projection and rejection of the user's input on the collision normal allowing virtual objects to be oriented in any direction. A user study was conducted comparing movement based feedback to constraint-based feedback and no feedback. Movement based feedback was favored by a third of the subjects and given the highest rating for enthusiasm by half of the subjects. This work presents a novel haptic rendering method that can easily be applied to existing video games.**

I. Introduction

Most haptic rendering algorithms follow a position correspondence model [1], [2], [3]. The virtual position is determined by the physical position of the device, possibly with scaling. This has the benefit of following a predictable pattern where moving the device to a certain physical position always corresponds with the same virtual position. In most cases, the reachable virtual workspace is predefined at design time. This is desirable in most applications as it provides a natural, intuitive means of interaction. There are, however, situations where a predefined virtual workspace is cumbersome or undesirable. In large environments a means of navigation is necessary. The proposed movement based method allows the user to navigate through the environment while also providing haptic feedback. The workspace is infinite along any controllable direction.

Video games make extensive use of the visual and auditory modalities, but with the exception of limited vibratory feedback, largely exclude haptic interaction. A richer sense of presence and immersion could be achieved by adding haptic feedback. In this paper the proposed movement based haptic rendering method is applied to a two-dimensional side-scrolling video game.

Section II provides a summary of related work. Section III details the movement based haptic rendering technique. The haptic interface and implementation details are described in Section IV. A user study, conducted to compare the presented method with existing methods, is detailed in Section V. The results of the experiment are presented and discussed in Sections VI and VII, followed by concluding remarks in Section VIII.

II. Background

A. Haptic Rendering

The *penalty* method applies a force proportional and opposite to the amount of penetration into a virtual volume. The simplicity of this approach has led to extensive study and expansion. There are, however, a number of drawbacks to this method [4]. It is often difficult to determine which exterior surface to associate with a given volume when multiple primitives touch or intersect. Force discontinuities appear when approaching other surfaces of the same object. Lastly, thin objects are unable to generate sufficient force to prevent the device from passing through [2]. To overcome these limitations, Zilles and Salisbury [1] proposed a *constraint-based* method. This method employs a god-object which is constrained by the virtual environment and controlled by physics. Vector field force shading [5], analogous to Phong shading for graphic display, was incorporated into haptic rendering by Morgenbesser and Srinivasan. Ruspini et al. [2] extended these ideas with a finite virtual proxy. The virtual proxy is able to model force shading, friction, surface stiffness, and texture. The finite size of the virtual proxy also prevents it from slipping through any tiny numerical gaps present in most polygonal meshes. In addition to rigid objects, uids [6] have also been simulated using unified particle models.

III. Movement Based Haptic Rendering

Unlike the position based control of the constraint-based method, movement based rendering functions under acceleration control. Acceleration control is commonly used for navigation among physics driven video games. Acceleration is applied based on deviation from center, similar to a joystick. More acceleration is applied as the device moves further from center. As the acceleration is high and the maximum velocity low, the scheme approximates velocity control. This method of input is analogous to arrow key input. Joysticks and arrow keys are very common input methods for games so the control scheme can easily be retrofitted to apply movement based haptic rendering.

Rendered forces are based on the projection and the rejection of the input vector on the collision normal. The input vector is a normalized Euclidean representation of the position of the haptic device centered at the device origin. Moving the haptic device to the furthest right would result in an input vector of [1, 0, 0], the furthest downward would result in [0, -1, 0]. The collision normal is the normalized sum of the normals of the impacting colliders. The projection is the orthogonal projection of the input vector onto a line parallel to the collision normal. The projection is a vector parallel to the collision normal:

$$\mathbf{p} = \frac{\mathbf{i} \cdot \mathbf{n}}{|\mathbf{n}|^2}\ \mathbf{n} \tag{1}$$

The rejection is the orthogonal projection of the input vector onto the plane orthogonal to the collision normal:

$$\mathbf{r} = \mathbf{i} - \mathbf{p} \tag{2}$$

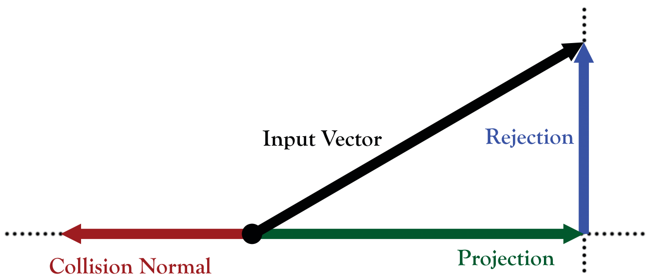

Fig. 1. Projection and rejection of the input vector on the collision normal.

The rendered normal force is simply the negative of the virtual object stiffness multiplied by the projection. The friction force is the negative of the virtual damping multiplied by the rejection. Virtual object weight is proportional to the projection of the collision normal on the downward direction if the projection is not antiparallel. Water drag and inertia are applied when in water. Drag is the negative of the virtual drag coefficient multiplied by the player's velocity. Water inertia is a constant force applied along the ow direction. Impact forces are generated based on decaying sinusoids [7].

$$\mathbf{F_n} = -k\ \mathbf{p} \tag{3}$$
$$\mathbf{F_f} = -b\ \mathbf{r} \tag{4}$$
$$\mathbf{W} = m\ g\ (\mathbf{n} \cdot \mathbf{d})\ \mathbf{d} \tag{5}$$
$$\mathbf{F_{wd}} = -d\ \mathbf{v} \tag{6}$$
$$\mathbf{F_{wi}} = w\ \mathbf{f} \tag{7}$$

IV. Implementation

The movement based haptic rendering method was implemented on a Sensable Phantom Omni. A two-dimensional side-scrolling video game was developed to assess the interaction method. Unity was used as the game engine. Secondary game assets were included to provide a more holistic, entertaining gaming experience.

The developed game utilized Unity's existing physics engine for collision detection. Unity's physics engine updates at 50 frames per second. Humans are able to perceive vibrations in the finger up to 5-10 kHz with a maximum sensitivity around 200-300 Hz [8]. As such haptic rendering should update at 1000 Hz [9]. Because of the discrepancy between the two update rates the normal, friction, weight, drag, and inertia forces are summed and smoothed to reduce erratic motions. The impact force is overlaid raw in order to better capture the transient response.

A dynamic link library is used with interoperation to control the Omni from managed code. The C code used to control the Omni is wrapped in a dynamic link library and called in C# through Platform Invocation Services. Methods to initialize, deactivate, get position, and set force were developed. In order to avoid complicated marshaling only blittable types are used. The position is returned using three separate functions, one for each axis, instead of an array. In addition to complex marshalling, returning an array requires the user to release the array memory manually, an unnecessary complication as the size of position is held constant at three.

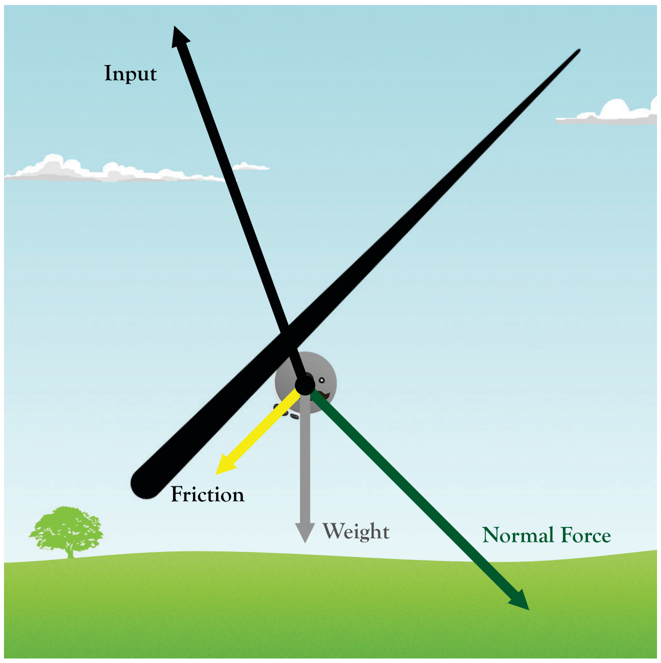

Fig. 2. Normal force, friction, and weight applied to the user.

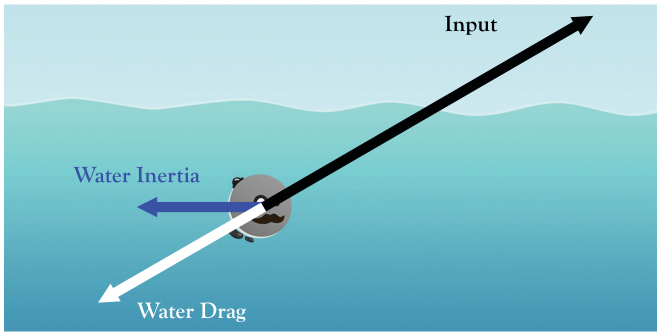

Fig. 3. Water drag and inertia applied when the user enters water.

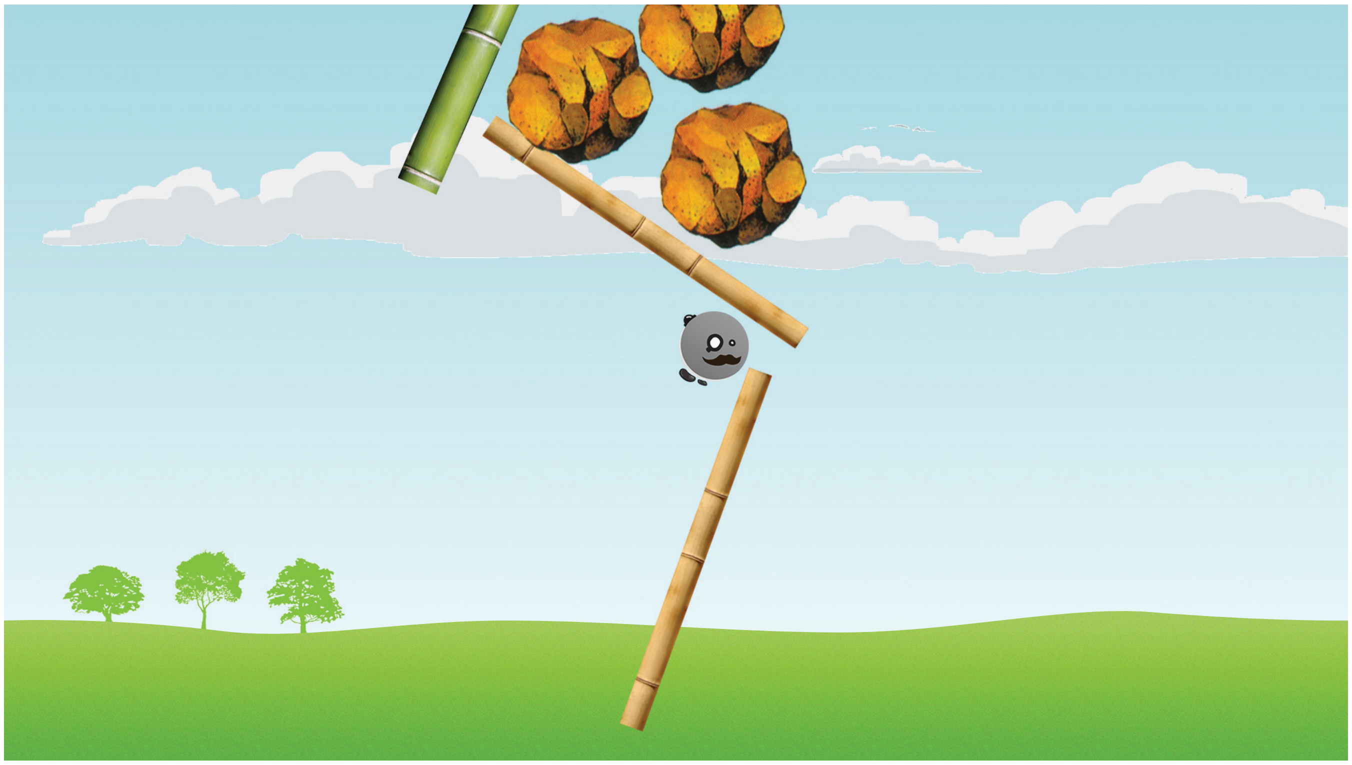

Fig. 4. Implemented scene with secondary game assets. The user is able to interact with static and dynamic obstacles in the virtual world. Collision with any object generates forces which are governed by the movement based haptic rendering algorithm. All objects generate normal force and friction. Dynamics objects impart their weight and impact forces. Water confers drag and inertial forces.

V. Evaluating Interaction

A subjective human experiment was developed to test the partiality and aptness of the proposed movement based haptic interaction method. Subjects were presented with three different interaction methods and asked to rate them on a ten point scale based on six different criteria. The three interaction methods presented were: no feedback, movement based feedback, and constraint-based feedback. The six criteria are listed in Table I. In the methods with haptic feedback only normal force was rendered. Other than removing the extraneous forces, the movement based feedback method presented in the experiment was equivalent to what was described in Section III. The constraint-based feedback method followed that described in [2] with only normal force rendered at a frame rate of 50 Hz. The no feedback method was equivalent to the movement based feedback method only with no feedback rendered. The movement based method and the no feedback method were both acceleration control. The constraint-based method was position control.

At the beginning or end of each experiment data on the subject's age, gender, handedness, and gaming experience was collected. Before beginning the experiment, subjects were introduced to the system by allowing them to explore two demonstration levels. When they felt comfortable with the system the experimentation level was loaded. The experimentation level consisted of a maze where subjects were asked to navigate from the center to the lower right colliding with walls as they went. The different interaction methods were presented once in a random order.

After completing and rating each interface subjects were asked to explain any differences they felt between the three methods and if they had a favorite interface. The entire experiment lasted less than five minutes.

VI. Results

The user study included 10 subjects, aged in their 20s or 30s with one female and nine males. One subject was left handed. The experiment was performed with the left hand due to space constraints. Five subjects had little to no gaming experience where the remaining five had some, no one had extensive gaming experience.

The average rating for each criteria for each interface, compiled across users, is shown in Figure 6. Higher values correspond to a better rating. The total mean rating incorporating all the criteria is shown by the dashed lines.

TABLE I
Criteria

Realism	how realistic is the control scheme
Ease	how easy is it to control the character
Aptness	how appropriate does the control feel
Intuitiveness	how natural is the control
Enthusiasm	how much did you enjoy this method
Partiality	how much do you prefer this method

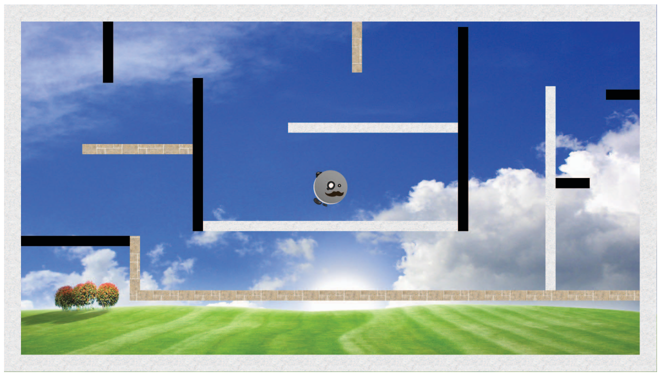

Fig. 5. Experimentation level. A simple maze user's were asked to traverse from the center to the lower right.

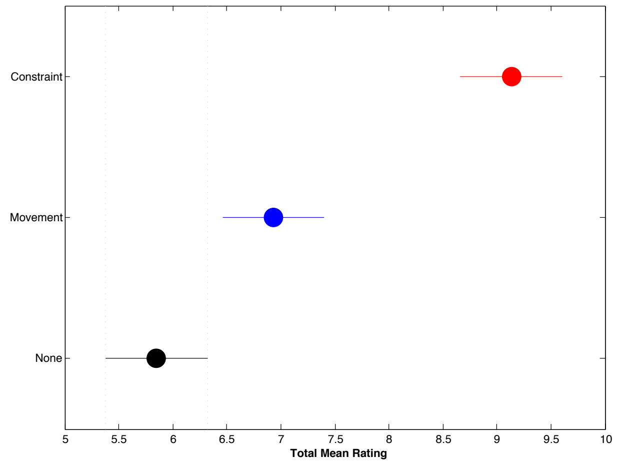

Fig. 7. Total mean rating and standard error for each interface. Statistically significance is present for each interface.

VII. Discussion

Statistically significant differences were present in the total mean rating among all three interfaces, F (2, 163) = 34.73, $p < 0.001$, as shown in Figure 7.

The constraint-based method had by far the highest total mean rating, 9.13, followed by the movement based method, 6.93. No feedback garnered a rating of 5.85. The constraint-based method had the highest average rating for each criteria, with all but partiality being above 9. Movement based rendering had the second highest average rating in all but ease and intuitiveness. No feedback was rated as easier than movement based feedback and of equal intuitiveness. All the subjects were able to identify that no feedback was given in one of the trials and most were able to correctly recognize the differences between the other two methods. No feedback was preferred by one subject, and three and four votes were given to the movement and constraint methods respectively. Some subjects did not vote and others voted for two methods, each method was given half a point in these cases. It was clear that some subjects did not understand the movement based rendering scheme and consequently conferred poor ratings. Other subjects opted for an all-or-nothing approach, either awarding a rating of 1 or 10 with little in-between. The constraint method was the biggest beneficiary of this approach and no feedback the most diminished by it. Many subjects were enthusiastic about the movement based method with half of the subjects giving the method a rating of ten for this category.

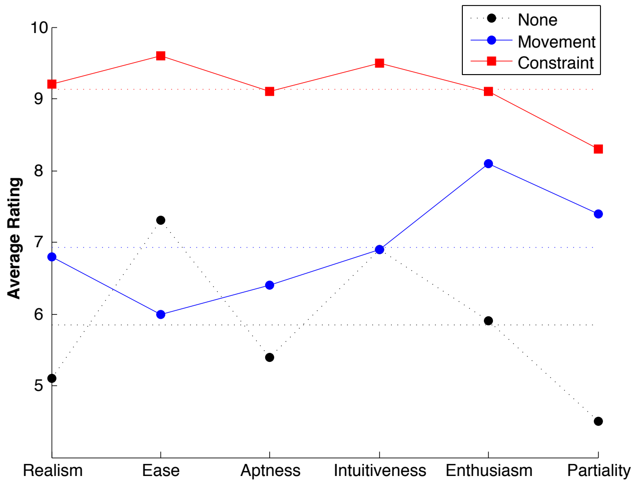

Fig. 6. Average rating for each criteria for each interface.

VIII. Conclusion

While not as realistic or intuitive as constraint-based methods, movement based haptic rendered provides a viable option for haptic interaction. Built around acceleration control, movement based rendering allows for haptic interaction in a control scheme other than position control. Incorporating normal force, friction, object weight, water drag, water inertia, and impact forces movement based rendering can easily be extended further to incorporate any number of surface or material properties. Human subject testing reveals that most people enjoy movement based feedback and find it entertaining.

A timer callback which updates at 1000 Hz could be added to the system to help improve force rendering and reduce the need for smoothing. The system currently operates at 50 Hz which is well below the recommended haptic update rate. Movement based haptic feedback could also be retrofitted to existing games to see if any improvement is gained by adding haptic feedback.

References

[1] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *IEEE International Conf. on Intelligent Robots and System, Human Robot Interaction, and Co-operative Robots*, pp. 146–151, 1995.

[2] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *Proc. 24th Annual Conf. Computer Graphics and Interactive Techniques*, pp. 345–352, Aug 1997.

[3] C. Basdogan and C. H. Ho, "Principles of haptic rendering for virtual environments." Online.

[4] K. Salisbury, F. Conti, and F. Barbagli, “Haptic rendering: Introductory concepts,” *IEEE Computer Graphics and Applications*, vol. 24, pp. 24–32, Mar-Apr 2004.

[5] H. B. Morgenbesser and M. A. Srinivasan, “Force shading for haptic shape perception,” *Proc. ASME Dynamic Systems and Control Division*, vol. 58, pp. 407–412, 1996.

[6] S. H. G. Cirio, M. Marchal and A. Lécuyer, “Six degrees-of-freedom haptic interaction with uids,” *IEEE Trans. Vis. Comput. Graphics*, vol. 17, pp. 1714–1727, Nov 2011.

[7] K. J. Kuchenbecker, J. Fiene, and G. Niemeyer, “Improving contact realism through event-based haptic feedback,” *IEEE Trans. Vis. Comput. Graphics*, vol. 12, pp. 219–230, Mar 2006.

[8] T. A. Kern, “Biological basics of haptic perception,” in *Engineering Haptic Devices*, ch. 3, pp. 35–58, Berlin: Springer-Verlag, 2009.

[9] M. C. Lin and M. Otaduy, eds., *Haptic Rendering: Foundations, Algorithms, and Applications*. A K Peters/CRC Press, July 2008.