VECIMS 2007 – IEEE International Conference on
Virtual Environments, Human-Computer Interfaces, and Measurement Systems
Ostuni – Italy , 25-27 June 2007

A Device Independent Haptic Player

Mohamad Eid[1], Mohamed Mansour[1], Rosa Iglesias[2], and Abdulmotaleb El Saddik[1]

Multimedia Communications Research Laboratory
University of Ottawa, Canada
E-mail: [1]{eid, mansour, abed}@mcrlab.uottawa.ca, [2]riglesias@discover.uottawa.ca

Abstract

This paper presents the design, implementation, and quantitative evaluation of a haptic player system that is independent from the haptic device, in the context of our HAML-based authoring tool project at the University of Ottawa. The system comprises three components: a haptic recorder, a feature extractor, and the haptic player. The haptic recorder is capable of retrieving, filtering, and storing the data sent by a haptic device in a local database. The feature extractor generates a standardized description of the stored data by utilizing the HAML schema. Finally, the haptic player component recreates a haptic stimulus by utilizing the HAML description of the corresponding stimulus. Haptic stimuli are played back according to the haptic device's features. The experimental evaluation of the proposed HAML-based player showed that the haptic player system is independent from the device used during data recording.

1. INTRODUCTION

The sense of touch plays a prominent role in our daily life activities. It is vital for: holding and moving objects, typing on computer keyboards, providing information about textures and surfaces and expressing emotions, among others. It is also important for teaching a skillful task for fast and accurate performance, such as, handwriting and any training assistance mode (i.e. in medical procedures, drawing, and in sports, like tennis, to learn how to strike the ball with your racket). Teaching a sensorimotor skill has been proven as an expensive and difficult task [1]. Teaching skillful tasks generally requires active assistance, that is, an expert trainer to physically interact with the trainee. However, such training is time consuming and requires a high trainer/trainees ratio. A system capable of presenting complex physical tasks to a group of students will not only cut the cost drastically, but also improve the quality of teaching by allowing the trainer to trace the performance, and students to perform tasks as many times as required. The haptic technology can be utilized to create such systems and it provides physical interaction with trainees using haptic interfaces; thus decreasing the trainer/trainees ratios.

On the other hand, the sense of touch contributes to the expression of emotions and perceptions. Lately, commercially available instant messaging systems have addressed this issue by integrating visual-audio emoticons, winks, nudges or animations. In this framework, touch can be seen as a new mechanism for conveying emotions in a more natural and realistic way. For instance, some researchers have designed instant messaging systems where users can send each other a haptic message for expressing emotions [2]. We envision a system where files containing haptic data can be played back according to user actions.

The term haptics is derived from a Greek term 'haptesthai', meaning of or relating to the sense of touch. Nowadays, this term refers to the expanding discipline that studies the sense of touch and the human interaction with an environment via touch. A recent literature review that describes the fundamentals of haptic technologies and their potential applications can be found in [3]. A potential application of haptic technology is the ability to record and playback haptic stimuli. As an example, a haptic player could be useful in training where the motions of an expert may be recorded and saved for later 'playback' by a trainee. Several haptic playback systems/prototypes have been designed and evaluated for training, for instance, epidural injection techniques [4, 5], back palpation procedures [6] and handwriting [7, 8]. The ability to play back haptic data can also be introduced in instant messaging systems for expressing emotions. In this context and with the increasing diversity of commercial haptic devices currently available, haptic players should be decoupled from the haptic device used.

Haptic playback refers to the ability of displaying prerecorded haptic information using a haptic interface [6]. Ideally, a haptic player should have the ability to display information regarding the trajectory to be followed and the forces that needs to be exerted. However, one of the major issues in haptic playback is that position and force have contradictory causality. Until now, most of the studies have focused on playing

1-4244-0820-2/07/$20.00 ©2007 IEEE

back the data to guide the user throughout a prerecorded path. Other research, in turn, has focused on providing controlled forces to users along a defined trajectory. For instance, the force applied plays an important role in medical surgery applications [5]; however, in handwriting applications following a trajectory is more significant [7, 8]. On the other hand, several efforts have been made for simultaneously displaying both a prerecorded trajectory and the exerted force. Controlling this relation poses a challenge and methods, such as impedance control, have been used [9, 10]. However, it continues to be an issue to be addressed.

In this paper, we show the potential of using the Haptic Application Meta Language (HAML) [11] to standardize haptic data description and thus decouple the data from the haptic player. The player becomes highly independent from the haptic device; meaning that any haptic player compatible with the HAML standard will be able to playback the data in the same way commercial video players play audio or video clips. Users can play back haptic stimuli regardless of the haptic device used, for instance for learning language calligraphies or for playing haptic-emoticons in an instant messaging system.

In the next section, we first describe the related work and the contributions of this paper. In Section 3, we present the architecture of the player system and how we addressed the issues related to haptic playback. Section 4 presents our performance evaluation of the proposed player. Finally, in Section 5 we summarize our findings and future work.

2. RELATED WORK

The theory of haptic playback has been investigated by several researchers. The idea was initially proposed in [12], where several schemes were proposed to simultaneously display force and position trajectories. The authors claimed that it is impossible to haptically display both the force and position trajectories at the same time with the same haptic device. Consequently, most researches have focused on displaying one of the two aspects and providing information about the other through another modality such as vision.

Several attempts have been made to build systems for teaching sensorimotor skills through force feedback. For instance, the authors in [13] investigated the possibility of providing gesture training using visual and/or haptic guidance. The results showed that the visual-only training mode does not significantly improve learning as compared to the haptic-only mode. This acknowledges the fact that haptic modality is a potential channel for human learning. Another constraint-based training system is presented in [5] to guide users by restricting their movements from deviating from a reference golden path – usually the path followed by an expert. In a similar effort, Yokokohji and his colleagues discussed haptic trajectory playback for the purpose of training for simple tasks [14]. The developed system actively drags a user through a task to provide learning in completing the task.

Similar techniques have also been applied to teach handwriting. For instance, the authors in [7] demonstrated a system where the position of a teacher can be recorded and played back to help students forming the Chinese characters. Nonetheless, haptic playback has been intensively researched for medical training scenarios. As an example, Williams and his colleagues [6] described a haptic playback system for the Virtual Haptic Back Project, developed at Ohio University. The goal was to create a haptic simulation of the human body to assist students in learning palpatory techniques. Recently, Corno and Zefran proposed a novel engineering analysis of haptic playback that enables simultaneous display of force and position data to a user [10]. The authors developed control strategies where an explicit model of the user is derived and validated by experimental results.

Although the cited authors have contributed significantly to better understanding the theory and implementation of haptic playback systems, their work is not aimed at defining a standard for haptic players. Their results are only applicable to their applications and haptic devices used for recording data. In contrast, this paper discusses a device independent haptic player by utilizing standardized HAML descriptions of the captured data. Users can independently record and/or play haptic stimuli regardless of the used device.

3. HAML-BASED PLAYER SYSTEM

The proposed haptic player system comprises three major components: the haptic recorder, the HAML feature extractor, and the haptic player. Figure 1 shows a high level overview of the system architecture.

3.1 The Haptic Recorder

The haptic recorder is capable of retrieving, filtering, and storing compacted sets of data in a local database. By haptic data, we refer to spatial (position and orientation) data and force (force and torque) data and their temporal relationships (timestamps). Recording haptic data has been proven problematic due to several issues such as the high bandwidth requirements (sampling rates equal or higher than 600 Hz), data accuracy and relevance, and data filtration and presentation. By combining an intelligent filtering algorithm, a thread-management algorithm, as well as an Input/Output optimization algorithm (Figure 1), the haptic recorder is able to save accurate haptic data even at sampling rates of over 1 kHz [15].

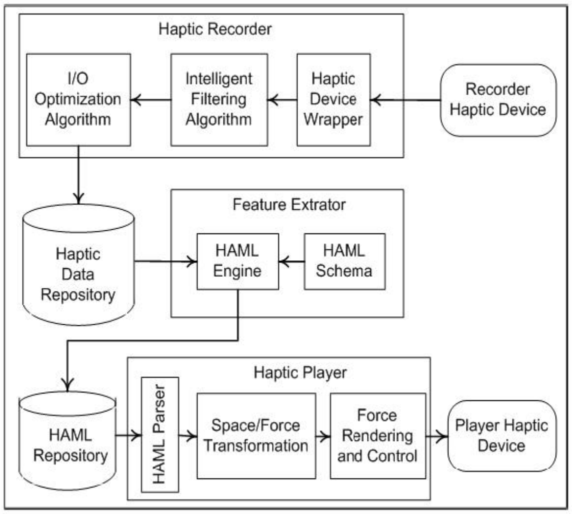

Figure 1. The HAML-based playback system

When the haptic recorder retrieves the data from the haptic device (accessed through the haptic API), it begins filtering the streaming data. The filtration is a two-steps process, which first checks the timestamp of the received sample to ensure that it is unique. If so, the intelligent filtering algorithm checks the relevance of the sample. Relevance is first and foremost a user-based choice. The idea is to test if the difference between the current sample (position, velocities, or orientation) and the previous one falls within a user-defined threshold, the data will be treated as 'redundant' (for example due to hand tremors or device-related variables), and thus the sample is dropped.

If the sample is deemed unique and relevant, it is stored in a large, circular memory buffer awaiting its turn to be flushed to disk. This process is realized using the Invictus algorithm [15]. This algorithm relies on a large memory-resident circular, one-way memory buffer, and a shorter intermediate write buffer, which is used to flush data to disk. The data is stored into the write buffer to enhance managing cross-thread calls. If we were to write the data from the memory buffer directly to disk, then the memory buffer needs to be locked for larger time; hence dropping a larger number of incoming samples. A pseudo code that summarizes the algorithm is shown in Figure 2.

```
Create new ManagementThread();
StartManagementThread();
While(Simulation is Active)
{
  RetrieveDataFromHapticDevice();
  FilterRetrivedData();
  SaveFilteredDataInMemory();
  if(SaveToDisk Flag is Triggered)
  {
    SaveToDisk()
  }
}
```

Figure 2. Pseudo code for the Invictus algorithm

3.2 The HAML Feature Extractor

Once the stimulus is recorded, the feature extractor generates standardized HAML descriptions for the captured data. The HAML file describes the repository location (i.e. database name, type and connection string), the haptic interface used to capture the data, the force computation method, the haptic data structure (i.e. time, position, orientation, velocity, force, torque), measurement units (i.e. centimeters/meters, seconds/ milliseconds), and stimulus features. The haptic interface description includes the quality and performance parameters of the interface, such as the device workspace, the maximum and minimum forces and/or torques, the haptic refresh rate, and the number of degrees of freedom. The force computation technique is essentially the approach used to determine the force feedback (i.e. spring, spring-damping, or spring-damping-inertia).

The stimulus features include the stimulus workspace, the maximum and minimum forces, the sampling rate, and reference frame. Once generated, the HAML file is stored in the HAML repository for later use, as shown in Figure 1. An excerpt of the generated HAML document that describes a recorded stimulus is shown in Figure 3.

```
<Data Type = "Database">
   <DataFormat> MSACCESS </DataFormat>
   <DBSchema>Haptic_Session</DBSchema>
   <DBFile> C:\\Users\\ test.mdb
</DBFile>
 <Data Structure>
   <Item Name="TimeStamp" Type="int"
Unit="ms"/>
   <Item Name="force" Type="float"
Unit="N"/>
   <Item Name="position" Type="float"
Unit="N"/>
   <Item Name="velocity" Type="float"
Unit="m/s"/>
 </Data Structure>
</Data>
<Force_Computation>
   <Name>Spring-Damper</Name>
   <Stiffness> 0.5 N/cm</Stiffness>
   <Damper> 0.5 Nsec/cm</Damper>
</ Force_Computation >
<Stimulus>
   <Name>Sample Recording</Name>
   <Workspace> 77.314x42.6497x36.28484
   cm</Workspace>
   <Center> 38.657x21.32485x18.14242
cm</Center>
   <Sampling_rate>1000</Sampling_rate>
   <Orientation>0x0x0</Orientation>
   <Duration>343535</Duration>
</Stimulus>
<Device>
   <Name>Phantom Omni</Name>
```

Figure 3. HAML description of the recorded stimulus

3.3 The Haptic Player

The haptic player enables the user to select the HAML file that describes the haptic stimulus, makes the necessary transformation to the player device frame, and plays it back using the player device. The player comprises three subcomponents: the HAML loader, the transformation component, and the haptic rendering component. The HAML loader parses the HAML file and imports the stimulus features. Eventually, the transformation component converts the coordinates and forces of the stimulus to the player device space. The transformation matrix between the two spaces is derived based on the relative orientation between the recorder device frame and the player device frame. The haptic rendering component computes the driving forces that must be sent to the device to reconstruct the recorded trajectory.

3.3.1 Transformation

The transformation component maps the recorder device frame $\{B\}$ (position and orientation) to that of the playback device $\{A\}$. A point position in frame $\{B\}$ is represented as ${}^{B}P$ whereas its equivalent in frame $\{A\}$ is ${}^{A}P$. The transformation between the two frames is defined as shown in equation (1).

$$ {}^{A}P = {}^{A}_{B}T \times {}^{B}P \tag{1} $$

Where ${}^{A}_{B}T$ is the transformation matrix from frame $\{B\}$ to frame $\{A\}$. It is expressed as a concatenation of a rotation matrix ${}^{A}_{B}R$ and a translation vector ${}^{A}P_{B_{ORG}}$, as shown in equation (2). Notice that ${}^{A}P_{B_{ORG}}$ is the position of the origin of frame $\{B\}$ relative to frame $\{A\}$.

$$ {}^{A}_{B}T = [{}^{A}_{B}R, {}^{A}P_{B_{ORG}}] \tag{2} $$

The rotation matrix, for an angle θ and around a general vector $K = [K_x, K_y, K_z]^T$, is given in equation (3) [16].

$$ K = \begin{bmatrix} K_{xx}V(\theta) + C(\theta) & K_{xy}V(\theta) - K_z S(\theta) & K_{xz}V(\theta) + K_y S(\theta) \\ K_{xy}V(\theta) + K_z S(\theta) & K_{yy}V(\theta) + C(\theta) & K_{yz}V(\theta) - K_x S(\theta) \\ K_{xz}V(\theta) - K_y C(\theta) & K_{yz}V(\theta) + K_x S(\theta) & K_{zz}V(\theta) + C(\theta) \end{bmatrix} \tag{3} $$

Where $V(\theta) = (1 - Cos(\theta))$, $K_{xx} = K_x.K_x$, $K_{xy} = K_x.K_y$, $S(\theta) = Sin(\theta)$, $C(\theta) = Cos(\theta)$

To map the recorded stimulus to the player device workspace, the transformation component performs position and/or force scaling whenever necessary. If the stimulus workspace W_B is larger than that of the player device W_A, the data is divided by the factor W_B / W_A. The same scaling applies for forces (by comparing the maximum force of the stimulus and the maximum force property of the player device).

The last step before force computation is the position interpolation. If the recorded stimulus sampling rate f_B is not equal to the player rate f_A, then a Sampling Rate Conversion (SRC) is needed. If f_B is higher than f_A, the SRC is achieved using linear interpolation (to fill the gaps between successive samples). Otherwise, a fraction of the recorded samples will be dropped.

3.3.2 Haptic Rendering

The player reads the transformed position information of the recorded stimulus, retrieves the current device position, and computes the driving force *(F)* using equation (4). This force drives the device back through the previously recorded position for playback. Notice that the player can use different playback techniques based on the available data and/or device. This is enabled by describing a specific technique in the HAML description file of the haptic stimulus.

$$ F = K_p \Delta X + K_d v \text{, where } \Delta X = X_n - X_c \tag{4} $$

In equation (1), X_n represents the next playback position to move to whereas X_c refers to the current position of the haptic device. The variable (v) represents the velocity of the haptic device. The constants K_p and K_d are the stiffness and damping constants, respectively. Notice that in equation (1), X represents a position vector $\{X, Y, Z\}^T$ and F represents the force vector $\{F_x, F_y, F_z\}^T$.

The implemented playback algorithm can be summarized as follows: when a new playback position data is read, the current position of the device is retrieved and a new driving force is computed and sent to the device. This loop repeats itself at the rate of 1kHz until the last playback position. One problem we have encountered was that a large difference between the initial device position and the startup playback position may result in an unstable movement. This issue happens only at the starting of the playback. To solve this problem, we used an interpolation technique that applies gradual forces to smoothen the movement of the device to its playback starting position. This is shown as a pseudo code in Figure 4.

```
//move device to playback starting point
Xs = GetPlaybackStartupPosition(Database);

Do {
     SendForceToDevice(Fs); //Fs is a step force
     Xd = GetCurrentPosition (Device);
     e = Xs - Xd;
   } while (e < K) // K is a minimum error

//start the playback
while (Xn = GetNewPosition(DataBase))
   {
     Vn = GetNewVelocity(DataBase);
     Xd = GetCurrentPosition (Device);
     F = Kp*(Xn - Xd) + Kd*Vn;
     SendForceToDevice(F);
   }
//Kp and Kd are stiffness and damping factors
```

Figure 4. A pseudo code for the playback approach

4. Performance Evaluation

In this section, the player performance analysis and the obtained results are presented and discussed. The devices used for this study were the Phantom Omni device [17], the MPB Freedom 6S [18] and the CyberForce system [19].

4.1 Haptic Devices

The MPB Freedom 6S device is a high fidelity force feedback device that is characterized by a position workspace of 17Wx 22Hx 23D cm, high position resolution (2 μm), and a sampling rate of 2 kHz [18]. The Phantom Omni device, in turn, has a smaller workspace, 16W x 12H x 7D cm, a position resolution of 0.055 mm, and a sampling rate of 1 kHz [17]. The CyberForce system is a hand exoskeleton device with a larger workspace (approximately 30.5x51 cm vertically oriented rectangle swept through 133 degrees) and a sampling rate of 170 Hz [19].

4.2 Player Quality

To evaluate the quality of the player, the Mean-Square Error (MSE) for trajectory reconstruction and its standard deviation were measured. The MSE represents the average error between the desired playback position and the recorded position. The MSE for the position is computed using equation (5) where (X_{ip}, Y_{ip}, Z_{ip}) and (X_{ir}, Y_{ir}, Z_{ir}) represent the playback and recorded positions, respectively. The standard deviation gives a measure of the spread of position errors over the playback trajectory.

$$MSE = \frac{\sum_{i=1}^{n} \sqrt{(X_{ip} - X_{ir})^2 + (Y_{ip} - Y_{ir})^2 + (Z_{ip} - Z_{ir})^2}}{n} \quad (5)$$

Figure 5 shows the MSE and standard deviation for the test trajectories. The trajectories represent the five Latin vowels (a, e, o, u, and i) and two numbers, one (1) and five (5). Each trajectory was recorded and played 5 times with the Omni device, and the average and the standard deviation of the MSE between recorded and playback trajectories were computed. We have noticed that the error was almost the same for uniform trajectories (around 6 mm), regardless of the character geometry. Under all tested conditions, the player remained stable.

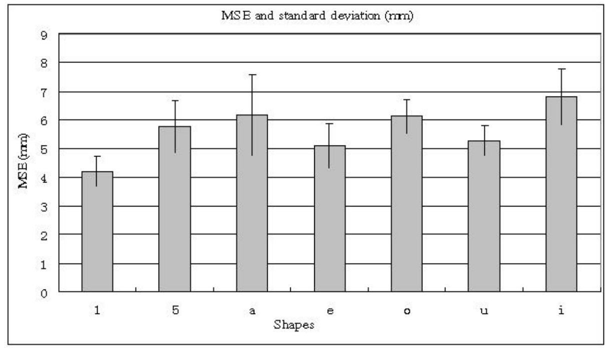

Figure 5. MSE and standard deviation for various shapes

4.3 Device Independence

To test the player independence of the haptic interface, we have recorded stimuli using the CyberForce system and the MPB Freedom 6S device and played them back with the Phantom Omni device. Figure 6 shows the MSE average errors when playing back data captured by the three haptic devices: the MPB Freedom 6S, the CyberForce system, and the Phantom Omni device. Even though the MSE and standard deviation are a bit larger when recording data using the CyberForce system, the stimulus pattern remained perceivable. A user can distinguish which vowel/number is played back.

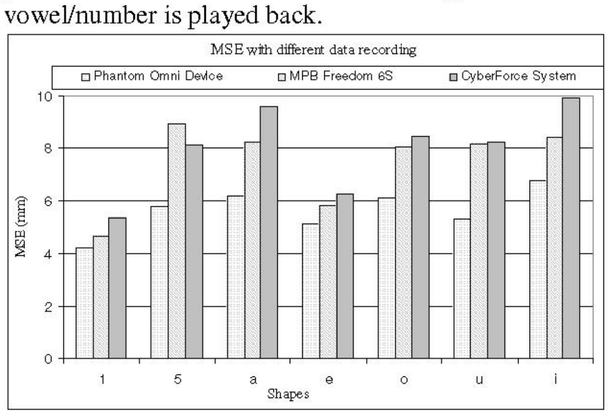

Figure 6. MSE error when the recording and player devices are different

When playing back the data collected from the CyberForce system we found that, compared to the MPB device, the playback accuracy had decreased. The average MSE of a trajectory captured by the MPB Freedom device was around 7.5 mm, whereas the average of a trajectory captured by the CyberForce system was around 8.5 mm. This difference in error was introduced due to the significant differences in the sampling rate and workspace of the two devices (170 Hz for the CyberForce system and 1 kHz for the Phantom Omni). Therefore, we conclude that the closer the specifications (particularly the workspace and sampling rates) of the recorder and player haptic devices, the better the playback quality.

5. Conclusion

In this paper, we have designed, implemented, and tested a device independent haptic player. The quality of the player has also been evaluated using various geometric trajectories, such as, vowels and number shapes. The results show the potential of this player with high-quality playback (around 6 mm error). Also, we show the ability of the player to play haptic stimuli recorded using the CyberForce system and the MPB Freedom 6S device and play them back using the Phantom Omni device. As discussed, stability is achieved in all tested conditions. On the other hand, it is shown that the quality of the player decreases when the workspace or sampling rate between recorder and player devices differ significantly (such as, the case with the CyberForce system and the Omni device).

As per future work, we plan to integrate audio/visual feedback in the player. This can be achieved by extending the HAML description to include audio/visual media descriptions. HAML is MPEG 7 compliant, so integrating audio/video should be a straightforward task. Moreover, synchronizing these media with haptic feedback is one of our immediate future works. Finally, we plan to explore the potential of haptic playback in the education sector by building a learning system for different calligraphies such as Arabic or Japanese.

References

[1] B. Gillespie, S. O'Modhrain, P. Tang, C. Pham, and D. Zaretsky, "The virtual teacher", In Proceedings of the ASME IMECE, Vol. 64, pp. 171–178, 1998.

[2] I. Oakley and S. O'Modhrain, "Contact IM: Exploring asynchronous touch over distance". In Proceedings of CSCW 2002, November, 2002, New Orleans, LA.

[3] M. Eid, M. Orozco and A. El Saddik, "A Guided Tour in Haptic Audio Visual Environment and Applications", Accepted for publication in the International Journal of Advanced Media and Communication, vol.1, n.3, 2007.

[4] C. Basdogan, S. De, J. Kim, M. Manivannan, H. Kim, and M.A. Srinivasan, 'Haptics in minimally invasive surgical simulation and training', IEEE Computer Graphics and Application, Vol. 24, Issue 2, pp 56 – 64, 2004.

[5] T. Dang, T.M. Annaswamy, and M.A. Srinivasan, "Development and evaluation of an epidural injection simulator with force feedback for medical training", Medicine Meets Virtual Reality, pp. 97–102, 2001.

[6] R.L. Williams, M. Srivastava, R. Conaster, and J.N. Howell, "Implementation and Evaluation of a Haptic Playback System', The Electronic Journal of Haptics Research, Vol. 3, No. 3, 2004.

[7] C. Teo, E. Burdet, and H. Lim, "A robotic teacher of chinese handwriting", In Symposium for Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 335–341, 2002.

[8] B. Bayart, A. Pocheville, and A. Kheddar, "An adaptive haptic guidance software module for I-Touch: example through an handwriting teaching simulation and a 3D maze", In proceedings of the Haptic Audio Visual Environments and their Applications, Ottawa, Canada, 2005.

[9] N. Hogan, "Impedance control: An approach to manipulation: Part i - theory, part ii - implementation, part iii – applications", ASME Journal of Dynamic Systems, Measurement and Control, 107:1–24, 1985.

[10] M. Corno and M. Zefran, "Haptic Playback: Modeling, Controller Design, and Stability Analysis", Proceedings of Robotics: Science and Systems, Philadelphia, USA, 2006.

[11] M. Eid, A. Alamri, and A. El Saddik, "MPEG-7 Description of Haptic Applications Using HAML", Proceedings of the HAVE2006, Ottawa, Canada, 2006.

[12] Y. Yokokohji, R. L. Hollis, and T. Kanade, "What you see is what you feel – development of a visual/haptic interface to virtual environment", In Proceeding of the IEEE Virtual Reality Annu. Int. Symp, pp. 46–53, 1996.

[13] D. Feygin, M. Keehner, and F. Tendick, "Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill", Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 40–47, 2002.

[14] Y. Yokokohji, R. L. Hollis, T. Kanade, K. Henmi, and T. Yoshikawa, "Toward machine mediated training of motor skills -skill transfer from human to human via virtual environment", In RO-MAN, pp. 32–37, 1996.

[15] Fayez El-Far, "Invictus: A Real-Time Haptic Data Repository Architecture", Masters Thesis, University of Ottawa, 2007.

[16] J. Craig, "Introduction to Robotics", Third Edition, Pearson Prentice Hall, ISBN-13: 978-0201543612, 2005.

[17] SensAble Technologies official Website: www.sensable.com, accessed March 3, 2007.

[18] MPB Technologies official Website: http://www.mpb-technologies.ca/, accessed March 3, 2007.

[19] Immersion Corporation official Website: http://www.immersion.com, accessed March 3, 2007.