

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



## Haptic Data Transmission based on the Prediction and Compression

Yonghee You and Mee Young Sung

*Department of Computer Science and Engineering  
University of Incheon  
South Korea*

### 1. Introduction

Even state-of-the-art haptic technology still suffers from a number of limitations, such as the high price and weight or size of the haptic interface, in addition to the limitations in work space and the lack of force feedback to the body. Moreover, when it comes to networking, the high bandwidths, low network latency, high stability, and the synchronization requirements of haptics are not met yet (Saddik, 2007).

Computer haptics is an emerging area of research that deals with the techniques for generating and displaying the “touch” of virtual environments and objects (Saddik, 2007), (You et al., 2007). Haptic collaborative virtual environment (HCVE) is an enhanced virtual reality space that supports sense of touch, which is called “haptic”. In HCVE, remote users connected over networks are able to collaborate by sharing touching experiences in addition to well-established audio and visual interfaces. The success of HCVE largely depends on timely transmission of haptic data despite time-varying network conditions such as delay, loss, and jitter.

One of the important challenges in HCVE data transmission over the Internet is the limited bandwidth. Haptic data is too bulky, relative to the available bandwidth. The situation should improve when better haptic based compression techniques are introduced. The demands for real-time simultaneous recording and transmission of voluminous data produced by multimedia haptics are pushing toward the exploration of haptic data compression. However, despite the stringent need for haptic data compression, the field is still in its infancy and many possibilities have emerged (Saddik, 2007).

The fact that the data generation frequency of haptic interface devices is extremely high, e.g. 1 KHz, makes the realization of successful HCVE more challenging. For seamless haptic data communication even under adverse network conditions, we propose a data compression method which contributes to solve the bandwidth limitations of haptic data transmission over the internet (You et al., 2007). Our method is based on mesh compression. Some related work is presented in the next section. In section 3, the details of our compression method are described, some experimental results are discussed in section 4, and we conclude our work in section 5.

## 2. Floating-Point Compression for Haptic Data Transmission

In this section, we briefly describe haptic rendering and some technologies such as haptic data transmission (You et al., 2007) and floating-point compression.

### A. Haptic Rendering

Haptic rendering refers to the group of algorithms and techniques that are used to compute and generate forces and torques in response to interactions between the haptic interface avatar inside the virtual environment and the virtual objects populating the environments (Saddik, 2007). The goal of haptic rendering is to enable a user to touch, feel, and manipulate virtual objects through haptic interfaces as realistically as possible (Mark et al., 1996), (Ruffaldi et al. 2006). A force-feedback device can generate kinesthetic information and temporal tactile information. By using these perceptual cues such as shape, stiffness, texture and friction, haptic rendering can render various properties of a virtual object. Unlike visual rendering, a minimum update rate to achieve realistic haptic rendering depends on the properties of virtual objects and a force-feedback device. Recommended update rates are 1 KHz and 5 KHz - 10 KHz for a rigid surface and a textured surface respectively. For a transformable object, it is advised to keep the rate as fast as you can.

### B. Haptic Data Transmission

In networked haptic applications, there exist various requirements such as high bandwidths, the low network latency, the high stability, and the synchronization. However, there has been extensive research related to the problems caused by the bandwidth limitations and latency (Hikichi et al., 2001), (Ishibashi & Asano, 2007).

Time-varying network conditions pose challenges to successful communication of haptic data (Hikichi et al., 2001). Adverse network links sometimes cause irregular force-feedback which deteriorates the haptic experiences (You et al., 2007). The transmission of the haptic data, which mainly consists of the position information of haptic device pointers, is basically similar to multimedia streaming. However, it is much more demanding because the haptic rendering rate required for satisfactory haptic experience is quite higher than that of graphic rendering; 1 KHz for haptics, in contrast to 30 Hz for graphics. To meet such challenges, there have been various research efforts. For group synchronization control, Y. Ishibashi et al propose virtual time rendering algorithms (Ishibashi & Asano, 2007). Hikichi et al employ a queue monitoring algorithm (Hikichi et al., 2001) designed for efficient haptic collaboration. However, these approaches have limitations that they are not able to cope with delay, loss and jitter at the same time.

### C. Floating-point Number Compression

In some environments, the fast transmission or the storage of large amounts of numerical data are required. Examples involve storing 3D geometry, sending statistical results over a network, and also sending haptic data which is composed of three floating-point numbers  $x$ ,  $y$  and  $z$ . Since such data are too bulky, it is necessary to find a way to compress them efficiently in a lossless manner. There are some approaches for compressing the floating-point numbers (IEEE 754, 1985).

Martin Isenburg, Peter Lindstrom, Jack Snoeyink from University of North Carolina have proposed a predictive geometry coding used for 3D mesh compression (Lindstorm & Isenburg , 2006), (Isenburg et al. , 2005), (Isenburg & Alliez, 2002). Paruj Ratanaworabbhan, Jian Ke, and Martin Burtscher have also designed a compression method of scientific floating-point data using a predictor and hash tables (Raranaworabhan et al., 2006). In addition, Costa Touma and Craig Gotsman from Israel Institute of Technology have developed a compression algorithm using entropy coding for triangle mesh compression (Touma & Gotsman, 1998). Even though all of these methods work differently, the main ideas of compression are mostly based on the characteristics of floating-point arithmetic. Those characteristics will be explained in the latter section.

3. Floating-Point Compression for Haptic Data Transmission

In networked haptic virtual environments, there exist various types of haptic devices. Some of those devices called stylus-type generate 3D positional information. The positional information usually consists of  $x$ ,  $y$  and  $z$  coordinates which are represented in single precision floating-point numbers. The positions of haptic pointer play an important role to produce the force-feedback, since the position itself or the collision between the haptic pointer and an object are main factors for calculating the magnitude and the direction of force-feedback.

Therefore, we usually need to transmit the positional information at the haptic rendering rate 1 KHz over the network in any networked haptic environment. In this case, we need to send at least 12 bytes per millisecond.

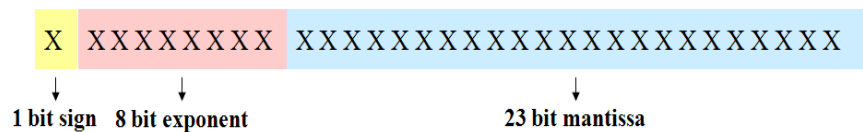


Fig. 1. Configuration of a single precision IEEE floating-point number

Floating-point number	Bit Representation
-81240646	1 10000101 01000100111101100110110
-80.796814	1 10000101 01000011001011111111000
11.638161	0 10000010 01110100011010111101000
11.478516	0 10000010 01101111010100000000000

Table 1. Examples of floating-point numbers in bit representation

Our approach for compressing the haptic data is to compress the bit-representations of single precision floating-point numbers. It is to reduce the size of the haptic data packet (12 bytes) that consists of three floating-point numbers  $x$ ,  $y$  and  $z$ . IEEE floating point is the most commonly used representations for real numbers on computers, including Intel-based PC's, Macintoshes, and most Unix platforms. For a single-precision 32-bit IEEE floating-point number, the sign  $s$  is a single bit that specifies whether the number is positive ( $s=0$ ) or negative ( $s=1$ ), the exponent  $e$  is an eight bit number with an added bias of 127 where 0 and

255 are reserved for un-normalized near-zero and infinite values, and the mantissa  $m$  is a twenty-three bit number that is used to represent 223 uniformly-spaced numbers within the range associated with a particular exponent (IEEE 754, 1985). Figure 1 shows the configuration of a single precision 32-bit IEEE floating-point number. As shown in Table I, we can recognize that when there are small differences between floating-point numbers. There occur only small changes in mantissa. Due to the configuration of a single precision 32-bit IEEE floating-point number, if the absolute difference between two floating-point numbers is small, the difference in mantissa is also small (IEEE 754, 1985).

We propose an algorithm to compress haptic data using the idea discussed above. If we only transmit the different bits of 3D positions of a haptic pointer, the packet may vary from 0 byte to 12 bytes (0~4 bytes for  $x$ ,  $y$  and  $z$ ), contrary to the fact that without the compression, we have 12 bytes every millisecond.

As discussed before, the position of the haptic pointer is sampled at 1 KHz and this indicates that the differences between consecutive floating-point numbers are relatively small. Figure 2 is the result from the experiment where we calculated the differences between consecutive positions of the haptic pointers. The number of positions that were used in the experiment was 6000. The maximum difference was 1.937969 while the average difference was 0.390084. This means that the differences are relatively small so changes in mantissa are also small. From the results of our experiment we found out that if we can make the differences smaller, then the amount of haptic data we need to transmit over a network will be smaller.

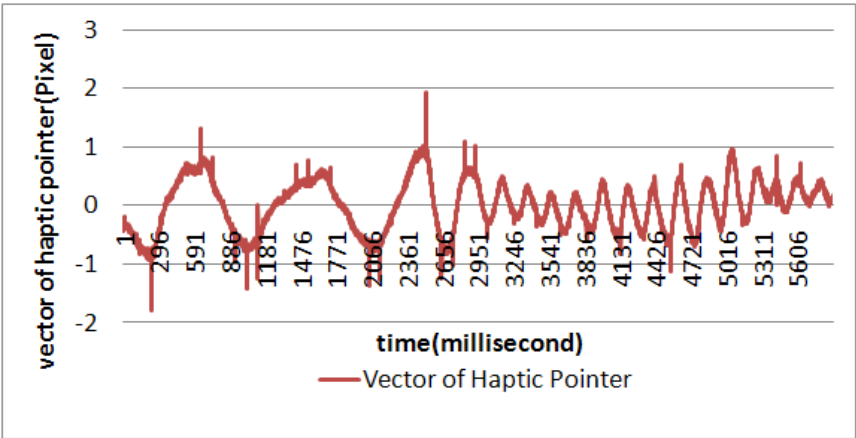


Fig. 2. Differences of consecutive haptic pointers

**A. Position Prediction for Haptic Pointer**

As discussed above, if we can get smaller differences we have smaller sizes of packets to send. In this section, we describe how the differences can be minimized by prediction.

The haptic pointer, which represents the movement of the haptic device moves freely according to the user’s will, so it seems to be impossible to predict the next position based on the previous positions. However, we can predict the next position with a certain acceptable error using the previous vector of the haptic pointer since haptic pointers are sampled very densely.

We propose a simple position prediction method in order to minimize the difference between two floating-point numbers which are used for compression. Our method works as in Figure 3. If we can calculate the previous vector of the haptic pointer, we can predict the

next position of the haptic pointer more precisely rather than just using the previous position

The prediction method is executed as follows:

$$x_n^p = x_{n-1} + v_n$$

(1)

$$v_n = x_{n-1} - x_{n-2}$$

(2)

where  $x_n^p$  is the predicted position while  $x_{n-1}$  corresponds to the previous position.  $v_n$  denotes the velocity of the haptic pointer's position. The velocity  $v$  is computed by averaging the previously received positions.

In order to verify the efficiency of applying the linear prediction method to the prediction of haptic pointer, we performed an experiment comparing the error of using the prediction value  $x_n^p$  and using the previous position  $x_{n-1}$  to compare with the next position  $x_n$ . The results of the experiment are shown in Figure 4. If we category one using the previous position as type 1 and the other using its prediction value as type2. The average errors are 0.249083 and 0.038881 for type 1 and type 2 respectively. Since type 2 shows a small error this means that using a linear prediction method will result in the efficiency of compression.

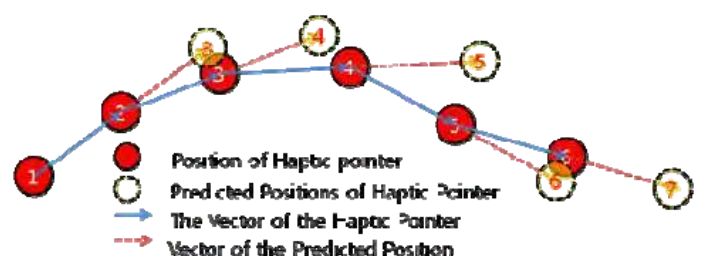


Fig. 3. Prediction of Haptic Pointer

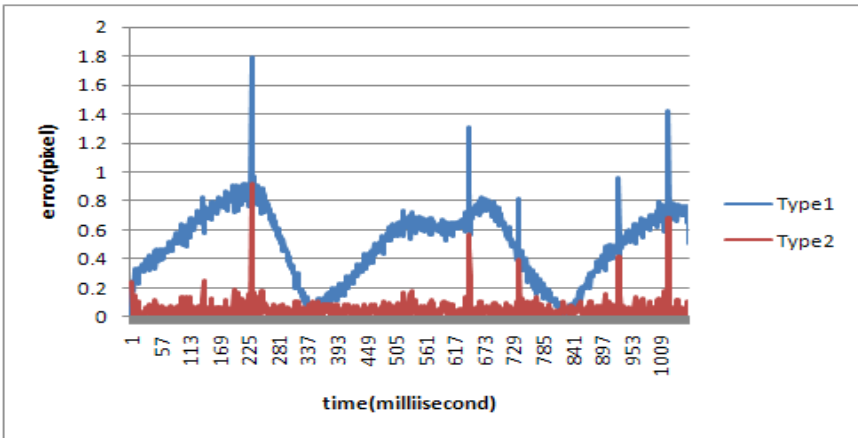


Fig. 4. Results of an Experiment on Prediction of Haptic Pointer

**B. Predictive Haptic Data Compression**

We compress the floating-point number by extracting the efficient bits from a floating-point number and its prediction, since the small difference in numbers results in a small difference



in bit-representation of a floating-point number (IEEE 754, 1985). If the absolute difference between a floating-point number and its prediction value is small, the difference in mantissa is also small and we only need to transmit the difference in bit representation. In this section we examine how the compression of the haptic data proceeds for networked haptic applications. For the ease of explanation, we assume that there is a networked haptic application based on the master-slave model. Figure 5 shows the process of the haptic data

Number of Effcient bits		1~8bit	9~16bits	17~24bits	The whole data
Allocate d bytes		1byte	2bytes	3bytes	4bytes
Header	X	0x00	0x40	0x44	0x08
	Y	0x00	0x20	0x22	
	Z	0x00	0x10	0x11	

Table 2. Examples of floating-point numbers in bit representation

transmission.

**a) Initialization**

To let both the master and the slave have the initiation values, a master sends the whole positional information at least twice. The init value will be stored in Prediction Module for further prediction.

**b) Prediction**

Prediction Modules in a master and a slave predict the next position based on the previously received position. For example as in Figure 5, the Prediction Module produces the prediction value -77.916992.

**c) Encoder**

With the actual position from the haptic device and the predicted position, the encoder performs an exclusive-or operation to compress the floating-point number  $x$  to the smaller number of bits; for example, the actual number -77.916916 which is 32bit long, can be reduced to 8 efficient bits and only those 8 bits are required to be sent. Encoder also performs the exclusive-or operation iteratively for  $y$  and  $z$ .

**d) Packet Module**

Packet Module is responsible for organizing the compressed data into a packet and putting a header onto the packet. Packet Module first calculates the number of the efficient bits of the results from Encoder. In Figure 5, only 11110110 has to be sent to the receiver. However, for the convenience of packetizing, we provide a packetizing policy as in Table 2. If the number of efficient bits are between 1~8, 2~16 and 17~24, we put the efficient bits in 1, 2, and 3 bytes. In addition, if the exponents of the actual position and the predicted position are different, the master transmits all of the positional information without compression for reliability. Packet Module also add to a packet a one byte header which describes the size of the efficient bits for  $x$ ,  $y$  and  $z$ .

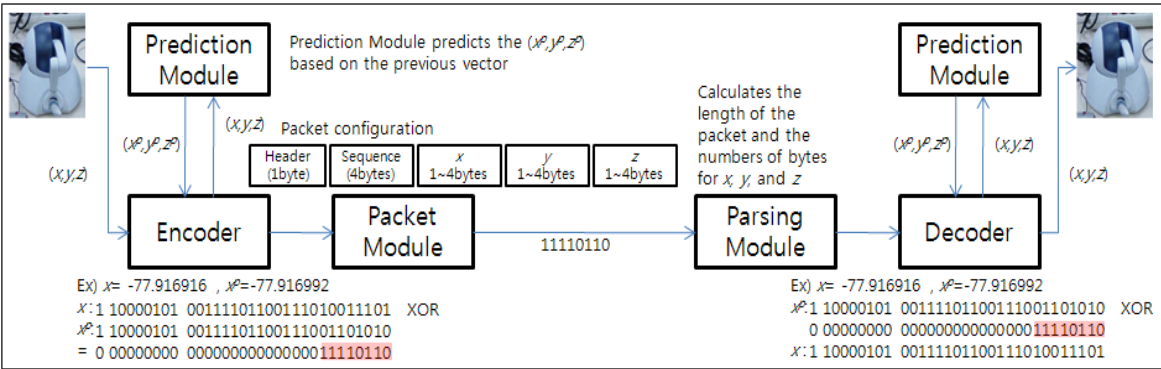


Fig. 5. The Process of the Haptic Data Transmission

- e) **Parsing Module**  
Once the slave receives a packet, Parsing Module disassembles the packet into a header and efficient bits of  $x$ ,  $y$  and  $z$ .
- f) **Decoder**  
Decoder gets a predicted position and performs an exclusive OR operation with the efficient bits of  $x$ ,  $y$  and  $z$  added with 0s in the front. Then, we finally regenerate the original position from the relatively small amount of bits.  
As discussed above, if we can use the prediction and Exclusive Or operation properly, we don't have to send the 12 bytes( $x$ ,  $y$  and  $z$ ) of long floating-point data for bulky haptic data which often cause network delays, losses, and jitters.

4. Experiment

We performed an experiment to evaluate the efficiency of the proposed compression method. In this section, we explain the test application, software configuration, hardware setup, experiment and results.

A. Test Application

The test application used in the experiment is a haptic device tele-operation application. It is composed of two parts: a master and a slave. The master sends the motion data to the slave and the slave outputs the force feed-back according to the motion data which consists of three floating-point numbers. The slave sometimes sends control messages to the master such as connect, ready, NAK and so forth to connect or control errors. The overview of the application is shown in Figure 6. This application is suitable for the test because we can feel the effects of network errors through both the sense sight and touch.



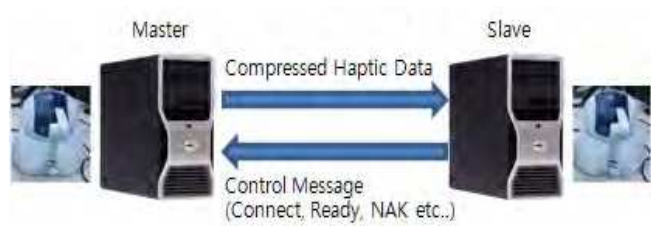


Fig. 6. Overview of the Test Application

**B. Software Configuration**

The software configuration of the test application is shown in Figure 7. The compression method and network module is developed dependant on C++. The OpenHaptics Toolkit (SDK) from SensAble for haptic rendering from Sensable Technologies, and the QUANTA (The Quality of Service Adaptive Networking Toolkit) are used in the application.

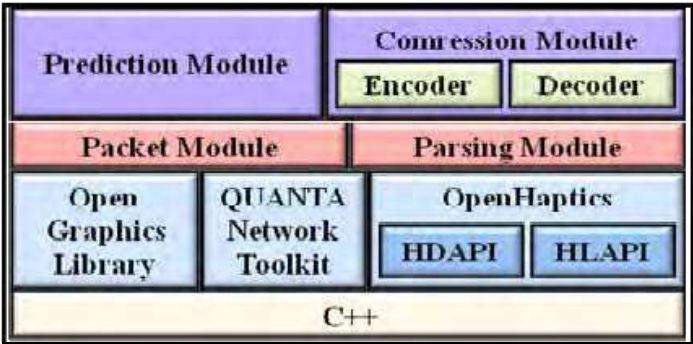


Fig. 7. Software Configuration

**C. Hardware Setup**

- Sensable PHANToM Omni
- Dell Precision PWS380 Intel® Pentium 4 CPU 3.20GHz, 1.00 GB RAM
- NVIDIA Quadro FX 1400
- Microsoft Windows XP Professional Version 2002 Service Pack 2

**D. Experiments**

We performed an experiment to evaluate the efficiency of the proposed method. The master mainly transmits its positional information to the slave and the haptic device of the slave follows the motion of the master. In order to assess our method, we measure the size of packets sent in six cases; five cases depending on the motions of the haptic pointer (1: stay static, 2: move slow, 3: draw 8, 4: draw \*, 5: move fast and irregularly) and one without compression (6).

Figure 8 shows the result of the transmission experiment for approximately four minutes. If we do not consider the UDP header, the average bandwidths used for the transmission are 7992.59bps (compression ratio 47.06%), 10574.76bps (62.27%), 11201.43bps (65.96%), 11441.18bps (67.37%), 11808.37bps (69.53%), and 16983 bps in the cases of 1, 2, 3, 4, 5, and 6 respectively. The result leads us to confirm that our compression method works efficiently for networked haptic applications.

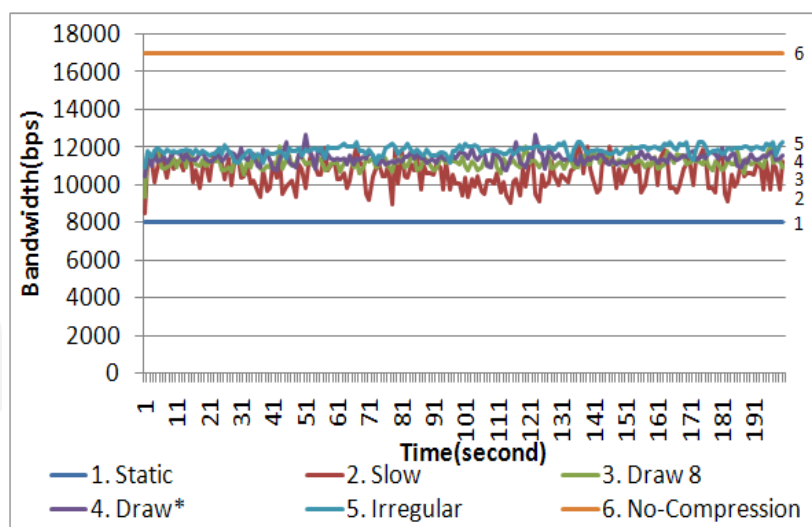


Fig. 8. Results of the experiment

## 5. Conclusion

We propose a floating-point haptic data compression in order to reduce the bandwidth which sometimes causes packet losses, jitters, and delays when being exceeded. Our compression method is based on the idea that the bit representations of consecutive floating-point numbers change slightly from the most significant bit. The exclusive OR operation helps extract the efficient bits from floating-point numbers and our prediction method produces a smaller difference between two consecutive floating-point numbers. Some experiments on our proposed method lead us to conclude that our method provides relatively high compression ratio, approximately 47.06%~69.53%. This will help the haptic data transmission even under limited bandwidth.

For future work, we plan to work on the QoS algorithms which cope with significant delay, jitter, and loss for haptic applications (Cruz-Neira et al., 1993), (Marsh et al., 2004), (Fujimoto & Ishibashi, 2005).

## 6. References

- Abdulmotaleb El Saddik. (2007). "The Potential of Haptics Technologies," IEEE Instrumentation & Measurement Magazine, IEEE, pp. 10-17
- Yonghee You, Mee Young Sung, and Kyungkoo Jun. (2007) "A scheme for Haptic Data Transmission under Various Network Conditions," In Proceedings of IEEE-ICME 2007 (IEEE International Conference on Multimedia & Expo), IEEE, Beijing, China, pp. 2238-2242, July 2-5.
- W. Mark, S. Randolph, M. Finch, J. V. Verth, and R. M. Taylor. (1996) "Adding force feedback to graphics systems: Issues and solutions," Proceedings of ACM SIGGRAPH '96 Conference, ACM press, pp. 447-452.
- Emanuele Ruffaldi, Dan Morris, Timothy Edmunds, Federico Barbagli and Dinesh K. Pai. (2006) "Standardized Evaluation of Haptic Rendering Systems," In Proceedings of the Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, IEEE, pp. 225-232.

- K. Hikichi et al. (2001) "Architecture of Haptics Communication System for Adaption to Network Environments," In Proceedings of IEEE International Conference on Multimedia and Expo Proceedings, IEEE, pp. 744-747.
- Yutaka Ishibashi and Toshio Asano. (2007) "Media Synchronization Control with Prediction in a Remote Haptic Calligraphy System," In Proceedings of ACE 2007, ACM press, pp. 79-86.
- IEEE 754 (1985) : Standard for binary floating-point arithmetic, IEEE.
- Peter Lindstrom and Martin Isenburg. (2006) "Fast and Efficient Compression of Floating-Point Data," In IEEE Transactions on Visualization and Computer Graphics, IEEE, vol. 12, No. 5, pp. 1245-1250.
- Martin Isenburg, Peter Lindstrom and Jack Snoeyink. (2005) "Lossless Compression of Predicted Floating-Point Geometry," In Computer-Aided Design, Elsevier Science, volume 37, pp. 869-877.
- Martin Isenburg and Pierre Alliez. (2002) "Compressing Polygon Mesh Geometry with Parallelogram Prediction," In Proceedings of Visualization, IEEE, pp. 141-146.
- Paruj Ratanaworabhan, Jian Ke, and Martin Burtcher. (2006) "Fast Lossless Compression of Scientific Floating-Point Data," In Proceedings of Data Compression Conference, pp. 133-142.
- Costa Touma and Craig Gotsman. (1998) "Triangle Mesh Compression," In Proceedings of Graphics Interface '98, pp. 26-34.
- E. He, J. Alimohideen, J. Eliason, N. K. Krishnaprasad, J. Leigh, O. Yu, and T. A. DeFanti. (2003) "Quanta a toolkit for high performance data delivery over photonic networks," In Future Generations Computer Systems, Special Issue: iGRID 2002, FGCS, volume 19, number 6, pp. 919-933.
- C. Cruz-Neira, D. Sandin, and T. A. DeFanti. (1993) "Surround-screen projection-based virtual reality: The Design and Implementation of the CAVE," In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, ACM press, pp. 135-142.
- J. Marsh, M. Glencross, S. Pettifer, R. Hubbard, J. Cook, and S. Daubrebet. (2004) "Minimizing latency and maintaining consistency in distributed virtual prototyping," In Proceedings of ACM SIGGRAPH Conference on the Virtual Reality Continuum and its Applications in Industry (VRCAI), ACM press, Singapore, pp. 386-389.
- M. Fujimoto, Y. Ishibashi. (2005) "Packetization Interval of Haptic Media in Networked Virtual Environments," In ACM NetGames, ACM press, pp. 1-6. Information



## **Advances in Haptics**

Edited by Mehrdad Hosseini Zadeh

ISBN 978-953-307-093-3

Hard cover, 722 pages

**Publisher** InTech

**Published online** 01, April, 2010

**Published in print edition** April, 2010

Haptic interfaces are divided into two main categories: force feedback and tactile. Force feedback interfaces are used to explore and modify remote/virtual objects in three physical dimensions in applications including computer-aided design, computer-assisted surgery, and computer-aided assembly. Tactile interfaces deal with surface properties such as roughness, smoothness, and temperature. Haptic research is intrinsically multi-disciplinary, incorporating computer science/engineering, control, robotics, psychophysics, and human motor control. By extending the scope of research in haptics, advances can be achieved in existing applications such as computer-aided design (CAD), tele-surgery, rehabilitation, scientific visualization, robot-assisted surgery, authentication, and graphical user interfaces (GUI), to name a few. Advances in Haptics presents a number of recent contributions to the field of haptics. Authors from around the world present the results of their research on various issues in the field of haptics.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yonghee You and Mee Young Sung (2010). Haptic Data Transmission Based on the Prediction and Compression, Advances in Haptics, Mehrdad Hosseini Zadeh (Ed.), ISBN: 978-953-307-093-3, InTech, Available from: <http://www.intechopen.com/books/advances-in-haptics/haptic-data-transmission-based-on-the-prediction-and-compression>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen