International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 24 (2017) pp. 14948-14954
© Research India Publications. http://www.ripublication.com

Haptic Interactions for Probing Real Objects in Remote Places

A Ram Choi*, Chan Woo Kim*, Migyeong Gwak and Mee Young Sung***

**Incheon National University, Incheon, Korea.*
***Computer Science, University of California, Los Angeles, Los Angeles, U.S.A.*

Abstract

Past efforts on haptic interactions were only concerned with the haptic feedback for simulated virtual objects. In this paper, we propose a method for haptic interactions with real objects in remote places. We developed a tele-haptics system including a server and a client distant probing. This paper explains the system architecture and the haptic rendering method based on the data obtained from remote sites. A number of user studies on rigid objects and deformable objects are performed in order to evaluate the quality of haptic transmission and tactile perception of the proposed method. As a result, we have concluded that our method is adequate for perceiving different stiffness of objects in distance and tactile feedbacks of rigid objects. However, the remote probing on deformable objects causes some unstable vibration. It is also observed that the responsiveness is insufficient at the moment of the first collision. For our future works, we will solve the problems indicated in our experiments such as the unstable vibration and the low responsiveness. We will also focus on the quantitative analysis of the proposed method for remote haptic probing and its responsiveness for haptic interactions.

Keywords: Tele-haptics, Haptic rendering, Haptic feedback, Perception test, Responsiveness

INTRODUCTION

Recently, there has been an exponential increase in the demand of virtual reality technology in the IT (information technology) industries. The IT industries mainly demand HMD (Head Mount Display) devices which take play role in the visual aspect of virtual reality. Most applications of virtual reality can enable users to experience virtual reality with only visual and auditory senses. However, in certain applications, tactile feedbacks may create the more realistic virtual environment. Haptics is a technique that allows users to perceive tactile sensations regarding force and movement. Haptics technology plays an important role in simulations in virtual environments, such as in virtual surgery training [1] or haptic gaming system [2]. In the implementation of haptics technology, haptics devices take an integral part. Typical haptic devices include Touch (former PHAMTOM Omni) [3] and Touch X (former PHANTOM Desktop) [4] of 3D Systems, and Falcon [5] of Novint Technologies, as shown in Figure 1.

(a) **(b)** **(c)**

Figure 1: Haptic devices: (a)Touch, (b) Touch X, (c) Novint Falcon

When a haptic device collides with a virtual object, the haptic device performs a haptic rendering [6] process by calculating the reaction force of the moment. Then it provides a haptic feedback to users, thereby enabling users to feel the touch. To deliver smooth, sophisticated tactile feedback to users, an update rate of 1000 Hz or higher should be maintained. But in the case of real-time haptic application through the network, the responsiveness of haptic feedback may be degraded due to delay, loss, etc. on the network.

In order to solve this haptic feedback degradation problem, various studies are being conducted to enable smooth feedback in network-based haptics applications [7][8][9][10][11][12][13]. One of the earliest studies is the “Algorithms for Network-Based Force Feedback” by researchers at MIT [9]. This study proposed a “Passive Transmission Line Modeling” and a “Haptic Dead-Reckoning” to reduce errors of haptic feedback caused by network delay. Also, researchers at Ottawa University proposed a smoothed synchronous collaboration transport protocol (SCTP) for tactile data transmission [10]. Experimental results of this study show that the proposed protocol can increase the efficiency of tactile based collaborations compared to the existing transmission protocols such as SCTP or Light TCP (transmission control protocol). The University of Waterloo and Handshake VR Inc. has commercialized the Handshake proSENSE™ 1.3, which is a tele-haptic toolbox [11]. This study provides various network delay compensation techniques and enables users to create easily network-based tactile collaboration programs in drag-and-drop fashion.

One of our studies on haptic collaboration in a networked environment is the “An Integrated Haptic Data Transmission in Haptic Collaborative Virtual Environments” [12]. This study analyzes the changes of delay, jitter, and loss in networked haptic collaborative applications. It also proposes an algorithm that can be applied to the real network traffic. The lossy and jittered packets are compensated by a simple linear prediction method to reduce errors caused by the packet loss and jitters. The concurrency problem among clients caused by the delay in network collaborations is solved by setting the buffering time.

Another study of ours titled "MuseSpace: a touchable 3D museum with maximum usage of haptics" [13] provides diverse haptic interactions in networked environments. This study implements a virtual exhibition space in 3D and enables the exhibitions represented by 3D objects which can be experienced by visual, auditory, and tactile sensations. It also allows users to interact with other users on a real-time basis through the network.

Usually, a haptic device is used for transmitting the sensation of touch of an object in the virtual environment to users. However, unlike such general usage, in this paper, we propose a method to use devices in transmitting tactile sensations of actual objects in distant places, not in the virtual environment, to users. Onward in this paper, Section 2 describes the outline of the proposed system; Section 3 describes the implementation; Section 4 describes the experiments; finally, Section 5 states our conclusion.

SYSTEM OVERVIEW

The purpose of this study is to enable users to sense the feeling of touch of real objects located in remote places using a haptic device. The proposed system is divided into a server and a client. The server takes charge of user interactions and the client is responsible for user's control to objects in remote locations. The server consists of a computer and a haptic device, and the client is additionally equipped with a force sensor and a webcam.

Figure 2 illustrates the system configuration of the proposed system.

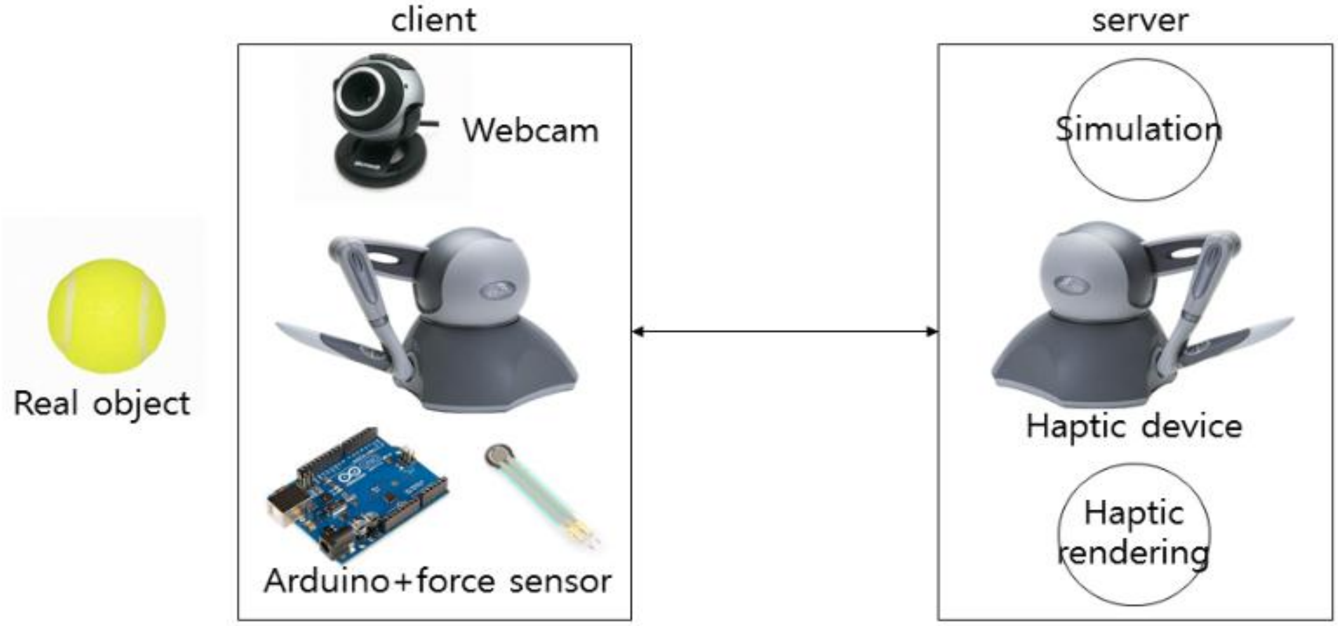


Figure 2: System configuration

The following Figure 3 summarizes the conceptual structure of the proposed system.

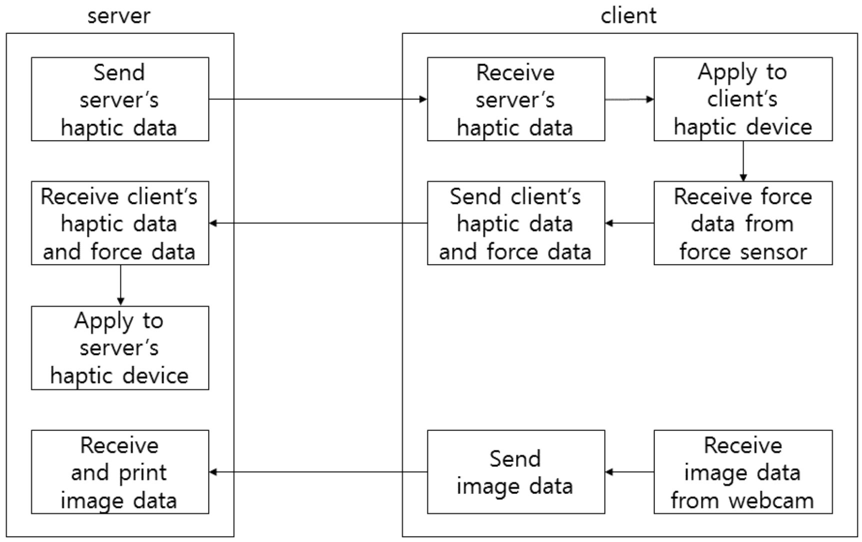


Figure 3: Conceptual structure

The process of live video transmission is performed separately from the haptic data transmission. It is a process of receiving video information captured by the camera of the client and displaying the live video information on the server's monitor. The user can recognize the situation of the remote place via live video and can perform some adequate control to the remote situation.

The camera of the client can work as an eye of a remote mobile probing robot and the haptic device can be an arm of it. Since the client's haptic device only needs to indicate the position of the server's haptic device, it can be replaced by another mechanical device such as robotic arm rather than the the haptic device. Therefore, this study can be applied to autonomous mobile robots or robot vehicles for probing dangerous places.

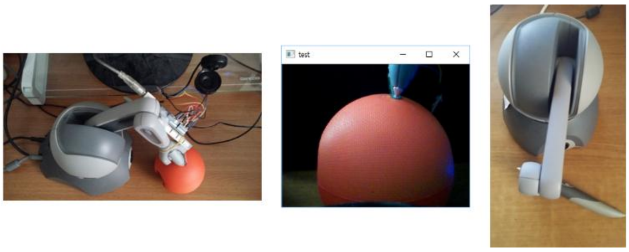
IMPLEMENTATION

This system uses CHAI3D [14], which is a cross-platform C++ simulation framework, to construct our software environment for the proposed system. Communication between the server and the client is implemented using the Windows socket library. Since the protocol requires real-time fast processing, UDP (user datagram protocol) [15] is chosen rather than TCP (transmission control protocol) [16]. To obtain the reaction force against the object at the remote place, the client's haptic pointer is attached to a force sensing register (Interlink Electronics FSR 400) [17] which is controlled by the Arduino Uno microcontroller [18]. The process of capturing, encoding, and decoding video information from a webcam is implemented in OpenCV [19]. The computing environment for our experiments is shown in Table 1.

Table 1: Computing environment for experiments

Heading level	Server	Client
OS	Windows 10	Windows 10
CPU	Intel i7-5820K 3.30GHz	AMD FX-8370 4.00GHz
RAM	8GB	4GB
haptic device	PHANTOM Omni	PHANTOM Omni
force sensor	Windows 10	Arduino Uno + Interlink Electronics FSR 400
webcam	Intel i7-5820K 3.30GHz	Skydv 300 cam

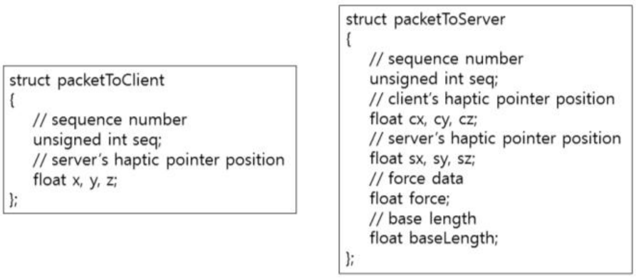
The pictures of haptic devices are presented in Figure 4. Figure 4 (a) is the haptic device of the client, Figure 4 (b) is the screenshot of the server monitor, and Figure 4 (c) is the haptic device of the server.



(a) **(b)** **(c)**

Figure 4: System devices (a) haptic device of the client, (b) screenshot of the server monitor, (c) haptic device of the server

The information transmitted from the server to the client over the network is defined as shown in the left data structure of Figure 5. The right data structure of Figure 5 illustrates the information transmitted from the client to the server.

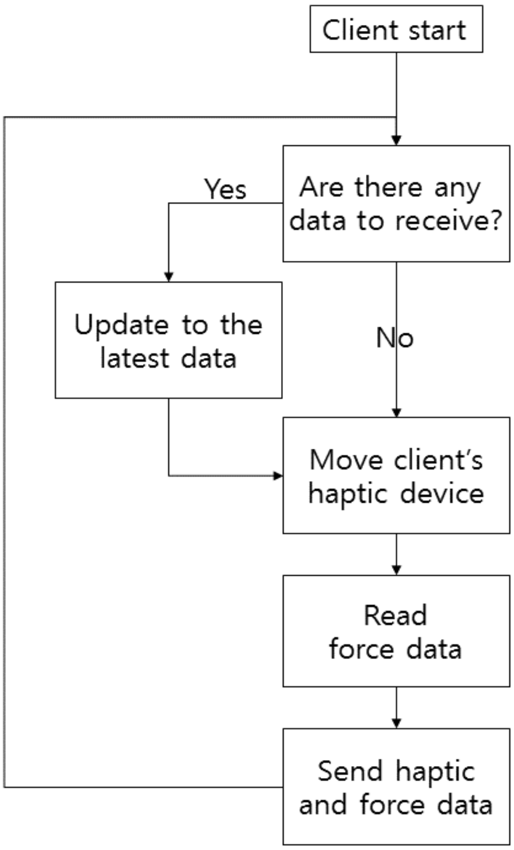


```
struct packetToClient
{
    // sequence number
    unsigned int seq;
    // server's haptic pointer position
    float x, y, z;
};
```

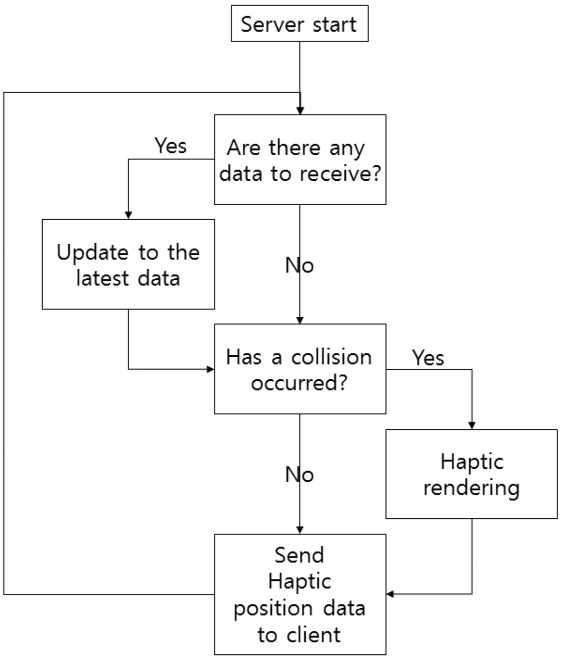
```
struct packetToServer
{
    // sequence number
    unsigned int seq;
    // client's haptic pointer position
    float cx, cy, cz;
    // server's haptic pointer position
    float sx, sy, sz;
    // force data
    float force;
    // base length
    float baseLength;
};
```

Figure 5: Definition of the haptic data packets (left: the packet structure from server to client, right: the packet structure from client to server)

The process of transmitting, receiving, and processing of the above haptic data is presented in Figure 6.



(a)



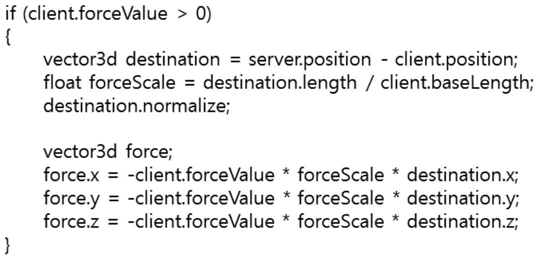
(b)

Figure 6: Process Flows of the client and the server for the haptic data transmission

The client repetitively examines whether the data inputted from the server exist. If it detects the corresponding data, it updates its position data with the most recent data from the server and moves its haptic pointer to the most recent position. It reads the force value from the force sensor at the position where the haptic pointer is moved to, and it transmits that force value and the current position value to the server. This process is repeated until the application ends.

Meanwhile, the server continuously checks whether the data received from the client. If the collision occurs on the client side, the server performs the haptic rendering according to the client's probing data. If no collision occurred, the current position of the server's haptic pointer is transmitted to the client. This process is also repeated until the application ends. The haptic rendering of the server is the process of calculating the force feedback so that the user can feel the tactile sensation of actual objects at the remote sites.

The algorithm is summarized in the following Figure 7.



```
if (client.forceValue > 0)
{
    vector3d destination = server.position - client.position;
    float forceScale = destination.length / client.baseLength;
    destination.normalize;

    vector3d force;
    force.x = -client.forceValue * forceScale * destination.x;
    force.y = -client.forceValue * forceScale * destination.y;
    force.z = -client.forceValue * forceScale * destination.z;
}
```

Figure 7: Haptic rendering algorithm of the server

The haptic rendering algorithm of the server starts with the examination whether the collision is detected or not, by checking the values of the force sensor received from the client. If this value is 0, no collision has occurred. When a collision occurs, the direction vector is obtained by subtracting the current server position and the current client position. The force value is calculated by multiplying the normalized value of the direction vector by the force vector. The force vector is calculated by the production of the force sensor value and the force scale value. The force scale value is the length of the current direction vector divided by the length of the base vector that corresponds to the direction vector of the moment of the first collision.

Figure 8 illustrates the calculation of force values. If the haptic pointer of the server moves from *sp*0 (server position 0) to *sp*1, the after *sp*2, along with the current *cp* (client position), the force scale values are calculated by dividing the current force vector by the length of the base vector.

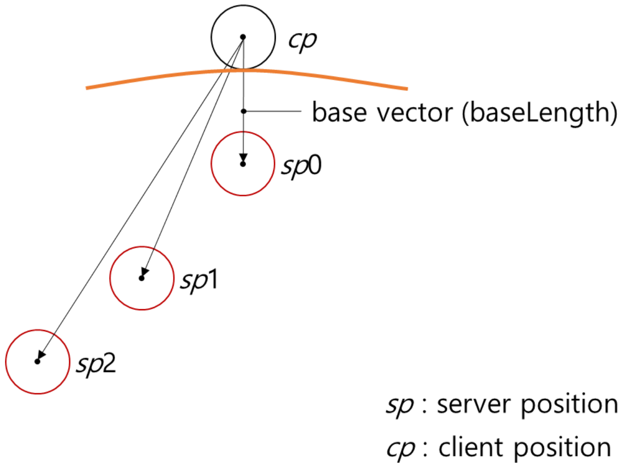


Figure 8: Calculation of the force scales

The purpose of multiplying the force scale is to interpolate the discrete force values received from the client to the continuous haptic values according to the fast haptic rendering frequency (1000Hz). The reason why the discrete force values occur is that the haptic rendering speed is faster than the update rate of the most recent position data. The final value of the force calculation is applied to the server's haptic device.

EXPERIMENTS

With different degrees of rigidity; a solid book, a rubber ball, and a sponge in Figure 9. The user tests were conducted according to the following survey questions.

- *The position synchronization*: Is the position synchronized between the server's haptic pointer and the client's haptic pointer?
- *The smoothness of the haptic rendering*: Is the change of force feedback smooth?
- *The sensibility of the stiffness*: Does the degree of rigidity feel similar to reality?
- *The haptic reality for deformable objects*: In the case of deformable objects, is the feeling of the deformation realistic?

Figure 9: Solid book, rubber ball, sponge

Through the user studies, various feedbacks are gathered from the test participants. In general, test participants felt adequate tactile sensations on the three different objects (a solid book, a rubber ball, and a sponge). They also said that the synchronization of positions of two haptic pointers was excellent enough. In addition, they evaluated that the haptic feedback was transmitted smoothly over the rigid objects, but it was not satisfactory over the deformable objects. Also, they pointed out that some unstable vibration occurred occasionally, and the responsiveness of the system at the moment of the first collision was poor.

Table 2 presents the results of the experiment for calculating the Euclidean distance between the two haptic pointers of the server and the client. The data is collected at intervals of 0.1 second (10Hz) during 15 seconds. In our experimental workspace, the haptic pointer can move between -0.09 to 0.0945 (the range of 0.1845) with respect to the x-coordinate, -0.22 to 0.2145 (the range of 0.4345) with respect to the y-coordinate, and -0.1 to 0.21 (the range of 0.31) with respect to the z-coordinate. The results of this experiment lead us to conclude that the proposed system can synchronize relatively well with the average error of 0.00679891 (Euclidian distance) between two haptic pointers. Note that the maximum Euclidean distance of this experimental space is 0.56473935 and the simple calculation of the probable error ratio is 1.2×e-10.

Table 1: Error of the position synchronization

Minimum error	Maximum error	Average error
0.00143685	0.03468801	0.00679891

Figure 10 represents the distance errors between two haptic pointers of the server and the client.

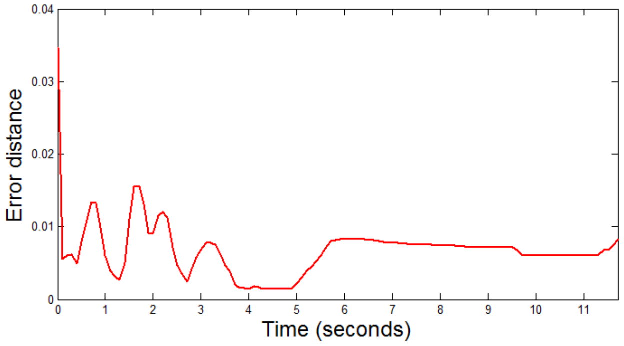


Figure 10: Distance error graph

CONCLUSION

In this paper, we propose a system that allows users to directly feel the tactile sensations of real objects located at remote sites using networked haptic devices. The system is composed of a server and a client and it performs a haptic rendering process based on the data obtained from the remote client. We conducted a number of experiments to evaluate the proposed system. As a result, we have concluded that our system can provide relatively good haptic feedbacks on a rigid object, while some unstable vibrations may occur for deformable objects. The experimental results also indicate some latencies of the responsiveness at the moment when the first collision occurs.

This study can contribute to the development of autonomous mobile robots or robot vehicles for probing dangerous places. The camera of the client in our system is analogous to the eye of the robot, while the haptic device of the client is similar to a functional robotic arm.

In our future works, we plan to solve the problems that arose during the experiments and to perform quantitative evaluation of the proposed haptic rendering and of the responsiveness test.

ACKNOWLEDGEMENT

This work was supported by Incheon National University Research Grant in 2013 (No. 20130756). The authors would like to express sincere thanks to Mr. Doug Jae Yoo for numerous enlightening discussions and helpful comments.

REFERENCES

[1] LapSim®: The Proven Training System. *https://surgicalscience.com/systems/lapsim/* (Retrieved June 1, 2017).

[2] S. M. Kim, M. Y. Sung, “A Haptic Gaming System for Tactile Textures and 3D Shapes Discrimination,” International Journal of Multimedia & Ubiquitous Engineering, Vol. 9 Issue 9, pp. 319-334, September 2014.

[3] The Touch™ Haptic Device. *https://www.3dsystems.com/haptics-devices/geomagic-touch/* (Retrieved June 1, 2017).

[4] The Touch X Haptic Device. *https://www.3dsystems.com/haptics-devices/geomagic-touch-x* (Retrieved June 1, 2017).

[5] Novint Falcon, *http://www.novint.com/index.php/products/novintfalcon/* (Retrieved June 1, 2017).

[6] K. Salisbury and F. Barbagli, “Haptic rendering: introductory concepts,” Computer Graphics and Applications, IEEE vol. 24, no. 2, pp. 24-32, 2004.

[7] G. Vineet, N. Jayakrishnan and C. Subhasis, “Opportunistic Adaptive Haptic Sampling on Forward Channel in Telehaptic Communication,” 2016 IEEE Haptics Symposium, Philadelphia, Pennsylvania, USA, April 8-11, 2016.

[8] D. Sabine, W. A. William and F. A. Aldo, “Gaze-based teleprosthetic enables intuitive continuous control of complex robot arm use: writing & drawing,” 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics, UTown, Singapore, June 26-29, 2016.

[9] P. W. John, J. K-S. Robert, A. K. Marc and H. Neville, “Algorithms for Network-Based Force Feedback,” Proceedings of the fourth PHANTOM User Group Workshop, MIT, (1999) October 9-12.

[10] B. Azzedine and M. Haifa, “An Efficient Hybrid Multicast Transport Protocol for Collaborative Virtual Environment with Networked Haptic,” International Workshop on Haptic Audio Visual Environments and their Applications, Ottawa, Canada, (2006) November 4-5.

[11] Mehran A. and Kevin T., Haptic enabled robotics training system and method, U.S. Patent 20,090,253,109, October 8, 2009.

[12] Y. H. You, M. Y. Sung, K. K. Jun, “An Integrated Haptic Data Transmission in Haptic Collaborative Virtual Environments,” Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference, July 11-13, 2007.

[13] Y. H. You, M. Y. Sung, K. K. Jun, and S. R. Lee, “MuseSpace: a touchable 3D museum with maximum usage of haptics,” Proceeding SIGGRAPH '07 ACM SIGGRAPH 2007, San Diego, California, posters, Article No. 154, ACM, August 05-09, 2007.

[14] CHAI 3D. *http://www.chai3d.org/* (Retrieved June 1, 2017).

[15] J. Postel, “User Datagram Protocol,” RFC 768, 1980.

[16] G. C. Vinton and E. K. Robert, "A Protocol for Packet Network Intercommunication," Transactions on Communications, IEEE vol. Com-22, no. 5, (1974), pp. 637-648.

[17] FSR ® 400 Series Data Sheet. *https://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf* (Retrieved June 1, 2017).

[18] Arduino UNO. *https://www.arduino.cc/en/Main/ArduinoBoardUno/* (Retrieved June 1, 2017).

[19] OpenCV. *http://opencv.org/* (Retrieved June 1, 2017).