Hindawi Publishing Corporation
Advances in Human-Computer Interaction
Volume 2016, Article ID 8608972, 6 pages
http://dx.doi.org/10.1155/2016/8608972

Research Article

Capturing the Perceived Phantom Limb through Virtual Reality

Christian Rogers,[1] Jonathan Lau,[1] Denver Huynh,[2] Steven Albertson,[2] James Beem,[3] and Enlin Qian[4]

[1]*Computer Information & Graphics Technology, Purdue School of Engineering and Technology, Indiana University-Purdue University Indianapolis (IUPUI), 799 West Michigan Street, Indianapolis, IN 46202, USA*
[2]*Department of Computer & Information Science, Purdue School of Science, Indiana University-Purdue University Indianapolis (IUPUI), 799 West Michigan Street, Indianapolis, IN 46202, USA*
[3]*Department of Computer Engineering, Purdue School of Engineering and Technology, Indiana University-Purdue University Indianapolis (IUPUI), 799 West Michigan Street, Indianapolis, IN 46202, USA*
[4]*Department of Biomedical Engineering, Purdue School of Engineering and Technology, Indiana University-Purdue University Indianapolis (IUPUI), 799 West Michigan Street, Indianapolis, IN 46202, USA*

Correspondence should be addressed to Christian Rogers; rogerscb@iupui.edu

Received 18 April 2016; Revised 23 June 2016; Accepted 5 July 2016

Academic Editor: Marco Mamei

Phantom limb is the sensation amputees may feel when the missing limb is still attached to the body and is still moving as it would if it still existed. Despite there being between 50 and 80% of amputees who report neuropathic pain, also known as phantom limb pain (PLP), there is still little understanding of why PLP occurs. There are no fully effective long-term treatments available. One of the struggles with PLP is the difficulty for amputees to describe the sensations of their phantom limbs. The sensations may be of a limb that is in a position that is impossible for a normal limb to attain. The goal of this project was to treat those with PLP by developing a system to communicate the sensations those with PLP were experiencing accurately and easily through various hand positions using a model arm with a user friendly interface. The system was developed with Maya 3D animation software, the Leap Motion input device, and the Unity game engine. The 3D modeled arm was designed to mimic the phantom sensation being able to go beyond normal joint extensions of regular arms. The purpose in doing so was to obtain a true 3D visualization of the phantom limb.

1. Introduction

Amputation of a limb is a procedure of last resort that may result from trauma, prolonged constriction, or surgery. It is usually required to control pain or the spread of disease for the survival and quality of life of the patient. Amputation often causes sensations that refer to the missing limb. The ghostly sensation of the missing limb is called the phantom sensation. It has been revealed that both peripheral and central nervous systems are the major factors of phantom limb pain (PLP) and work has been done to study this cause. "What is curious is that phantom phenomena were seemingly ignored for centuries, despite their occurrence throughout time immemorial" [1]. One of the challenges of studying PLP is that patients with PLP are not able to describe the pain precisely and the sensation of a missing limb does not aid others to help diagnose the specific pain. The current method used is to describe the degree of pain is by utilizing a scale of 0–10 and drawing a depiction of the arm. However, due to the fact that drawing skills may vary from patient to patient this method is not precise enough for research studies. The aim of this project was to develop a friendly interface through which patients can adjust a 3D model to best fit their phantom limb position. The process of adjusting the model can take longer than the patient intends so virtual reality technology like Leap Motion was implemented to take a realistic mapping of the existing limb of the patient. "There has been a tendency to regard the syndrome as a clinical curiosity, and very little experimental work has been done on it" [2]. We believe that the implementation of the proposed

work will help the further study of PLP by providing a reliable method to measure the perceived position of the phantom limb and therefore aiding in the treatment of PLP.

2. Materials and Methods

Phantom limb pain (PLP) is any painful sensation that refers to the absence of a limb. This type of pain is most often found with individuals who have lost a body part through an amputation. It is estimated that 80% of patients who have a total or partial limb loss develop PLP. PLP is known to be highly distressing and can cause feelings of shooting pain, burning, throbbing, or cramping. Usually, resolution of PLP is slow and can take months or even years to dissipate [2].

Mirror therapy is often one known way to alleviate phantom limb pain. Mirror therapy involves the induction of limb imagery where individuals see imagery of the existing limb imaged onto the area where the missing limb is located. Chan et al. [3] conducted a randomized sham-controlled trial that involved mirror therapy compared to imagery therapy. All patients in the trial had the condition of PLP. Pain intensity decreased with mirror therapy as well as the number of pain episodes and direction of each. After four weeks of treatment, all patients in the mirror therapy group reported a decrease in pain. Two patients did have reactions of grief as they saw the viewing of their reflected lower limb from the mirror therapy.

A similar method to mirror therapy is mirror box therapy. Mirror box therapy has been used with stroke patients who have lost control of a hand. Patients are given the illusion that they have a working and functional hand by reflecting the healthy hand. Robertson et al. [1] developed a mirror therapy tool utilizing the Kinect motion sensor. The goal of this simulation was to enable the part of the brain that is unaffected by the stroke to learn control of the impaired hand. They evaluated five different finger tracking SDKs using a controlled environment that resembled a clinic. They also discussed the strengths and weaknesses of each SDK and compared them in a trade-off matrix to quantify the individual performance.

When amputees perceive the existence of an amputated limb it often comes with unpleasant, annoying, or distressing pain (PLP). While a phantom limb may appear normal in the mind, often the person perceives the limb as having odd sizes (thinner, thicker, or swollen) or in various shapes and telescoping issues (shortened limb or lengthened), proprioceptive sensations (touch, pressure, temperature, itching, vibration, pins, and needs). The limb may feel abnormal like "fingers twisted out of shape or grossly intertwined or the thumb pushing through the palm" [4]. Giummarra et al. [4] conducted a study with 283 amputees, who administered a structured questionnaire to systematically determine the frequency and nature of the phantom limb. They suggested that the intern limb image and the limb schemata would play a role in the continued perception of the phantom limb. This system will continue that research, providing those with PLP the opportunity to not only describe the sensations, shape, position, and rotation of the phantom limb but also provide tools for visualization.

The system being developed and tested for this project offers a low-cost alternative to the previous treatments stated. Not only will the system provide the same conditions as mirror therapy, but also it will lead to stronger communication between the physician and the patient, allowing the patient to further manipulate the structure of hand using a basic user interface which will aid in the accuracy of the missing limb.

3. Results and Discussion

3.1. Rationale and Specific Aims. PLP affects a large percentage of amputees but the understanding for the cause of PLP is limited. For this project we applied advanced virtual reality technology to capture the perceived arm position of the phantom limb. The hypothesis was that, with Leap Motion (a computer hardware sensor device that captures the motion of a hand and fingers) and Unity, a 3D model of the phantom limb could be created and adjusted to best fit the position of the patient's phantom limb and also reduce time. By providing others with a better understanding of the position of the phantom limb it could therefore lead to better treatment of PLP.

3.2. Limb Position Capture. The procedure that is proposed through this system captures the position of the existing hand. When utilized the subject is asked to put the arm behind a curtain that can blind it. The hidden hand is then captured utilizing the Leap Motion device. The Leap Motion is a motion controller for virtual touch of a computer interface and for controlling objectives in a virtual environment. Some limitations exist with the Leap Motion. The first is that it loses track of the hand if the hand moves too fast. The second is that it requires the hand in a certain position to recognize all the joints and fingers. The Leap Motion controller software is being improved to decrease these limitations by offering better control of the hand and easier access to appendages in different positions. Currently, the developers circumvented the limitations by having the subject slowly move their hand into the Leap Motion capture area, keeping the hand in the position as Figure 1 shows for five seconds. Then the subject may move their hand freely.

3.3. Initial Testing. Initial testing was conducted on one or two individuals that did not suffer limb loss. In this testing visible and hidden limb positions were captured simultaneously to ensure consistency of the Leap Motion device. Two Leap Motion devices and two computers were utilized to capture the position and process the data for each limb. After the subject confirmed the position of both visible and hidden limbs the researcher captured the position. A 3D Unity model of the subject's visible limb in the captured position was shown on the computer screen for the subject to adjust. Another 3D Unity model of the subject's hidden limb was shown on the other computer behind the curtain where the subject was not able to see. The researcher was able to see both 3D models. The whole process took approximately 10 minutes per patient. In a situation where individuals with PLP are tested the existing limb will be captured and viewable to both the subject and the researcher. The subject will then

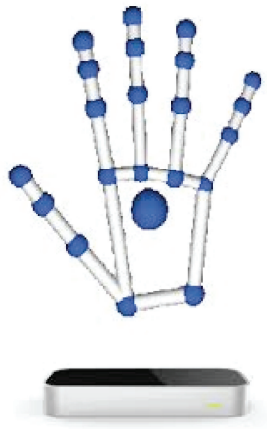

Figure 1: The position of hand that Leap Motion used to recognize all the joints and fingers.

have the ability to adjust the modeled limb to more greatly mimic the visualization of the missing limb.

3.4. 3D Model Adjustment. This procedure aims to let the subject adjust the 3D model of the visible limb so that the 3D model fits the perceived arm position. After the limb position capture process is completed, a 3D Unity model of the subject's visible arm appears on the computer screen. The subject is asked to adjust the 3D model to the position that he/she thinks the hidden arm is in. The subject can do so by clicking the arm model. To extend the subject can click on the part of the hand, hold the mouse button, and drag to extend the appendage. The same can be done to rotate and bend the model. All the joints can be rotated, fingers bent, rotated, and extended, the wrist rotated, and the arm extended and bent. The subject can reset any changes by pressing the "r" button. The subject confirms the adjustments are complete. The process takes approximately 5 to 10 minutes depending on the subject.

3.5. Error Evaluation. This procedure aims to evaluate the error between the position of the adjusted model and the position of the perceived arm. After the researcher fixes the 3D model, a C# code is run in Unity to extract the position of all joints as one Excel file. Then a Matlab code is used to read the Excel file and calculate the error. The error for each joint is then averaged to get the mean error using the following formula:

$$\text{ERROR} = \frac{\left|\text{position of adjusted model} - \text{position of perceived model}\right|}{\left|\text{position of perceived model}\right|} * 100\%. \quad (1)$$

3.6. Statistical Considerations. The only data collected in this study is the position of the arm. Standard statistical methods will be employed in this study. Mean and standard deviations will be calculated. The sample size is determined by the sample size needed for sufficient analysis.

3.7. The Arm Model. The arm model (Figure 2) was created from scratch starting with multiple references of the arm at different angles in order to achieve a good perspective of what an arm looks like. In order to get a better perspective while creating the arm a couple of quality references were picked to be inserted into Maya for the front and side image planes to be traced over for the basic shape. The model itself started from a single polygon cube. In order to shape the model the vertices were moved and stretched to shape out the arm while maintaining a proper resolution for the model. With the references on the image plane, the vertices were moved to try to match them as close as possible to the images, paying close attention to both views to maintain a proper shape. While stretching out the model's vertices, it was important to continue adding edge loops to fill in "empty" spaces to maintain the resolution. For the fingers, they had to be modeled separately and combined later on to the hand. This was created with a cylinder polygon and shaped into a finger form by moving the vertices accordingly. In order to make a proper rig the model had to have more resolutions in areas that are intended to move and stretch, especially in parts of the finger that were going to bend.

With the basic shape of the arm complete the model was brought into ZBrush which is a digital sculpting and painting program. The arm model was imported into ZBrush seamlessly and was given better details and more resolution that would be otherwise impossible inside Maya. After completion of the model's appearance (Figure 3), it was then imported back to Maya.

In order to have the arm model moved, it had to be given a skeleton with controls. Joint controls were given accordingly to parts of the arm that are rotatable. In order to have joints control the arm, the mesh was bound to the joints via skin bind. Skin bind creates a relationship between the joint controls and the arm model's mesh, allowing each joint to control nearby vertices of the mesh. At this current stage the controls are not user friendly, so in order to remedy each joint control was given ring NURBS to easily select and rotate (Figure 4). The skeleton is then hidden so it cannot be altered.

The joint controls were not controlling the appropriate parts but they were not controlling the mesh properly. It would often break the model and look unnatural. This was fixed with skin weight painting. This process tells the joints how much influence they should have on each vertex of the mesh; with the skin weight tool the arm was given much smoother controls on each joint. This repetitive process was repeated for each joint with various tests of trial and error. Each control was animated in order to see which spots needed attention (Figure 5).

With the arm now controllable and able to control its mesh more appropriately, it was ready to be brought into Unity for further testing.

3.8. Programming of the Model's Features

3.8.1. Movement of the Model. Sliders control movement of each component in the model. Both the position and rotation may be changed. Utilizing the built-in horizontal controls in Unity, sliders were created that were divided into 360 slices.

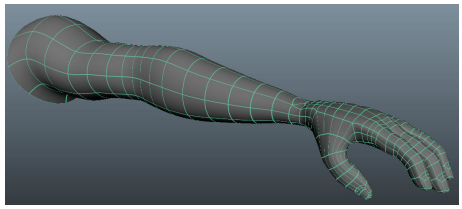

Figure 2: The arm model.

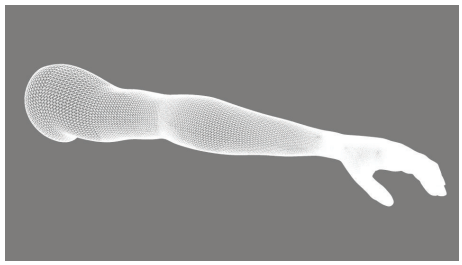

Figure 3: Model after ZBrush application.

The slider itself was initialized to the center, that is, 180. 360 was chosen so that movement could be thought of as rotation along a certain 360-degree axis. When a component is clicked on its sliders are activated. Once per frame the current values of the sliders are taken and recorded. For each frame the rotation and position are changed according to the sliders by making new vectors out of the difference between the current rotation and coordinate and the old rotation and coordinates. Figure 6 is a capture of the interface used to adjust the model.

3.8.2. Leap Motion Programming Component. The program incorporates a Leap Motion camera to help speed the process of manipulating the arm model in to the position that the patient perceives their hands to be and collect positional data of hidden hand. The Leap Motion is bundled with an application programmer interface for use with the Unity software package that the software side of the project is built on. This application programmer interface or API for short provides methods that allow developers to interface with the software used to program the Leap Motion device. When properly implemented the Leap Motion will apply positional and rotational data on to the arm model. This will be accomplished by first implementing a virtual version of the Leap Motion device into three-dimensional space where the arm model is located. It then receives position and rotation data from the test patient in the real environment and translates that information into three-dimensional space where the virtual device is located. The process of recreating the position and rotation of a real object in virtual space involves creating a quaternion (a vector that not only includes directional information but also includes information to tell the object it is attached to which direction is "up"). A quaternion is applied to each individual object within the arm model such as the joints and the bones and when they are applied simultaneously it results in fluid motion. The API

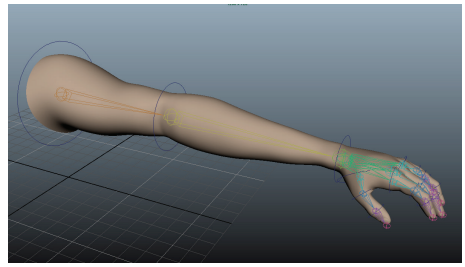

Figure 4: Model with ring NURBS.

does this work. The setup can be accomplished by applying the finger and hand scripts to the corresponding objects on the model.

3.9. Recording of the Data. Documentation of the data required consultation with the biomedical researcher for the project. The rotational coordinates and position coordinates in the three-dimensional space were agreed on as the standard for data comparison. In the initial attempts at this process the names associated with each component of the arm model were collected into a data structure known as an array. For each component, the name, x rotational coordinate, y rotational coordinate, z rotational coordinate, x position coordinate, y position coordinate, and z positional coordinate were written on a file. In order to do this for every component without having to write the same code continuously a "for each" loop was used to repeat the code.

After observing that the components in Unity were organized in a tree structure a more efficient approach was taken. In the second attempt at recording the data, the rotational coordinates and positional coordinates of each component were documented. Instead of collecting the names of each component in an array the tree of components was instead traversed recursively. In order for this to work, the script was attached to the uppermost component. When the "s" key is pressed, every child of the uppermost component is retrieved and the name, rotational coordinates, and position coordinates are written on file. For every child, this process is repeated. This continues until there are no more children.

3.10. Model Reset. The data documentation mechanism was used in order to get the default rotational and positional coordinates for the model reset. Once this data was acquired, a similar approach in comparison with the recording of the data was taken. Again, each component of the arm model was traversed recursively. However, this time the position and rotational coordinates were retrieved from the file containing the default coordinates. By reading line by line, the x, y, and z coordinates were assigned into variables that were then used to create new vectors. The rotational and positional properties of the current component are then set to the vectors. The process is then repeated with the children of the current component until the entire hand is reset to the state of Figure 7.

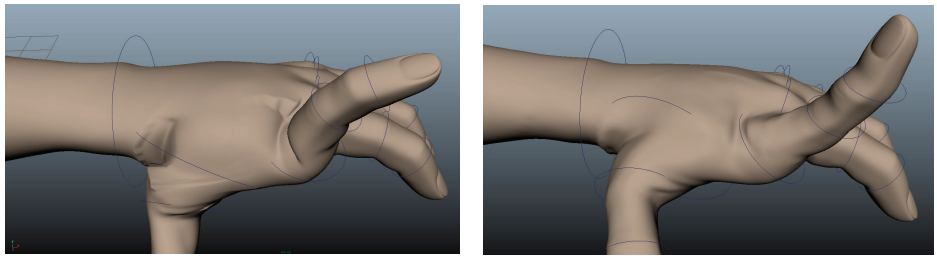

Figure 5: Before and after adjusting skin weights.

Figure 6: Figure within interface.

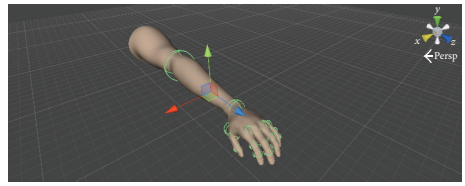

Figure 7: Finalized model.

4. Conclusion

In initial findings we concluded that although the model hand was not 100% identical to the actual hand when initially captured, one hundred percent accuracy is impossible as this is the first version of the program and certain positional capabilities will be learned about and implemented into later versions.

The first study will be a series of clinical trials which involves amputees with PLP. This protocol applies only to round 1 and will likely to be modified for round two based on results gained in round one.

4.1. Cost. The cost of the system is low. Items that are needed are a laptop or desktop computer and a Leap Motion which currently retails for $69.99 on the Leap Motion website.

4.2. Enrollment/Randomization. Recruitment for this study will be self-referrals from persons that heard of the study and qualify. Enrollment will occur between potential subjects and the investigators. Interested potential participants that meet the inclusive criteria of suffering from PLP due to missing one upper extremity can contact the investigators and schedule for a prescreening visit to determine final eligibility. Potential participants who qualify will be invited to the study. Those that may not be included are those that either do not have PLP or are missing a lower extremity.

4.3. Final Considerations. While more testing must be conducted with the system, initial development concludes that some things must be considered when operating the system.

(1) Some training will need to be conducted. This training will be basic and will be primarily focused on setting up the system hardware (the Leap Motion and computer).

(2) The system will never be 100% accurate due to communicative error between the patient, the Leap Motion, and the clinician. The purpose of this system is merely to provide another opportunity to communicate the physical state of the missing limb but it will do so with error.

(3) Previous studies have stated there can be an emotional response when a patient suffering from PLP sees a limb as if he has two limbs available. This needs to be considered for future testing and production [5].

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Authors' Contributions

Jonathan Lau, Denver Huynh, Steven Albertson, James Beem, and Enlin Qian contributed equally to this work.

References

[1] C. Robertson, L. Vink, H. Regenbrecht, C. Lutteroth, and B. C. Wunsche, "Mixed reality Kinect Mirror box for stroke rehabilitation," in *Proceedings of the 28th IEEE International Conference on Image and Vision Computing New Zealand (IVCNZ '13)*, pp. 231–235, Wellington, New Zealand, November 2013.

[2] F. H. Kiabi, M. R. Habibi, A. Soleimani, and A. E. Zeydi, "Mirror therapy as an alternative treatment for phantom limb pain: a short literature review," *The Korean Journal of Pain*, vol. 26, no. 3, pp. 309–311, 2013.

[3] B. L. Chan, R. Witt, A. P. Charrow et al., "Mirror therapy for phantom limb pain," *The New England Journal of Medicine*, vol. 357, no. 21, pp. 2206–2207, 2007.

[4] M. J. Giummarra, N. Georgiou-Karistianis, M. E. R. Nicholls, S. J. Gibson, M. Chou, and J. L. Bradshaw, "Corporeal awareness

and proprioceptive sense of the phantom," *British Journal of Psychology*, vol. 101, no. 4, pp. 791–808, 2010.

[5] V. S. Ramachandran and W. Hirstein, "The perception of phantom limbs. The D. O. Hebb lecture," *Brain*, vol. 121, no. 9, pp. 1603–1630, 1998.