

# The Virtual Acupuncture Based on Natural Gesture and Force Feedback

Sijie Lin, Guanglong Du and Ping Zhang  
South China University of Technology, Guangzhou, China

**Abstract**—the paper presents a virtual acupuncture system which uses digital virtual human model to carry out the virtual acupuncture experiments by using natural gesture and force feedback. In the paper, five Leap Motions are employed to measure the position and orientation of the operator's hand. Due to the tracking errors and the noise of equipment, the measurement errors will increase over time. So, Kalman Filter and Particle Filter are introduced to estimate the position and orientation of the hand. Then we use 3D Max to build the joint model and OSG (OpenSceneGraph) to refactor the model of the hand. Because it is needed to detect the collision of two models including hand model and human model, here the K-DOPs collision detection algorithm is used to achieve it. When the hand model and the human model collide, we need to use noncontact force feedback which using Lorentz Force to let the operator experience the drag force like it comes from the human model through simply wearing a small magnet. Therefore, the virtual Acupuncture experiment can be carried out successfully on the digital virtual human model.

**Keywords**-virtual acupuncture; nature gesture; force feedback

## I. INTRODUCTION

Chinese acupuncture is an important part of traditional Chinese medicine. However, in the traditional study of Chinese acupuncture, the method of operating samples mechanically is used universally, which is direct but wasting time and energy. Besides, this method has the disadvantages of low repeatability and low utilization rate of samples.

With the development of virtual reality technology in the field of modern medicine and the development of visualization technology, more and more researchers have done a lot of work in the virtual acupuncture system. P.-A. Heng presented a 3-D interactive virtual environment which allows the operator performs acupuncture, and it also supports a force feedback interface with needle insertion [1]. With the update of information technology, some researchers combine multiple technologies to construct a 3-D digital human model for acupuncture, like Heng P A, and Xie Y [2]. In Ren Kanehira's acupuncture training system, to construct a virtual 3D human acu-points model in the computer, the virtual reality technology was employed. And the system also quantitatively indexed the position and depth of acu-points with the model [3].

However, the method proposed above does not interact well with human and not so convenient. We hope that the system can make the operator feel more reality when interact with the virtual environment while do not need the complex operation or equipment. Therefore, this paper presents a virtual acupuncture system based on natural gesture and force feedback system. In

the system, the operator can use their finger as a needle to perform an acupuncture practice on the virtual human model in computer, which does not require the complex operation and facilities, and all the operator needed are just wearing a small magnet and use his finger to carry out the virtual acupuncture.

The remainder of this paper is organized as follows. Section II gives an overview of the virtual acupuncture system based on natural gesture and force feedback. Section III is about the position and orientation of gesture. Section IV describes the modeling of operator's hand. And Section V describes the collision detection of models and the force feedback system. Section VI details the experiment and section VII makes a conclusion.

## II. OVERVIEW



FIGURE I VIRTUAL ACUPUNCTURE SYSTEM

Fig. 1 shows the structure of acupuncture system. An operator who wears a small magnet stands in front of the table and uses his finger as a needle to make a virtual acupuncture on the virtual human model which occurs in the computer. And in front of the computer there are five Leap Motions and four coils. In the whole process of our virtual acupuncture system, firstly, by five Leap Motions, we get the position as well as orientation of the hand. Because the error will exist and accumulate over time during measuring the data of the hand, to solve this problem, the Kalman Filter [4] algorithm and the Particle Filter [5] algorithm which can estimate the position and the orientation of the hand respectively are introduced. Then the 3D Max can be used to build the joint model of the hand. And through the OSG, we can get an accurate three-dimension model of the hand. After we get the three-dimensional of the hand, we can use it to make a virtual acupuncture. When the hand model collides with the human model, we need to use noncontact force feedback which using Lorentz Force to let the operator experience the drag force like it comes from the human model through simply wearing a small magnet. And a K-DOPs algorithm is employed to detect the collision in the paper.

### III. THE POSITION AND ORIENTATION OF GESTURE

#### A. The Coordinate System of Hands

To collect the position and orientation data of the hand and let it more accurate, we use a hand tracking system with five Leap Motions. Through allocating a unique ID to each LM, the system can know that the data comes from which LM, and calculate the position and orientation of the centroid on the hand. In each LM, there are three coordinate systems  $X_W Y_W Z_W$ ,  $X_L Y_L Z_L$  and  $X_H Y_H Z_H$ .  $X_W Y_W Z_W$  is the world-fixed frame.  $X_L Y_L Z_L$  is the Leap Motion frame and it is translated from the world-fixed frame. And  $X_H Y_H Z_H$  is the hand frame which is a translation of the Leap Motion frame and at the center of the palm. We can use the translation from frame  $X_H Y_H Z_H$  to the frame  $X_L Y_L Z_L$  to represent the position of the operator's hand. And as for the orientation of the operator's hand, we can use the roll-pitch-yaw frame rotation angles ( $\phi, \theta, \psi$ ) to represent it. In this way, the position, velocity, acceleration, angular velocity and rotation angles of the hand can be collected.

In the hand-tracking platform with five Leap Motions, one LM is placed at the center and the other four LMs are arranged around it. And the surrounding LM is placed in an oblique angle about  $45^\circ$  and toward the center, it is to increase the workspace of the bottom. Moreover, the measured data are more accurate and stable if we can fuse the measured data from multiple LMs. So in this paper we build the workspace with five LMs.

#### B. The Estimation of Position

The Kalman filter is the optimal recursive state estimation algorithm under the condition of the Gauss distribution, which can predict a true teaching reproduction state parameter from a set of limited, incomplete, and observed teaching reproduction data including noise. The Kalman filter model can be described through a linear stochastic equation containing the system state model and the measurement model, it is defined as follow:

$$\begin{cases} x_k = A_k \cdot x_{k-1} + B_k \cdot u_{k-1} + w_{k-1} \\ z_k = H_k \cdot x_k + v_k \end{cases} \quad (1)$$

where  $x_k \in \mathbb{R}^n$  is the state vector,  $A_k$  is the  $n \times n$  matrix and represents the state transition model which is used in the previous state  $x_{k-1}$ ,  $B_k$  is the  $n \times l$  matrix and represents the control-input model which is used to input vector,  $u_{k-1} \in \mathbb{R}^l$  is defined as the deterministic input vector,  $b_{k-1}$  and  $v_k$  represent the process and measurement noise respectively.  $z_k \in \mathbb{R}^m$  represents the measurement vector, and the observation model  $H_k$  maps the true state space to the observed space.

Assume that  $P(p_x, p_y, p_z)$  is the coordinates of the end of the gesture, and the lossless Kalman filter can estimate the position state data of the gesture end. Then the direction cosine matrix which is transformed from the coordinate system of the robot hand gesture to the space coordinate system is as follows [6]:

$$M_{H2W} = \begin{bmatrix} m_{X_x} & m_{Y_x} & m_{Z_x} \\ m_{X_y} & m_{Y_y} & m_{Z_y} \\ m_{X_z} & m_{Y_z} & m_{Z_z} \end{bmatrix} \quad (2)$$

So, the acceleration of the end of the gesture in the space coordinate system can be expressed as [7]:

$$\begin{aligned} \square V_x &= m_{X_x} \square A_x + m_{Y_x} \square A_y + m_{Z_x} \square A_z \\ \square V_y &= m_{X_y} \square A_x + m_{Y_y} \square A_y + m_{Z_y} \square A_z \\ \square V_z &= m_{X_z} \square A_x + m_{Y_z} \square A_y + m_{Z_z} \square A_z - |g_l| \end{aligned} \quad (3)$$

Where  $(A_x, A_y, A_z)$  represents the component of acceleration of the hand,  $|g_l|$  is the moduli of the acceleration of gravity in the space coordinate system. So the velocity component of each coordinate axis can be described as:

$$V_x = p_x \quad V_y = p_y \quad V_z = p_z \quad (4)$$

When using lossless Kalman filtering to estimate the state, the state vector at the end of the gesture at time k can be defined as:

$$x_{k,pos} = [p_{x,k}, V_{x,k}, A_{x,k}, p_{y,k}, V_{y,k}, A_{y,k}, p_{z,k}, V_{z,k}, A_{z,k}] \quad (5)$$

Since there is no control input in the system, the acceleration measurement is only dependent on gravity, and the Z-axis at the end of the gesture is always parallel to the direction of the gravity. Therefore, the system input matrix is as:

$$\Gamma_{pos} \square u_{k-1} = [0, 0, 0, 0, 0, 0, -|g_l| \square^2/2, -|g_l| \square, 0]^T \quad (6)$$

Because the gravity and system input matrix do not change at this time, the process noise vector is:

$$w_k = [0, 0, w_x, 0, 0, w_y, 0, 0, w_z]^T \quad (7)$$

According to the derivation describe above, the measurement matrix can be written as:

$$H_{pos} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

The position data from the sensor will be input to the lossless Kalman filter system model as a variable, and finally we can get the estimated value of the output position.

### C. The Estimation of Orientation

A quaternion can be used to express the orientation of the end of the gesture. Suppose that the quaternion  $q$  is differentiable for time, and can be represented by:

$$\begin{bmatrix} \partial q_0 / \partial t \\ \partial q_1 / \partial t \\ \partial q_2 / \partial t \\ \partial q_3 / \partial t \end{bmatrix} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} 0 \\ \omega_x/2 \\ \omega_y/2 \\ \omega_z/2 \end{bmatrix} \quad (9)$$

Where  $q_0, q_1, q_2, q_3$  are the quaternion components at the end of the gesture.  $\omega_x, \omega_y, \omega_z$  represent the angular velocity components. Therefore, the state described the end of the gesture which contains the quaternion state and the angular velocity state can be defined as:

$$x_{k,ori} = [q_{0,k} \ q_{1,k} \ q_{2,k} \ q_{3,k} \ \omega_{x,k} \ \omega_{y,k} \ \omega_{z,k}] \quad (10)$$

Where  $q_{0,k}, q_{1,k}, q_{2,k}, q_{3,k}$  represent the state component of the quaternion at time  $k$ ,  $\omega_{x,k}, \omega_{y,k}$  and  $\omega_{z,k}$  represent the angular velocity component at time  $k$ . So the state transition matrix of the system can be obtained, it is shown as follow:

$$\Phi_{ori} = \begin{bmatrix} 1 & 0 & 0 & 0 & -q_{1,k}\Delta t/2 & -q_{2,k}\Delta t/2 & -q_{3,k}\Delta t/2 \\ 0 & 1 & 0 & 0 & q_{0,k}\Delta t/2 & q_{3,k}\Delta t/2 & q_{2,k}\Delta t/2 \\ 0 & 0 & 1 & 0 & q_{3,k}\Delta t/2 & q_{0,k}\Delta t/2 & -q_{1,k}\Delta t/2 \\ 0 & 0 & 0 & 1 & -q_{2,k}\Delta t/2 & q_{1,k}\Delta t/2 & q_{0,k}\Delta t/2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Where  $\Delta t$  is the sampling frequency of the IMU measurement unit.  $\Gamma_{ori}$  is a zero matrix since the system have no input operation, so it can be ignored. At this point, we can estimate the quaternion state at the end of the gesture from the angular velocity. Thus, the process noise vector is written as:

$$w_k = [0 \ 0 \ 0 \ 0 \ w_x \ w_y \ w_z]^T \quad (12)$$

Where  $w_x, w_y$  and  $w_z$  represent the components of the process noise on each axis. Because the sensor has been calibrated, the observation matrix can be written as:

$$H_{ori} = [0^{n \times p} \ I^{n \times n}] \quad (13)$$

Because the quaternion component must satisfy the condition that the sum of the squares is 1, the quaternion state at time  $k$  should be normalized:

$$q_k = [q_{0,k}/M \ q_{1,k}/M \ q_{2,k}/M \ q_{3,k}/M] \quad (14)$$

$$M = \sqrt{q_{0,k}^2 + q_{1,k}^2 + q_{2,k}^2 + q_{3,k}^2}$$

### IV. THE MODELING OF THE HAND

Our fingers can be divided into 19 parts in the human joint recognition by the API of Leap Motion. Here, 3DMax can be employed to construct a cylinder which represents the hand joint, and then converted it to a node model in the OpenSceneGraph. Because we just use the index finger as a needle, so we only need to use 4 cylinders to build the human index finger model.

Due to the difference between the coordinate system in Leap Motion and in OpenSceneGraph, when leap motion inputs the coordinate, we need to convert it. Leap Motion uses a coordinate system that the horizontal direction is the X axis and the right side is the positive axis. And the Y axis is the vertical direction, with the upward direction as the positive axis. The horizontal direction is the Z axis and direction pointed to the user is a positive axis. In OpenSceneGraph, the horizontal direction is the X axis, which is the same as Leap Motion. But it uses the Z axis as the vertical direction of the window device and its upward is the positive axis. And the Y-axis is used to represent the direction perpendicular to the X-axis in the horizontal plane, and the direction which is directed to the operator is the positive axis. Therefore the input data need to pass the Y-axis and Z-axis data exchange in the OpenSceneGraph. Beside, a Group pointer is defined as the root node in the OpenSceneGraph, and then we can use the inner matrix transformation function and the incoming coordinates in conjunction with the processing of each cylinder. And finally 4 cylinders were added as a node to the root node.

### V. NONCONTACT FORCE FEEDBACK USING LORENTZ FORCE

#### A. K-DOPs Collision Detection Algorithm

Before we use the force feedback, the collision between the human model and gesture model must be detected. The K-DOPs detection technique detects the collision through the divisions of the movement path. Before dividing the movement path, the algorithm needs to build the K-DOPs bounding boxes, which is used to calculate the bounding boxes of the movement paths. And then the K-DOPs algorithm divides the movement path into two parts that have the same length. Then it checks where the collisions occur. If they occurred in the first half, the algorithm will divide the first half into two equal parts once again, if they occurred in the second half, and then divide the second. The algorithm will divide the path all the way until the length less than a threshold which is given before. And when happen collision, if it reaches the finished condition, in order to limit the triangle which possibly has a collision in a bounding box, a recursive traversal should be performed on of the bounding box tree in the completed interval. Then it performs the K-DOPs detection against the triangles in the bounding box. We can find more details of the K-DOPs algorithm in [8].

#### B. Structure of Force Control

In the system, through the synthesis of multi-directional forces, six-dimensional force feedback effects can be achieved. Multiple coils are mounted on the four corners of a platform and they are not in the same orientation. Every coil is placed on a three-dimensional rotatable foothold, in this way we can adjust the angle between the coil and the permanent magnet,

which ensure that the permanent magnet is at a proper position. Moreover, we also can adjust the angle of the axis of the coil which has an advantage that altering the overlapping electromagnetic field more flexibility. So that it can create a three-dimensional space force with the appropriate magnitude and direction. According to the bilateral control model [9], the kinetic model consisting of the operator's hand and the finger model in the computer can be obtained from:

$$\begin{aligned} f_h - \tau_m &= m_m \ddot{x}_m + b_m \dot{x}_m \\ \tau_s - f_e &= m_s \ddot{x}_s + b_s \dot{x}_s \end{aligned} \quad (15)$$

where  $x_m$  and  $x_s$  represent the positions of the master-slave hand respectively,  $\tau_m$  and  $\tau_s$  represent the forces of the master-slave hand,  $f_h$  is the applied force of the master hand while  $f_e$  represents the force acted by the environment,  $b_s$  and  $b_m$  refer to the velocity damping coefficient of the master-slave hand. We can get the information of operator's hand including position, velocity, and acceleration before. So, according to the force acting on the hand and the distance between the hand and the coil, we can calculate the three-dimensional moment of the force of the operator's hand.

### C. Open Loop Force Control

Through Maxwell's equations, the magnitude of the electrical current and the distribution of the magnetic field of each coil can be calculated if given magnitude of the electromagnetic force. Moreover, if given distribution of the electromagnetic field, the magnitude and direction of the force acting on the permanent magnet can be achieved by the Lorentz force equation. Besides, based on the molecular circulation hypothesis, the permanent magnet can be regarded as a solenoid which is composed by the magnetizing current, so we can get the differential equation of the force on the magnet as follow:

$$dF = I' dl \times \left( \oint_L \frac{\mu_0}{4\pi} \frac{I dl \times e_r}{r^2} + B' \right) \quad (16)$$

where  $I'$  and  $I'$  represent the solenoid electrical current,  $B'$  is the electromagnetic field which is generated by the magnetized iron, and  $L$  is the path of the solenoid wire. Hence, the force of the magnet can be written as:

$$F = \oint_L df = \oint_L I' [dl \times \left( \oint_L \frac{\mu_0}{4\pi} \frac{I dl \times e_r}{r^2} + B' \right)] \quad (17)$$

The calculation of the magnetic force is applied by the finite element idea in mathematical which discretizes the calculation of the force of the magnet, and then according to the above formulas, the force of the permanent magnet can be calculated.

## VI. EXPERIMENT

### A. The Environment of Experiment

In our experiment, there several devices that we need, including a computer, five Leap Motion, four coils and a small magnet which will be wore by the operator. As shown in Fig. II,

we can see the whole virtual acupuncture system. In the computer, there is a virtual human model. And In front of the computer, one LM is placed at the center and the other four LMs are arranged around it and pluralities of coils are mounted around the four corners but not in the same orientation.

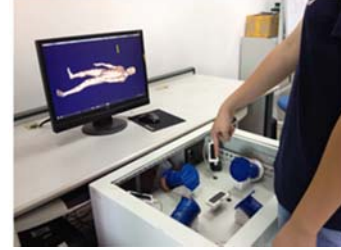


FIGURE II THE PROCESS OF VIRTUAL ACUPUNCTURE SYSTEM

### B. The Result of Experiment

The result of our experiment shows that the operator wearing a small magnet is able to use his hand to carry out a virtual acupuncture practice. At present, our system is convenient and has good human-computer interaction. As the operator's hand moves down, we can see the model of the finger also moves down, when the finger model collides with the human model, the operator can feel the drag force come from the human model, which actually generated by the electromagnetic field in the coil. And the force that the operator feels about is change all the way. The result proved that our system is successful. Through this virture acupuncture system, the operator can carry out acupuncture practices repeatedly while not waste too much samples. The result also shows that the system can provide a sense of reality of the acupuncture to the operator.

## VII. CONCLUSION

The paper introduces a virtual acupuncture system which based on natural gesture and the force feedback and shows the experiment. The experimental results proved that when performing a virtual acupuncture by using his/her hand in the system, the operator can feel the sense of reality that likes performing the acupuncture on the real human body. Besides, the operator also can repeatedly carry out a virtual acupuncture practice, which can increase the utilization rate of samples.

## ACKNOWLEDGMENT

Project funded by "National Natural Science Foundation of China (Grant No: 61403145)" "Guangdong Provincial Science and Technology Project (2014B090921007)", "Guangzhou Municipal Science and Technology Program (20150810068)" and Haizhu District of Guangzhou City Science and Technology Project (2014-cg-02).

## REFERENCE

- [1]. Heng P A, Wong T T, Yang R, et al. Intelligent inferencing and haptic simulation for Chinese acupuncture learning and training [J]. IEEE Transactions on Information Technology in Biomedicine, 2006, 10(1): 28-41.
- [2]. Heng P A, Xie Y, Wang X, et al. Virtual acupuncture human based on chinese visible human dataset[J]. Studies in health technology and informatics, 2005, 119: 194-197.

- [3]. Kanehira R, Shoda A, Yagihashi M, et al. Development of an acupuncture training system using virtual reality technology[C]//Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08. Fifth International Conference on. IEEE, 2008, 4: 665-668.
- [4]. S.P.Won, W.W.Melek, and F. Golnaraghi, "A fastening tool tracking system using an IMU and a position sensor with Kalman filters and a fuzzy expert system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp.1782-1792, Nov. 2009.
- [5]. T. Khalid, Z. Mourad, C. Jean-Bernard, and B. Mohammed, "Bayesian bootstrap filter for integrated GPS and dead reckoning positioning," in *Proc. IEEE Int. Symp. Ind. Electron.*, Vigo, Spain, 2007, pp. 1520–1524.
- [8]. Ping Zhang, Bei Li, and Guanglong Du. A Wearable-base and Markerless Human-Manipulator Interface with Feedback Mechanism and Kalman Filters [J]. *Industrial Robot*, 2015, 42(5):485-495.
- [9]. S.P.Won, W.W.Melek, and F. Golnaraghi, "A fastening tool tracking system using an IMU and a position sensor with Kalman filters and a fuzzy expert system," *IEEE Trans. Ind. Electron.*, vol. 56, no. 5, pp.1782-1792, Nov. 2009.
- [10]. Klosowski J T, Held M, Mitchell J S B, et al. Efficient collision detection using bounding volume hierarchies of k-DOPs[J]. *IEEE transactions on Visualization and Computer Graphics*, 1998, 4(1): 21-36.
- [11]. Anderson R J, Spong M W. Bilateral control of teleoperators with time delay[J]. *IEEE Transactions on Automatic control*, 1989, 34(5): 494-501.