

The Virtual Assembly Technology Based on Natural Gesture

Yanjiao Chen, Guanglong Du and Ping Zhang

School of Computer Science & Engineering, South China University of Technology, PR China

Abstract—the purpose of this paper is to present a method called virtual assembly technology based on natural gesture. In the proposed method, five Leap Motions are fixed on the table to measure the position of the hands. Due to the tracking errors and the noise of equipment, the measurement errors will increase over time. Therefore, we use Kalman Filter and Particle Filter to evaluate the position and orientation of our hands. Then we use 3D Max to build the joint model of our hands, and OSG (OpenSceneGraph) to refactor the model of the hand. We use k-DOPs collision detection algorithm to detect collisions between hands model and human model. Then we use noncontact force feedback to make the operator experience the drag force like in a real environment, so that the operator can use natural gesture to do various assembly operation on the parts of the products in an interactive virtual assembly environment.

Keywords-virtual assembly; kalman filtering; leap motion; force feedback.

I. INTRODUCTION

As a significant application of virtual reality technology in the field of design and manufacture, virtual assembly technology has made rapid development in recent years, some of the virtual assembly system has been applied in industry.

According to the current understanding, the development of the virtual assembly[1] can be divided into three stages, the first stage is the theory of virtual assembly being proposed, the second stage is the simple duplicate and establishment of the virtual assembly system, the third stage is the deep study on development direction of the virtual assembly. At present, the development of virtual assembly technology abroad has entered the third stage, however, the research of virtual assembly in our country is currently in the transition from the second stage to the third stage. Although the study of virtual assembly technology develops rapidly, and has achieved a lot of research results, but compared with foreign countries, there is still a gap of about 5 years.

The development of virtual assembly technology is a key part of virtual manufacturing technology[2], it has become the main object of the current research in the field of virtual manufacturing technology. Therefore, the paper proposed a virtual assembly technology based on natural gesture. Operators can use their own hands to simulate virtual assembly operations. The hand tracking system consists of five Leap Motions, it can obtain the position date and the orientation date of gestures. Also, Kalman Filter[3] and Particle Filter[4] are employed to reduce the data error. And then we can apply the noncontact force feedback system with four coils using Lorentz force to feel the drag force like in a real environment. Our

method does not require the operator to learn much about virtual assembly, people just need to wear a small magnet, and then they can do various types of assembly operations like in a real environment.

The remainder of this paper is organized as follows. Section II gives an overview of the virtual assembly technology based on virtual reality. Section III describes gesture acquisition. Section IV details the posture estimation and Section V describes the force feedback system. Section VI details the experiment. And then Section VII makes a conclusion.

II. OVERVIEW



FIGURE I VIRTUAL ASSEMBLY SYSTEM

Fig. I shows the virtual assembly system in our method. A man stands in front of the table and uses his gesture to make virtual assembly operations. Firstly, the orientation and the position of gestures will be collected by the hand tracking system using five Leap Motions. Then we use the Kalman Filter[3] algorithm and the Particle Filter[4] algorithm to evaluate the orientation and position of our hands. After the data of our hands fused, we can use 3D Max to build the joint model of our hands, and then we can get an accurate three-dimension model of our hands. Before using force feedback, we need to detect the collision using k-DOPs collision detection algorithm if the hand model collides with the human model. Then we use force feedback system with four coils which use Lorentz force to feel the drag force like in a real environment.

III. GESTURE ACQUISITION

A. The Structure of the Platform

In our methods, we use a hand tracking system with five Leap Motions. To build a larger and regular workspace, one LM is placed at the center of the platform and the other four LMs are placed around it. The workspace of two LMs are overlapping so that the operator's hand can move from the workspace of one LM to that of another one. So, the measured

date is more accurate and stable in the overlapping workspace, as we can fuse the measured data from the workspace of five LMs.

The coordinates of the surrounding LMs are registered with the coordinate of the center LM. To obtain the postures with respect to the frame of the center LM in real time, the coordinate registration should be completed before the system runs. The measured data in the overlapping workspace are used to register the corresponding two LMs.

B. Coordinate System of Leap Motion

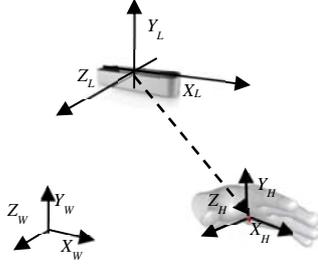


FIGURE II THE COORDINATE SYSTEM OF HANDS

Five Leap Motions are combined to track the orientation and position of the operator's hand in the tracking system. The coordinate system of hands is shown in Fig.II. We define $X_W Y_W Z_W$ as the world fixed frame. And $X_L Y_L Z_L$ is defined as the Leap Motion frame which is translated from the world-fixed frame. And the $X_H Y_H Z_H$ is defined as the hand frame which is a translation of the Leap Motion frame. These three frames are parallel in all axes. The center of the hand is the original point of the hand coordinate system. X_H is the straight line from the origin points to the middle finger and it points out. Y_H is a straight line from the center of the hand to the outside of the palm and points upward, it is perpendicular to the back of the hand. And Z_H is perpendicular to the X_H and Y_H . The position of operator's hand can be calculated as the translation from frame $X_H Y_H Z_H$ to the frame $X_L Y_L Z_L$. And the orientation of operator's hand is represented by the roll-pitch-yaw frame rotation angles (ϕ, θ, ψ) , which means the rotation between the frame $X_L Y_L Z_L$ and the frame $X_H Y_H Z_H$ about X, Y, Z axes. Therefore, the position, velocity, acceleration, angular velocity and rotation angles of the human hand can be collected by the Leap Motion.

IV. POSTURE ESTIMATION

A. The Estimation of Position

The Kalman filtering is a data processing technique for removing noises and restoring the real data. It is often used to estimate the state of the dynamic system, especially in the presence of white noise and non-linear system.

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 $x_k \in \mathfrak{R}^n$ is the state vector and $z_k \in \mathfrak{R}^m$ is the measurement vector, $u_{k-1} \in \mathfrak{R}^l$ is the deterministic input vector and v_k is a measurement noise vector. the $n \times n$ matrix A_k corresponds to a state transition model which is used in the previous state x_{k-1} , the $n \times l$ matrix B_k is applied to the input vector which is the control-input model. H_k maps the true state space into the observed space.

Suppose 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. It is assumed that the direction cosine matrix transformed from the coordinate system of the robot hand gesture to the space coordinate system is as follows[5]:

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

Therefore, the acceleration of the end of the gesture in the space coordinate system can be expressed as[6]:

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

Where $|g|$ represents the moduli of the acceleration of gravity in the space coordinate system, (A_x, A_y, A_z) represents the component of acceleration in the coordinate system of the hand gesture. Therefore, the velocity component of each coordinate axis in the space coordinate system can be written as:

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

In the process of state estimation using lossless Kalman filtering, the state vector at the end of the gesture at time k is 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 are no control inputs in the system, the acceleration measurement is only dependent on gravity. In general, the Z-axis of the coordinate system at the end of the gesture is always parallel to the gravitational direction. Therefore, the system input matrix is:

$$\Gamma_{pos} \square_{k-1} = [0, 0, 0, 0, 0, 0, -g_x \square_{k-1}^2/2, -g_y \square_{k-1}, 0]^T \quad (6)$$

The gravity and system input matrix remain constant at this time. Therefore, the process noise vector is:

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

Based on the above derivation, the measurement matrix can be written as:

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

The position data obtained from the sensor is input as a variable to the lossless Kalman filter system model, and finally the estimated value of the output position is obtained.

B. The Estimation of Orientation

The orientation of the end of the gesture can be expressed as a quaternion. The partial derivative equation of the quaternion q with respect to time t can be expressed as:

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

Where q_0, q_1, q_2, q_3 represent the quaternion component at the end of the gesture, and $\omega_x, \omega_y, \omega_z$ represent the angular velocity in x, y and z axes, respectively. Thus, the state that defines the end of the gesture contains the quaternion state and the angular velocity state:

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

Where $q_{0,k}, q_{1,k}, q_{2,k}, q_{3,k}$ represents the state component of the quaternion at time k and $\omega_{x,k}, \omega_{y,k}$ and $\omega_{z,k}$ represents the angular velocity component at time k . Therefore, we can get the state transition matrix of the system:

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

Where Δt is the sampling frequency of the IMU measurement unit. Γ_{ori} is a zero matrix, so it can be ignored. At this point, the quaternion state at the end of the gesture can be evaluated from the angular velocity. Thus, the process noise vector at time k is defined 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. As the sensor has been calibrated. Therefore, the observation matrix is expressed as:

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

Since the quaternion component needs to satisfy the condition that the sum of the squares is 1, it is necessary to normalize the quaternion state at time k :

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

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

C. The Three-dimensional Model of Hands

The API of Leap Motion divided our fingers into 19 parts in the human joint recognition. Tail finger, ring finger, middle finger and index finger all have four joints while thumb has three joints.

To represent one human hand joint, we need to use 3DMax to construct one cylinder and then we need to converted it to a node model in the OpenSceneGraph. In our method, we use 19 cylinders in OpenSceneGraph.

There exists much differences between the coordinate system in Leap Motion and the coordinate in OpenSceneGraph. The Y axis and the Z axis are exactly the opposite of each other, So in the OpenSceneGraph coordinate system, the input data needs to be exchanged between the Y axis and the Z axis.

In the OpenSceneGraph, a pointer is created as the root node, and then we can use the inner matrix transformation function and the incoming coordinates in conjunction with the processing of each cylinder. And finally 19 cylinders are added as a node to the root node.

D. The k-DOPs Collision Detection Algorithm

In this paper, we use the k-DOPs collision detection algorithm to detect the collision.

Penetration occurs during the movement when the cylinder object moves through the cube object. In this paper, we adopt the dynamic k-DOPs collision detection algorithm to check whether exist the penetration. This algorithm detects the whole process of the movement. At first, the algorithm divides the movement path into two equal parts, and then the algorithm detects the both parts to find out if there exist collisions. If the collision occurs in one of the two parts, the algorithm will divide the exact part into two parts once again. This operation

will continue to circulate until the path has reached the final condition and cannot be divided any more. In order to determine the collision area quickly and accurately, we adopt the bounding box tree which is a binary tree based on the bounding box. Collision detection between bounding box trees is a recursive traversal process. Perform a recursive traversal and keep the triangles having potential conflicts in a bounding box, then the k-DOPs algorithm will detect the collision against those triangles limited in the bounding box.

V. FORCE FEEDBACK SYSTEM

A. Structure of Force Control

In our method, we use four coils with iron core to fulfill six-dimensional force feedback effects. At the center of each edge of the platform, there is a coil placed between two LMs, and there are four coils in total. Each of them is placed in a three-dimensional rotation point, the operator can experience the forces through wearing a small magnet. The coil can adjust the angle between the coil and the permanent magnet for checking the position of the magnet is appropriate, it can also adjust the angle of the axis of the coil to change the overlapping electromagnetic field.

We have obtained the position of operator's hand in the hand tracking system introduced before, so we can easily calculate the three-dimensional moment of the force of the operator's hand through the force acting on the hands and the distance between the hand and the coil.

According to the bilateral control model[7], 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 f_h and f_e represent the applied force of the master hand and the force acted by the environment respectively, x_m and x_s represent the position of the master- slave hand, while τ_m and τ_s are the forces of the master-slave hand, b_s and b_m refer to the velocity damping coefficient of the master-slave hand.

B. Open Loop Force Control

Open loop force control system is a one-way control process, the control results will not feedback to influence current control system. The coils with iron core produce an electromagnetic field, this field will generate a kind of force, the operator wears a permanent magnet to experience the force and the magnet cannot block the hand motion. Maxwell's equations and the Lorentz force equation are the basic equations of classical Electromagnetics. If the magnitude of the electromagnetic force is given, the magnitude of the electrical current and the distribution of the magnetic field of each coil can be calculated through Maxwell's equations. In addition, if the distribution of the electromagnetic field is known, the

magnitude and direction of the force acting on the permanent magnet can be calculated by the Lorentz force equation. The permanent magnet can be regarded as a solenoid consisting of magnetizing current, so the differential equation of the force on the magnet can be expressed as:

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

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

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

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

VI. EXPERIMENT

A. The Environment of Experiment

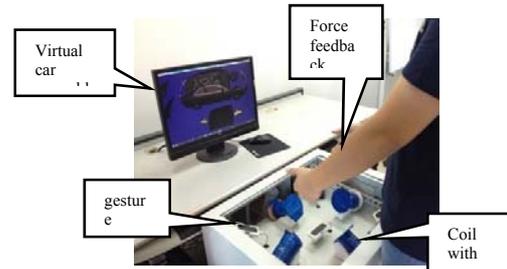


FIGURE III THE ENVIRONMENT OF EXPERIMENT

Fig. III shows the environment of our experiment. We use five Leap Motions to track the position of the hand and four coils with iron core to fulfill the force feedback and a computer to complete the virtual assembly operation. The operator stands in front of the table and wears a small permanent magnet to feel the force feedback. He uses three-dimensional gesture to make a virtual car assembly.

B. The Result of Experiment

In the process of operation, system provides real-time collision detection and assembly constraint processing, assembly path and sequence processing and other functions, so that we can analysis the assembly ability of the product, we can verify and plan the assembly sequence of product and so on. At the end of the assembly (or disassembly), the system can record all the information of the assembly process, and generate the evaluation report, video recording and so on for subsequent analysis.

VII. CONCLUSION

The paper presents a virtual assembly technology based on natural gesture. The operator can easily do a variety of assembly operations on the parts of the car using his/her 3D gesture like in a real environment.

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]. Kostić Z, Cvetković D, Jevremović A, et al. The development of assembly constraints within a virtual laboratory for collaborative learning in industrial design[J]. Tehnicki Vjesnik, 2013, 20(5):747-753.
- [2]. Jayaram S, Connacher H I, Lyons K W. Virtual assembly using virtual reality techniques[J]. Computer-Aided Design, 1997, 29(8):575-584.
- [3]. 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.
- [4]. 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.
- [5]. 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.
- [6]. 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.
- [7]. Anderson R J, Spong M W. Bilateral control of teleoperators with time delay[J]. IEEE Transactions on Automatic control, 1989, 34(5): 494-501.