Development of Kinect based teleoperation of Nao robot

Li, Chunxu

http://hdl.handle.net/10026.1/15867

10.1109/icarm.2016.7606908
2016 International Conference on Advanced Robotics and Mechatronics (ICARM)
IEEE

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.
Development of Kinect based Teleoperation of Nao Robot

Chunxu Li1, Chenguang Yang2, Peidong Liang3, Angelo Cangelosi1, Jian Wan4

Abstract—In this paper, an online tracking system has been developed to control the arm and head of a Nao robot using Kinect sensor. The main goal of this work is to achieve that the robot is able to follow the motion of a human user in real time to track. This objective has been achieved using a RGB-D camera (Kinect v2) and a Nao robot, which is a humanoid robot with 5 degree of freedom (DOF) for each arm. The joint motions of the operator’s head and arm in the real world captured by a Kinect camera can be transferred into the workspace mathematically via forward and inverse kinematics, realistically through data based UDP connection between the robot and Kinect sensor. The satisfactory performance of the proposed approaches have been achieved, which is shown in experimental results.

I. INTRODUCTION

With the rapid development of robot technology, the application of teleoperation with robots for social activities has been widely used. The field of research with respect to motion capture of human based on tracking system attracted great attention during the past decades [1]. In [2], the author used a Microsoft Kinect sensor intelligently to schedule a platform of multiple RGBD cameras especially in the Bayesian object tracking system. The kinematic validity of a Kinect based skeletal tracking system has been evaluated, which is designed to use with an upper limb virtual reality rehabilitation system [3]. In [4], a work on tracking the fingertips and palm center were developed based on Kinect sensor. However, researches focusing on head following is also necessary.

Head following technology has a wide range of applications, especially in the video face recognition authentication, security, surveillance and human-robot interaction [5]. Due to various factors such as, the low resolution of human face in video, illumination change and motion blur factors, the head following involves a number of challenges [6]. According to [7], a multilayer neural-net (NN) controller for limb following is developed. In [8], a research team presents a model-based method for analyzing a human body motion in order to follow a robot arm model which represents a human body motion.

In recent years, the Kinect sensor with depth information has increasing number of applications in some special areas, such as, the game entertainment, health and fitness [9]. In early 2014, Microsoft company developed a new generation of sensor, which is Kinect v2, while providing Kinect development kit SDK, which makes Kinect v2 can collect more accurate color information with depth, become more suitable for the head following project [10].

NAO robot has been widely investigated owing to its opening programming framework [11]. In the field of scientific research, researchers use the Nao robot in computer science, mathematics, physics and artificial intelligence [12]. At the same time, people develop the programming of robots into the education with Nao robot because of its convenience, for example, one Swedish university encourages students to understand the knowledge of voice recognition and visual feedback through the establishment of laboratory with Nao robots [13]. In commercial field, Nao robots have been used in some large production launches to highlight the high technology of the products [14].

II. PRELIMINARIES

The control system consists of two main parts: Kinect v2 camera and the Nao robot. Fig.1 shows the control block of the system, which has been implemented to verify the performance of our approach.

A. Kinect v2

Kinect v2 produced by Microsoft, is an RGB-D device, which can capture depth, colour, and IR images (also sound) [15]. The depth, IR and colour image resolutions are significantly improved from that of the first generation Kinect v1. Raw colour, depth and IR images can be captured from the device using the Kinect 2.0 SDK (Software Development Kit) from Microsoft. Using the SDK, captured colour and depth information can be consolidated (transformed) into real-world co-ordinates, called Camera Space; these co-ordinates are referenced to the centre of the depth sensor [16]. Skeletal tracking can also be achieved with the use of the Kinect 2.0 SDK; although this feature is not used in this
project, it could be used by other researchers to track the position of an operator in front of device. Since the point specified by the operator on the Kinect colour image is 2 dimensional, the Kinect depth space image is required to determine the depth of this point. Both the colour image, and depth image transformed into a common frame, the origin of which is located at the centre of the depth camera, called Camera space; the co-ordinate system of this space follows a right-handed convention [17](as shown in Fig.2).

![Fig. 2. Illustration of the origin of the Kinect v2s Camera space; the same as the depth sensor origin, modified from [18]](image)

**B. Nao Robot**

In this project, we choose the Nao robot (shown in Fig.3), which is produced by the Aldebaran-Robotics in France and it is a humanoid robot with 5 DOF joints [19]. The Nao robot supports multiple sensors and controllers, such as, the head and jaw cameras, the chest sonar sensors, movement motors on the neck, hands and feet, three colour LEDs (in red, green and blue) on the eyes, and the head and feet tactile sensors [20]. NAO has a body mass index (BMI) of about 13.5 kg/m², hence, compared to other robots with the same height, Nao is quite light [20]. According to [21], Nao has a total of 25 degrees of freedom (DOF), wherein 11 joints are for the legs and pelvis and the rest are for the trunks, arms and head. In addition, each arm is supported by a 2 DOF shoulder, a 2 DOF elbow, a 1 DOF wrist and a 1 DOF hands gripper. The head is able to rotate on both yaw and pitch axes [21].

![Fig. 3. Illustration of Nao Robot and coordinate system](image)

**III. CALIBRATION**

The two angles of head for the Nao robot can not be used directly, which needs to be placed into the same coordinate system as the one read from Kinect, where we use the coordinate transformation. We use the following transformation as shown in (1) and then we can obtain Nao’s coordinate transformation from the robot coordinates to the Kinect coordinates.

\[ X_i = TX'_i \]

where \( T \) represents the transformation matrix, we will give it concept as following \( X_i = [x_i \ y_i \ z_i \ 1]^T \), \( X'_i = [x'_i \ y'_i \ z'_i \ 1]^T \), which are the coordinates under both the robot and Kinect systems.

We can retain the transformation matrix \( T \) by calculating four points under both the robot and Kinect coordinate systems. Giving the assumption there are four points \( p_1, p_2, p_3 \) and \( p_4 \), whose coordinates are under the robot coordinate system and the Kinect coordinate system respectively. Fig.4 shows an example for one of the four points in relationship between the Nao robot and the Kinect coordinate system. \( T \) is the equation to calculate the transfer matrix \( T \) from the Kinect coordinates to the robot coordinates.

\[
T = \begin{bmatrix}
  x_1 & x_2 & x_3 & x_4 \\
  y_1 & y_2 & y_3 & y_4 \\
  z_1 & z_2 & z_3 & z_4 \\
  1 & 1 & 1 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
  x'_1 & x'_2 & x'_3 & x'_4 \\
  y'_1 & y'_2 & y'_3 & y'_4 \\
  z'_1 & z'_2 & z'_3 & z'_4 \\
  1 & 1 & 1 & 1
\end{bmatrix}
\]

In this project, we choose four positions under the same point on the central point of right hand for Nao robot which are placed on both Nao coordinate system and Kinect system as shown in Fig.5. To obtain the positions under Kinect coordinate system, we arrange a person stand in front of a Kinect sensor, which is used to obtain the coordinate data of human’s right hand, and keep his right hand place on the same position as the right hand of Nao robot. The four positions under Nao robot coordinate system are \((1, 1, 1), (2, -0.5, 0.5), (1, 0.5, 1.5), (1, -2, -3)\), under Kinect coordinate system are \((-0.0790, 0.3053, 0.3908), (0.0600, 0.2180, 0.5385), (-0.0012, 0.4000, -0.0703), (0.2364, -0.1443, 0.3090)\).

Substituting all the values of positions both on Nao and Kinect system into (2) we can calculate the transformation matrix of Kinect under ground based coordinate system \( T^{K}_0 \), where we assumed the coordinate system of Nao robot is ground based because the Nao robot keeps stationary in this project.
with Nao’s origin as its reference point. In order to match the orientations of the Camera space co-ordinate system and Nao co-ordinate system, the relevant axis of the Kinect co-ordinate system have been matched correctly with the axis of the Nao co-ordinate system.

A. Method to Get Orientation Angles of Head

The pose of a user’s head, namely the orientation angles $\alpha_f^k$, $\beta_f^k$, $\gamma_f^k$ are able to be collected from the statistic of the experiment by using visual studio based on Kinect v2. In order to send the data of orientation angles for user’s head, the angles need to be transformed into those under the basic ground coordinate, which is shown commonly in Fig.6. In this experiment, we keep the flat of Kinect camera be always parallel of the user’s head, where the value of $\beta_f^k$ equals zero. Then we have the matrix as follow, which represents rotation matrix:

$$\begin{bmatrix}
\alpha_f^k & \beta_f^k & \gamma_f^k
\end{bmatrix} = \begin{bmatrix}
\frac{\cos \alpha_f^k \cos \beta_f^k - \sin \alpha_f^k \sin \beta_f^k \cos \gamma_f^k}{\sin \beta_f^k} & \frac{-\cos \alpha_f^k \sin \beta_f^k - \sin \alpha_f^k \cos \beta_f^k \cos \gamma_f^k}{\sin \beta_f^k} & \frac{\sin \alpha_f^k \cos \gamma_f^k}{\sin \beta_f^k}
\end{bmatrix}$$

According to $R_0^f = R_0^K \times R_K^f$, wherein the $R_0^K$ has already been obtained through calibration process, as shown in (5), the rotation matrix $R_0^f$ for user’s head under the ground base coordinate system could be found by (7), where we denote the abbreviations $s_a$, $c_a$, $s_b^k$, $c_b^k$, $s_c^k$, $c_c^k$, $s_d^k$, $c_d^k$ for $s_a$, $c_a$, $s_b^k$, $c_b^k$, $s_c^k$, $c_c^k$, $s_d^k$, $c_d^k$ respectively.

Finally, the orientation angles of user’s head under the basic ground coordinate can be calculated, which are shown in (9)-(11), where we give the abbreviation:

$$\begin{bmatrix}
\alpha_f^0 = \arctan \frac{r_{21}}{r_{11}} \\
\beta_f^0 = \arctan \left( \pm \frac{\sqrt{r_{11}^2 + r_{21}^2}}{r_{31}} \right) \\
\gamma_f^0 = \arctan \frac{r_{32}}{r_{33}}
\end{bmatrix}$$

The image stream is captured from the Kinect v2 colour camera, such as the operators view of the Nao workspace. In order to select a point in the colour image which the operator desires the end effector of Nao to be transformed from the Kinect colour space to the Nao co-ordinate space. The pixels of the colour image could be transformed into the Camera space as follows:

$$[x_c, y_c] = R \cdot [x_k, y_k] + T_{\text{Trans}}$$

where $x_c$ and $y_c$ give the position of a transformed colour frame pixel (in Camera space), $x_k$ and $y_k$ give the position of the colour frame pixel, $R$ is the rotational matrix, and $T_{\text{Trans}}$ is the translation matrix, between the colour frame and Camera space. However, the Kinect v2 depth camera, suffers from inherent camera intrinsics, such as radial distortion (due to curvature of lens). Intrinsic parameters are given in one of the Kinect SDK examples, and were used to calculate the undistortion transformation matrix. It is now possible to transform any point in Camera space to Nao workspace,
in the same direction of vector \( \mathbf{i} \) to frame Hip-Left as the origin of basic coordinate, while seen in the right panel of Fig.8. After that, we take the point relative to the camera coordinate as shown in the left panel of initial position of each joint angle description of the coordination transformation from frame arm. The D-H parameters of the kinematic model of human arm shown in Fig.7 to reflect the 7-DOF model of our human.

**B. Method to Get 7 Joint Angles of Arm**

We create the Denavit-Hartenberg (D-H) coordinate system shown in Fig.7 to reflect the 7-DOF model of our human arm. The D-H parameters of the kinematic model of human arm are listed in Tab.I. According to the D-H method, the description of the coordination transformation from frame \( i \) to frame \( i - 1 \) has been shown in (2).

![Fig. 7. Human arm model and its primary D-H coordinate system and the initial position of each joint angle](image)

By using the Kinect sensor, we can receive the skeleton data of human directly, which include 25 joints 3D positions relative to the camera coordinate as shown in the left panel of Fig.8. Then, we create the geometry model of human arm as seen in the right panel of Fig.8. After that, we take the point Hip-Left as the origin of basic coordinate, while \( x - axis \) is in the same direction of vector \( \overrightarrow{AO} \). And \( y - axis \) is along with vector \( \overrightarrow{OC} \). It is easy to get the normal vector of each axis of the base coordinate, \( \overrightarrow{X_0}, \overrightarrow{Y_0} \) and \( \overrightarrow{Z_0} \):

\[
\overrightarrow{X_0} = \frac{\overrightarrow{AO}}{|\overrightarrow{AO}|}; \overrightarrow{Y_0} = \frac{\overrightarrow{OC}}{|\overrightarrow{OC}|}; \overrightarrow{Z_0} = \overrightarrow{X_0} \times \overrightarrow{Y_0}
\]  

(12)

In order to calculate the angle of each joint, we consider two conditions: i) the left thumb must be in the same plane with the palm because we need the plane which consisted by points of thumb, wrist and hand to calculate some other joints angles. ii) The angles of two vectors can only change from 0 to 180°, so there must be some other solutions we ignored.

\[
\begin{bmatrix}
7.87c_ac_b - 10.27(c_asb_s + s_asg) + 1.25(c_asb_s + s_asg) \\
7.87s_ac_b - 10.27(s_asb_s - c_acp) + 1.25(s_asb_s - c_acp) \\
-7.87s_b - 10.27c_ac_s + 1.25c_ac_s \\
2.37c_ac_b - 0.73(c_asb_s - s_acp) + 1.09(c_asb_s - s_acp) \\
2.37s_ac_b - 0.73(s_asb_s + c_acp) + 1.09(s_asb_s + c_acp) \\
-2.37s_b - 0.73c_ac_s + 1.09c_ac_s
\end{bmatrix}
\]

(7)

**TABLE I**

**MODEL REPRESENTATION OF THE DH PARAMETER TABLE [22]**

<table>
<thead>
<tr>
<th>Link Number</th>
<th>( \theta )</th>
<th>( d )</th>
<th>( a )</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \theta_1 )</td>
<td>0</td>
<td>0</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>2</td>
<td>( \theta_2 )</td>
<td>0</td>
<td>0</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>3</td>
<td>( \theta_3 )</td>
<td>0</td>
<td>0</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>4</td>
<td>( \theta_4 )</td>
<td>0</td>
<td>0</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>5</td>
<td>( \theta_5 )</td>
<td>( d_5 )</td>
<td>0</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>6</td>
<td>( \theta_6 )</td>
<td>0</td>
<td>0</td>
<td>( \pi/2 )</td>
</tr>
<tr>
<td>7</td>
<td>( \theta_7 )</td>
<td>0</td>
<td>( a_7 )</td>
<td>0</td>
</tr>
</tbody>
</table>

These seven angles are all in the joint space, and their initial positions are displayed in Fig.7. The geometry vector method we proposed is based on the principle of cosine value of two vectors shown in (17), in addition, the angle between two planes can be got by using their normal vector [22].

It is simple to find that angle \( \theta_1 \) is the supplement of the angle between plane COD and xOy.

\[
\theta_1 = \pi - \angle \overrightarrow{CO} \times \overrightarrow{CD}, \overrightarrow{CB} \times \overrightarrow{CO} >
\]

(13)

Angle \( \theta_2 \) is the angle between \( y - axis \) and vector \( \overrightarrow{CD} \), which is given below:

\[
\theta_2 = \angle \overrightarrow{CO}, \overrightarrow{CD} >
\]

(14)

In the same way, we can also get other angle: \( \theta_3 \) is the angle of plane BCD and CDE.

\[
\theta_3 = \angle \overrightarrow{CB} \times \overrightarrow{CD}, \overrightarrow{CB} \times \overrightarrow{CE} >
\]

(15)

Angle \( \theta_5 \) is the angle of plane CDE and DEH.

\[
\theta_5 = \angle \overrightarrow{CD}, \overrightarrow{DE} >
\]

(16)

Angle \( \theta_6 \) is the angle between vector \( \overrightarrow{CD} \) and plane EHF.

\[
\theta_6 = \Pi/2 + \angle \overrightarrow{EH} \times \overrightarrow{EG}, \overrightarrow{ED} >
\]

(17)

However, \( \theta_7 \) IS the yaw angle of wrist and it is difficult to calculate using the above method. \( \theta_7 \) can be seen as the angle between \( \overrightarrow{X_5} \) and \( \overrightarrow{Y_7} \). While,

\[
\overrightarrow{X_7} = \frac{\overrightarrow{EF}}{|\overrightarrow{EF}|}; \overrightarrow{Y_7} = \frac{\overrightarrow{EF} \times \overrightarrow{EH}}{|\overrightarrow{EF} \times \overrightarrow{EH}|}; \overrightarrow{Z_7} = \overrightarrow{Y_7} \times \overrightarrow{X_7}
\]

(19)
So now the problem became to solve $\vec{X}_5$. We know that $\vec{X}_5$ must be in the plane of EFH, in addition, $\vec{X}_5$ is Perpendicular to $\vec{D\bar{E}}$. Thus when suppose that:

$$\vec{X}_5 = k_1 \vec{EF} + k_2 \vec{EH}$$

(20)

There are:

$$(k_1 \vec{EF} + k_2 \vec{EH}) \times \vec{D\bar{E}} = 0; |k_1 \vec{EF} + k_2 \vec{EH}| = 1$$

(21)

Therefore,

$$\theta_7 = \theta_5, \theta_7$$

(22)

Up to now, all the joint angles can be calculated.

V. EXPERIMENTAL STUDIES

In this section, some experiments have been designed to test the performance of following system. Experimental platform: computer system is windows 10, Kinect SDK for windows, Visual studio 2013 and OpenCV library. The experiment environment is an indoor and adequate illumination environment. Only one person stands forward to Kinect in the distance about 2 meters. Because the degree of freedom for Nao robot’s arm is only 5, thus we choose the ShoulderPitch, ShoulderRoll, ElbowRoll, ElbowYaw and WristYaw joints from human arm to take the experiment.

A. Head Following Experiment

In this part, we run this experiment to test the stability of the head following system where we place a person in front of the Kinect camera. Then we run the program developed by the Kinect SDK 2.0 and send the command to an additional computer, which controls the Nao robot, to imitate the movement of human head. It has been shown in Fig.9, we run 4 times experiments with different postures in order to get the total performance, which are rotating $30^\circ$ to the left direction, rotating $30^\circ$ to the right direction, pitching up with $30^\circ$ and pitching down with $30^\circ$ respectively. The Fig.10 below shows the process of the first experiment. As a result, a graph can be drawn (seen in Fig.11), wherein the data has been collected from head angles for both operator and Nao robot. From the result of head following experiment, the movement of user’s head can be followed accurately.

![Fig. 9. Illustration of operator’s head for different postures](image)

Fig. 9. Illustration of operator’s head for different postures

From the first test above, we are able to conclude that the user’s head with different postures can be captured quite smoothly. The Nao robot imitates the movement of human head well. The result illustrates that there is a fast and accurate response to follow the user’s head and imitate the familiar movement with human in different postures.

B. Limb Following Experiment

In the second experiment, the operator keeps his body standstill and only puts his limb forwards at $90^\circ$ and raises up at $180^\circ$ and stretches his limb sidewards at $90^\circ$, with a low speed from the origin position to final position, for both of his right and left arms. Then we use the same method in the first experiment, after obtaining the data, we send it to the additional computer, which controls the Nao robot, to imitate the movement of human limb. The Fig.? below shows the process of the second experiment. As a result, a graph can be drawn (seen in Fig.13), wherein the data has been collected from arms for both operator and Nao robot. From the result above, we can conclude that all the movements of those 5 selected joints from user’s arm can be followed completely.

![Fig. 10. Process of imitation for Nao robot head following: Rotating 30° to left and right directions and pitching up and down with 30°](image)

![Fig. 11. Result of head following experiment for both human and Nao robot](image)

VI. CONCLUSION

A system for tracking user’s head and arm to interact with Nao robot via a Kinect v2 sensor has been developed in this paper. Kinect v2 is controlled to track the head’s and arm’s motion of the human operator. The operator can move their head and arm in a natural way to manipulate the position of the Kinect device. Additionally, the kinematics equations
In the near future, we would focus on teleoperation of Nao robot, which would be added to the project. We would like to thank Junshen Chen for his technical assistance.

**REFERENCES**


