Robotics and Autonomous Systems 79 (2016) 108–121

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

NAO humanoid robot: Analysis of calibration techniques for robot sketch drawing

Avinash Kumar Singh *, G.C. Nandi

Robotics and Artificial Intelligence Laboratory, Indian Institute of Information Technology Allahabad, Allahabad, U.P., India

HIGHLIGHTS

- The paper proposed 3 novel techniques to calibrate humanoid camera plane to its end-effector position. The solution is generalized consider the experimental setup is same.
- Extensively error analysis and time complexity analysis is performed to evaluate each proposed technique.
- Proposed techniques are computationally sound and require only few points (4 points) to calibrate the system. The error involved in calibration is also very less.

ARTICLE INFO

Article history:
Received 14 April 2015
Received in revised form
9 December 2015
Accepted 15 January 2016
Available online 12 February 2016

Keywords:
NAO humanoid robot
Sketch drawing
NAO calibration
DH principle

ABSTRACT

The strength of the NAO humanoid robot is discussed with several challenges in the arena of human portrait and sketch drawing. These challenges include extracting feature points from the input image, defining these points with respect to end effector, finding the inverse kinematics solution and designing a visual feedback system. This paper mainly addresses the fundamental issue of defining a relationship between the points of the image plane and NAO end effector position. This relationship enables NAO to perceive points of image plane with respect to its body coordinate system. Three different techniques based on the principles of fundamental matrix, pseudo inverse and Artificial Neural Network based regression analysis are applied to handle the calibration difficulties on NAO robot. A comprehensive study on sample points collected from NAO end effector position and corresponding image points has been made to understand the effectiveness of each technique. The degree of performance ability of each technique has been measured using time complexity, and mean square error metrics.

1. Introduction

Humanoid robots have apparently similar body structure like human beings. Their human like design enables them to share the same workspace with humans [1]. Apart from the appearance similarity they do have similar kind of human like sensors (e.g. they have vision sensors analogous to the human eyes, tactile sensors analogous to human touch sensors and they have speakers and microphones to produce as well as to hear sound, similar like human mouth and ear). The rapid development in the field of humanoid robotics during the last few decades shows the presence of several different kinds of humanoid robots in our society. Although they are coming under the same umbrella (Humanoids) yet their shapes, size and specification vary between manifestation [2–5]. For example ASIMO (Advanced Step in Innovative MObality) [6] from Honda Research Laboratory has 57 degree of freedom (DOF), HRP-4C [2], (Humanoid Robotics Platform-4(Cybernetic human)) from National Institute of Advanced Industrial Science and Technology (NIST) Japan has30 DOF, iCub [3] built by Italian Institute of Technology has 53 DOF while the other two popular robots Humanoid Open Architecture Platform -2 (HOAP2) [7] and NAO [8] developed by fujitsu and Aldebaran Robotics respectively have DOF as 25. One possible reason for these differences could be that they all have designed for different specific purposes and they are still in the development process. We have not yet able to discover the best shape and size of these humanoid robots.

Apart from the physical similarities with human beings, they do also have computation power which helps them to perceive and process different objects, gestures and human commands [6,9]. They can recognize human beings as well can do dialogs with them. WioNA [10](Wikipedia Ontology NAO) can communicate with you in its native Japanese language. It has also built in Japanese

* Corresponding author.
E-mail addresses: rs110@iiita.ac.in (A.K. Singh), gcnandi@iiita.ac.in (G.C. Nandi).

http://dx.doi.org/10.1016/j.robot.2016.01.009
0921-8890/

Wikipedia Ontology and Robot Action Ontology, which gives liberty to NAO to discuss various topics as well as express them in their different gestures. The other humanoid robots ASIMO [11] is inherited with various data analysis (audio/video/sensors) and processing tools which helps it to understand human intentions. The technology is getting transferred from the single standalone systems to these mobile robots, moreover research is also getting popular in the field of their cognitive architecture design [11,12] which facilitate them to learn things by themselves. A brief discussion of humanoid cognition and learning can be found in [13]. These advancements establish their presence in various household applications (cleaning, cooking, washing, etc.) [14,15]. They can play various sports like soccer, golf, etc., keeping their interest in sports a Robocup [16] is being organized every year. Robocup is a soccer game organized between two teams of humanoid robots having 5 members in a team. The objective of Robocup is to beat a Federation of International Football Association (FIFA) world cup winner's team under the rules and protocol of FIFA in 2050. They are also helping in educating by assisting their supervisor [17,18]. They are becoming a good housekeeper specially for assisting physically disabled persons. Their role in fighting against autism is very appreciable [19]. Their specially designed program for Autism Spectrum Disorder (ASD) children helps them in making interaction with them and to find other ways to deal with Autism [19]. They have also been tried in the military field where they can be used to rescue as well as to combat with the enemies [20,21], but there is always a question regarding risk, ethics and design to use these robots in the war field [22]. They are also turning to be a good entertainer [23].

Moving from the war, sports and household application they are creating their presence in the field of painting and sketch drawing. Every human wants to be drawn either for leaving trails to its successors or to satisfy their own ego. All of the approaches discussed in the following subsections are developed to create human portraits were inspired by this idea however some of them were designed to perform accurate painting for industrial purpose. The first work in this direction of portrait drawing is noticed in 1970s. A program designed by Harold Cohen named as "AARON" [24] to a pictorial representation of visual scenes. AARON has been in the continuous development phase and it has learnt the morphological representation of different objects like plants, animals and persons. There are a small number of robots specified to create drawing in an open loop without having any feedback system. The other advancement in this field is categorized into two wings (a) the first wing discusses the development with respect to manipulators while the (b) second wing focus more on humanoid robot drawing.

1.1. Painting/sketch drawing by manipulators

There are several inspiring as well as successfully running robotic machines which prove that robots can do better drawing and painting if they are trained properly. Haeberli [25] has developed different techniques such as sampling geometry using ray-painting, approximating images by relaxation, etc. to create static and animated images which are the abstract representation of the photograph and synthetic scenes. The type of brush used in the painting has great impact on the quality; therefore Hertzmann [26] varied the brush size to paint rough approximation of given images. The painting process is divided into several layers and for each layer different size of brush is used to keep the gradient of image into consideration. PUMA paint [27,28] was such an online painting project started in late 1998. The PUMA robot was connected to the internet which allows you to draw anything with the help of its Java based interface. The PUMA paint has accurate drawing capability; however it is not a fully automatic tool. The previous semi-automated techniques require human involvement. Later, several fully automated systems came into the picture which can create sketches in a fully automated manner. Collomosse [29] defined an automatic system to control brush stroke and sizes based on the image salience and gradient. The other issue associated with these systems was to accumulate feedback system. Therefore the vision based system has been taken in use. The vision guided system helps in providing the feedback to the system as well as helping in achieving the full automation [30–32]. Rectification of the input image and proper gripping are the two major challenges in sketch drawing/painting. The 3 step solution has been discussed by Shunsuke et al. [33] to discuss these challenges. The first step used to create the 3D model of the object in order to recognize it; the second module used to extract the edge, line, etc. like features from the transformed silhouette images and the third module discuss how multi fingered hand paints the geometry. Paul a robotic hand–eye system designed by Tresset et al. [34,35] does the aesthetic human face sketching more accurate than all the previous approaches. It has 4 degrees of freedom with the camera placed at different body. It has efficient feedback control and facial feature extraction modules. The other kind of robotic manipulator based painting has also been developed in order to paint cars and other objects [36,37].

1.2. Painting/sketch, drawing by humanoids

The first fully automated humanoid system which could be tried as sketch artist is developed at EPFL taking help of HOAP-2 robot [38]. The robot can draw the sketch of any person who is sitting in front of him. The primitive techniques of face detection, boundary and edge extraction with the trajectory planning are used in this application. The other effort has been made by Srikaew et al. [39] to create artistic portraits using the ISAC robot. The beauty of using ISAC as a sketch artist is due to their soft hands who can mitigate the drawing as fine as humans. They have McKibben artificial muscle, which is an excellent actuator while the stereo vision helps it in a 3D representation of the object.

After analyzing the existing literature it has been observed that development in the field of painting/ sketch drawing by manipulators has gained much attention in comparison to humanoids. One of the possible reasons could be the extra and redundant degrees of freedom associated with robotic hands. Moreover reachable work space of humanoid robot and solving the inverse kinematics for at least 5 degrees of freedom also act as an inhibitor. We have provided a solution (calibration) which defines the image plane with respect to an NAO end effector. Once NAO define each point of the image plane with respect to its body coordinate system, the only thing left is to calculate the feasible inverse kinematic solution. This paper restricts itself for estimating the content of the calibration metric only. The workflow of the paper is summarized in Fig. 1. The first step is to extract the boundary point using any boundary extraction technique. This problem can be solved by applying the geometrical feature extraction techniques of image processing. The second problem of representing the image points corresponds to NAO end effector is divided over two sub problems. The first problem is to map the image points with respect to table coordinate (the place where NAO supposed to draw sketches) and the second issue is to represent table coordinate system with respect to an NAO end effector. The first problem is solved by applying the linear transformation while the second problem has been tackled by applying three different approaches. These approaches are derived from the concept of the fundamental matrix, pseudo inverse and regression analysis. Fundamental matrix approach basically used in the stereo vision is to find the popular lines. These epipolar

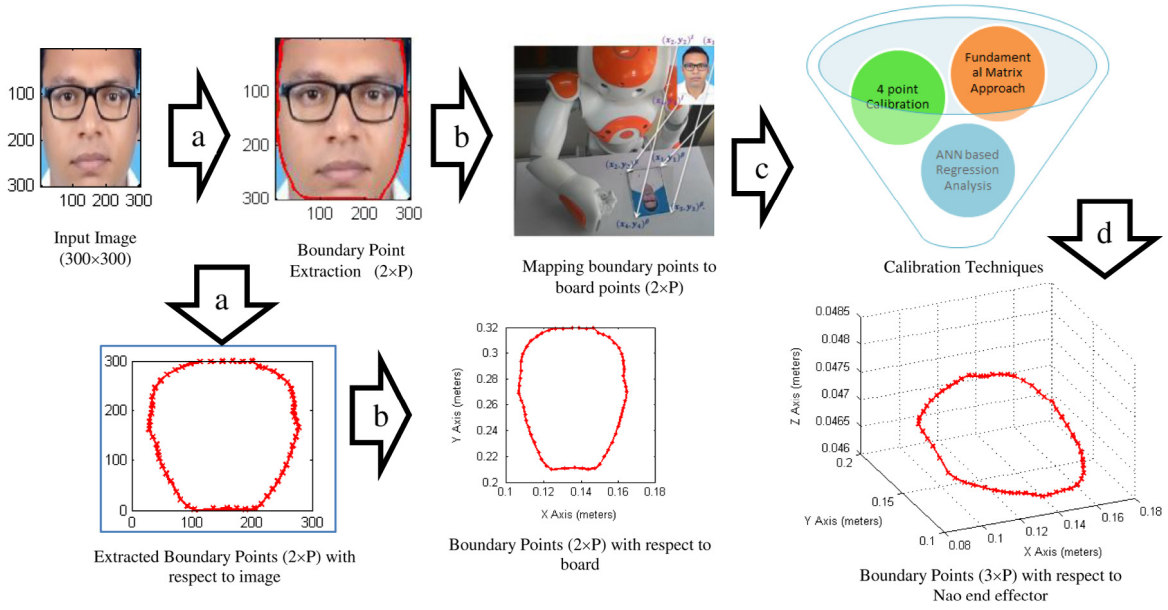

Fig. 1. Flow chart of the proposed work.

lines help to determine the corresponding points in right image given by the points in left image. We have exploited the same concept to align these two different coordinate systems. Harley 8 point algorithm is applied at the points collected from the NAO end effector position and corresponding point lies on the table coordinate system. The second approach named as four point calibration used only four points to find the content of calibration matrix. The third technique uses Artificial Neural Network to find the regression line which helps in approximating the given table point with respect to NAO body coordinate system. The performance of these techniques has been evaluated on the basis of their respective time complexity and the mean square error involved in the project.

The paper is constructed as follows. Section 2 presents the background of NAO robot and its DH specification. Section 3 describes the problem definition in details. In Sections 4 and 5, we proposed the solution using three different transformations based on the concept of fundamental matrix, four point calibration and regression analysis. Section 6 performed results and discussion of our proposed transformations and shows the critical analysis and comparison with each other. In the last Section 7, we conclude the paper with its contribution towards the sketch drawing by Humanoids and its future prospects.

2. Estimation of NAO DH parameters

Jacques Denavit and Richard Hartenberg in 1955 have given the method which is widely known as DH principle to establish the relation of one link to another link by a 4×4 transformation [40]. If there are different links connected in a sequential chain function and assume that any of the link is moving then it will be applying its effect on the rest of the joints connected with this. DH principles can help us to build a relation between these connected links in understanding their relative motion. We are using DH principles to define the end effector position with respect to base. There are four parameters defined by Denavit and Hartenberg which can fully describe the content of the 4×4 transformation matrix. These four parameters are link length (a), twist angle (α), link offset (d) and joint angle (θ). The descriptions of all four parameters are given below.

- a_i = the distance between Z_i to Z_{i+1} measured along X_i, it is called common normal.

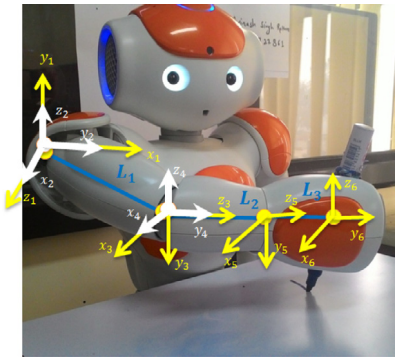

Fig. 2a. Coordinate frame attached on NAO's right hand. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- α_i = the angle from Z_i to Z_{i+1} measured about X_i.
- d_i = the distance between x_{i-1} to x_i measured along z_i.
- θ_i = the angle from x_{i-1} to x_i measured about Z_i.

The first step in DH principle is to attach the coordinate frame for each axis. There are two mandatory conditions which should be followed at the time of coordinate attachment.

1. The Z_i-axis is set to the direction of the joint axis (the rotation direction).
2. The X_i-axis is parallel to the common normal between Z_i and Z_{i-1}.

NAO has five degrees of freedom in its right as well as left hand [5]. The specification of both right and left hand are same except their joint's angle of rotation [41]. For making it simple, we have only configured the DH parameter table for the NAO right hand. In order to establish the DH parameter for NAO, it is necessary to define the different degree of freedom associated with each link. Figs. 2a and 2b briefly describes different joints and their rotation axis.

Fig. 2b is the sketch representation of Fig. 2a for giving more insight of the problem. We have added link length and their axis of rotation additionally for better explanation of the NAO DH parameters. We have used two different colors, yellow and white in Fig. 2a and blue and orange in Fig. 2b for symbolizing two different joints at the same place. The DH parameters estimated based on above specification are summarized in Table 1.

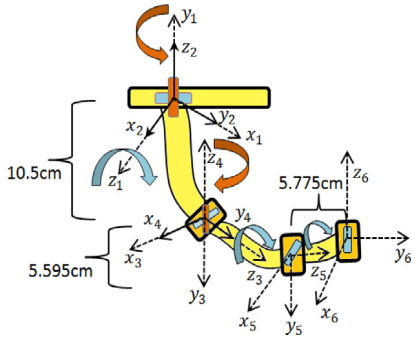

Fig. 2b. Sketch representation of NAO's right hand with axis of rotation and link specifications. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
NAO's right hand DH parameters.

Link i	Twist angle α_{i-1}	Common normal a_{i-1}	Offset length d_i	Joint angle θ_i
1	0	0	0	θ_1
2	$-90°$	0	0	$\theta_2 - 90$
3	$-90°$	0	10.5 cm	θ_3
4	$+90°$	0	0	θ_4
5	$-90°$	0	5.595 cm	θ_5
6	$+90°$	0	5.775 cm	0

The twist angle is measured between Z_i to Z_{i+1}, we have assumed Z_0 and Z_1 At the same location, therefore the twist angle for first link is set to 0. The other links have their twist angle as $\mp 90°$ based on their orientation with respect to frames. The common normal is also set to zero for all links because the z axis of all consecutive links are orthogonal to each other. The link offset is measured along x_{i-1} to x_i. Therefore, it is set to zero for all links that have a common origin. Joint angle is variable because each joint has its own rotation range. The transformation between links $i-1$ to i is defined as:

$$T_1^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_2^1 = \begin{bmatrix} \sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ -\sin\theta_3 & -\cos\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_4^3 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_5^4 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ 0 & 0 & 1 & l_2 \\ -\sin\theta_5 & -\cos\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_6^5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

There are six transformation matrices in existence, which are capable of deriving the relation between the end effector position with respect to the first link of right hand. We can define a relation of 6 links to 1st link using Eq. (1).

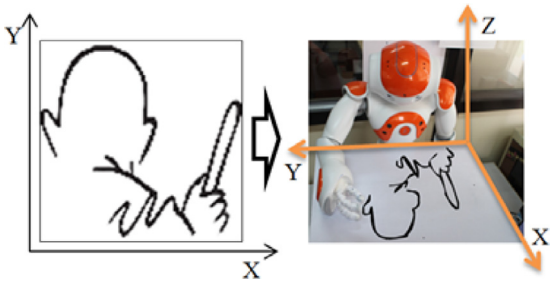

Fig. 3. Problem definition (transformation from image plane to NAO end effector position).

$$T_{endEffector}^{1st\ link} = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 \times T_5^4 \times T_6^5. \quad (1)$$

3. Problem definition

Given the point $(x_i, y_i)^I$ corresponds to image plane where $i \in$ imageHeight, $i \in$ imageWidth find out the transformation matrix T such that it will define each point of image plane with respect to NAO end effector position.

The problem is also described in Fig. 3. The first image shows its presence in the image plane later it will be transformed and NAO will perceive it in the form of its end effector position. If $(x', y', z')^{(\eta)}$ are the points corresponding to NAO end effector and $(x, y)^I$ are the points corresponding to Image then the transformation matrix would be:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}^{\eta} = [T] \times \begin{bmatrix} x \\ y \end{bmatrix}^{I}. \quad (2)$$

The direct transformation from image plane to NAO end effector could be decomposed into two sub problems. First, we can transform the point of the image with respect to table coordinate system (NAO supposed to draw image points on the table) and then we can define the second transformation which we will transform to each point of the table with respect to an NAO end effector. If we introduce the $(x, y, z)^{\beta}$ with respect to table the whole problem will reduce as given below.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}^{\eta} = [T_2] \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}^{\beta} \quad (3)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^{\beta} = [T_1] \times \begin{bmatrix} x \\ y \end{bmatrix}^{I}. \quad (4)$$

4. Proposed solution

In order to map points from image plane to NAO end effector, it is required to first transform the point with respect to table coordinate and then transform the point in terms of NAO end effector position. In order to do that first we have to physically estimate to what region NAO end effector can reach with respect to the table origin. Once the reachable region's coordinate are retrieved with respect to table origin, the next task is to map every pixel of the image to this reachable region. Let the four reachable corner points defined after the physical measurement of the table with respect to table origin are $(\boldsymbol{x_1}, \boldsymbol{y_1})^{\beta}$, $(\boldsymbol{x_2}, \boldsymbol{y_2})^{\beta}$, $(\boldsymbol{x_3}, \boldsymbol{y_3})^{\beta}$, $(\boldsymbol{x_4}, \boldsymbol{y_4})^{\beta}$ Similarly, we can define the image with four corner points

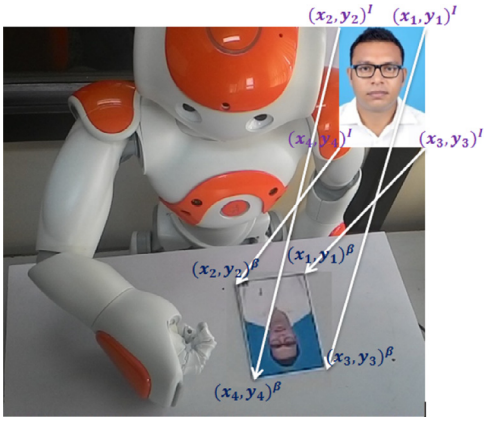

Fig. 4. Image and table coordinate system.

$(\boldsymbol{x_1}, \boldsymbol{y_1})^{\boldsymbol{I}}$, $(\boldsymbol{x_2}, \boldsymbol{y_2})^{\boldsymbol{I}}$, $(\boldsymbol{x_3}, \boldsymbol{y_3})^{\boldsymbol{I}}$, $(\boldsymbol{x_4}, \boldsymbol{y_4})^{\boldsymbol{I}}$. Both image and table's four corners and their mapping is presented in Fig. 6. The z axis of the table have been kept as constant. The relationship between these coordinate systems can be derived with the linear transformation given in Eqs. (4) and (6).

$$\beta_X = \min(\beta_X) + \frac{X - \min(I_X)}{\max(I_X) - \min(I_X)} * (\max(\beta_X) - \min(\beta_X)) \tag{5}$$

$$\beta_Y = \min(\beta_Y) + \frac{Y - \min(I_Y)}{\max(I_Y) - \min(I_Y)} * (\max(\beta_Y) - \min(\beta_Y)) \tag{6}$$

$$\theta^\beta = \left\{\max(x_1, x_2, x_3, x_4)^\beta\right\}, \quad \Theta^\beta = \left\{\min(x_1, x_2, x_3, x_4)^\beta\right\},$$
$$\theta^I = \left\{\max(x_1, x_2, x_3, x_4)^I\right\}, \quad \Theta^I = \left\{\min(x_1, x_2, x_3, x_4)^I\right\},$$
$$\rho^\beta = \left\{\max(y_1, y_2, y_3, y_4)^\beta\right\}, \quad \sigma^\beta = \left\{\min(y_1, y_2, y_3, y_4)^\beta\right\},$$
$$\rho^I = \left\{\max(y_1, y_2, y_3, y_4)^I\right\}, \quad \sigma^I = \left\{\min(x_1, x_2, x_3, x_4)^I\right\}$$

θ^β, Θ^β, ρ^β, σ^β are the maximum and minimums along the x and y coordinates with respect to table while θ^I, Θ^I, ρ^I, σ^I are the maximum and minimums along the x and y axis with respect to Image coordinate system. A mapping from image plane to table coordinate system is shown in Fig. 4.

Here we have taken the input image dimension as 300 × 300 pixel and the reachability physically measured by using NAO with respect to table coordinate is 15 × 10 cm. The image points are extracted from the raw input image with the help of classical canny edge detection [42]. The extracted edge points are shown in Fig. 5(b) and the transformed points retrieved after applying the linear transformation as described in Fig. 5(c).

4.1. Estimation of homogeneous transformation matrix

The first step towards solving the inverse kinematic problem for NAO right hand is to establish a relation between the points

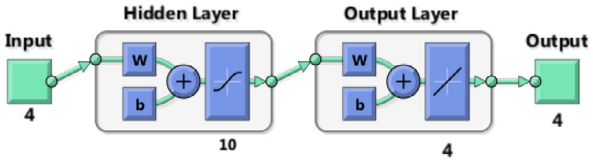

Fig. 6. Artificial neural network structure.

lies on the table surface with respect to NAO body coordinate. Both coordinate systems can vary with respect to each other in terms of scaling, translation and rotation. Hence, in order to make a proper alignment with these three parameters should be estimated first. As the table coordinates and NAO coordinate both are represented in the same unit (meters) we assume that scaling parameter will not play any role here. Hence, it is worthy to estimate only the remaining two parameters. The calibration matrix which would transform the points from one coordinate frame to another will be made up of rotation and translation parameters. In the given circumstances, this is the simple homogeneous transformation matrix given in Eq. (7).

$$(x'y'z')^{(\eta)} = H \times (x, y, z)^\beta \tag{7}$$

where, $(x'y'z')^{(\eta)}$ represents points on the NAO coordinate frame. H is the homogeneous transformation matrix and $(x, y, z)^\beta$ are the points represented with respect to table coordinate frame.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 & T_x \\ r21 & r22 & r23 & T_y \\ r31 & r32 & r33 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}. \tag{8}$$

The component of the calibration matrix is in the form of $H = \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0_{1\times3} & 1 \end{bmatrix}$. R represents the rotation along x, y and z axis, whereas T represents the translation along this axis. The combined rotation of all three axes is given in Eq. (2) while the translation is shown in Eq. (9) given in Box I.

Similarly

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \tag{10}$$

Three different approaches fundamental matrix, 4 point calibration and artificial intelligence based regression has been introduced to estimate the content of the homogeneous transformation matrix H. In order to solve the above equation we have collected 60 points from the table coordinate frame and corresponding points represented with respect to NAO end effector.

5. Calibration techniques

Three different approaches derived from the concept of fundamental matrix, Pseudo inverse and non-linear regression

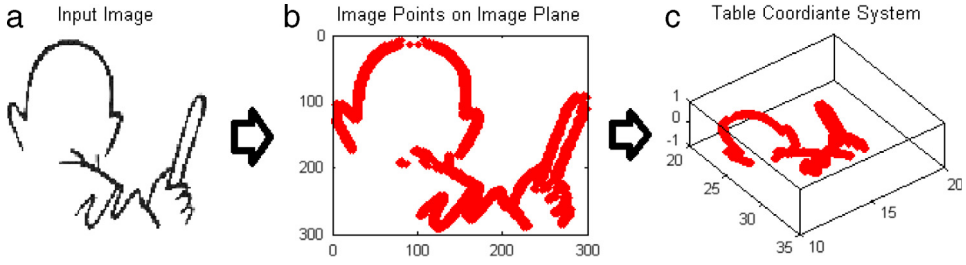

Fig. 5. Transformation from image Plane to table coordinate system. (a) Raw Input image. (b) Boundary points (x_i, y_i) extracted with canny edge detection. (c) Points corresponding to table coordinate system (after transformation).

$$R = \begin{bmatrix} \cos\theta_y \cos\theta_z & -\cos\theta_x \sin\theta_z + \sin\theta_x \sin\theta_y \cos\theta_z & \sin\theta_x \sin\theta_z + \cos\theta_x \sin\theta_y \cos\theta_z \\ \cos\theta_y \sin\theta_z & \cos\theta_x \cos\theta_z + \sin\theta_x \sin\theta_y \sin\theta_z & -\sin\theta_x \cos\theta_z + \cos\theta_x \sin\theta_y \sin\theta_z \\ -\sin\theta_y & \sin\theta_x \cos\theta_y & \cos\theta_x \cos\theta_y \end{bmatrix}. \quad (9)$$

Box I.

fitting using artificial neural network has been discussed in this section. Calibration is the process of establishing the coefficient of the homogeneous transformation matrix explained in the above section.

5.1. Fundamental matrix approach

The problem of estimating the value of $r11, r12, r13, \ldots, r33$ expressed in Eq. (8) is analogous to estimation of fundamental matrix in stereovision calibration. Fundamental matrix has the great impact in finding the correct epipolar line with respect to the given point either from left frame or right frame [43]. The same way estimation of fundamental matrix will help us to find the representation of table coordinate frame with respect to NAO coordinate frame or vice versa. To do, this is mandatory that both the coordinate frames satisfy the equation of fundamental matrix given in Eq. (12).

$$(P^{\eta})_{4\times1} = (F)_{4\times4} \times (P^{\beta})_{4\times1} \quad (11)$$

$$P^{\eta} \times F \times P^{\beta} = 0. \quad (12)$$

Here P^{η} represents the point of the same object corresponding the NAO end effector position, F is the fundamental matrix shown in Eq. (13) while P^{β} represents the points corresponding to table coordinate frame.

$$F = \begin{bmatrix} f1 & f2 & f3 & f4 \\ f5 & f6 & f7 & f8 \\ f9 & f10 & f11 & f12 \\ f13 & f14 & f15 & f16 \end{bmatrix}. \quad (13)$$

Let the point $(x'y'z') \in \eta$ and $(x, y, z) \in \beta$ then putting these points in Eq. (14) we will get:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}^T \times \begin{bmatrix} f1 & f2 & f3 & f4 \\ f5 & f6 & f7 & f8 \\ f9 & f10 & f11 & f12 \\ f13 & f14 & f15 & f16 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0. \quad (14)$$

The above equation can be further simplified and represented as given in Eq. (15)

$$\begin{aligned} &f1xx^1 + f2yx^1 + f3zx^1 + f4x^1 + f5xy^1 + f6yy^1 \\ &\quad + f7zy^1 + f8y^1 + f9xz^1 + f10yz^1 + f11zz^1 + f12z^1 \\ &\quad + f13x + f14y + f15z + f16 = 0. \end{aligned} \quad (15)$$

This is the linear equation of 16 unknown; it can be solved by applying Hartley 8 point calibration [43] described in Algorithm 1 while the description of the symbol is presented in the Table 5 listed in Appendix. Hartley 8 point calibration [43] is the heart of the estimation of the fundamental matrix. It starts with collecting n points from both the coordinate system and normalization of these points to achieve the same scale. So that the solution will be scale independent. Singular value decomposition (SVD) of the collected points helps to find the characteristic root of the equation. Further the selection of minimum Eigen vector and representation in the form of a 4×4 matrix reduces the solution size. The degree of solution should be 3 as we have the 3 dimensional data. Therefore, it is required to reduce rank of the matrix and to make it three. SVD is applied on the 4×4 matrix in order to make the rank 3 as well as to find the final fundamental matrix.

5.2. 4 Point calibration

The cost of the transformation presented in Eq. (14) can be further estimated with the help of 4 point calibration. We need to only have 4 points of table and the corresponding points defined in NAO's body coordinate system. If we expand Eq. (14) we will get following equations.

$$x' = f1 * x_i + f2 * y_i + f3 * z_i + f4 \quad (16)$$

$$y' = f5 * x_i + f6 * y_i + f7 * z_i + f8 \quad (17)$$

$$z' = f9 * x_i + f10 * y_i + f11 * z_i + f12. \quad (18)$$

Here $i \in [1, n]$, n is the number of calibration points (we kept $n = 4$). After expanding Eqs. (16), (17), (18) for all four calibration points, we will get.

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ x'_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{bmatrix} \times \begin{bmatrix} f1 \\ f2 \\ f3 \\ f4 \end{bmatrix}. \quad (19)$$

$A = T * B.$

If we extend this solution for rest of the two axes, we will see that the transformation matrix contents are same. This shows that there is no need to estimate the transformation matrix for each independent axis. The solution of Eq. (19) can be obtained by simply solving for $B = T^{-1} * A$.

5.3. Regression analysis with artificial neural network

A linear regression analysis has been used to find out the predicted value of any of the table coordinate in terms of NAO's body coordinate. This is the case of linear regression analysis having one independent and one dependent variable. Table's coordinates are considered to be the independent coordinates whereas the NAO's coordinate system is assumed to be the dependent. If we use the regression analysis to estimate the value of dependent variable, a prediction function always exists which transforms the value of independent variable to the predicted value. As this is the case of linear regression, the prediction function would be the equation of line. Given as: $y = m * x + c \ldots(1)$, here y is the predicted value of the input variable x, m is the magnitude and c of the line and it is a constant value. We took help of artificial neural network to find the coefficient of the line. Artificial Neural Network widely known as the abbreviation of ANN, it has the similar analogical structure which we are having in our brain. It is basically three layer architecture shown in Fig. 6, where at the first layer we supply input to the system, the second layer is the hidden layer where learning takes place and at the third layer output is defined. The structure of neural network, it consists of one input layer which accepts a four dimensional table coordinates as an input vector. One hidden layer having 10 neurons, one output layer having four neurons in the output corresponds to NAO's body coordinate frame. There are 60 samples of tables and of NAO's body coordinates, we kept 75% of the population as training while 15% for both the testing and validation. We have used random function to split the population over training, testing and validation population. We have used Levenberg–Marquardt back propagation [44] algorithm in order to train the network. Sigmoidal function ($F(X) = 1/(1+e^{-x})$) is used

Algorithm 1: Pseudo code for the Estimation of Fundamental Matrix

Input: $(P'_i)^{\eta}$ and $(P_i)^{\beta}$ where $i \in [1, N]$, $(P', P) \in \Re^4$ *and* $\forall (P', P)\ \exists\ (x, y, z, 1)$

Output: $Fundamental\ Matrix : F_{4\times4}$.

Begin:

Step1: Calculate $\mu'_x \leftarrow \frac{1}{N}\sum_{i=1}^{N} x_i$, $\mu'_y \leftarrow \frac{1}{N}\sum_{i=1}^{N} y_i$, $\mu'_z \leftarrow \frac{1}{N}\sum_{i=1}^{N} z_i\ \forall\ (P')^{\eta}$, similarly calculate $\mu_x, \mu_y, \mu_z\ \forall\ (P)^{\beta}$

Step2: Calculate $(P^{1'}{}_i)^{\eta}$: $P^{1'}{}_i(x) \leftarrow P'_i(x) - \mu'_x$, $P^{1'}{}_i(y) \leftarrow P'_i(y) - \mu'_y$, $P^{1'}{}_i(z) \leftarrow P'_i(z) - \mu'_z$, similarly calculate $(P^{1}{}_i)^{\beta}$.

Step3: Calculate $(S_{1i})^{\eta} \leftarrow \sqrt{(x_i^2 + y_i^2 + z_i^2)}$ & $(S_{2i})^{\beta}\sqrt{(x_i^2 + y_i^2 + z_i^2)}$ where $i \in [1, N]$

Step4: $(\overline{\mu_1})^{\eta} \leftarrow \frac{1}{N}\sum_{i\leftarrow1}^{N}(S_{1i})^{\eta}$ *and* $(\overline{\mu_2})^{\beta} \leftarrow \frac{1}{N}\sum_{i\leftarrow1}^{N}(S_{2i})^{\beta}$

Step5: $Scale^{\eta} \leftarrow \sqrt{3}/(\overline{\mu_1})^{\eta}$ and $Scale^{\beta} \leftarrow \sqrt{3}/(\overline{\mu_2})^{\beta}$

Step6:
$$T^{\eta} \leftarrow \begin{bmatrix} Scale^{\eta} & 0 & 0 & -Scale^{\eta} \times \mu'_x \\ 0 & Scale^{\eta} & 0 & -Scale^{\eta} \times \mu'_y \\ 0 & 0 & Scale^{\eta} & -Scale^{\eta} \times \mu'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and}$$
$$T^{\beta} \leftarrow \begin{bmatrix} Scale^{\eta} & 0 & 0 & -Scale^{\eta} \times \mu_x \\ 0 & Scale^{\eta} & 0 & -Scale^{\eta} \times \mu_y \\ 0 & 0 & Scale^{\eta} & -Scale^{\eta} \times \mu_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Step7: $(P^{new'}{}_i)^{\eta} \leftarrow T^{\eta} \times (P'_i)^{\eta}$ and $(P^{new}{}_i)^{\beta} \leftarrow T^{\beta} \times (P_i)^{\beta}$

Step8: *from equation* (9) // $A \times F \leftarrow 0$

$$[x_i x_i' \quad y_i x_i' \quad z_i x_i' \quad x_i' \quad x_i y_i' \quad y_i y_i' \quad z_i y_i' \quad y_i' \quad x_i z_i' \quad y_i z_i' \quad z_i z_i' \quad z_i' \quad x_i \quad y_i \quad z_i \quad 1] \times \mathrm{F} \leftarrow 0$$

Step9: Calculate $(eigen\ value)\lambda_i$ & $(eigen\ vector)\ \Psi_i\ of\ A$.

Step10: Find Ψ *corresponds to minimum* λ and assign it as F.

Step11: $F \leftarrow \Psi_{4\times4}$ // reshape Ψ *to make it a matrix of dimension* 4×4.

Step12: *Apply singular value decomposition* (SVD) *on F to calculate upper tringular matrix* (U), $(eigen\ value)\lambda$ & $(eigen\ vector)\Psi$

Step13: $\mathrm{F} \leftarrow U_{4\times4} \times [\lambda_{1,1} \quad \lambda_{2,2} \quad \lambda_{3,3} \quad 0]^{t}{}_{4\times1} \times \Psi^{t}{}_{1\times4}$

Step14: $\mathrm{F} \leftarrow T^{\beta} \times F \times T^{\eta}$

End:

Table 2
ANN specification.

Technical specifications	
Input	4 dim
Hidden Layer Neurons	10
Output	4 dim
Training function	trainlm
Learning Constant	0.001
Stopping Criteria	MSE
Training, Test, Validation population	75%,15%,15%

as the transfer function form input to hidden layer, while linear transfer function ($f\ (x) = a{*}x{+}b$) is used as the activation function from hidden layer to output layer. Here a & b are the coefficient of the line. The learning constant is 0.001 and the stopping criteria is the mean square error, if the mean square error is less than the threshold, we will stop the weight updating as well as training phase. The specification of the Artificial Neural network Structure is expressed in Table 2.

Design of Neural Network Learning Algorithm for estimating Calibration matrix

The primary aim of training algorithm is to update the weight matrix and biases involved in the network. As soon as the weight and biases are updated and there is no further change is them, we believe that the network is trained. The training section has several sub modules define below.

Step 1: Feed Forward (Input to output): Let "$\underline{p}$" is the input to the network and a^k is the output at layer k, $k+1, \ldots, k+2, \ldots, M$. The computation performed for each neuron at each layer will be:

$$\underline{a^{k+1}} = \underline{f^{k+1}}(w^{k+1}\underline{a^k} + \underline{b^{k+1}}) \quad \text{where } k = 0.1.2 \ldots, M-1.$$

Here "w" is the weight matrix, M represents the number of layer in the neural network, "f" denotes the transfer function and "b" represents the bias corresponding to layer "k". At the first layer output will be equal to the input vector: $(\underline{a^0} = \underline{p})$. The principle of learning algorithm is to minimize the gap between the input vector $(\underline{p})$ and the target vector $(\underline{t})$. The training of the network is evaluated based on the fitness or performance function. In this case the performance function is evaluated with:

$$V = \frac{1}{2}\sum_{i=1}^{Q}(\underline{t}_q - \underline{a}_q^M)^T(\underline{t}_q - \underline{a}_q^M) = \frac{1}{2}\sum_{q=1}^{Q}\underline{e}_q^T\underline{e}_q.$$

Here $\underline{a}_q^M$ is the output at the given input $\underline{p}_q$ while the error is evaluated based on the difference between the target and the desired. For the qth input it is calculated by: $\underline{e}_q = \underline{t}_q - \underline{a}_q^M$.

Step 2: Weight update: The error could be minimized by taking the derivative of the error the same way as we do in case of gradient descent learning algorithm. The change in weight (Δw) and bias (Δb) will be calculated by:

$$\Delta w^k(i,j) = -\alpha \frac{\partial \hat{V}}{\partial w^k(i)} \quad \text{and} \quad \Delta b^k(i,j) = -\alpha \frac{\partial \hat{V}}{\partial b^k(i)}.$$

Here α is the learning constant defined by:

$$\partial^k(i) \equiv= \frac{\partial \hat{\hat{V}}}{n^k(i)}.$$

Similarly the change with respect to each layer with respect to the performance function V is:

$$\frac{\partial \hat{V}}{\partial w^k(i,j)} = \frac{\partial \hat{V}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial w^k(i,j)} = \partial^k(i) a^{k-1}(j)$$

$$\frac{\partial \hat{V}}{\partial b^k(i)} = \frac{\partial \hat{V}}{\partial n^k(i)} \frac{\partial n^k(i)}{\partial b^k(i)} = \partial^k(i).$$

Step 3: Back Propagation (Output to Input): Let $\underline{x}$ is the input vector and V is the fitness function. In order to fit a best curve it is required that the $V(\underline{x})$ should be minimum. If we apply the Newton's method it would be:

$$\Delta \underline{x} = -[\nabla^2 V(\underline{x})]^{-1} \nabla V(\underline{x})$$

where $\nabla^2 V(\underline{x})$ is the Hessian matrix and $\nabla V(\underline{x})$ is the gradient. As required the intention of the fitness function is to minimize the sum of square error between the target and the desired and it can be calculated by:

$V(\underline{x}) = \sum_{i=1}^{N} e_i^2(\underline{x})$ where "e" represent the error for the corresponding input "i". The error can also be minimized by calculating the first and second derivative of the error and then by compensating it by sending it back in the network.

$$\nabla V(\underline{x}) = J^T(\underline{x}) \underline{e}(\underline{x})$$

$$\nabla^2 V(\underline{x}) = J^T(\underline{x}) J(\underline{x}) + S(\underline{x}).$$

Here $J(\underline{x})$ represents the Jacobian matrix for the input vector ($\underline{x}$).

$$J(\underline{x}) = \begin{bmatrix} \frac{\partial e_1(\underline{x})}{\partial x_1} & \frac{\partial e_1(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_1(\underline{x})}{\partial x_n} \\ \frac{\partial e_2(\underline{x})}{\partial x_1} & \frac{\partial e_2(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_2(\underline{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\underline{x})}{\partial x_1} & \frac{\partial e_N(\underline{x})}{\partial x_2} & \cdots & \frac{\partial e_N(\underline{x})}{\partial x_n} \end{bmatrix}.$$

In this case the sum of square can also be represented in the form of:

$$S(\underline{x}) = \sum_{i=1}^{N} e_i(\underline{x}) \nabla^2 e_i(\underline{x}).$$

If the sum of square is almost zero or tending to be zero ($S(\underline{x}) \cong 0$) then Eq. (15) will become (when $S(\underline{x}) \cong 0$, this is the special case of Gauss–Newton method).

$$\Delta \underline{x} = [J^T(\underline{x}) J(\underline{x})]^{-1} J^T(\underline{x}) \underline{e}(\underline{x}).$$

But when the $S(\underline{x}) \cong 0$, the problem can be modified and solved by using Marquardt–Levenberg [44].

$$\mu \Delta \underline{x} = [J^T(\underline{x}) J(\underline{x}) + \beta]^{-1} J^T(\underline{x}) \underline{e}(\underline{x}). \tag{20}$$

Here μ is the learning constant multiplied to a compensating factor β whenever sum of square error increases and divided whenever the sum of square error reduces.

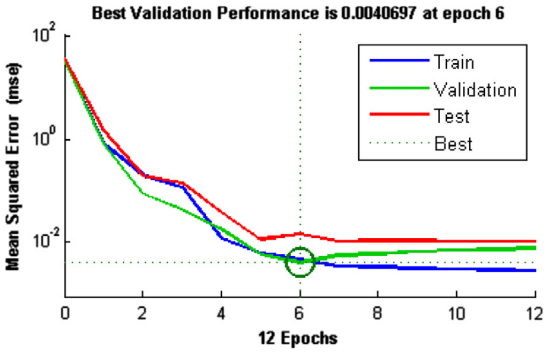

Fig. 7. Convergence of the network for training, testing and validation points (number of epochs vs. mean square error).

Performance Evaluation: The network has converged very fast, just within 12 epochs. The sum of square error is minimized and the training stopped when it reached to 0.0040 of mean square error. Fig. 7 shows the convergence rate of the network. A cross validation technique which divides the total population into their respective proportionate of training, testing and validation has been put to control over learning. Usually the sample points are distributed randomly over these three sets. Each set has individual role in the design of neural net like training taking care of the weight updating, validation is responsible for the controlled learning (to avoid the over fitting) and testing module ensures the reliability of prediction/approximation. The quality of the approximation can be evaluated by:

Mean Square Error (MSE): Mean Square Error is a parameter to evaluate the error in each epoch. In total 75% of the total population has been feed forwarded in each epoch. A definite error is involved against each sample point. The MSE of these error shows the global error of that particular epoch. If the MSE approaches to zero, we believe that the error is going to minimize and soon it will converge. The MSE can be calculated using the following expression

$$MSE = \sqrt[2]{e1^2 + e2^2 + e3^2 + e4^2 + \cdots + en^2}.$$

Here $e1^2 + e2^2 + e3^2 + e4^2 + \cdots + en^2$ are the error with respect to each sample point and each can be calculated by ($error = Output_{desired} - Output_{actual}$). In our case the desire out put is the NAO end effector output while the actual output is what we are getting from the network at the output layer.

Pearson's Correlation Coefficient (R): The second evaluation criteria is the correlation between the points of NAO end effector and table points. If the correlation between these two are approaching to one, this means their differences are Turing to zero. It means the errors is going to minimize and network is going to converge. The Pearson correlation can be calculated based on the variance of Table Points (σ_{table}), variance of respective NAO's end effector points (σ_{nao}) and the co-variance between the table points and NAO end effector points ($\Psi_{table,nao}$).

$\sigma_{table} = \sum (x_i - \bar{x})^2$ here x is the point belongs to *table* and $\bar{x}$ is the mean of all $x \in table$, similarly

$\sigma_{nao} = \sum (y_i - \bar{y})^2$ here y is the point belonging to *nao* and $\bar{y}$ is the mean of all $y \in nao$. For both the populations $i = 1$ to n, after expanding both the equations we will get

$\sigma_{table} = \sum x^2 - n\bar{x}^2$ and $\sigma_{nao} = \sum y^2 - n\bar{y}^2$, now th co-variance between these two can be calculated as:

$$\Psi_{table,nao} = \sum (x_i - \bar{x})(y_i - \bar{y})$$

$$\Psi_{table,nao} = \sum xy - n\bar{x}\bar{y}$$

$$R = \Psi_{table,nao} / \sigma_{table} \times \sigma_{nao}$$

$$R = \sum xy - n\bar{x}\bar{y} \Big/ \sum x^2 - n\bar{x}^2 \times \sum y^2 - n\bar{y}^2.$$

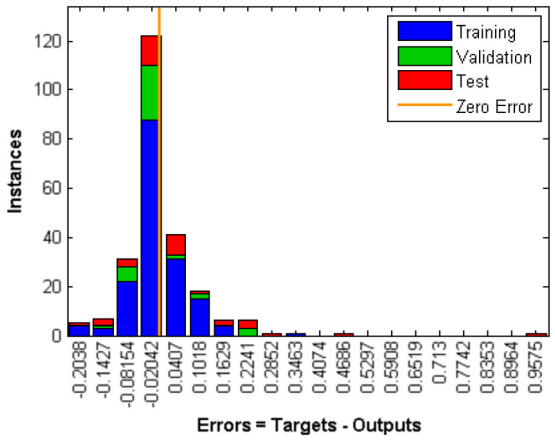

Fig. 8. Error histogram (window size 0.0612) for the training, testing and validation points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

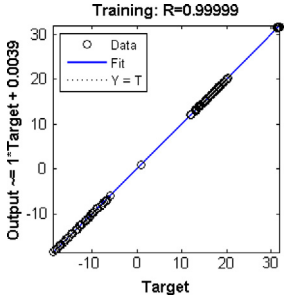

Fig. 9a. Regression plot between target and actual output for the training points.

Fig. 8 shows the error involved in different stages of training, testing and validation. Each one is symbolized with the different color i.e. blue is representing the training, green for validation and red for testing. The error involved in approximation is grouped on the interval of 0.0612. There are 20 bins used to represent the error histogram where most of the error lies between the −0.2038 to 0.2241. Apart from showing the distribution of error, the histogram also helps to identify the outliers, for example the testing point with an error of 0.9575 and 0.4686 and training point with an error of 0.3463 can be considered as outliers. It is always good to remove the outlier from the training set otherwise it can mislead the network, and enforce it to fit the outlier by extrapolating the given input points.

The regression plots of each of the phases (train, test and validation) are also depicted in Figs. 9a–9d which confirms that the error is minimized. All the input points are falling on the regression line, which proves that the variance between the points are less and it has been minimized. There are 60 points corresponding to each axis, further these 60 points are modulated between the training, testing and validation. We have considered 75% of the population as training while the 15% and 10% population is used for validation and testing. In total 45 number of points is being used for training, 9 points used for validation and 6 points are used for testing the network. The Pearson correlation coefficient "R" is almost 1 in all the cases which ensures the efficiency of approximation.

The regression plot shown in Figs. 9a–9d reflects the error of projection involved in training, testing and validation phases. A generalized regression line has been developed which can fit every test point corresponding to all x, y and z axes. In total we have 27 validation points, 18 testing points and 135 training points corresponding to x, y and z axis.

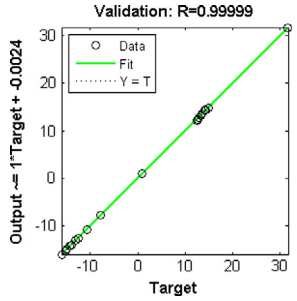

Fig. 9b. Regression plot between target and actual output for the validation points.

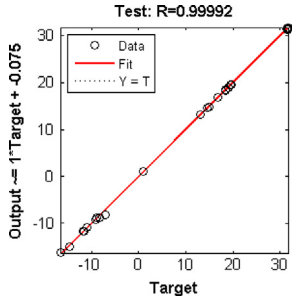

Fig. 9c. Regression plot between target and actual output for the testing points.

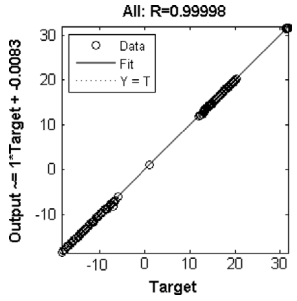

Fig. 9d. Regression plot between target and actual output for the all training, validation and testing points.

6. Comparative analysis of proposed solution

The results are summarized in the form of projection error involved after projecting each test point over the NAO's body coordinate frame. We have calculated absolute error which shows that how the projected points differ from the actual points along each coordinate axis. The statistical analysis of standard deviation and mean square error depicted in Table 3 shows the efficiency of the calibration matrix. All these three measurements shown in Table 3 are calculated on the basis of all 60 samples.

The mean and mean square error shows the errors involved in projection while the standard deviation and variance shows how the error are distributed along the mean. For fundamental matrix approach the Y axis has the mean and mean square error less than other two axes, therefore it has the good projection direction. The beauty of four point calibration is its less number of calibration points. Only four points are sufficient enough to establish the relation between these two coordinate frames. However the validation of the transformation matrix is being evaluated with the help of four measurement units presented in Table 3. As the mean, and variance along the z axis is very less hence the projection error involved along z axis is very less for each sample point. After applying the regression analysis with the help of Artificial Neural Network, we have achieved very good approximation results. The gap between the projected table points over the NAO's body

Table 3
Statistical analysis of absolute error involved along each axis for all three proposed techniques.

Measurements	Fundamental matrix			4 point calibration			Regression analysis		
	X axis	*Y* axis	*Z* axis	*X* axis	*Y* axis	*Z* axis	*X* axis	*Y* axis	*Z* axis
Mean	1.4508	1.0057	1.3379	0.4928	0.3881	0.0895	0.0079	−0.0089	−0.0025
Standard deviation	0.6326	0.2365	0.2977	0.4421	0.3269	0.0617	0.0922	0.1175	0.0636
Variance	0.4002	0.0559	0.0886	0.1955	0.1068	0.0038	0.0085	0.0138	0.0040
Mean square error	2.4984	1.0664	1.8771	0.4352	0.2558	0.0118	0.0084	0.0137	0.0040

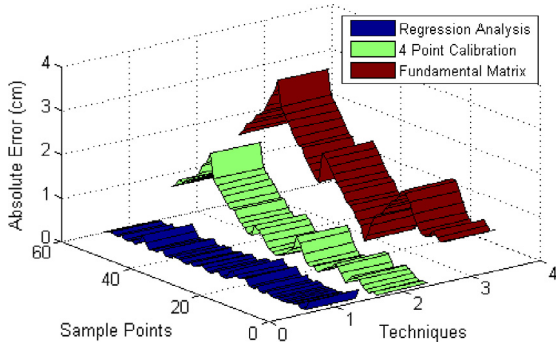

Fig. 10a. Absolute error along *X* axis for each sample point with respect to each transformation technique.

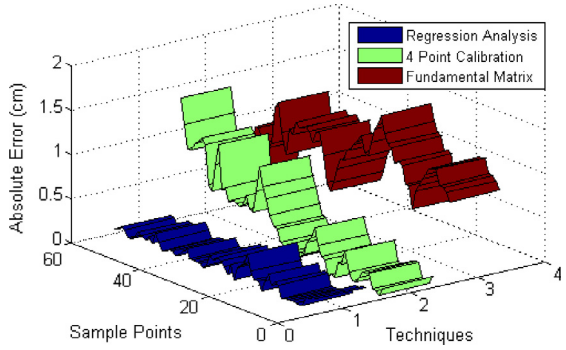

Fig. 10b. Absolute error along *Y* axis for each sample point with respect to each transformation technique.

coordinate frame and the NAO's actual points are very less. The mean square error which was also the stopping criteria for ANN has the minimum values, which ensures that the variations between these two are very less.

We have considered three parameters (a) mean square error involved in projection, (b) time complexity analysis, and (c) feasibility of the solution to test the performance of each technique. The mean square error reported for each technique along their *X*, *Y* and *Z* projection directions are depicted in Fig. 11. If the mean square error is less and closer to zero then the projection along that direction is considered to be good. It is clearly visible from the comparative analysis of all the techniques that ANN based regression analysis has minimum mean square error. Hence it could be considered as the best method to calibrate. The absolute error along *X* direction is depicted in Fig. 10a, which shows for each sample point how much projection error involved for regression analysis, 4 point calibration and fundamental matrix. Almost for every point the absolute error of regression analysis is less tending to zero; in comparison to other two approaches. The same has been reported along *Y* and *Z* direction in Figs. 10b and 10c. The performance of all three techniques can be ranked on the basis of mean square error and their absolute error for every point. These two parameters indicate that regression analysis is the best way to estimate the calibration matrix followed by the 4 point calibration and the fundamental matrix approaches. The other two performance measurement techniques are discussed below.

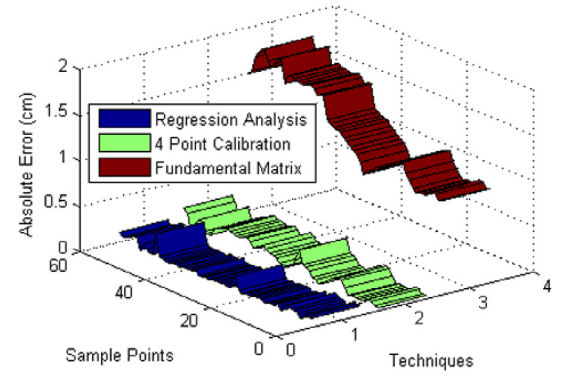

Fig. 10c. Absolute error along *Z* axis for each sample point with respect to each transformation technique.

6.1. Time complexity analysis

Time complexity of each technique is evaluated based on the time involved for establishing the homogeneous transformation matrix. The time complexity of the algorithm is estimated based on its each individual steps. These algorithms are less iterative in nature while more mathematical. The time taken to execute the mathematical equation is usually less than the iterative approaches. Therefore the time complexity of these algorithms is not very high. Number of sample data points play major role in estimating the algorithm complexity. If the number of sample points is less, it will require less time to estimate contents of calibration matrix. Therefore the time complexity of the 4 point calibration is less than the other two techniques. The time

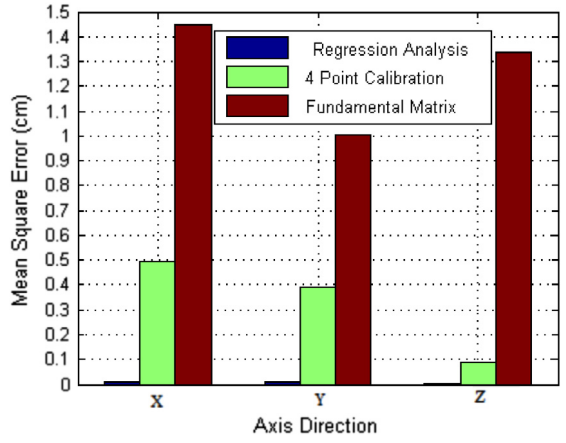

Fig. 11. Mean square error of each transformation matrix along each projection direction.

complexity is estimated only on the basis of steps required for establishing the calibration matrix. It does not consider the time involved in projecting the input points to the output.

Table 4
Time complexity of fundamental matrix.

Steps	Complexity
1	$T_1 + T_2 + T_3$ // *meancalculation*
2	$T_4 + T_5 + T_6$ // *meanalignment*
3	$T_7 + T_8$ // *magnitudealongtableandnao*
4	$T_9 + T_{10}$ // *meanofmagnitudes*
5	$T_{11} + T_{12}$ // *scallingforeach*
6	$T_{13} + T_{14}$ // *ScaleTransformation*
7	$n + n$ // *pointtransformation*
8	n // *pointmultiplication*
9	n^3 // *svd of* $n \times 16$*matrix*
10	n // *findingminimum*
11	T_{15} // *reshaping*
12	4^3 // *svd of* 4×4*matrix*
13	T_{16} // 4×4*matrixmultiplication*
14	T_{17} // *assignment*

6.1.1. Fundamental matrix approach

The total complexity of the algorithm is evaluated after summing up all the individual steps listed in Table 4. The algorithm is composed in 14 steps. The complexity of most of the steps are constant while for step 8, 9 and 10 it is variable and dependent on the number of points. As the number of samples points are known to us, the total time complexity is also assuming to be somewhat constant.

Total Running Time: $T_1+T_2+T_3+T_4+T_5+T_6+T_7+T_8+T_9+T_{10}+T_{11}+T_{12}+T_{13}+T_{14}+n+n+n+n^p+n+T_{15}+n^p+T_{16}+T_{17}$.

$T_1, T_2, T_3, \ldots, T_{17}$ are having the unit constant time $O(1)$. So we can eliminate them as they have little effect on the overall complexity. If we leave these unitary constants we will left with: $4n + 4^3 + n^3$. There are two terms involved in the complexity one is linear and the other one is polynomial of degree 3. The degree of the polynomial depends on the number of unknown variables. The first term is linear hence its complexity will be $O(n)$, while the complexity of polynomial will be n^3. Therefore the complexity of the fundamental matrix approach is n^3.

6.1.2. Four point calibration

There are two steps involved in the four point calibration. First step is to find the inverse of a 4×4 square matrix and second step consists of a simple matrix multiplication. Therefore the complexity of the four point calibration would be: $n^3 + O(1)$. n^3 is the complexity of the $n \times n$ dimension matrix interspersion and $O(1)$ is the complexity of matrix multiplication. Our matrix is having the dimension 4, so the above complexity will reduce to 4^3. This is also a constant term hence the total time complexity of the four point calibration will be constant.

6.1.3. Artificial neural network based regression analysis

A multilayer neural network consists of two major building blocks (1) feed forward and (b) back propagation. The feed forward module feeds inputs from input layer to the output layer and the back propagation modules propagate the errors from the output layer to the input layer. Hence the complexity for both the modules are evaluated separately and combined at the end.

(a) **Complexity of Feed Forward**: There are 4 inputs, 10 hidden neurons and 4 output neurons in the output layer. The neurons present in the hidden layer as well as the output layer are the replicas of each other. The complexity of the feed forward network would be:

for each neuron at hidden layer(y)

$$= sigmoid\left[\sum_{i=1}^{n} x_i \cdot w_{ij} - \theta_j\right]$$

for each neuron at the output layer (output)

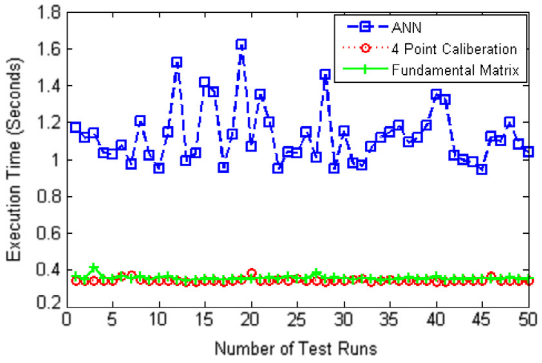

Fig. 12a. Execution time of different test runs.

$$= a \times \left[\sum_{j=1}^{k} y_{jk} \cdot w_{jk} - \theta_k\right] + b.$$

In the hidden layer there are 10 neurons, so $j \in [1, 10]$, similarly i is the number of inputs which is a four dimensional vector, therefore $i \in [1, 4]$, θ is representation of bias at the hidden layer. So the complexity of the hidden layer will be $10 \times O(n^2)$, where $n = 4$. The complexity of the output layer would be measured as: $4 \times O(k^2)$, where k is the number of output neurons, a and b are the coefficient of the line (predictor). So the total complexity of the feed forward network is: $10 \times O(n^2) + 4 \times O(k^2)$, which will be a constant as n and k are fixed.

(b) **Complexity of Back Propagation**: We have used Levenberg–Marquardt [44] as a back propagation algorithm which helps in weight updating and training the network. Levenberg–Marquardt is a heuristic based optimization method widely known as L–M optimization method. It is a mixture of Gauss–Newton and gradient descent algorithm. The complexity of the Levenberg–Marquardt algorithm based on the numbers of weight assigned in the network. If the numbers of weight are less it will converge faster otherwise it can take longer time. The main problem in convergence lies in finding the inverse of the Jacobian matrix of the error function, inverse of the Hessian matrix which scale up to N^3, where N is the number of weights. Hence the complexity of the back propagation will be: $O(N^3)$.

The total time complexity of Artificial Neural Network will be: $O\left(n^2\right)+4\times O(k^2)+O(N^3)$. As $O(N^3)$ supersedes the behavior of rest two elements. The time complexity of ANN will be treated as $O(N^3)$.

After analyzing the theoretical computation complexity of all the above techniques we can conclude that four point calibrations will take less time in comparison to ANN and fundamental matrix approach. In order to confirm our theoretical analysis we have conducted a practical test. In our pen test we have executed each algorithm and recorded the CPU time for each algorithm. Total 50 test cases have been generated for each algorithm. A plot between the number of test cases and time taken by each algorithm is presented in Fig. 12a. The average time is also being calculated based on the given 50 test cases shown in Fig. 12b. The average running time of four point calibration is 0.3441 s which is very less in comparison to fundamental matrix of 0.3559 s and ANN of 1.1256 s. The test runs are executed on a 32 bit (Windows 7) machine having Intel Core I-5 processor with 2.50 GHZ CPU speed. Machine has 8 GB of Internal Memory (RAM).

6.2. Feasibility of the solution

The feasibility of the solution is guaranteed by checking the belongingness of projected end effector point with respect to

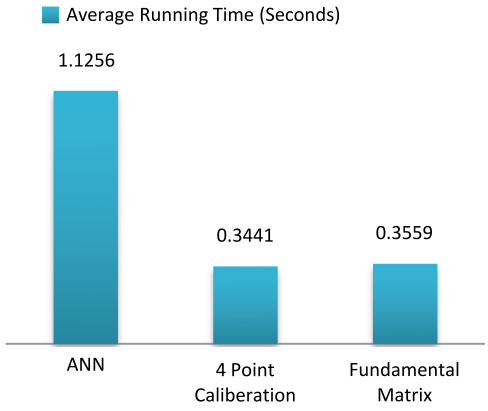

Fig. 12b. Average running time of each algorithm.

reachable workspace generated by NAO's right hand. The reachable workspace is generated by using the NAO's right hand forward kinematics. Forward kinematics of the right would provide the reachable points for the given input joint values. The movement (angle of rotation) of each joint is predefined and fixed [5]. For example the right shoulder pitch has degree of freedom −119.5 to +119.5°; similarly right shoulder roll cam move from −76 to 18°. The equation used for generating the forward kinematics is given below:

$$T^{torso}_{endEffector} = T^{torso}_0 \times T^0_1 \times T^1_2 \times T^2_3 \times T^3_4 \times T^4_5 \times T^5_6.$$

The transformation matrices $T^0_1, T^1_2, T^2_3, T^3_4, T^4_5, T^5_6$ are same as described in Section 2 except this T^{torso}_0 transformation. T^{torso}_0 is a transformation used to define end effector position with respect to torso. The transformation has only translation parameters which translate the point define with respect to right hand to torso position.

$$T^{torso}_0 = \begin{bmatrix} \cos(90) & -\sin(90) & 0 & 0 \\ \sin(90) & \cos(90) & 0 & -d_1 \\ 0 & 0 & 1 & -d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Here d_1 and d_2 are the translation involved along the Y and Z axis from the torso the center point of right hand. This translation along Y (d_1) is equal to the sum of SholderOffsetY + ElbowOffsetY, but in negative direction, similarly the translation along the Z axis (d_2) is equal to the Sholderoffset Z in the negative direction. Putting values of cos(90) and sin(90) into the equation; it will be reduced to;

$$T^{torso}_0 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -113 \\ 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The points generated after applying the forward kinematics is depicted in Fig. 13a. The plot is covering all three axis and the points fall on these three axis. The reachability over X axis varies from −181.8947 mm to +130.5467 mm. Similarly the distance it can travel along Y and Z axes are varies from −294.8649 to −19.8064 and −81.8947 to 281.8947. It is observed from the previous literature that in most of the cases of drawing Z direction is almost constant. Therefore the graph showing the points of X and Y axis is described in Fig. 13b.

Fig. 13b shows points spread over the XY plane. The range of each axis has been calculated to demonstrate that the reachability area is satisfying the primary criteria i.e. $fk^x \leq \pm(L1 + L2 + L3)$, $fk^y \leq \pm(ShoulderOffsetY + L1 + L2 + L3)$, $fk^z \leq \pm(ShoulderOffsetZ + L1 + L2 + L3)$. The value of ShoulderOffsetY and ShoulderOffsetZ are 98 mm and 100 mm respectively. The link L1, L2 and L3 are 105 mm, 55.95 and 57.75 mm long. fk^x, fk^y, and fk^y could be any reachable point which will satisfy the above constraints. The points projected after applying each transformation

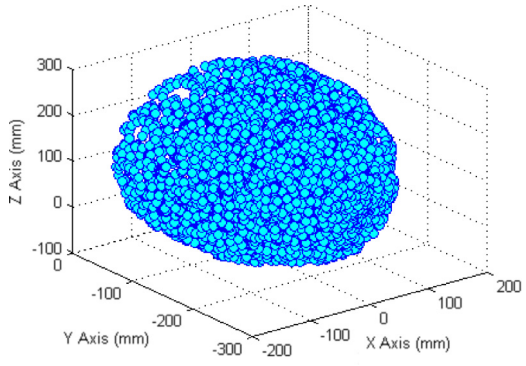

Fig. 13a. The working envelop of right hand with respect to its torso position.

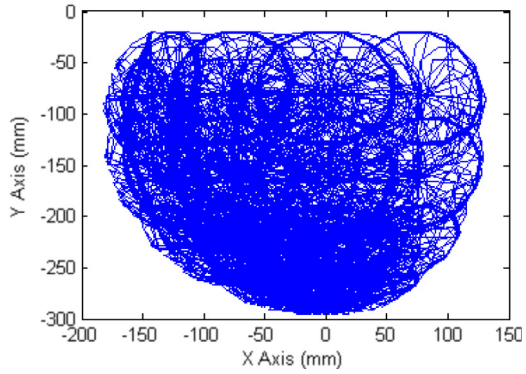

Fig. 13b. Region covered along X and Y axis by the right hand with respect to its torso position.

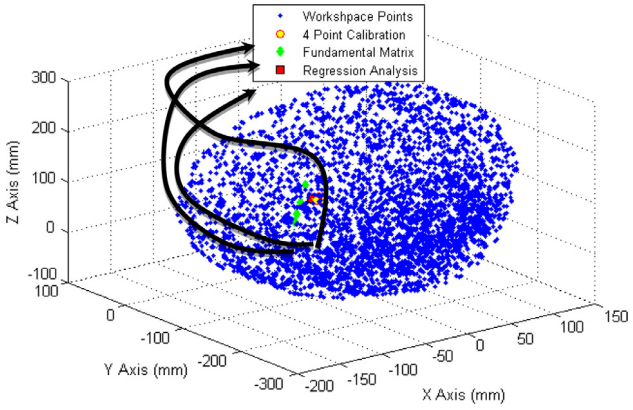

Fig. 14. Projection of test points over the reachable workspace generated by each technique.

are depicted in Fig. 14. All projected points are falling under the reachable workspace which describe that all the techniques are satisfying these criteria. The mean square error in regression analysis is very less along each direction which directly means the little deviation from the actual end effector position. Thus it can be concluded that regression analysis could be considered as best projection matrix.

7. Conclusion and future research direction

In this research, different calibration techniques have been used to establish a correlation between the image plane of NAO robot and its end effector position. These techniques help to measure the orientation of the NAO end effector and to solve the inverse kinematic problem. Three different approaches (a) Fundamental matrix (b) 4 point calibration and (c) ANN based regression analysis have been introduced to find the coefficient

of calibration matrix. The experiment has been performed with 60 data points collected from the NAO end effector and from its image plane. The projection validity has been ensured using four statistical metrics (a) error mean, (b) standard deviation, (c) variance and (d) mean square error. The minimum mean square error and variance along each projection direction improves the strength of these calibration techniques. The regression analysis technique produces 0.0084 cm, 0.0137 cm, and 0.0040 cm of mean square error (MSE)along X, Y and Z direction which is less than the MSE value (2.4984 cm, 1.0664 cm, 1.8771 cm) of fundamental matrix approach and MSE value (0.4352 cm, 0.2558 cm, 0.0118 cm) of 4 point calibration approach. Two other metrics such as time complexity and transformation reliability are also introduced to evaluate the performance of these techniques. We have obtained a computational complexity of $O(1)$ for 4 point calibration which is minimum with respect to computational complexity of fundamental matrix ($O(n^3)$) and regression analysis techniques ($O(n^3)$). The average computational time for all the three techniques have been verified on 50 iterations where regression analysis takes 1.1256 s, 4 point and fundamental matrix takes 0.3441 s and 0.3559 s to establish the coefficient of the calibration matrix. In second evaluation criteria, the transformations of all three techniques are verified as the transformed values of all respective techniques are residing under the working envelope of NAO right hand. On the basis of all three performance metrics, it is concluded that regression analysis is better way to estimate calibration matrix with less mean square error while 4 point calibration takes less time than others.

In future research direction, an artistic excellency of NAO robot may be introduced in the field of criminal investigations to help Police sketch artist. In India like many other countries, the crimes especially against women are rising. One of the many reasons perhaps is the low conviction rate. The initial stage of criminal investigation starts with the exploration of evidences and eyewitnesses. An eyewitness can act as a guide to trace out the suspect. Her/his perception about the suspect can be useful to identify the criminal. Based on the descriptions of the eyewitnesses normally a manual sketch is prepared and released in the Newspapers. In such cases these Humanoids with the additional ability of sketch drawing could be very useful.

Acknowledgments

Author would like to thank Dr. Rahul Kala Assistant Professor in Information Technology at Indian Institute of Information Technology, Allahabad, India for his critical review and feedback on this paper. Author would also extend his thanks to Gopal Lal Chaudhary, M.Tech. Final Year student for assisting over this project.

Appendix A

Table 5
The description of the symbols used in Algorithm 1.

Symbol	Meaning
$(P')^{\eta}$	Point corresponds to NAO end effector
$(P)^{\beta}$	Point on table coordinate
N, R^4	Number of points here ($N = 60$), Real space (4 dimensional)
μ_x', μ_y', μ_z'	Mean corresponding to x, y and z axis with respect to NAO coordinate frame
μ_x, $\mu_{y,}\mu_z$	Mean corresponding to x, y and z axis with respect to table coordinate frame
$(\overline{\mu_1})^{\eta}$, $(\overline{\mu_2})^{\beta}$	Mean corresponds to x, y and z axis for both NAO and table coordinate frame
T^{η}, T^{β}	Scale Translation matrix for both NAO and table coordinate frame

Appendix B. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.robot.2016.01.009.

References

[1] Rüdiger Dillmann, Regine Becher, Peter Steinhaus, ARMAR II—A learning and cooperative multimodal humanoid robot system, Int. J. Humanoid Rob. 1 (01) (2004) 143–155.
[2] Kenji Kaneko, et al., Humanoid robot HRP-3, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, IEEE, 2008.
[3] Giorgio Metta, et al., The iCub humanoid robot: an open platform for research in embodied cognition, in: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, ACM, 2008.
[4] Yu Ogura, et al., Development of a new humanoid robot WABIAN-2, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, IEEE, 2006.
[5] Aldebaran Robotics, NAO Datasheet H25—Corporate—Aldebaran Robotics| Discover NAO, 2012.
[6] ASIMO instruction manual by Honda Motor Co., Ltd.
[7] HOAP-2 instruction manual by Fujitsu Automation Co., Ltd.
[8] David Gouaillier, et al., Mechatronic design of NAO humanoid, in: IEEE International Conference on Robotics and Automation, 2009. ICRA'09, IEEE, 2009.
[9] S. Brown, Meet Pepper, The Emotion Reading Robot. *TECHNOLOGY*, 2014.
[10] Shotaro Kobayashi, et al., Intelligent humanoid robot with japanese Wikipedia ontology and robot action ontology, in: 2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI), IEEE, 2011.
[11] Yoshiaki Sakagami, et al., The intelligent ASIMO: System overview and integration, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, Vol. 3, IEEE, 2002.
[12] Catherina Burghart, et al., A cognitive architecture for a humanoid robot: A first approach, in: 2005 5th IEEE-RAS International Conference on Humanoid Robots, IEEE, 2005.
[13] Stefan Schaal, Is imitation learning the route to humanoid robots? Trends Cogn. Sci. 3 (6) (1999) 233–242.
[14] Kei Okada, et al., Vision based behavior verification system of humanoid robot for daily environment tasks, in: 2006 6th IEEE-RAS International Conference on Humanoid Robots, IEEE, 2006.
[15] Kazuhiko Kawamura, et al. Humanoids: Future robots for home and factory, in: International Symposium on Humanoid Robots, 1996.
[16] Hiroaki Kitano, et al., Robocup: The robot world cup initiative, in: Proceedings of the First International Conference on Autonomous Agents, ACM, 1997.
[17] Minoo Alemi, Ali Meghdari, Maryam Ghazisaedy, Employing humanoid robots for teaching English language in Iranian Junior high-schools, Int. J. Humanoid Rob. 11 (03) (2014).
[18] Aude Billard, Robota: Clever toy and educational tool, Robot. Auton. Syst. 42 (3) (2003) 259–269.
[19] Syamimi Shamsuddin, et al., Initial response of autistic children in human–robot interaction therapy with humanoid robot NAO, in: 2012 IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA), IEEE, 2012.
[20] Patrick Lin, George Bekey, Keith Abney, Autonomous military robotics: Risk, ethics, and design. California polytechnic State Univ. San Luis Obispo, 2008.
[21] Brian M. Yamauchi, PackBot: a versatile platform for military robotics *Defense and Security*. International Society for Optics and Photonics, 2004.
[22] Gary E. Marchant, et al., International governance of autonomous military robots, Columia Sci. Technol. Law Rev. 12 (2011) 272–276.
[23] Masahiro Fujita, Yoshihiro Kuroki, Tatsuzo Ishida, A small humanoid robot sdr-4x for entertainment applications, in: 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings, Vol. 2, IEEE, 2003.
[24] Harold Cohen, How to Draw Three People in a Botanical Garden, Vol. 89, AAAI, 1988.
[25] Paul Haeberli, Paint by numbers: Abstract image representations, ACM SIGGRAPH Comput. Graph. 24 (4) (1990).
[26] Aaron Hertzmann, Painterly rendering with curved brush strokes of multiple sizes, in: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1998.
[27] Matthew R. Stein, The pumapaint project, Auton. Robots 15 (3) (2003) 255–265.
[28] Matthew R. Stein, Christopher P. Madden, The pumapaint project: Long term usage trends and the move to three dimensions, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005, IEEE, 2005.
[29] J.P. Collomosse, P.M. Hall, Painterly rendering using image salience, in: Eurographics UK Conference, 2002. Proceedings. The 20th, IEEE, 2002.

[30] Ka Wai Kwok, Yeung Yam, Ka Wah Lo, Ga-based homography transformation for vision rectification in robot drawing system, in: 44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05, IEEE, 2005.
[31] Yan Lu, Josh H.M. Lam, Yeung Yam, Preliminary study on vision-based pen-and-ink drawing by a robotic manipulator, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2009. AIM 2009, IEEE, 2009.
[32] Oliver Deussen, et al., Feedback-guided stroke placement for a painting machine, in: Proceedings of the Eighth Annual Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging, Eurographics Association, 2012.
[33] Shunsuke Kudoh, et al., Painting robot with multi-fingered hands and stereo vision, Robot. Auton. Syst. 57 (3) (2009) 279–288.
[34] Patrick Tresset, F. Fol Leymarie, Generative portrait sketching, in: Proceedings of VSMM, 2005.
[35] Patrick Tresset, Frederic Fol Leymarie, Portrait drawing by Paul the robot, Comput. Graph. 37 (5) (2013) 348–363.
[36] Fiammetta Ghedini, Massimo Bergamasco, Robotic creatures: Anthropomorphism and interaction in contemporary art *RO-MAN*, 2010.
[37] Leonel Moura, A new kind of art: The robotic action painter, in: X Generative Art Conferece, Politecnico di Milano University, 2007.
[38] Sylvain Calinon, Julien Epiney, Aude Billard, A humanoid robot drawing human portraits, in: 2005 5th IEEE-RAS International Conference on Humanoid Robots, IEEE, 2005.
[39] A. Srikaew, et al. Humanoid drawing robot, in: IASTED International Conference on Robotics and Manufacturing, 1998.
[40] John J. Craig, Introduction to Robotics: Mechanics and Control, Vol. 3, Pearson Prentice Hall, Upper Saddle River, 2005.
[41] N. Kofinas, Forward and inverse kinematics for the NAO humanoid robot, Technical University of Crete, 2012.
[42] John Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1986) 679–698.
[43] Richard I. Hartley, In defense of the eight-point algorithm, IEEE Trans. Pattern Anal. Mach. Intell. 19 (6) (1997) 580–593.
[44] Martin T. Hagan, Mohammad B. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Netw. 5 (6) (1994) 989–993.

Avinash Kumar Singh obtained his B.Sc. in I.T. and M.Sc. in Information Technology degree from Kumaun University, Nanital (Uttarakhand) in the year of 2007 and 2009 respectively. He received his M.Tech in Information Security from Kalinga Institute of Industrial Technology, Bhubaneswar (Orissa) in the year of 2011. He has secured first rank in his M.Sc. at university level. Currently, he is working as a full time research scholar at Indian Institute of Information Technology Allahabad, India. His research interests include biometrics, computer vision, pattern recognition, human robot interaction, web and network security.

G.C. Nandi graduated from the Indian Institute of Engineering, Science & Technology (formerly Bengal Engineering College, Sibpore, Calcutta University) in 1984 and from Jadavpur University, Calcutta in 1986. He obtained his Ph.D. degree from the Russian Academy of Sciences, Moscow in 1992. He was awarded a National Scholarship by the Ministry of Human Resource Development (MHRD), Govt of India in 1977 and a Doctoral Fellowship by the External Scholarship Division, MHRD, Govt. of India in 1988. During 1997, he was a visiting research scientist at the Chinese University of Hong Kong and he also visited the Faculty at the Institute for Software Research, School of Computer Sciences, Carnegie Mellon University, USA (2010–2011). Currently, he is the senior Professor and Dean of Academic Affairs at the Indian Institute of Information Technology, Allahabad. From January to July 2014, he served as the Director-in-Charge of the Indian Institute of Information Technology, Allahabad. Professor Nandi is a senior member of the ACM, senior member of IEEE, Chairman, ACM-IIIT-Allahabad Professional Chapter, (2009–2010), Chartered Member of Institute of Engineers (India), and a Member of the DST (Department of Science and Technology, Govt. of India) Program Advisory Committee Member of Robotics, Mechanical and Manufacturing Engineering. He has published more than 100 papers in various refereed journals and international conferences. His research interest includes robotics, especially biped locomotion control and humanoid push recovery, artificial intelligence, soft computing, and computer controlled systems.