Open access • Journal Article • DOI:10.1007/S10055-016-0303-Y

A hybrid optical---mechanical calibration procedure for the Scalable-SPIDAR haptic device — Source link

M'hamed Frad, Hichem Maaref, Samir Otmane, Abdellatif Mtibaa

Institutions: University of Évry Val d'Essonne, University of Monastir

Published on: 01 Sep 2017 - Virtual Reality (Springer London)

Topics: Initialization and Haptic technology

A hybrid optical-mechanical calibration procedure for the Scalable-SPIDAR haptic device.

[1,2] M'hamed Frad, [1]Hichem Maaref, [1]Samir Otmane, [2]Abdellatif Mtibaa
[1] IBISC Laboratory, University of Evry Val-d'Essonne, France
[2] EµE Laboratory, University of Monastir, Tunisia
E-mail : mhamed.frad@gmail.com
Tel: (+216) 53 84 07 04
Fax: (+216) 73 68 33 99

Abstract: In this research, a simple, yet, efficient calibration procedure is presented in order to improve the accuracy of the Scalable-SPIDAR haptic device. The two-stage procedure aims to reduce discrepancies between measured and actual values. First, we propose a new semi-automatic procedure for the initialization of the haptic device. To perform this initialization with a high level of accuracy, an infrared optical tracking device was used. Furthermore, audio and haptic cues were used to guide the user during the initialization process. Second, we developed two calibration methods based on regression techniques that effectively compensate for the errors in tracked position. Both Neural Network (NN) and Support Vector Regression (SVR) methods were applied to calibrate the position errors present in the haptic device readings. A comparison between these two regression methods was carried out to show the underlying algorithm and to indicate the inherent advantages and limitations for each method. Initial evaluation of the proposed procedure indicated that it is possible to improve accuracy by reducing the Scalable-SPIDAR's average absolute position error to about 6 mm within a 1m x 1m x 1m workspace.

Keywords: Scalable-SPIDAR, Virtual reality, Tracking, Calibration, Neural Network, Support Vector Regression.

1. Introduction

Haptic feedback is a natural part of our daily experience when we manipulate objects in our surroundings. With advances in haptic technology, virtual reality (VR) systems with integrated haptic devices are becoming increasingly popular in assembly training (Xia et al. 2012), virtual machining (Fletcher et al. 2013) and virtual biology training (Faroque et al. 2015). Being able to touch, feel and manipulate virtual objects improves the realism of virtual environments and ,therefore, intensifies the user's feeling of immersion and presence (Srinivasan & Basdogan 1997; Ramsamy et al. 2006). Various studies have addressed the development of haptic devices in virtual reality, for instance Burdea (1996), Hayward et al. (2004) and Fuchs et al. (2011). Unfortunately, most of the current devices are either bulky, intrusive, expensive or only work within a limited space (Srinivasan 1995), (Ikits et al. 2000).

To alleviate the mentioned shortcomings, Buoguila et al. (2000) proposed a human-scale haptic device named Scalable-SPIDAR for **SP**ace **I**nterface **D**evice for **A**rtificial **R**eality. The device is derived from the original desktop SPIDAR device which was developed by Hirata and Sato (1992). According to the state-of-the-art haptic interface technology given by Kunzler and Runde (2005), Scalable-SPIDAR is transparent, unbulky and suitable for large-scale haptic immersion. However, it has certain limitations in interacting with virtual environment, especially manipulation difficulties. In fact, the Scalable-SPIDAR is a single-point interface and, thus, is not well suited to perform specific two-handed tasks such as the bimanual exploration of large virtual environments and grasping of virtual objects. The device uses tensioned string techniques to track and measure the motion of the user's hands as well as to display force feedback sensations. Nevertheless, due to various limitations, Scalable-SPIDAR is subject to tracking inaccuracies and, thus, cannot always provide faithful data rendering. The question, then, is whether any discrepancy caused by tracking inaccuracies matters to the user's experience or performance in virtual environments. Several studies have shown that large tracking inaccuracies lead to inconsistency between physical and virtual working environments which may contribute to detracting from the sense of presence, causing sickness or degrading the task performance (Meyer et al. 1992; Welch and Foxlin 2002; Jayaram and Repp 2002).

Therefore, it is important to improve the tracking accuracy of the Scalable-SPIDAR haptic device. To this end, accurate calibration is indispensable. In this work, we focus on the development of a comprehensive calibration procedure for improving the accuracy of the Scalable-SPIDAR haptic device. As a design goal, the procedure needs to be robust and flexible enough to be applied to other similar systems. In the next section, we provide an overview of related works regarding methods used to improve tracking accuracy in virtual and augmented reality environments. Thereafter, we introduce our visual-haptic VR setup, followed by a brief description of its main components. In section 4, we detail the developed calibration procedure. We present some of the key aspects concerning the characterization and calibration of the Scalable- SPIDAR haptic device. Finally, we conclude by evaluating the performance of the proposed calibration techniques.

2. Related work

Before launching in a further discussion, it is important to grasp what is meant by the term "accuracy". In fact, the accuracy of a measurement system refers to the degree of closeness of measurements of a quantity to its actual value (ISO 5725-1 1994). Similarly, tracking accuracy refers to the closeness of position tracker measurements to the actual position value. Unfortunately, tracking devices are often inaccurate and imperfect. Distortions are introduced into the tracking data so that the data reported by the tracker only loosely corresponds to the actual data. If this distortion is repeatable, it can be characterized and then the actual data can be inferred from the distorted data. This will be called calibrating the tracking device.

In the following section, we provide an overview of related activities regarding methods used to calibrate tracking devices in virtual and augmented reality environments. The majority of previous works has focused on calibration issues for electromagnetic trackers. It has been reported that these trackers have an inherent accuracy problem due to the dependence of their reports on the local electromagnetic field that can be easily distorted by conductive or ferrous materials. To overcome this issue, various calibration methods have been proposed. survey Kindratenko (2000) conducted a detailed survey of these methods . In each of the reviewed methods, measurement data was collected to determine the relationship between actual and distorted data. Several error correction methods were, then, applied so that distorted measurement data can be effectively compensated.

Ghazisaedy et al. (1995) used a local correction technique based on trilinear interpolation to correct errors in the position reported by the tracker. Calibration data was collected using an ultrasonic tracking system and a regular grid and a look up table of corrections were built. The technique compensated large errors in position readings but did not significantly improve on small errors. Similarly, Livingston and State (1997) applied trilinear interpolation to compensate for errors in the reported position and a sequence of spherical linear interpolations between quaternions representing the reported orientation to correct errors in orientation. A look up table was built by collecting a very large irregular distributed set of samples and re-sampling it into a rectilinear grid to form calibration cells. The authors determined that position accuracy could be improved by roughly 80% although they had difficulty with orientation compensation.

Bryson (1992) developed a weighted look up table technique with two different weight functions (one with linear interpolation and one with a Gaussian-weighted look up table). He measured errors in the tracked position on a rectilinear grid defined by a pegboard and found that the Gaussian-weighted look up table performed best. Kindratenko and Bennett (2000) extended this technique to calibrate both the position and orientation reports. The authors built a look up table by placing the tracker at regularly spaced locations and with a known constant orientation and then recording both the true position/orientation and the reported by the tracker. They conclude that weighted look up table technique could also significantly reduce the errors in the reported orientation. Ellis et al. (1999) proposed to correct errors in the reported position by using an adaptation of an algorithm developed by Kenwright and Lane (1996). The proposed technique used a local interpolation based on a simple form of shape function derived for tetrahedrons. Similarly, Briggs (1999) used a weighted look up table technique to correct position errors. This technique is similar to the one presented by Livingston and State (1997), however, it differs in the way the look up tables are re-sampled from data taken on an irregular grid. In addition, Jayaram an Repp (2002) used various inverse weighting interpolation methods to obtain correct values from the distorted electromagnetic tracker readings. The authors mentioned that the look up table based correction method dramatically improves tracking system accuracy but that the improvement varied based on the applied interpolation method.

Another way of calibrating errors for electromagnetic tracking systems is to use the high-order polynomial fit method. In fact, Bryson (1992) computed polynomials of the order 1 through 8 for each reported position from the calibration table via the least-square fit method. He found via the correlation analysis that the fourth-order polynomials were the most suited. Kindratenko (1999) and Ikits et al. (2001) extended this technique to compensate the errors in the reported orientation. Kindratenko (1999) used Euler angles, while, Ikits et al. (2001) used quaternionial representation to compute corrected orientations. Furthermore, Zachmann (1997) introduced a scattered data interpolation using Hardy's Multi-Quadric method with LU matrix decomposition to solve the interpolation equation. The motivation for using this method is that Hardy's Multi-Quadric polynomials tend not to oscillate as polynomial interpolations like Newton or Lagrange interpolation, since the degree of interpolating function does not depend of the number of samples. So far, the majority of reported calibration methods are based either on local interpolation or polynomial function methods. Local interpolation method requires a look up table of the closest points or the enclosing cell, which needs some additional data, structures if the look up is to be faithful and fast. Polynomial calibration works very well when distortion is smoothly spread through the tracking workspace but is not that effective when an abrupt, localized nonlinearity occurs in the mapped space. Hence, an alternative error correction approach with a multi-layer feed-forward neural network was proposed by Kindratenko and Sherman (2005). The calibration procedure in this method is both to obtain a calibration table and to train the network until a certain accuracy level is reached. Recently, Moreira et al. (2014) tested three calibration methods (linear interpolation, higher-order polynomial and Hardy's Multi-Quadric) in order to implement a suitable strategy for the evaluation of new dental impression techniques.

With regard to mechanical tracking system, only some attempts have been made to increase the accuracy of tracking. Most of these attempts examined the positioning accuracy and the calibration of the Phantom haptic device. A planar grid calibration was suggested by Reinig and Tracy (1997) to correct the haptic position and then extend this correction to the entire haptic workspace by extrapolation. Unfortunately, the results show that the position errors increase as the tip moves away from the grid. Ikits et al. (2003) proposed an enhancement of the grid approach by placing two perpendicular planar grids in the haptic volume. More recently, Knoerlein and Harders (2011) presented a comparison of tracker based to tracker-less haptic device calibration. They compare previously developed technique by Harders et al. (2009) using external optical tracking system to a new approach not relying on external metrology. Prior approaches focusing on the Phanthom haptic device calibration are based on model methods. However, important to these methods is the development of an accurate kinematic model. On the other hand, only few attempts have been made to explore the potential of model-less calibration method in haptic devices. Model-less method does not need any kinematic model. Boudoin et al. (2010) developed a neural network-based method to improve the position accuracy of the Scalable-SPIDAR haptic device. They used a two layered feed-forward neural network to correct non-linear position errors.

3. System overview

In order to immerse the user into the virtual environment, changes in the user's location have to be incorporated into the system. We used the infrared (IR) optical 6-DoF tracking system "Artrack1/Dtrack" from Advanced Realtime Tracking GmbH to get marker pose estimates. In fact, the "ARTrack1/Dtrack" system is based on two IR intelligent cameras that are able to recognize retroreflective markers and compute the marker pose in image coordinates (2D) with high precision. These data are transferred via Ethernet to the central Drack PC. The PC with the "Dtrack" software samples the 2D data coming from the two IR cameras and computes the 6-DoF pose targets, which are located in the measurement volume. Unfortunately, IR optical trackers are prone to occlusions (Welch and Foxlin 2002), which leads to significant loss of tracking.

To overcome the problem of tracking-loss, we combine the IR optical tracker with a haptic-based tracking approach. In fact, a string based haptic system called Scalable-SPIDAR has been integrated into our setup in order to provide stable tracking and to enable haptic interaction with virtual objects in the virtual environment. As well, haptic sensations are known to impart users with realistic feeling about physical interactions. Scalable-SPIDAR is composed of a cubic frame, which encloses a cave-like space, where the user can move around to perform large-scale movements. Within this space, various aspects of haptic feedback sensations associated mainly with weight, contact and inertia can be displayed to the user's hands by means of tensioned strings. The front side of the device holds a large screen where a computer generated virtual world is displayed. Providing such combination of haptic and visual feedback cues may strongly improve interaction with virtual objects in manner similar to those in real world. This combination is essential to let the user's eyes and hands work in concert to discover and manipulate objects populating the virtual environment (Bouguila et al. 2000). The device uses tensioned string techniques to track hands position as well as to provide force feedback sensation. The technique consists in applying appropriate tensions to the eight strings supporting each the end-effector worn by the user Kim et al. (2002). In order to control the tension and the length of each string, one extremity is attached to the end-effector and the other extremity wound around a pulley, which is driven by a DC motor. An encoder is fixed to the DC motor to detect the string length variation. The set of DC motor, pulley and encoder controlling each string are mounted in the corners of the frame. A general overview of the system's components is shown in Figure 1.

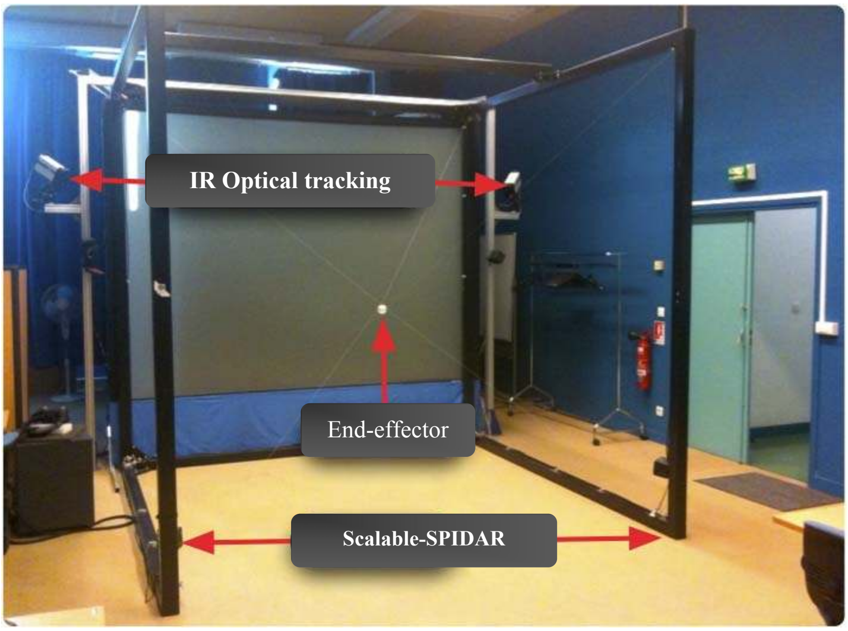

Fig. 1 Components of the visual-haptic VR setup

4. Calibration procedure

The driving force of our current research is the development of a simple, yet, efficient calibration procedure for improving the accuracy of the Scalable-SPIDAR haptic device. Therefore, we developed a hybrid tracking approach using the IR optical tracker available within our setup. Accuracy and robustness of the calibration procedure is greatly influenced by the quality of the optical tracker's calibration. To this end, two types of calibration have to be performed with the software DTrack before tracking is started: room calibration and body calibration. During room calibration, the axes and the origin of camera coordinate system are defined. For this procedure a "wand kit "is put into the field of the IR cameras. Following the process described in the ARTrack1/Dtrack manual (A.R.T. GmbH 2003), a valid room calibration can be performed. As shown in figure 2, the calibration angle defines the origin and the axes of the coordinate system:

- The center of the marker lying in the crossing point of the two sides of the angle defines the origin of the coordinate system.
- The long side of the calibration angle defines the +X direction
- The short side of the calibration angle defines the +Y direction.
- The Z-axis corresponds to the right-handed coordinate system.

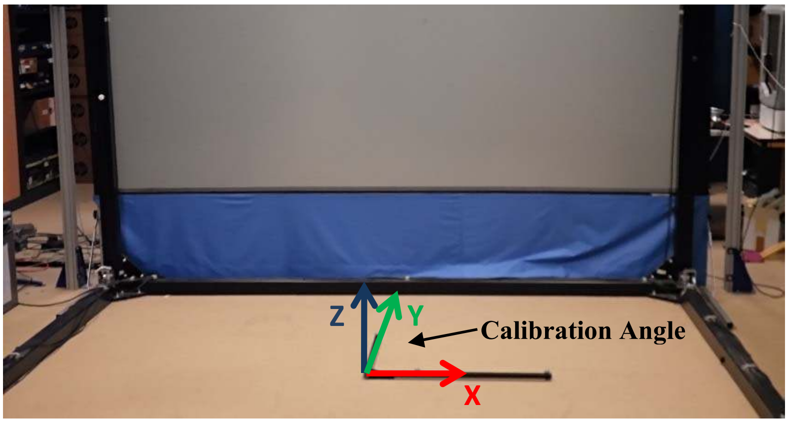

Fig. 2 Optical tracking system calibration (Room Calibration)

During body calibration, DTrack fixes a local coordinate system (body coordinate system) for each rigid body. Therefore, the position and orientation of a target are expressed by a transformation $(\vec{x}, R)$ that transforms a vector $\vec{x}$ from body coordinate system to room coordinate system

$$\vec{x}_{room} = R * \vec{x}_{body} + \vec{s} \tag{1}$$

Where the coordinate $\vec{x}$ gives the position of the origin of the body coordinate system, measured in room coordinate and R is the 3x3-rotation matrix describing the rotational part of the transformation. The next step focuses on the calibration of the haptic interface. Scalable-SPIDAR is a human-scale haptic device, based on tensioned string techniques, to display force feedback sensation in a large space, while allowing smooth movement and keeping the space transparent. However, it provides only a low tracking accuracy. This problem is mainly caused by two reasons. First, scalable SPIDAR has a built-in initialization procedure that is based on manually moving the end-effector to home position, then calling some particular API functions (Group 2005). Figure 3 shows the discrepancy between the position provided by the built-in initialization procedure and the true initialization position that has to be reached. As revealed by picture 3, the built-in procedure is unable to accomplish and repeat the initialization of the Scalable-SPIDAR in a precise manner. Therefore, a custom initialization procedure that allows the user to accurately place the end-effector to a fixed known location within the workspace of the haptic device is needed.

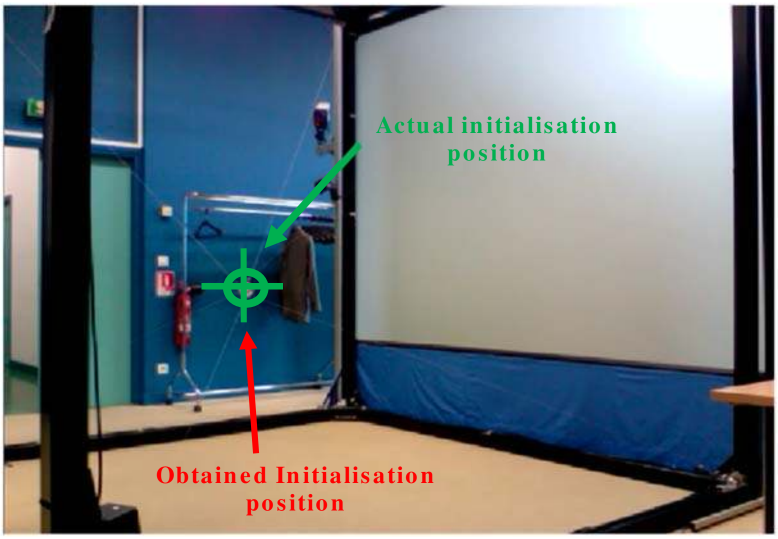
Actual initialisation position
Obtained Initialisation position

Fig. 3 Built-in initialization procedure of Scalable-SPIDAR

The position and force computations within the High Definition Haptic Controller HDHC firmware (figure 4) are based on the nominal kinematic model of the Scalable-SPIDAR haptic device, which is different from the actual model. In fact, inaccuracies between nominal and actual kinematic models arise mainly form shortcomings in the mechanical design of the haptic device. These shortcomings cause errors in the string lengths, which in turn, lead to positional errors of the end-effector and therefore affects the tracking accuracy of the Scalable-SPIDAR.

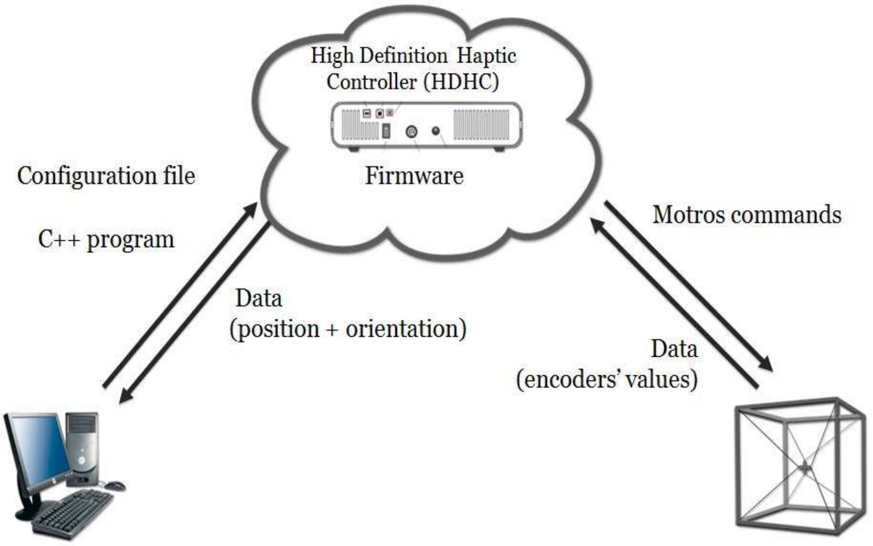

Fig. 4 Illustration of Scalable-SPIDAR **(**Motors and stings, HDHC and Computer)

The calibration process can now be carried out in two stages. We first determine a solution for the initialization problem. To this end, we propose a new procedure for initializing the haptic device, which will be outlined in the next section. Thereafter, we apply a calibration method to calibrate position readings from the haptic device within a defined working volume.

4.1 Multimodal initialization procedure

The built-in procedure conflicts with our need to a highly accurate initialization. The more accurate this initialization is, the higher tracking accuracy becomes. Therefore, we propose a new semi-automatic procedure for the haptic device initialization. To accomplish this initialization with a high level of accuracy, we use the IR optical tracking system. The underlying idea is to measure the actual initialization position (colored in green in figure 3) with respect to the room coordinate system. To this end, an IR marker is rigidly attached to the end of the end-effector. Here, the center of the cubic frame is taken to be the true initialization point. The vector $\vec{v}$, shown in figure 5, gives the distance of the IR marker attached to the end-effector relative to actual initialization point as well as the required direction to converge to it. In order to help the user to accomplish and repeat initialization in a precise manner, multimodal cues are needed. This can be derived by first converting spatial information into multimodal parameters. Let h be a function that takes distance from the origin as input and returns an index expressing the closeness of the end-effector to the origin:

$$h(d) = \begin{cases} \dfrac{d_{\min} - d}{d_{\min}} & if \quad d \le d_{\min} \\ 0 & if \quad d > d_{\min} \end{cases} \qquad (2)$$
$$d = \|\vec{v}\| \qquad with \qquad \vec{v} = [v_x, v_y, v_z]$$

Where d is the distance between the end-effector and the origin with respect to the room coordinate system and $d_{\min}$ is the minimal distance from which h starts fluctuating. In our case, d and $d_{\min}$ are expressed in millimeter.

In the following subsection, multimodal cues used during the initialization process are presented.

Audio cue: A discrete audio cue that increases in frequency when approaching the initialization point is used. Therefore, as the end-effector approaches the initialization point, the frequency of the beeping sound increases. In our case, beep frequency is linked to the distance d according to:

$$f(d) = 1000 * h(d) \quad if \quad d \leq d_{\min} \tag{3}$$

Haptic cue: First, the direction vector $\vec{v}$ is transformed into a force vector $\vec{F}$ according to:

$$\vec{F} = \vec{\Delta} * s(d) * F_{Max} \tag{4}$$

Where $s(x)$ is a sigmoid function that allows applying a progressive force to the end-effector given by:

$$s(x) = e^{-10((1-h(d))-0.5)^2} \quad \text{if} \quad x \in [-0.5, 0.5] \tag{5}$$

Δ is the unit vector given that points in the direction of the origin $\vec{\Delta} = \frac{1}{\|\vec{V}\|}.\vec{V}$ and F_{Max} is the maximum force that can be exerted on the end-effector. This force vector is then exerted on the end-effector as a way to provide haptic guidance cue to the user to reach the origin.

The initialization process can now be carried out in two steps. First, the built-in procedure places the end-effector in a limited region of the workspace close to the origin. However, as depicted in figure 3, this procedure results in a discrepancy of about a few centimeters. Therefore, it is called "unrefined initialization region". Due to the low sensitivity of the haptic device in small distance, the end-effector can’t be moved using force feedback capabilities. In order to improve initialization accuracy, one more step is needed. To this end, we perform the multimodal approach described above. Based on the high accuracy of the external tracker and multimodal cues, the user’s hand, holding the end-effector, is guided in a precise and repeated way to the origin. Consequently, a new initialization area that we name "precise initialization region", is obtained.

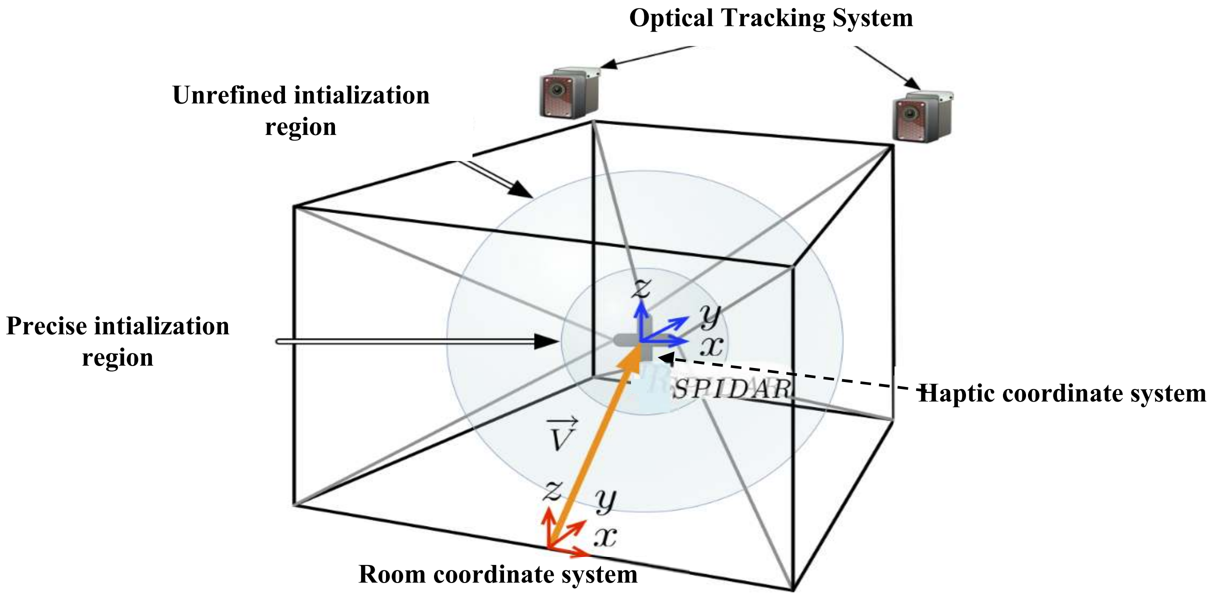

Fig. 5 Proposed intialization procedure

With the proposed initialization procedure described in the previous section, we are able to estimate the relationship between the haptic and the room coordinate system. Let p_i^H and p_i^R represent the position of the end-effector respectively in the haptic and room coordinate systems. The transformation between these points is then given by: $p_i^R = p_i^H + t$ where t denotes the translation vector from origin of the room coordinate system to the haptic workspace origin:

$$t = \begin{pmatrix} 0 \\ 0 \\ 1500 \end{pmatrix} \tag{6}$$

4.2 Shortcomings of the mechanical design

Unfortunately, additional errors in the tracked position remain due to discrepancy between the nominal kinematic model of the Scalable-SPIDAR and its actual realization. In fact, the actual realization does not quite meet the nominal design specifications required to accurately track the user's hands within the workspace. Problems arise mainly from shortcomings in the mechanical structure design (figure 6). In the next section, some of these shortcomings are investigated.

Encoders are secured to the motor shaft

Scalable-SPIDAR is a haptic device supported by strings in parallel that are actuated by tensioning motors. Each string is reeled on a pulley attached to a rotary encoder that is secured to the motor shaft. This configuration is identified as a source of problem. In fact, before running Scalable-SPIDAR, the user must specify the pulley radius in order to calculate changes in the length of strings. However, this radius is no longer constant, depending on the amount of string wound. Hence, this radius would change causing errors in length measurement-lengths are measured from the angular position of the pulley and its radius, which leads to errors in the reported end-effector position.

Pulley size is very small

The small radius of the used pulley further exacerbates errors in length measurement. It turned out that winding radius is not negligible if compared to the pulley radius especially when large amounts of string are reeled on pulley. As a consequence, the pulley radius will vary widely as the string is being wound resulting in a total discrepancy between the measured and the true length of string.

Badly-designed string guide

In the actual realization, string passes through a hole located in a plate and wraps around a pulley. This method conflicts with our need to a reliable winding control method. First, it increases system friction, damages the string and impedes it to wind around pulley on an ideal way. Second, it does not prevent string from sliding out of the grooves on the pulley. Indeed, if the pulley fails to wind the string fast enough, excessive string waiting to enter to the pulley will bend and forms a curve with a curvature large enough to slide out of the grooves on the pulley. Hence, the reeled string does not reflect the exact distance between the attaching point and the end-effector. Thus, the position measurement is no longer precise.

Large SPIDAR's dimensions

The length of string that can be wound greatly affects the workspace volume. The longer the wound length is, the larger the workspace volume is. However, due to design issues described above, length measurement errors will increase as the workspace volume increases. Thereby, the large dimensions of the haptic device exacerbate length errors.

It can easily be seen from the investigation above that shortcoming in the mechanical structure design lead to a significant loss in tracking. To address this problem, calibration method is used instead of changing the mechanical structure. The calibration problem can be tackled using model-based or model-less method. Since the HDHC (High Definition Haptic Controller) software of the Scalable-SPIDAR haptic comes as a black box, it does not allow full functionality or control of the system. Moreover, the black box structure conceals the inner kinematic model of the system used to compute end-effector pose. Therefore, we have to carry out a model-less haptic device calibration. In fact, the model-less method does not need any kinematic model or identification steps.

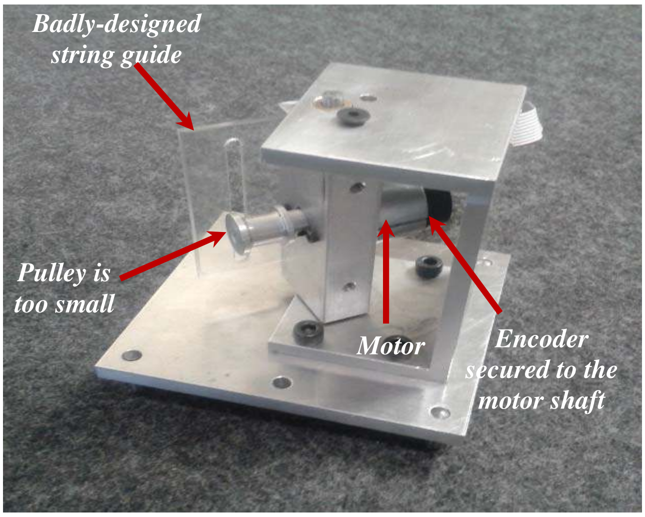
Badly-designed string guide
Pulley is too small
Motor
Encoder secured to the motor shaft

Fig. **6 B**asic structure of Scalable-SPIDAR's motor block

4.3 Calibration method

The calibration framework involves several steps: characterizing the Scalable-SPIDAR and providing methods for estimating the true end-effector position. This requires a reliable protocol for gathering large quantities of data in the working volume.

4.3.1 Data collection protocol

In order to characterize the Scalable-SPIDAR haptic device, one must collect tuples that consist of the measured data and the "truth values"-what a secondary accurate tracker should report. The accuracy of these actual values is crucial for the calibration method, since the goal of the calibration method is to map measured data into actual data. Therefore, we use the IR optical tracking device as a reference for the Scalable-SPIDAR. This is a legitimate choice because of its high accuracy. An overall view of the new procedure's components is depicted in figure 7. It can be seen that the custom initialization procedure is efficient and able to place the end-effector in the center of the cubic frame. Moreover, to provide for well-distributed data that can be collected, a volumetric calibration protocol is developed. The underlying idea is to develop a quasi-static data acquisition method where acquisition is guided by a calibration fixture. To this end, a virtual cubic lattice divided into a sequence of small cubic cells first fills the screen of our visual-haptic setup. Each cell corresponds to a sub-space of the Scalable-SPIDAR workspace.

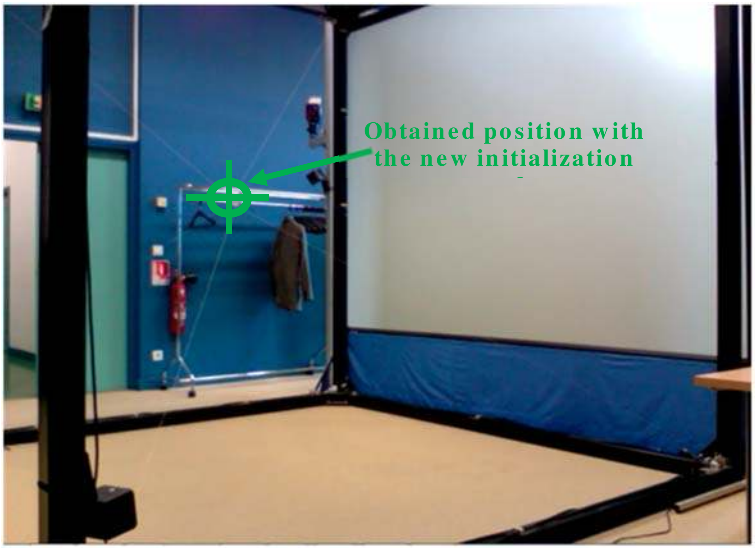

Fig. 7 The new procedure calibration

The set of small cubic cells covers the entire Scalable-SPIDAR workspace. On the other hand, a virtual end-effector (the red sphere in figure 8) shows the position of the marker attached to the end-effector with respect to the room coordinate system. The target cubic cell is highlighted by changing its color to pink. A user holding the end-effector straight and moving it until the displayed end-effector ranges inside the colored cubic cell. Then, the program records the position given by the IR optical tracking system and the haptic device. Once these positions are recorded, the cubic cell vanishes ensuring that only one measurement was associated to this sub-space. This procedure continues until all the small cubic cells inside the virtual cubic lattice are sampled. The sample points were collected starting at the front of the volume, processing through the same x-z plane and then moving forward to the next plane. In practice, a sub-volume of 1 m^3 is considered. This sub-volume is divided into 2197 small cubic cells. Hence, 2197-point measurements inside the cubic lattice are sampled. The developed protocol seems suitable for collecting a large number of data points with a predetermined data distribution using a quasi-static collection mode. Besides, it requires minimal user interaction. Indeed, to simplify human performance requirements in the data collection phase, we record data as soon the virtual end-effector collides with the cubic cell. Another big advantage of this protocol is that it allows collecting data over a flexible volume that can be changed by an expert. However, this protocol is time-consuming, especially when the number of samples increases significantly. In fact, the data collection takes about 80 minutes.

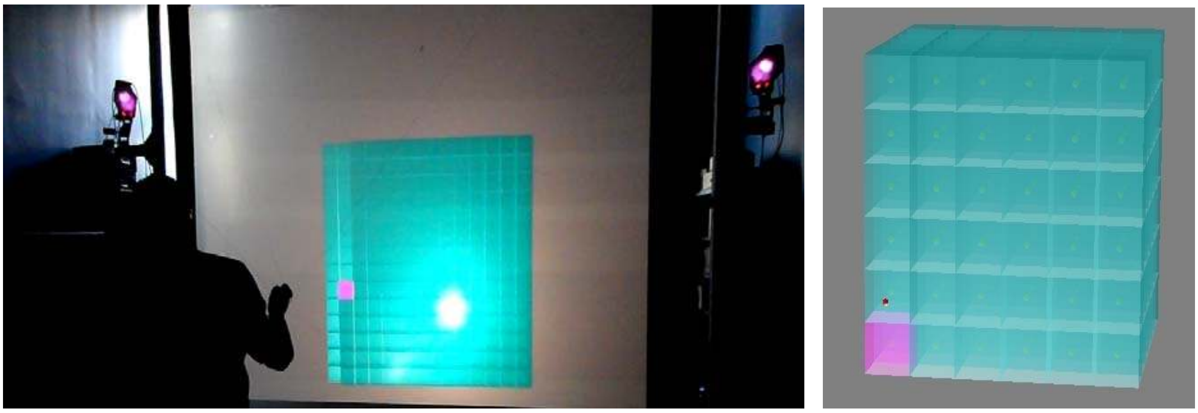

Fig. 8 Display in the Scalable-SPIDAR of boxes to collect. The virtual end-effector (the red sphere) is moved inside colored box (colored pink) to collect point

Seeing the characterization data obtained using the previous collection protocol, the absolute position errors for each of the 2197 point were calculated. The absolute position error is measured as the distance between the tracked position by the Scalable-SPIDAR haptic device and the corresponding reference position as reported by the IR optical tracking system.

$$\mathrm{err_{pos}}=\sqrt{(\mathrm{x_{SPIDAR}}-\mathrm{x_{OTS}})^2+(\mathrm{y_{SPIDAR}}-\mathrm{y_{OTS}})^2+(\mathrm{z_{SPIDAR}}-\mathrm{z_{OTS}})^2} \tag{7}$$

Where the $(x_{SPIDAR}, y_{SPIDAR}, z_{SPIDAR})$ position is reported by the Scalable-SPIDAR and $(x_{OTS}, y_{OTS}, z_{OTS})$ is the reference position.

Figures 9 and 10 illustrate position errors in two formats of varying details. Figure 9 shows the absolute position errors spatially at each reference position $(x_{OTS}, y_{OTS}, z_{OTS})$, with the error magnitude proportional to the corresponding colors. The plot clearly shows that errors are more pronounced when the end-effector is manipulated away from the center of the workspace toward the edge of the cubic frame.

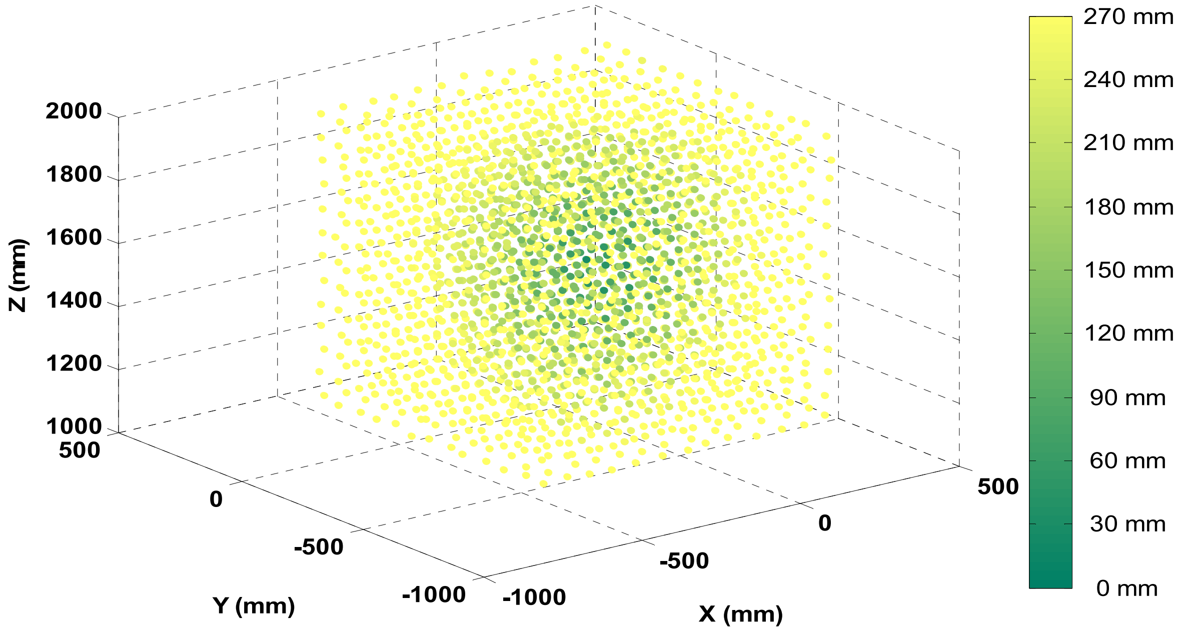

Fig. 9 Spatially-represented absolute position errors

In figure 10, absolute position errors are plotted as a function of the sequence in which they were collected from the front of the volume to the back. Plotting position errors as 1D plot results in the loss of a lot of spatial information, but still shows the same trend, and from the plot's periodicity we can infer that errors increase at the volume edges.

The clusters of large errors at the upper left and right corner of the plot, also called outliers, will be investigated in the next section.

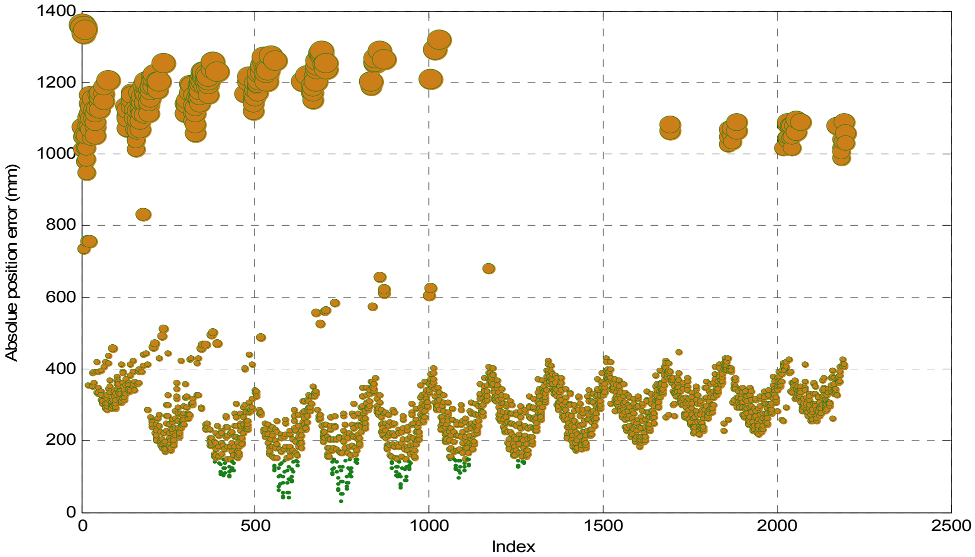

Fig. 10 Absolute position errors represented in sequence

4.3.2 Machine-learning techniques for calibration

The driving force of our current research is the development of a calibration procedure using regression techniques. In fact, the Scalable-SPIDAR haptic device is a multi-DOF device with complex-to model geometries. Moreover, some of the inaccuracy sources are difficult to model explicitly, which results in lack of model completeness and therefore in a loss of accuracy. Regression techniques provide attractive solutions for improving the tracking accuracy of the Scalable-SPIDAR. Indeed, the input-output relationship is performed in a black box method without the need for modelling explicitly the different errors sources affecting the device accuracy. To this end, we look for a regression function f that provides mapping from measured to actual data. The measured data is directly read from the haptic device being calibrated. The actual values are reported by the IR optical tracking system. Although the mapping can be performed using conventional nonlinear regression, the use of machine learning techniques such as neural networks and support vector regression is preferred because they can approximate any complex regression (Pao 1989).

The function-fitting paradigm from a machine learning point of view supposes that errors are additive and that the model Y=f(X)+ε is a reasonable assumption (Hastie et al. 2009). Supervised learning attempts to learn f based on training set of N input-output pairs $(x_i, y_i), i = 1, ...N$. Where the symbol X denotes an input variable and Y denotes a quantitative output. Observed values are written in lowercase; hence the ith observed value of X is written as x_i (where x_i is a scalar or vector).The input value x_i is fed into an artificial system, known as learning algorithm which also produces outputs $\widehat{f}(x_i)$ in response to the inputs. The learning algorithm has the property to modify its input/output relationship $\widehat{f}$ in response to the difference between the original and generated output $y_i - \widehat{f}(x_i)$. This process is known as learning by examples. Here the data pairs $(x_i, y_i), i = 1, ...N$ correspond respectively to tracked position and true position reported by the IR optical tracking system.

a. Neural network

Our goal is to design a neural network capable of estimating the actual end-effector position based on its tracked position. It has been shown that a feed-forward network with hidden layers is a universal approximator. This means that a multi-layer feed-forward network can approximate any continuous nonlinear function over a compact subset of $\mathbb{R}^n$ to an arbitrary degree of accuracy (Hornik et al. 1989). This is achieved by training the neural network with an error back propagation algorithm by presenting points' pairs of tracked position by Scalable-SPIDAR and actual position reported by the IR optical tracking system. Therefore, the neural network devised for position error correction should accept three input coordinates of the measured position and should produce three output x, y and z coordinates of the corrected position. We need to decide on the network architecture (number of hidden layers, number of neurons in each layer, type of transfer function, etc.) and on the type of the error back propagation training algorithm. There is no recipe for selecting the right network complexity for a given problem and the optimal solution is often found by experimentation. One of the first decisions to be made is how many layers are needed in order to obtain a good model. It is hard to say which topology is better. A reasonable answer would specify the cost function for neural network's performance, including the size of the neural network, accuracy and the like. According to Kecman (2001), it might be useful to try solving the problem at hand using a neural network with one hidden layer. In this work, we use a single layer perceptron also called the hidden layer back-propagation network with a tangent-sigmoid transfer function for the neurons in the hidden layer and linear transfer function for the neurons in the output layer (Bishop 1995). The right number of neurons in hidden layer required for an optimal solution is estimated experimentally by training networks of different sizes and determining the one with the best overall performances. Different error back-propagation training algorithms can be used to train feed-forward neural network. In order to obtain a fast training procedure in the sense that the network converges in a small number of iterations, we select Levenberg-Marquardt algorithm (Hagan and Menhaj 1994;Yu and Wilamowski 2011). Although neural network have good non-linear regression abilities, they also have some drawbacks. During the neural network optimization process, we need to move on to a surface having multiple local minima. Neural network's learning/optimizing algorithms cannot avoid being stuck in a local minima, which can lead to a sub-optimal solution (Suykens et al. 2002). Thus, neural networks have the problem that one cannot be sure that the global minimum is approaching, as there are exponentially many local minima, which can lead to a sub optimal solution. This problem is significantly mitigated by using support vector machines. In fact, SVM are not plagued with the problem of local minima as neural network are and provide a unique solution (Smola and Schölkopf 2004).

b. Support vector machine for regression

Initially developed for solving pattern recognition problems, support vector machine techniques can be successfully applied in regression for a functional approximation problem (Kwiatkowska and Fargion 2003; (Kecman 2005) Given a training data set $D = \{[\mathbf{x}_i, y] \in \mathbb{R}^n \times \mathbb{R}, i = 1, ..., l\}$ where the input **x** denotes n-dimensional vectors $\mathbf{x} \in \mathbb{R}^n$ and outputs $y \in \mathbb{R}$, support vector regression attempts to learn the input-output relationship in a black box modeling approach. It originally consists in finding a function $f(x)$ that has at most an epsilon deviation from the actually obtained targets for all the training data with the smallest complexity. Hence, support vector regression (SVR) amounts to solve a constrained optimization problem, in which the complexity, measured by the norm of the weight parameters, is minimized. Allowing for the cases where the constraints cannot all be satisfied leads to minimizing the Vapnik linear loss function with ε-insensitivity zone described by Melin and Castillo (2005) as:

$$E(\mathbf{x},y,f)=|y-f(\mathbf{x},\mathbf{w})|_{\varepsilon}=\begin{cases}0 & \text{if } |y-f(\mathbf{x},\mathbf{w})|\le\varepsilon\\ |y-f(\mathbf{x},\mathbf{w})|-\varepsilon & \text{otherwise}\end{cases}\tag{8}$$

where **w** denotes the weight vector.

The support vector regression algorithm can therefore be written as a quadratic programming problem where both the L1- norm of the errors larger than epsilon and the L2-norm of the weight parameters are minimized.

$$\frac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{l}|y_i-f(\mathbf{x}_i,\mathbf{w})|_{\varepsilon}\tag{9}$$

where C is a constant providing the trade-off between minimizing training errors and minimizing the model complexity term $\|\mathbf{w}\|^2$. To deal with nonlinear regression tasks, SVR uses various kernel functions such as polynomial, sigmoidal and Radial Basis Function RBF kernel, that allow to extend linear methods to nonlinear problems through an implicit mapping to higher dimensional feature space. Compared to neural network, SVR has many advantages: intrinsic regularization, no local minima (convex problem with a unique solution) and good generalization ability from a limited amount of samples (Bloch et al. 2008).

5. Experimental results and discussion

In this section, a series of experiments was conducted to evaluate the performances of both neural and support vector regression. Each experiment used gathered data to approximate regression function that provide mapping between the tracked position by the Scalable-SPIDAR haptic device and the reference position reported by the optical tracking system.

5.1 Data preprocessing

Before applying the regression technique, input data need to be prepared to simplify either the neural network or support vector training. One important task is elimination of outliers. Training with outliers without awareness may lead to fitting unwanted data and may jeopardize function approximation (Huang et al. 1992) (Vapnik et al. 1997). A close examination of the 1D box plot reveals that the points with large errors are isolated and inconsistent with the majority of data. This can be explained by the abnormal behavior of the motor placed at the upper left corner of the cubic frame. Therefore, we need a way to detect these data points and deflate their influence. We use informal box plots (Tukey 1977) to pinpoint the outlying points in the current dataset. Taking advantage of this method, we eliminate nearly 300 points of the 2197 points. The absolute position errors distribution without outliers is plotted as a frequency histogram (Figure 11). Some of the representative statistics that describe much of this error distribution are given in table 1. As can be seen from this table, the mean absolute error value is rather large. This means an important discrepancy of about 270 mm between the tracked position and the true position that ideally has to be reached. This poor spatial tracking accuracy may cause inconsistency between the physical and virtual environments, which in turn, can render training in a virtual environment ineffective, or, more seriously, result in negative skill transfer.

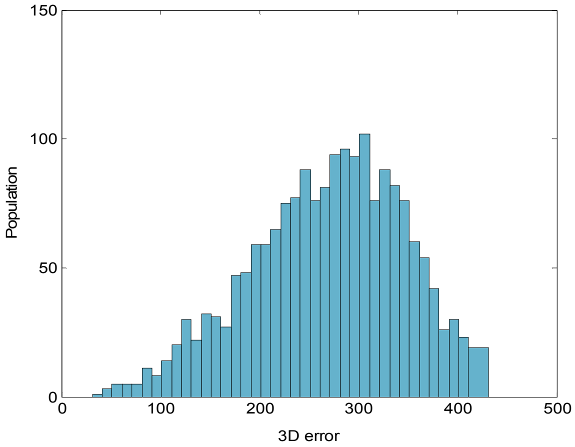

Fig. 11 Absolute position errors represented as a frequency histogram

Table 1 Error statistics for non-calibrated Scalable-SPIDAR

Absolute Position error	Scalable-SPIDAR
Mean (mm)	267.8350
Standard derivation (mm)	76.5174
Maximum (mm)	430.9926

Before applying regression techniques, we still need to scale data. In fact, the task of training regression algorithm is significantly simplified if data lie within a small range. We scale all inputs to have mean zero and standard deviation one.

$$\begin{cases} N_{SPIDAR} = ((P_{SPIDAR} - \Gamma_{SPIDAR})./\sum_{SPIDAR}) \\ N_{OTS} = ((P_{OTS} - \Gamma_{OTS})./\sum_{OTS}) \end{cases} \tag{10}$$

Where:

$\Gamma_{SPIDAR} = \gamma_{SPIDAR}.J$ and $\sum_{SPIDAR} = \sigma_{SPIDAR}.J$

$\Gamma_{OTS} = \gamma_{OTS}.J$ and $\sum_{OTS} = \sigma_{OTS}.J$

γ_{SPIDAR} : Matrix of mean components of the matrix of positions given by the Scalable-SPIDAR.

γ_{OTS} : Matrix of mean components of the matrix of positions given by the optical tracking system.

P_{SPIDAR} : Positions reported by the Scalable-SPIDAR.

P_{OTS} : Positions reported by the IR optical tracking system.

J: Identity matrix.

N_{SPIDAR} : Standardized positions of the Scalable-SPIDAR.

N_{OTS} : Standardized positions of the optical tracking system.

5.2 Calibration using neural network

In the present study, we use a single-hidden layer feed-forward network which accept three inputs-x, y and z coordinates of the tracked position by the Scalable-SPIDAR and produce three outputs x, y and z of the calibrated position. The selected network model had two layers, with a tangent-sigmoid transfer function $\Phi(u) = \tanh u = \frac{2}{1+e^{-2u}} - 1$ for the neurons in the hidden layer and linear transfer function $\Phi(u) = u$ for the neurons in the output layer (Figure 12).

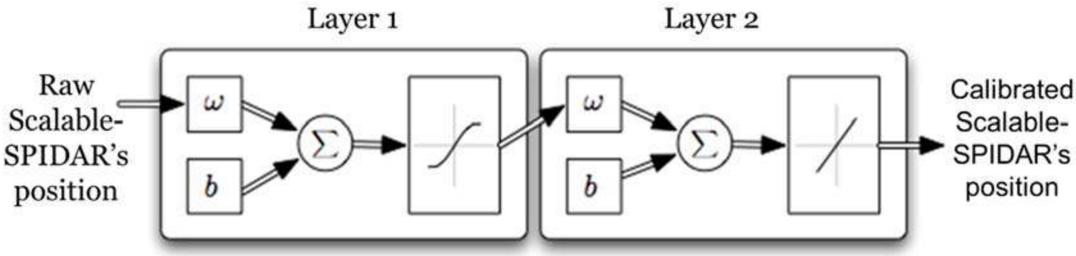
Layer 1
Layer 2
Raw Scalable-SPIDAR's position
ω
b
Σ
ω
b
Σ
Calibrated Scalable-SPIDAR's position

Fig. 12 Single-hidden layer network used for the Scalable-SPIDAR calibration

This network was then trained to provide the corrected outputs by modifying weights according to "training algorithm". The Lavender-Marquardt algorithm was preferred as it is saving time and well suited to function-approximation problems with networks of moderate size and number of parameters.

Furthermore, deciding on the number of neurons in the hidden layer is essential for deciding on our overall neural network architecture and, therefore, must be carefully considered. To get the right number of neurons, we train neural networks with 3 to 24 neurons in the hidden layer and compute the average errors between the expected actual position and the calibrated position. Based on these results, we select the most effective configuration estimating the values of the given data set. Thus, in essence, we calibrate the Scalable-SPIDAR with different configurations and pick up the one that gives the best results. By plotting the mean absolute error versus the number of neurons in hidden layer, we can see that the optimal solution is 20.

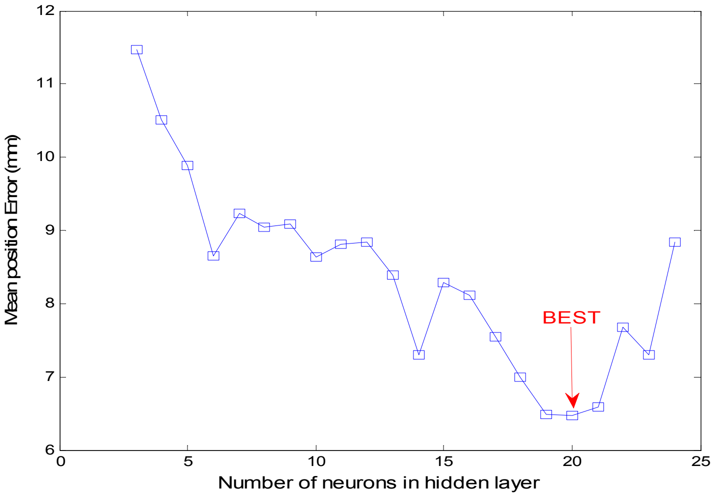

Fig. 13 Mean absolute position error versus number of neurons in hidden layer

Analyses were conducted to evaluate the performances of neural network-based calibration. Figure 13 represents the spatial distribution of absolute position errors after calibration with the error magnitudes proportional to the corresponding color. As revealed by this plot, neural network-based calibration is able to reduce the magnitude of absolute position errors from 267 mm to less than 30 mm in almost whole workspace.

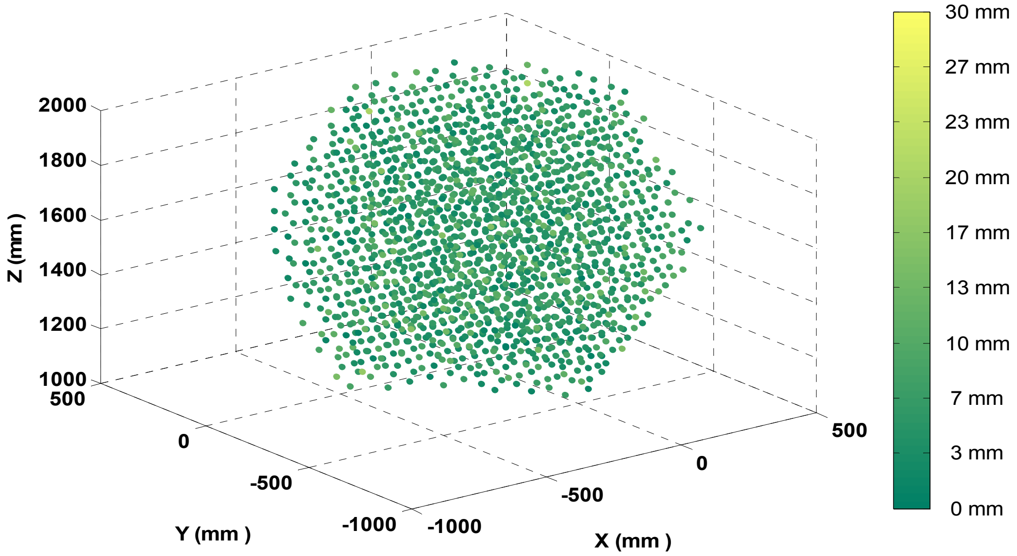

Fig. 14 Absolute position errors represented spatially after neural-network based calibration

As for figure 14, it depicts the distribution of absolute position errors as a function of distance to the Scalable-SPIDAR's origin before and after calibration. As shown by the plot, neural network-based method is efficient, and able to achieve fewer errors in the whole working space. It is apparent that error becomes less sensitive to distance and even the errors at distances greater than 600 mm were successfully corrected by calibration.

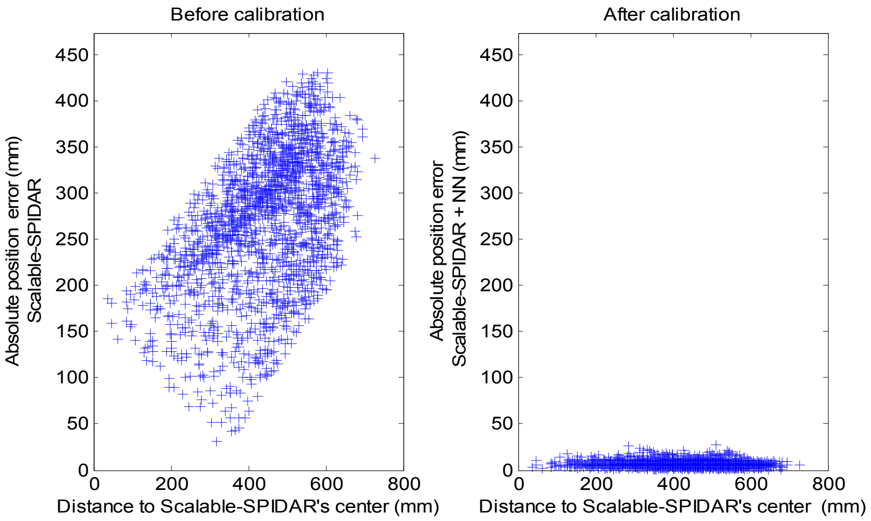

Fig. **15 D**istribution of absolute position errors as a function of distance to the Scalable SPIDAR's origin (before and after neural network-based calibration method)

To gauge more the importance of the improvements obtained with the neural network-based calibration method, some representative statistics before and after calibration is presented in table 2. The mean tracked position error from the training data set is 267.8350±76, 5174 mm (table 2), whereas the mean position error after calibration is 6.4760 ± 3.4331 mm. The statistics proved that neural network has significantly decreased errors in the tracked position.

Table 2 Error statistics before and after neural network-based calibration

Absolute Position Error	**Raw**	**NN**
Mean (mm)	267.8350	6.4760
Standard derivation (mm)	76.5174	3.4331
Maximum (mm)	430.9926	25.9110

After testing neural network with training data, we need to measure its ability to handle unseen data. The underlying idea is to construct a testing dataset. To this end, the end-effector of the Scalable-SPIDAR haptic device was moved in random trajectories of sequential points. These paths represent groups of data points within our predetermined working space. Following this, corresponding point measurements of both the Scalable-SPIDAR and the IR optical tracking system can be obtained, thus allowing evaluating the generalization capability of our network.

Table 3 Error statistics in generalization

Absolute Position Error	**Raw**	**NN**
Mean (mm)	263.0445	14.5080
Standard derivation (mm)	73.9771	8.3958
Maximum (mm)	415.8266	74.5346

Table 3 summarizes the results of the calibration based on the data from the testing dataset. A close examination shows that our neural network does not keep the same performances as in training, but it is still ranging within reasonable error band of 14.5080 ± 8.3958 mm.

5.3 Calibration using support vector regression

In this section, we applied support vector regression (SVR) for calibrating the Scalable-SPIDAR haptic device. The convergence of support vector machine depends on the selection of a kernel function. Kernel functions project the data into high dimensional feature space. This work uses that radial basis function to perform better than polynomial and linear functions in demining a relationship between Scalable-SPIADR and optical tracking system data. In the following, trials radial basis kernels were applied:

$$\text{radial basis}\, fucntion = \exp(-gamma * \left\| x_i - x_j \right\|^2) \tag{11}$$

where gamma is known as the kernel parameter.

A search was performed for the most effective capacity parameter C to enhance generalization accuracy of the regression method. The capacity measures the flexibility or richness of regression functions and gives the protection against over fitting. In experiments, the capacity was set to values between 30 and 100. Another parameter used in the training of support vector regression is epsilon, which checks the insensitivity of the regression. The algorithm assumes that estimations that lie within epsilon distance of their true values are accurate enough. Epsilon was chosen to be equal to 0.01.

To analyze the performances of the support vector regression, we plot the spatial distribution of absolute position errors as in section 5.2 but after calibration using support vector regression method.

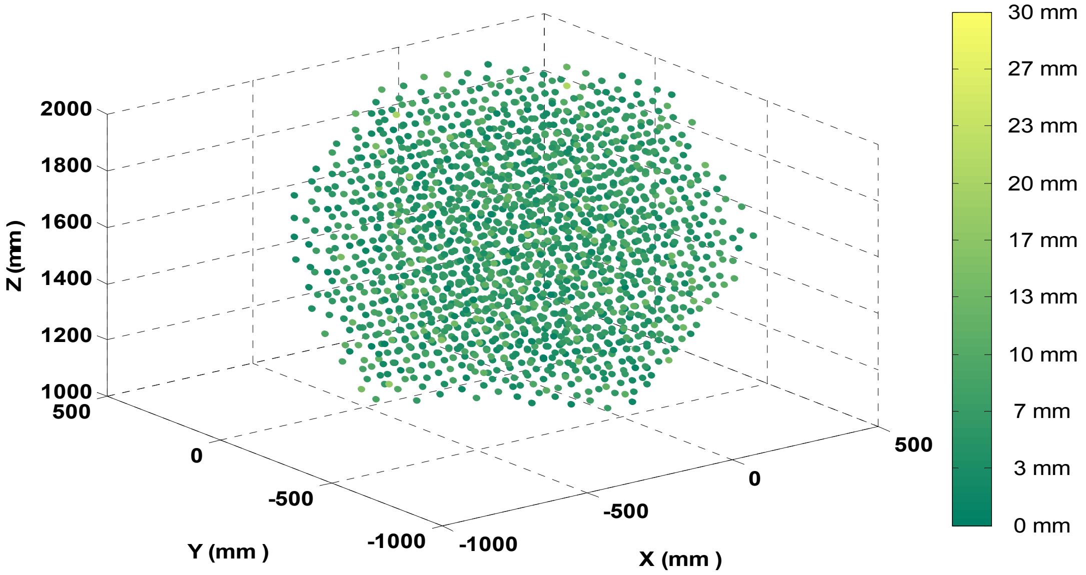

Fig. 16 Absolute position errors represented spatially after SVR calibration

As we can see, figure 15 shows that SVR is quite effective and greatly improves the Scalable-SPIDAR accuracy in comparison with figure 9. Support vector regression performs a good calibration in the whole workspace expect in its corners.

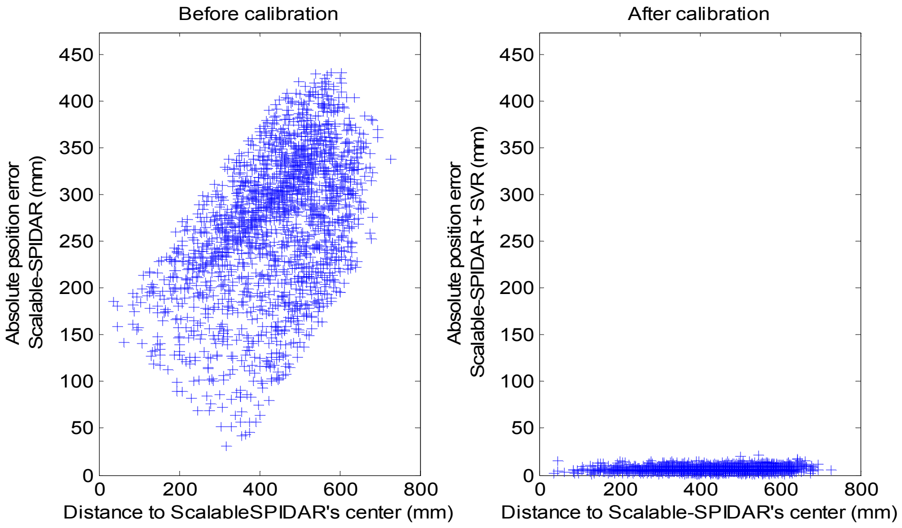

Fig. 17 Distribution of absolute position errors as a function of distance to the Scalable-SPIDAR's origin (before and after SVR calibration)

Figure 16 shows that SVR is quite efficient and exhibits fewer errors. The Scalable-SPIDAR accuracy is less sensitive to distance to the Scalable-SPIDAR's origin and even the errors at distances greater than 600 mm were corrected by calibration. We can see from table 4 that the SVR calibration was successful in reducing the mean absolute position error in tracked position from 267.8350±76.5174 mm to 6.2765±2.8626 mm. The maximum possible error is about 20.3819 mm

Table 4 Error statistics before and after neural support vector regression

Absolute Position Error	**Raw**	**SVR**
Mean (mm)	267.8350	6.2765
Standard derivation (mm)	76.5174	2.8628
Maximum (mm)	430.9926	20.3819

In order to measure the generalization ability of the support vector regression-based method, we use the same testing dataset created previously.

Table 5 Error statistics in generalization using SVR

Absolute Position Error	**Raw**	**SVR**
Mean (mm)	263.0445	5.4509
Standard derivation (mm)	73.9771	2.5501
Ma-baseximum (mm)	415.8266	22.3333

A close examination of table 5 reveals that SVR keeps convenient performance when handling unseen data. Both the mean and the standard deviation of absolute errors in position at the testing points are less than those by the neural network method. The SVR provided improved results at high error levels with a maximum position error of 22, 3333 mm.

5.4 Comparison of the calibration results by support vector regression and neural network-based calibration techniques

To stress the performances improvements obtained with machine-learning methods, we summarize the calibration results of the two methods discussed in this paper. The first method employs a single-layer feed forward network model as a calibration model. The second method uses a support vector regression to calibrate the Scalable-SPIADR readings. The design parameters were determined based on the guidelines described at the beginning of section 5.3.

Table 6 Summary of the calibration results obtained by the two learning-machine techniques discussed in this paper

Absolute Position Error	**Raw**	**NN**	**SVR**
Mean (mm)	267.8350	6.4760	6.2765
Standard derivation (mm)	76.5174	3.4331	2.8628
Maximum (mm)	430.9926	25.9110	20.3819

Based on the error statistics listed in table 6 and above (table 3 and 5), support vector regression method outperformed the neural network-based method when applied to calibrate the position readings form the Scalable-SPIDAR. The significance of the improvements obtained by SVR was demonstrated more by its ability to generalize when handling unseen data. Actually, SVR shows higher robustness and greater estimation ability for future data points. The obtained results suggest that the proposed SVR is better suited than the neural network-based method for the Scalable-SPIDAR calibration. This in accordance with our initial hypothesis. In fact, as we have mentioned above, neural network's learning algorithms may stuck in local minima, thus providing suboptimal solution and limiting the effectiveness of the neural network-based calibration method

The support vector regression has the advantage of being free from local minima, and thus can be used to find whether the selected neural network has reached the global minima. The major advantage of this approach is that it allows avoiding non-optimal calibration solution.

5.5 Recalibration at runtime

The one-time calibration of the Scalable-SPIDAR is a crucial step to provide accurate tracking data. However, issues influencing calibration such as the haptic controller behavior can change while using the Scalable-SPIDAR. Therefore, position estimation might deteriorate if haptic control failures occur. To keep tracking quality high, recalibration at runtime is required at suitable intervals. It seems more convenient to use data provided by the optical tracking system for unobtrusive calibration. In fact, if the Scalable-SPIDAR loses tracking, optical tracking data can then be used and calibration would be improved at runtime without disturbing the user. Combining the Scalable-SPIDAR and the Optical Tracking System into a hybrid tracking system aims to maintain an uninterrupted tracking data and the required spatial accuracy under varying conditions in the operating workspace.

6. Conclusions

In this paper, a novel calibration procedure was proposed to improve the tracking accuracy of a human-scale haptic device: the Scalable-SPIDAR. The basic idea of the procedure is to adopt the concepts of virtual reality and machine-learning techniques to calibrate the position errors present in the haptic device readings. In this paper, a visual-haptic VR setup was established to maintain an interrupted tracking and to enable haptic interaction with virtual objects in the virtual environment. Based on this setup, a multimodal intialization procedure was developed in order to help the user to accomplish and repeat the intialization of the Scalable-SPIDAR in a precise manner. Nevertheless, additional errors in the reported position remains due to serious problems in the mechanical structure design of the haptic device. To overcome this issue, a collection data protocol was developed first to characterize the Scalable-SPIDAR's tracking position errors. Then, a model-less approach using regression methods was adopted to correct these errors. Two nonlinear regression techniques, thus, have been used for the calibration of the haptic device. The first one is a neural network-based calibration method, while the second is based on the support vector regression.

The obtained results above indicate that the proposed calibration techniques work well for position error, reducing the scalable-SPIDAR's average absolute position error to about 6 mm. The position correction achieved by the calibration procedure thus offers a visible improvement in tracking accuracy in a VR system.

Compared with other calibration methods, the proposed method employs machine-learning techniques such as neural network and support vector regression to tackle problems of tracking inaccuracy. Based on their universal approximation capability, these methods were able to perform the calibration of the Scalable-SPIDAR without the need for modeling explicitly the main error sources affecting its accuracy. Even though the proposed procedure is limited by the use of the Scalable-SPIDAR system, it is possible to extend it to other tracking systems. The distinguishing features of scalability and convince make from this procedure a flexible method that can accommodate various virtual reality systems ranging from desktop, human scale and network environments.

Furthermore, since the Scalable-SPIDAR is intended to serve as a 6-DOF human-scale haptic device, a straightforward extension to this work is to build an orientation calibration method. Therefore, the Scalable-SPIDAR can be used to accomplish several applications ranging from the simple "pick and place" task to more physical based interactions.

References

A.R.T. GmbH, 2003. *ARTtrack1 & DTrack - Manual*,

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*, New York, NY, USA: Oxford University Press, Inc.

Bloch, G. et al., 2008. Support vector regression from simulation data and few experimental samples. *Information Sciences*, 178(20), pp.3813–3827.

Boudoin, P. et al., 2010. SPIDAR Calibration based on Neural Networks versus Optical Tracking. BT - Artificial Neural Networks and Intelligent Information Processing, Proceedings of the 6th International Workshop on Artificial Neural Networks and Intelligent Information Processin. , pp.87–98.

Bouguila, L., Ishii, M. & Sato, M., 2000. Effect of coupling haptics and stereopsis on depth perception in virtual environment. *... of the 1st Workshop on Haptic Human ...*, pp.54–62.

Briggs, W., 1999. Magnetic Calibration by Tetrahedral Interpolation. In *Proceedings of NIST-ASME Industrial Virtual Reality Symposium*. Chicago, pp. 27–32.

Bryson, S., 1992. Measurement and Calibration of Static Distortion of Position Data from 3D Trackers. , 1669(1992), pp.244–255.

Buoguila, L., Ishii, M. & Sato, M., 2000. Multi-Modal Haptic Device For Large-Scale Virtual Environment. *ACM Multimedia*, pp.277–283.

Burdea, G.C., 1996. *Force and Touch Feedback for Virtual Reality*, New York, NY, USA: John Wiley &amp; Sons, Inc.

Ellis, S.R. et al., 1999. Sensor spatial distortion, visual latency, and update rate effects\non 3D tracking in virtual environments. *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, pp.218–221.

Faroque, S. et al., 2015. Haptic Virtual Reality Training Environment for Micro-robotic Cell Injection. In H. Kajimoto, H. Ando, & K.-U. Kyung, eds. *Haptic Interaction: Perception, Devices and Applications*. Tokyo: Springer Japan, pp. 245–249.

Fletcher, C. et al., 2013. The Development of an Integrated Haptic VR Machining Environment for the Automatic Generation of Process Plans. *Comput. Ind.*, 64(8), pp.1045–1060.

Fuchs, P., Moreau, G. & Guitton, P., 2011. *Virtual Reality: Concepts and Technologies* 1st ed., Boca Raton, FL, USA: CRC Press, Inc.

Ghazisaedy, M. et al., 1995. Ultrasonic calibration of a magnetic tracker in a virtual reality space. *Proceedings Virtual Reality Annual International Symposium '95*, pp.179–188.

Group, S., 2005. SPIDAR-G/AHS1.0A User's Manual Ver.2.0. , pp.1–44.

Hagan, M.T. & Menhaj, M.B., 1994. Training feedforward networks with the Marquardt algorithm. *Neural Networks, IEEE Transactions on*, 5(6), pp.989–993.

Harders, M. et al., 2009. Calibration, Registration, and Synchronization for High Precision Augmented Reality Haptics. *Visualization and Computer Graphics, IEEE Transactions on*, 15(1), pp.138–149.

Hastie, T.J., Tibshirani, R.J. & Friedman, J.H., 2009. *The elements of statistical learning : data mining, inference, and prediction*, New York: Springer.

Hayward, V. et al., 2004. Haptic interfaces and devices. *Sensor Review*, 24(1), pp.16–29.

Hirata, Y. & Sato, M., 1992. 3-dimensional Interface Device For Virtual Work Space. *Intelligent Robots and Systems, 1992., Proceedings of the 1992 lEEE/RSJ International Conference on*, 2, pp.889–896.

Hornik, K., Stinchcombe, M. & White, H., 1989. Multilayer Feedforward Networks Are Universal Approximators. *Neural Netw.*, 2(5), pp.359–366.

Huang, J.-N. et al., 1992. A Comparison of Projection Pursuit and Neural Network Regression Modeling. *Advances in Neural Information Processing Systems 4*, pp.1159–1166.

Ikits, M. et al., 2001. An improved calibration framework for electromagnetic tracking devices. *Proceedings IEEE Virtual Reality 2001*, pp.63–70.

Ikits, M. et al., 2000. The Visual Haptic Workbench. In *In Proc. PHANToM Users Group Workshop*. pp. 46–49.

Ikits, M., Hansen, C.D. & Johnson, C.R., 2003. A Comprehensive Calibration and Registration Procedure for the Visual Haptic Workbench. In *Proceedings of the Workshop on Virtual Environments 2003*. EGVE '03. New York, NY, USA: ACM, pp. 247–254.

ISO 5725-1, 1994. *Accuracy (trueness and precision) of measurement methods and results — Part 1: General principles and definitions.*

Jayaram, U. & Repp, R., 2002. Integrated Real-Time Calibration of Electromagnetic Tracking of User Motions for Engineering Applications in Virtual Environments. *Journal of Mechanical Design*, 124(December 2002), p.623.

K. Reinig, R. Tracy, H.G. and T.M., 1997. No TitleSome Calibration Information for a Phantom 1.5 a. In *Second PHANToM Users Group Workshop*.

Kecman, V., 2001. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, Cambridge, MA, USA: MIT Press.

Kecman, V., 2005. Support Vector Machines – An Introduction BT - Support Vector Machines: Theory and Applications. In L. Wang, ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–47.

Kenwright, D.N. & Lane, D.A., 1996. Interactive Time-Dependent Particle Tracing Using Tetrahedral Decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 2(2), pp.120–129.

Kim, S. et al., 2002. Tension based 7-DOF force feedback device: SPIDAR-G. *Virtual Reality, 2002. Proceedings. IEEE : Virtual Reality, 2002. Proceedings. IEEE : Virtual Reality, 2002. Proceedings. IEEE*, 2002, pp.283–284.

Kindratenko, V., 2000. A survey of electromagnetic position tracker calibration techniques. *Virtual Reality*, 5(3), pp.169–182.

Kindratenko, V., 1999. Calibration of electromagnetic tracking devices. *Virtual Reality*, 4, pp.139–150.

Kindratenko, V. & Bennett, A., 2000. Evaluation of Rotation Correction Techniques for Electromagnetic Position Tracking Systems. In J. Mulder & R. van Liere, eds. *Virtual Environments 2000 SE - 3*. Eurographics. Springer Vienna, pp. 13–22.

Kindratenko, V. V & Sherman, W.R., 2005. Neural Network-based Calibration of Electromagnetic Tracking Systems. *Virtual Real.*, 9(1), pp.70–78.

Knoerlein, B. & Harders, M., 2011. Comparison of tracker-based to tracker-less haptic device calibration. *World Haptics Conference (WHC), 2011 IEEE*, pp.119–124.

Kunzler, U. & Runde, C., 2005. Kinesthetic haptics integration into large-scale virtual environments. *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pp.551–556.

Kwiatkowska, E.J. & Fargion, G.S., 2003. Application of machine-learning techniques toward the creation of a consistent and calibrated global chlorophyll concentration baseline dataset using remotely sensed ocean color data. *IEEE Transactions on Geoscience and Remote Sensing*, 41(12), pp.2844–2860.

Livingston, M.A. & State, A., 1997. Magnetic tracker calibration for improved augmented reality registration. *Presence: Teleoperators & Virtual Environments*, 6(5), pp.532–546.

Melin, P. & Castillo, O., 2005. Studies in Fuzziness and Soft Computing , Volume 172. *Soft Computing*, 18(3-4), p.318.

Meyer, K., Applewhite, H.L. & Biocca, F.A., 1992. A Survey of Position Trackers. *Presence:*

Teleoperators and Virtual Environments, 1(2), pp.173–200.

Moreira, A.H.J. et al., 2014. Electromagnetic tracker feasibility in the design of a dental superstructure for edentulous patients. *IEEE MeMeA 2014 - IEEE International Symposium on Medical Measurements and Applications, Proceedings*, pp.1–6.

Pao, Y.-H., 1989. *Adaptive Pattern Recognition and Neural Networks*, Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Ramsamy, P. et al., 2006. Using Haptics to Improve Immersion in Virtual Environments. In V. Alexandrov et al., eds. *Computational Science – ICCS 2006 SE - 81*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 603–609.

Smola, A.J. & Schölkopf, B., 2004. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3), pp.199–222.

Srinivasan, M.A., 1995. Virtual Reality: Scientific and Technical Challenges. *Report of the Committee on Virtual Reality Research and Development, National Research Council. N. I Durlach and A. S. Mavor ,National Academy Press*, pp.161–187.

Srinivasan, M.A. & Basdogan, C., 1997. Haptics in virtual environments: Taxonomy, research status, and challenges. *Computers & Graphics*, 21(4), pp.393–404.

Suykens, J.A.K. et al., 2002. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1–4), pp.85–105.

Tukey, J. ~W., 1977. *Exploratory Data Analysis* Tukey, J.~W., ed., Reading, Mass.: Addison-Wesley.

Vapnik, V., Golowich, S.E. & Smola, A.J., 1997. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. In M. Mozer, M. I. Jordan, & T. Petsche, eds. *Advances in Neural Information Processing Systems 9 --- Proceedings of the 1996 Neural Information Processing Systems Conference (NIPS 1996)*. Dever, CO, USA: MIT Press, Cambridge, MA, USA, pp. 281–287.

Welch, G. & Foxlin, E., 2002. Motion tracking: no silver bullet, but a respectable arsenal. *Computer Graphics and Applications, IEEE*, 22(6), pp.24–38.

Xia, P. et al., 2012. A new type haptics-based virtual environment system for assembly training of complex products. *The International Journal of Advanced Manufacturing Technology*, 58(1), pp.379–396.

Yu, H. & Wilamowski, B.M., 2011. Levenberg-Marquardt training. *Industrial Electronics Handbook, vol. 5 – Intelligent Systems*, pp.12–1 to 12–18.

Zachmann, G., 1997. Distortion correction of magnetic fields for position tracking. *Proceedings Computer Graphics International*, pp.213–220,.