Virtual Reality: Research, Development, and Applications, 2000, vol. 5, no. 3, pp. 169-182.

A survey of electromagnetic position tracker calibration techniques

Volodymyr V. Kindratenko
National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign
405 North Mathews Avenue, Urbana, IL 61801, USA
kindr@ncsa.uiuc.edu

Abstract

This paper is a comprehensive survey of various techniques used to calibrate electromagnetic position tracking systems. A common framework is established to present the calibration problem as the interpolation problem in 3D. All the known calibration techniques are classified into local and global methods and grouped according to their mathematical models. Both the location error and the orientation error correction techniques are surveyed. Data acquisition devices and methods as well as publicly available software implementations are reviewed too.

Keywords

Tracker calibration, Electromagnetic position tracking, Virtual reality

1 Introduction

Several technologies [1] can be used to track user's position in virtual reality (VR) application s such as Cave Automated Virtual Environment (CAVE) [2]. So far, electromagnetic position tracking systems have been the most widely used technology. They are relatively inexpensive, provide a reasonably large working range, have fairly good resolution with acceptable jitter, are convenient to use, and do not suffer from the line of sight problem. However, they have an inherent accuracy problem due to the dependence of the measurements on the local electromagnetic field that can be easily distorted. Also, as the distance between the transmitter and the sensor increases, their accuracy decreases. One way to overcome these problems is to calibrate the system so that the position tracking errors can be effectively compensated.

Several tracker calibration techniques have been proposed in the past decade. However, they are scattered in time and across different publications and frequently repeat each other without knowing it. Also, each author defines the problem of tracker calibration in his own application-specific framework, thus making it difficult to interpret the results and compare various techniques. No unifying concept has been proposed so far. Therefore, there is a need for a comprehensive review of the state of the art in this field. In this work, the tracker calibration techniques are surveyed with the goals of defining a common framework, identifying and classifying different techniques reported in the literature, giving a detailed description of different approaches, and providing an in-depth analysis of their advantages and limitations.

Up until now, electromagnetic tracker calibration has seldom been done in practice for two main reasons: absence of the software and the difficulties involved with data acquisition. This survey shows that this is not the case anymore. The research conducted in the past decade resulted in the development of improved calibration methods as well as simplified data acquisition techniques and software. Using a freely available software and a simple and inexpensive equipment, it is now possible to calibrate an electromagnetic tracking system just in a few hours and with little effort.

The article is organized in the following way. First, an overview of electromagnetic tracking principles is given followed by the description of different types and sources of errors. Next, tracker calibration is introduced followed by a detailed description of both the location and orientation error correction techniques presented in a common framework. Next, data acquisition techniques used by different authors are described. The final part deals with the software implementations.

2 Electromagnetic position tracking

2.1 Principles and systems

Six degree of freedom (DoF) electromagnetic tracking is based on the application of orthogonal electromagnetic fields [3, 4, 5]. The system consists of a magnetic field transmitter and a receiver coupled via driving circuits. The electromagnetic transmitter contains 3 orthogonal coils that are pulsed in a sequence and the receiver has 3 coils that measure the field generated by the transmitter. The strength of the received signals is compared to the strength of the sent pulses to determine the position and compared to each other to determine the orientation.

To date two varieties of electromagnetic position trackers have been implemented: one implementation uses altering current (AC) to generate the magnetic field, the other uses direct current (DC). In an AC system, a current is sent to the emitter coils in a sequence so that 3 mutually perpendicular magnetic fields are generated. The field induces currents in the receiver that also consists of 3 passive mutually perpendicular coils. Sensor location and orientation therefore are computed from the 9 induced currents by calculating the small changes in the sensed coordinates and then updating the previous measurements [3, 4]. Carrier frequencies are typically in the 7 to 14 kHz range. The excitation pattern and processing are repeated typically at 30 to 120 Hz rates.

In contrast to the continuous wave generated by the AC systems, DC systems emit a sequence of DC pulses, which is in effect equals to switching the transmitter on and off [5]. This design is intended to reduce the effect of the field distortion due to the eddy currents induced in nearby metals when the field is changing. The initial measurements are performed with all 3 antennas shut off so that (x, y, z) components of the Earth's magnetic field are measured. Next, each transmitter coil is pulsed in a sequence and the induced current is recorded on each receiving coil after a short delay allowing the eddy currents to die out. Earth's magnetic field components are then subtracted from the 9 measured values generated in each receiver coil by each pulse and the resulting 9 measured values are then used to compute the location and orientation of the receiver relative to the transmitter.

Polhemus, Inc. and Ascension Technology Corporation are the major manufacturers of electromagnetic tracking systems that are currently in use in VR applications. Polhemus manufactures two varieties of AC-based VR application-oriented systems: 3Space Isotrak and 3Space Fastrak. The main difference between them is the number of sensors the systems can support. Fastrak is more powerful with the operational range up to 10 feet, but operation over a range of up to 30 feet is possible with the optional long range transmitter. 3Space Fastrak operates at 120 Hz achieving 0.06 mm location tracking and 0.025° orientation tracking resolution with 0.762 mm RMS location tracking and 0.15° RMS orientation tracking accuracy at 30.5 cm transmitter-receiver separation distance [6]. Ascension Technology manufactures several DC-based tracking systems including Flock of Birds (FoB), SpacePad, and pcBIRD. Flock of Birds links several processing units into one system allowing simultaneously track several receivers and is usually used in CAVE-like installations whereas SpacePad and pcBIRD are designed to operate within much smaller volumes. Flock of Birds operates at 144 Hz achieving 0.5 mm location tracking and 0.1° orientation tracking resolution with 1.8 mm RMS location tracking and 0.5° RMS orientation tracking accuracy at 30.5 cm transmitter-receiver separation distance [7].

2.2 Tracking errors and their sources

Due to the dependence of the measurements on the local electromagnetic field, the tracking systems are sensitive to the ambient electromagnetic environment. If there is a metal, or other conductive materials, or equipment that produces an electromagnetic field near the tracker's transmitter or receiver, the transmitter signals are distorted and the resulting measurements contain both *static* and *dynamic errors*. Dynamic errors are those that change over time and they are mainly due to the changing external electromagnetic fields. Static errors are those that remain constant over time. They are mainly caused by the electromagnetic filed distortions due to the surrounding metal.

M. Nixon et al. [8] conducted a study of the effects of metals and interfering fields on electromagnetic trackers. In one experiment, the effects of interfering fields present in a typical working environment were investigated. It was found that if the external fields are predominantly due to mains reticulation, the FoB is affected more severely than the Fastrak. Synchronizing Fastrak to the monitor frequency reduces the monitor interference more efficiently than when synchronized to the mains frequency. In the case of FoB, mains-synchronization should be used. Alternatively, setting the sampling rate at twice the nominal mains frequency and filtering the output may do the trick. In another experiment, noise measurements were made at varying transmitter-received distance with the sampling frequency synchronized to twice the mains frequency and an additional filtering. The results show that the error in the measured location is proportional to the 4th power of the distance from the receiver to the transmitter. Although the magnitude of the error for different systems is quite different, the relationship remains the same.

There are two metal-related phenomena that have some influence on the performance of electromagnetic trackers: eddy currents and ferromagnetism. Eddy currents are induced in metals by a changing magnetic field. AC-based trackers therefore induce eddy currents in nearby metals throughout each measurement period. DC-based trackers were developed in an attempt to eliminate this phenomenon. Although eddy currents are still induced by the rising or falling edges of the DC pulse, delaying the field measurements until some time after the rising edge allows the eddy currents to decay significantly. Fastrak is more sensitive to eddy currents and since all metals are conductors and produce eddy currents, the Fastrak is affected by all metals. The FoB is relatively insensitive to eddy currents induced in all metals with the exception of copper. The effects of stainless steel, brass and aluminum were found to be so insignificant that they could not be measured [8].

When the transmitter's field is applied, ferromagnetic materials produce a magnetization field the magnitude of which depends on the permeability of the material and the frequency of the electromagnetic field generator. Therefore ferromagnetic materials will affect both AC and DC systems. However, the trackers use different excitation frequencies and therefore the effects are different for different systems. With Fastrak, the permeability of steel is low and consequently the ferromagnetic effects are low too, but steel is a good conductor, therefore eddy current effects are predominant. With FoB, the eddy current effects of steel are low, but the permeability of steel at DC frequency is high. Therefore, ferromagnetic effects predominate. The same is true for ferrite, a material with low conductivity but high permeability for DC-generated field.

Amount of metal present in the working environment is a factor too. It was found [8] that the location error in the case of Fastrak increases proportionally to the 3rd power of the size of the metal plate and to the 2nd power in the case of FoB. However, the location error is undetectable in FoB for the aluminum plates.

To conclude, interfering fields and metals present in the operating environment are clearly a significant factor that affects the performance of any electromagnetic tracking system.

2.3 Tracker calibration

Synchronizing the tracker's sampling rate with the mains frequency or with the monitor frequency and applying additional filtering typically provides a sufficient compensation for the dynamic errors. These options are implemented in trackers hardware and therefore are not the subject of this survey. Static errors, the removal of which is the subject of this review, however are unique for each installation and their correction requires a prior knowledge of the shape of the field distortion.

One way to overcome static errors in electromagnetic tracking systems is to compensate the measurements for the errors through experimentally established dependencies between the actual receiver location/orientation and that reported by the tracking system. This procedure is commonly referred to as *tracker calibration*. Assuming that the transmitter's position is fixed and the surrounding metal does not move, the static error is a function of the position of the receiver and it can be corrected as long as the magnetic field does not "fold back" on itself. The procedure consists of experimentally obtaining a *calibration table* that contains *true* positions of the tracker sensor and the corresponding *tracked* positions at some limited number of locations within the system's operational space and, based on this limited set, deriving numerically a field distortion function. Several analytical techniques for computing a field distortion function have been proposed, their detailed overview is included below. Once the function is known, it can be used to compute the position (location and/or orientation) errors at any tracked location within the system's range of operation. This error then can be subtracted from the tracked values so that true undistorted position is found.

3 Location error correction methods

Assuming that the location error is the function of the tracked location only, the static location error correction problem can be described as the interpolation problem in 3D space [9]: given two limited sets of points $\{p_i\} \subset P \subset \mathbf{R}^3$ and $\{q_i\} \subset Q \subset \mathbf{R}^3$, $i = 1,...,n$, one is looking for a function $f : P \rightarrow Q$ such that $f(p_i) = q_i$ for $\forall i = 1,...,n$. This function should be at least C^1-continuous, there should be no oscillations, and ideally it should not depend on a certain topology of the data to be interpolated so that $\forall p \in P$ can be uniquely mapped into Q. Here $\{p_i\}$ denotes a known set of *tracked points* in tracked (distorted) space P and $\{q_i\}$ denotes a known set of *true points* in true (undistorted) space. Their coordinates are denoted as $p_i = (x_i^p, y_i^p, z_i^p)$ and $q_i = (x_i^q, y_i^q, z_i^q)$.

F. Raab et al. [4] suggested that correction of the distorted location measurements can take the form of additive vectors and can be stored in either a look-up table (LUT) or as polynomials in the location parameters. Much of the work done after [4] implements a variation of one of these two methods. All the location error correction techniques can be classified into two categories: *global methods* that take all points into account to derive an appropriate global mapping function, and *local methods* that use only some points to obtain a localized solution.

3.1 Tri-linear interpolation

M. Czernuszenko et al. [10], M. Ghazisaedy et al. [11], and M. Livingston et al. [12] used local error correction technique based on the tri-linear interpolation.

Tri-linear interpolation provides a unique mapping from any point inside a cube into a point inside an arbitrary polyhedron. Suppose, we place a tracker sensor at 8 regularly spaced locations $\{p_1,..., p_8\}$ that form a cube in the tracked space and determine the true coordinates $\{q_1,..., q_8\}$. Then, for any point p inside the cube we have [13]:

$$\begin{aligned} q = {} & (1-t)(1-u)(1-w)q_1 + t(1-u)(1-w)q_2 + \\ & (1-t)u(1-w)q_4 + tu(1-w)q_3 + (1-t)(1-u)wq_5 + \\ & t(1-u)wq_6 + (1-t)uwq_8 + tuwq_7 \end{aligned} \tag{1}$$

where

$$u = \frac{y^p - y_1^p}{y_4^p - y_1^p}\ t = \frac{x^p - x_1^p}{x_2^p - x_1^p},\ w = \frac{z^p - z_1^p}{z_5^p - z_1^p} \tag{2}$$

In order for this technique to work, the entire tracked volume to be calibrated has to be subdivided into small non-intersecting cubes (3D grid) whose vertices form sets $\{p_i\}_j$, $i = 1,...,8$, $j = 1,...,m$, and m is the total number of cubes. For each such cube, a corresponding set $\{q_i\}_j$ has to be known. For any point p in the tracked space, its surrounding cube $\{p_i\}_j$ can be located via LUT. It is then used to map p into the true space via equation (1). Mapping inside each individual cube is C^1-continuous, however the gradient of the interpolated function changes discontinuously at the boundaries of each grid square.

Perhaps the most severe limitation of the tri-linear interpolation approach is due to the requirements imposed on the topology of the calibration data set: each $\{p_i\}_j$ has to define a cube in the tracked space. The opposite mapping requires inverting (1), which contains non-linear products and therefore is typically solved numerically.

3.2 Shape functions

Tri-linear interpolation the equation of which can be easily derived analytically belongs to a brad class of so-called *shape functions* whose equations can only be obtained numerically. G. Zachmann [14] discusses a possibility of using shape functions for mapping an arbitrary hexahedron in the tracked space into a corresponding cube in the true space (in essence, the inverse of tri-linear interpolation). Although, he did not test this approach in practice, his theoretical framework is of interest for this review and therefore is discussed here.

Suppose a point $p \in P$ is located inside a hexahedron defined by $\{p_i\}$. The corresponding point $q \in Q$ is bounded by a cube defined by $\{q_i\}$. The shape function that maps any point p bounded by $\{p_i\}$ into a point q bounded by $\{q_i\}$ is defined as [14]:

$$q = f(p) = \sum_{i=1}^{8} f_i(p) q_i \tag{3}$$

Linear shape functions are considered:

$$f_i(p) = f_i(x, y, z) = f_{i1} + f_{i2}x + f_{i3}y + f_{i4}z + f_{i5}xy + f_{i6}xz + f_{i7}yz + f_{i8}xyz \tag{4}$$

where f_{ik} are the shape function coefficients. So, $f(p)$ is described by 8x8 matrix of coefficients. If we define $f_i(p_l)$ as a Kronecker δ-function, $f_i(p_l) = \delta(i,l)$, then $f(p_i) = q_i$. So, the coefficients f_{ik} can be obtained by solving 8 sets of linear equations $\mathbf{A}\mathbf{f}_i = \mathbf{e}_i$ where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_8]^{\mathrm{T}}$, $\mathbf{a}_i = [1, x_i^p, y_i^p, z_i^p, x_i^p y_i^p, x_i^p z_i^p, y_i^p z_i^p, x_i^p y_i^p z_i^p]$, $\mathbf{e}_i = [\delta(i,1), ..., \delta(i,8)]^{\mathrm{T}}$, and unknown coefficients $\mathbf{f}_i = [f_{i1}, ..., f_{i8}]^{\mathrm{T}}$.

S. Ellis [15] used local error correction technique based on a simpler form of shape function derived for tetrahedrons. His work is based on the solution found by D. Kenwright [16] who did not use the shape functions framework. However, as show below, Kenwright's analytically derived equation (7) is the shape function for an arbitrary tetrahedron.

Suppose $\{p_i\}$, $i = 1, ..., 4$, defines an arbitrary tetrahedron in the tracked space and $\{q_i\}$ defines a unit tetrahedron $\mathrm{T}^1 = (x, y, z) | x \geq 0,\ y \geq 0,\ z \geq 0,\ x + y + z \leq 1$ in the true space. Shape function for the unit tetrahedron is trivial [14]:

$$f_1(p) = x^p,\ f_2(p) = y^p,\ f_3(p) = z^p,\ f_4(p) = 1 - (x^p + y^p + z^p) \tag{5}$$

Kenwright's tetrahedral linear interpolation function that he uses to map from "natural" to "physical" coordinates can be obtained from (3) and (5):

$$f(p) = \sum_{i=1}^{4} f_i(p) = q_4 + (q_1 - q_4)x^p + (q_2 - q_4)y^p + (q_3 - q_4)z^p \tag{6}$$

Shape function for an arbitrary tetrahedron T can be obtained by applying the corresponding affine transformation that transforms T^1 into T. Kenwright instead analytically inverts (6) [16]:

$$[q]^{\mathrm{T}} = \frac{1}{V}\mathbf{A}[p - p_4]^{\mathrm{T}} \tag{7}$$

where $\mathbf{A}$ is the affine transformation matrix as defined by Zachmann [14] and V is its determinant.

In order for this technique to work, the entire true space that covers the tracked volume to be calibrated has to be split into unit-sized tetrahedrons $\{q_i\}_j$ and their corresponding tetrahedrons $\{p_i\}_j$ in the tracked space have to be known. For $\forall p \in P$, its surrounding tetrahedron $\{p_i\}_j$ is located via LUT. Then p can be mapped into the true space via equation (7). As with tri-linear interpolation, mapping inside each individual tetrahedron is C^1-continuous, however the gradient of the interpolated function changes discontinuously at the boundaries of each grid tetrahedron.

3.3 Other local interpolation schemes

S. Bryson [17], W. Briggs [18], and V. Kindratenko et al. [19] used local error correction techniques based on the localized interpolation from surrounding points.

Suppose, the tracker sensor is moved between 8 locations $\{q_1,...,q_8\}$ that form a cube in the true space and the corresponding tracked locations $\{p_1,...,p_8\}$ are recorded. They form a polyhedron in the tracked space. For each point p_i, the location error vector is computed as $v_i = q_i - p_i$. Then, for any point p inside the polyhedron defined by $\{p_1,...,p_8\}$, we have [17, 19]:

$$q = p + \sum_{i=1}^{8} w_i v_i \tag{8}$$

where w_i are normalized weights associated with each point so that $w_1 + ... + w_8 = 1$:

$$w_i = \frac{r_i}{\sum_{i=1}^{8} r_i} \tag{9}$$

Note that when $r_1 = ... = r_8$, the weighted sum of v_i in (8) equals to the average of all 8 error vectors.

S. Bryson [17] used two different weight functions: *linear* that defines each weight as a function of the distance and *bump* that defines each weight as an exponentially decaying function of distance between p and p_i. The exact equations are:

$$r_i = \prod_{j=1, j\neq i}^{8} d_j \text{ and } r_i = \begin{cases} e^{-d_i} & \mathit{if}\ d_i < d_{\max} \\ 0 & \mathit{if}\ d_i \geq d_{\max} \end{cases} \tag{10}$$

where d_i is the distance between p and p_i and $d_{\max}$ is a constant representing the distance for which weight r_i vanishes; it should be smaller than the smallest distance between any two points in the given $\{p_1,..., p\}_8$ [17].

V. Kindratenko et al. [19] used 3 different weight functions:

$$r_i = \begin{cases} 1 - \dfrac{d_i}{d_{\max}} & if\ d_i < d_{\max} \\ 0 & f\ d_i \geq d_{\max} \end{cases}, \qquad r_i = \begin{cases} 1 - \dfrac{d_i^2}{d_{\max}^2} & if\ d_i < d_{\max} \\ 0 & f\ d_i \geq d_{\max} \end{cases}, \text{ and}$$
$$r_i = \begin{cases} -\ln \dfrac{d_i}{d_{\max}} & if\ d_i < d_{\max} \\ 0 & f\ d_i \geq d_{\max} \end{cases} \tag{11}$$

I his case $d_{\max}$ equals to the half of the longest diagonal of the polyhedron.

W. Briggs [18] used a weight function based on the squared distance. His approach however differs in the way $\{q_i\}$ and $\{p_i\}$ are obtained. Thus, $\{p_i\}$ forms a cube in the tracked space and $\{q_i\}$ then defines a corresponding polyhedron in the true space. For each point p_i, the location error vector is computed as $v_i = p_i - q_i$. The final error correction equation is:

$$q = p - \sum_{i=1}^{8} w_i v_i \tag{12}$$

where w_i is the (supposedly normalized) weight function defined as [18]:

$$w_i = \frac{1}{d_i^2} \sum_{i=1}^{8} d_i^2 \tag{13}$$

where d_i is the distance between p and p_i. The author probably made a mistake in reporting the final set of equations because the above weight normalization equation does not guarantee that $w_1 + ... + w_8 = 1$ and therefore v_i is not properly scaled resulting in larger error estimation.

The method described by W. Briggs requires the topology of $\{p_i\}_j$ and $\{q_i\}_j$ be similar to the one used for tri-linear interpolation. Two other techniques however require the opposite: the entire true space that covers the tracked volume has to be subdivided into small non-intersecting cubes whose vertices form sets $\{q_i\}_j$. For each such set, the corresponding set $\{p_i\}_j$ has to be known. For $\forall p \in P$, its surrounding polyhedron (or cube) $\{p_i\}_j$ is located via LUT. Then p can be mapped into the true space using equations (8) or (12). The advantage of this type of interpolation is that less strict requirements can be imposed on the topology of $\{p_i\}_j$ and $\{q_i\}_j$, in fact, it can be generalized so that $\{p_i\}_j$ and $\{q_i\}_j$ can have any shape as long as they provide total coverage of the tracked volume and are non-intersecting. The interpolation quality however is sensitive to the choice of $d_{\max}$. Although the interpolation is C^1-continuous within each individual polyhedron, there is a possibility of a sadden jump when $d_i = d_{\max}$ because the contribution of v_i vanishes completely. Consequently, the gradient of the interpolated function may change discontinuously even inside each grid cell.

A common feature of tri-linear, tetrahedral, and local interpolation schemes is that they all require the tracked space to be subdivided into small subspaces of a certain size and therefore their quality is dependent on this size. Subdivision into large volumes can result in a lower overall calibration quality, smaller volumes require more measurements to be taken.

3.4 High-order polynomial fit

S. Bryson [17], V. Kindratenko [20], and M. Ikits et al. [21] used global error correction technique based on the high-order polynomial fit.

Suppose the sensor is moved on a rectilinear 3D grid $\{q_i\}$ in the true space Q and the corresponding tracked locations are stored as $\{p_i\}$ in the tracked space P. Here $i = 1,...,n$ where n is the total number of grid nodes where the measurements were taken. Location error vector v_i at the tracked location p_i can be found as $v_i = p_i - q_i$. A degree r vector polynomial of location $p \in P$ that fits the location error v can be formulated as [20, 21]:

$$v = f(p) = f(x, y, z) = \sum_{j=1}^{R} \begin{bmatrix} c_{xj} \\ c_{yj} \\ c_{zj} \end{bmatrix} x^{s_j} y^{t_j} z^{u_j} \tag{14}$$

where c_{xj}, c_{yj}, and c_{zj} are the polynomial coefficients, R is the number of terms in the polynomial, $R = (r+1)(r+2)(r+3)/6$, and s_j, t_j, and u_j are the nonnegative powers such that $0 \le s_j + t_j + u_j \le r$ and all permutations of $\{s_j, t_j, u_j\}$ are unique. Unknown coefficients c_{xj}, c_{yj}, and c_{zj} should be such that the fitted polynomial closely approximates the location error at $\{p_i\}$. They can be found by minimizing the following objective function [21]:

$$\sum_{i=1}^{n} \left\| v_i - f(p_i) \right\|^2 \tag{14}$$

which is a well-known least-square fit problem the solution of which can be found by solving the following simultaneous equations [20]:

$$\sum_{j=1}^{R} \begin{bmatrix} c_{xj} \\ c_{yj} \\ c_{zj} \end{bmatrix} \sum_{i=1}^{n} \left(x_i^{s_j} y_i^{t_j} z_i^{u_j} \right) \left(x_i^{s_k} y_i^{t_k} z_i^{u_k} \right) = \sum_{i=1}^{n} v_i \left(x_i^{s_k} y_i^{t_k} z_i^{u_k} \right) \tag{15}$$

for $k = 1,...,R$ and $n > R$.

The advantage of using high-order polynomials is that they provide C^1-continuous mapping over the entire space and are in fact R times differentiable, thus allowing for smooth interpolation. Error correction outside the pre-measured tracked space is also possible whereas all the interpolation schemes are ill-conditioned to extrapolation. Although all current implementations are based on a certain topology of $\{q_i\}$ and $\{p_i\}$, a non-uniformly scattered data can be used as well. Polynomial of the order 3 or 4 are typically most suitable for tracker calibration [20, 21], higher order typically results in undesirable oscillations. It has been noticed [20] that high-order polynomial fit best corrects large errors and occasionally introduces small additional errors when the original error is already small.

3.5 Hardy's multi-quadric method

G. Zachmann [9, 14] developed a global error correction technique based on Hardy's Multi-Quadrick (HMQ) scattered data interpolation method.

Suppose the tracker sensor is moved on a rectilinear 3D grid $\{q_i\}$ in the true space Q and the corresponding tracked locations in the tracked space P are stored as $\{p_i\}$. Here $i = 1,...,n$ where n is the total number of grid nodes where the measurements are taken. The general form of the HMQ interpolation function $f : \mathbf{R}^3 \to \mathbf{R}^3$ is [9]:

$$f(p) = \sum_{j=1}^{n} \alpha_j w_j(p) \tag{16}$$

where basis functions $w_i(p)$ are defined as

$$w_j(p) = \sqrt{\left\| p - p_j \right\|^2 + R^2} \tag{17}$$

Unknown expansion coefficients α_j can be found by solving the system of n simultaneous equations

$$\sum_{j=1}^{n} \alpha_j w_j(p_i) = q_i \tag{18}$$

using LU decomposition technique [13]. The smoothness of the interpolation depends on the parameter R^2. It has been found experimentally that $10 \le R^2 \le 1000$ is optimal and the exact value of R^2 within this range has a very little impact on the interpolation quality [9].

The advantage of this approach is that it provides C^1-continuous mapping and allows for smooth interpolation within the entire tracked space. Error correction outside the pre-measured tracked space is also possible. Although Zachmann's implementation requires a certain topology of $\{q_i\}$ and $\{p_i\}$, a non-uniformly scattered data can be used as well.

3.6 Neural network-based method

T. Saleh et al. [22] used global error correction technique based on the application of a multi-layer feed-forward neural network.

Suppose the tracker sensor is moved on a rectilinear 3D grid $\{q_i\}$ in the true space Q and the corresponding tracked locations in the tracked space P are stored as $\{p_i\}$. Here $i = 1,...,n$ where n is the total number of grid nodes where the measurements are taken. Sets $\{p_i\}$ and $\{q_i\}$ are used to train a neural network which is set up in such a way that once it is trained, it maps $\forall p \in P$ directly into $q \in Q$, thus eliminating the location error in p.

T. Saleh et al. [22] used a four-layer feed-forward network architecture with 3 input nodes and 3 output nodes, all using linear activation function $f(s) = s$. The first hidden layer contains 7-8 nodes, the second hidden layer contains 4 nodes, all using the sigmoidal activation function $f(s) = \left(1 + e^{-s}\right)^{-1}$. A bias unit is included with no incoming connections, and is fed into each hidden and output node. The justification for this particular choice of the network architecture is that “any well behaved function that maps from *n* ($n > 1$) dimensions to *m* dimensions can be approximated by a 2-hidden layer neural network” [22]. Although there is some discussion provided on the choice of the number of nodes in the hidden layers, it seems to be mostly based on experiments. Standard back-propagation algorithm is used to train the network until the average of the network output error for the entire training set falls below 0.01. The learning rate used to restrain the changes in weights during training was determined experimentally to be 0.01-0.1. The use of momentum, or the consideration of the previous value by a constant when updating a weight, is also included.

Neural network provides C^1-continuous smooth mapping within the entire tracked space. Error correction outside the pre-measured tracked space is also possible. Although the current implementation is based on a certain topology of $\{q_i\}$ and $\{p_i\}$, a non-uniformly scattered data can be used as well. It is not clear however if the used network architecture is the most appropriate one.

4 Orientation error correction methods

Assuming that the orientation error is the function of the tracked location only, the static orientation error correction problem can be solved using a modification of some of the location error correction techniques. There is only one study [12] that suggests that orientation error is the function of both the tracked location and orientation. However, more recent studies [19, 14], and [21] seem to be not able to confirm those findings.

Suppose the location of the tracker sensor is known on a rectilinear 3D grid $\{q_i\}$ in the true space Q and the corresponding locations and orientations in the tracked space P, denoted as $\{p_i\}$ and $\{o_i\}$, are also known ($i = 1,...,n$ where n is the total number of grid nodes). It is difficult to measure the actual sensor’s orientation in the true space, however it is possible to insure that it has the same orientation as the transmitter. In such case, o_i is, in essence, the error in the tracked orientation at the given location p_i. Sensor’s orientation can be represented in many ways. So far, three forms have been used for calibration purposes: Euler angles [20], rotation matrix [14], and quaternions [12, 15, 19, 21].

4.1 Local interpolation

M. Livingston et al. [12], S. Ellis et al. [15], and V. Kindratenko et al. [19] used local orientation error correction techniques based on some form of a localized interpolation scheme similar to the ones used for correcting location error. Very little information however is provided about the details of their implementations.

M. Livingston et al. [12]: “Position correction is performed by tri-linear interpolation of the error vectors for the corners of the cell into which the sample falls. Orientation correction is performed by an analogous interpolation of the error quaternions for the corners of the cell. We reduced the tri-linear problem to a sequence of simple linear interpolations, each between two quaternions; we used spherical linear interpolations.”

S. Ellis et al. [15]: “… error in orientation was measured by the quaternion that would rotate the measured local vertical, approximately a surface normal to each quadrilateral patch, into true vertical. These orientation errors within each calibrated cell were corrected by inverse rotations based on error measurements at adjacent calibration grid nodes."

V. Kindratenko et al. [19] used the approach identical to the one described for the location error correction. Quaternionial representation $\{[w,(x,y,z)]_i\}$ of $\{o_i\}$ was used. Instead of interpolating the location error vector v at the tracked location p , the orientation error quaternion $[w^e,(x^e,y^e,z^e)]$ is interpolated via equations (8), (9), and (11). Weights w_i are still based on the distance d_i , however they are used to scale the contribution of the orientation error quaternions rather than the location error vectors. The resulting error quaternion is normalized. Tracked quaternion $[w,(x,y,z)]$ is then multiplied by the inverse of the interpolated quaternion to obtain the true orientation of the sensor.

All three implementations impose certain topology limitations on $\{q_i\}$. C^1-continuous mapping from the tracked orientation into the true orientation is provided, but the gradient of the interpolated function may change discontinuously even inside each grid cell.

4.2 High-order polynomial fit

V. Kindratenko [20] and M. Ikits et al. [21] used global orientation error correction technique based on the high-order polynomial fit.

V. Kindratenko [20] used Euler angles representation $\{(\alpha,\beta,\gamma)_i\}$ for $\{o_i\}$. The polynomial fit equation (14) remains the same except that now it is used to fit the errors in tracked Euler angles instead of fitting the errors in tracked location p . Once its coefficients are known, orientation error $(\alpha^e,\beta^e,\gamma^e)$ at $\forall p\in P$ can be interpolated. This error however cannot be simply added to the tracked orientation because of the nature of the Euler angles. Instead, the tracked error should be rotated by the amount of interpolated rotation error. This is achieved by converting both the tracked orientation (α,β,γ) and the fitted orientation error $(\alpha^e,\beta^e,\gamma^e)$ into rotation matrixes $\mathbf{M}$ and $\mathbf{M}^e$, multiplying $\mathbf{M}$ by $\mathbf{M}^e$, and converting the resulting matrix back to the Euler angles.

M. Ikits et al. [21] used quaternionial representation $\{[w,(x,y,z)]_i\}$ for $\{o_i\}$. The polynomial fit equation (14) remains the same except that now it is used to fit the errors in the tracked vector part (x^e,y^e,z^e) of the error quaternion $[w^e,(x^e,y^e,z^e)]$ instead of fitting the errors in the tracked location p . The scalar part w^e of the error quaternion $[w^e,(x^e,y^e,z^e)]$ is then obtained from the constraint of unity: $\|[w^e,(x^e,y^e,z^e)]\|^2=1$. The corrected orientation quaternion can be obtained by multiplying the inverse of the fitted error quaternion by tracked quaternion $[w,(x,y,z)]$: $[w^e,(x^e,y^e,z^e)]^{-1}[w,(x,y,z)]$.

I both cases, a C^1-continuous smooth mapping from tracked orientation into true orientation is achieved. Although both implementations impose certain limitations on the topology of $\{q_i\}$ and require the censor to have 0 orientation in the true space, a non-uniformly scattered data at an arbitrary known orientation can also be used.

4.3 Hardy's multi-quadric method

G. Zachmann [14] used global orientation error correction technique based on the Hardy's multi-quadric interpolation. Orientation matrix representation $\{\mathbf{M}_i\}$ is used for $\{o_i\}$. Expansion coefficients α_i in (16) are computed from constrains $f\left(\mathbf{M}_i^p\right) = \mathbf{M}_i^q$. The resulting HMQ interpolation function is then used to interpolate the true orientation $\mathbf{M}^q$ from the corresponding tracked orientation $\mathbf{M}^p$. Of course, $\mathbf{M}^q$ has to be orthogonalized and normalized.

Just as in the polynomial fit, a C^1-continuous smooth mapping from the tracked orientation into the true orientation is achieved. Although Zachmann 's implementation imposes certain limitations on the topology of $\{q_i\}$ and require the censor to have 0 orientation in the true space, a non-uniformly scattered data at an arbitrary known orientation can also be used.

5 Calibration data acquisition techniques

All the calibration models, or at least their current implementations, impose certain restrictions on the topology of $\{p_i\}$ and $\{q_i\}$: the measurements have to be taken on the rectilinear grid either in the true space or in the tracked space. This requires an alternative tracking system, preferably inexpensive as compared to the electromagnetic tracking system to be calibrated. Several such systems and data acquisition approaches have been proposed. In one approach, the tracker sensor is moved on a rectilinear 3D grid with know node coordinates in the true space and the corresponding tracked coordinates are recorded. In another approach, the sensor is moved on the rectilinear 3D grid in the tracked space and the corresponding true coordinates of the tracker are measured and stored. Finally, the last approach is to move the sensor on an irregular random grid in either space, determine both its true and tracked coordinates at each location, and then re-sample them into a desirable topology.

S. Bryson [17]: "A standard 6'x4' pegboard on a stand was constructed. This pegboard was used to determine a set of actual positions for the sensor. The floor … was marked with positions for the pegboard so that a regular 8'x8'x6' volume could be measured. Certain holes were marked on the pegboard at 12" intervals as the measurement sights. At each measurement sight 60 measurements of the tracker were taken with the sensor at a fixed orientation. … The 60 measurements were then used to compute an average position for the time of the measurement and a standard deviation over that time." Although this technique is relatively accurate, it is time consuming. Also, an arbitrary orientation tracking is not possible.

V. Kindratenko [20], V. Kindratenko et al. [19], and T. Saleh et al. [22] used a sensor holder consisting of a 1'x1'x0.1' wooden platform with housing attached to the top and a set of plastic pipes of length 2, 3, 4, 5, 6, and 7 feet that can be plugged into the housing. Rectilinear 2D grid markers were placed on the floor. Moving the platform between the markers on the floor and changing the pipes allows to place the tracker sensor at the points whose coordinates can be precisely determined in the true space. After a careful alignment, the precision of this measuring technique is on the order of ±1 cm. Although this technique is relatively accurate, it is time consuming and an arbitrary orientation tracking is not possible.

W. Briggs [18]: "The samples were gathered with a jig made of non-ferrous PVC tubing constructed with a 1 foot square wooden base for discrete motion. … Multiple sensors were attached vertically to the jig and sampled simultaneously." Since Briggs's calibration technique requires the data to be available on the rectilinear grid in the tracked space, but his data acquisition technique produces just the opposite, an additional data re-sampling into rectilinear 3D grid in the tracked space was applied. For each point on the grid, its surrounding tetrahedron of the smallest volume is build from the measured points. Since the coordinates of the tracked points are known both in the true and tracked space, their location error vectors can be computed. The error vectors at the regularly spaced points in the tracked space are then interpolated using equation (12) and the found tetrahedrons. This of course introduces additional errors into the final calibration table. Also, an arbitrary orientation tracking is not possible.

G. Zachmann [9, 14] used similar device consisting of a wooden platform with a long wooden bar attached vertically. Several sensors can be attached to the bar at well-defined locations. Moving the bar on the floor with markers allows to position the sensors at known locations in the true space. This technique is both relatively accurate and is considerably faster as compared to [17] and [20]. An arbitrary orientation tracking however is not possible.

M. Livingston et al. [12] used a mechanical tracking system Faro Metrecom IND-1 (Faro arm) to determine the precise location and orientation of the electromagnetic tracker sensor in the true space. The measurements were taken on an irregular grid and re-sampled into a rectilinear grid in the tracked space. The re-sampling process uses a Gaussian kernel centered at each grid node and a cut-off radius equal to the tripled grid size. This of course introduces additional errors into the final calibration table. Sampling with Faro arm can be very fast and accurate and an arbitrary orientation tracking is possible, but its cost can be prohibitively high for many users.

M. Ghazisaedy et al. [11] used an ultrasonic measuring device (UMD) as the means of determining the sensor's location in the true space. UMD generates an ultrasonic sound signal and sends it towards an object. The sound reflected from the object is detected and the distance is computed from the time difference between when the signal was received and send. Four UMDs were combined in order to determine 3 DoF coordinates. The electromagnetic tracker sensor was attached to UMDs and the measurements were collected moving the sensor on the regular grid in the tracked space. This of course required a specialized software to guide the data acquisition process. Although the orientation tracking is not possible, the technique is relatively inexpensive, allows for a reasonably good precision, and fast. Its major drawback however that the tracked space must be surrounded by walls so that ultrasound can be bounced back, therefore it is only suitable for CAVE-like installations.

M. Ikits et al. [21] used NDI Optotrack 3020 optical tracking system as the means of tracking sensor's position in the true space. The system uses active infrared markers and specialized video equipment to determine marker's position. Electromagnetic tracker sensor and the infrared markers are attached to a flexible platform that can be moved around and oriented to an arbitrary angle. Sampling both systems at the same time allows to determine both the location and (limited) orientation of the sensor in the true and tracked space. Sampling with Optotrack is not as fast as with Faro arm, but it is certainly faster and more accurate that with the pegboard or jig. Limited orientation tracking is possible, but the cost of this device can be prohibitively high for many users.

The most unusual approach so far has been proposed by M. Czernuszenko et al. [10]. Its idea is based on superimposing real and virtual objects at the same location and visually determining the amount of displacement between them as a function of user's head location. Data acquisition procedure starts with placing a real object in the scene at known location and rendering a 3D virtual object at the same location. User wearing tracked stereo glasses moves to an arbitrary location p'_j in the environment and looks at the objects. If there would be no error in the head tracking, the objects would coincide. However they usually do not because of the error and the distance between them equals to the amount of the error at current user's head location. The user then uses a wand joystick to apply some translation v'_j to the virtual object until it coincides with the real one. Thus, error vector v'_j at the location p'_j in the tracked space is found. This procedure is repeated at m locations resulting in a non-uniformly scattered calibration data set. This data then is used to interpolate a set of the location error vectors $\{v_i\}$ at the rectilinear grid $\{p_i\}$ in the tracked space:

$$v_i = \sum_{j=1}^{m} \frac{w_{ij}}{\sum_{k=1}^{m} w_{ik}} v'_j \quad \text{where} \quad w_{ij} = \frac{1}{\left\| p_i - p'_j \right\|^2} \tag{19}$$

This technique of course requires a specialized software to guide the calibration process and 3D display environment. Its advantages are: fast calibration and low equipment cost. However the accuracy is limited and orientation correction is not available, although it can be added.

6 Software implementations

Although a number of tracker calibration techniques have been reported in the literature, very little has been done in making them readily available to the research community. Majority of the implementations are proprietary and are tightly integrated with specific systems. Up to date, only 4 software implementations have been made available to the research community.

The first software to offer tracker calibration capabilities was CAVElib [2]. The implementation is based on the tri-linear interpolation approach [10, 11], therefore only the location error correction is possible. No direct API calls to the calibration routine are provided; it can be used only as a part of CAVElib application. Because of the difficulties involved in acquiring calibration data suitable for CAVElib's implementation (rectilinear grid in the tracked space), it is seldom used. Source code is not available.

UNC Magnetic Tracker Calibration code [12] is available in the form of the source code freely downloadable from http://www.cs.unc.edu/~us/MagTrack/data.html. Tri-linear interpolation for the location error correction and quaternionial interpolation for the orientation error correction are the only techniques implemented. No data acquisition software is provided.

NCSA libTrCalibr ver. 2.00 [20, 19] is available in the form of the source code freely downloadable from http://www.ncsa.uiuc.edu/VEG/VPS/emtc/libtrcalib.html. It implements high-order polynomial fit and a local interpolation technique for both the location and orientation error correction. The calibration routines are organized in a library, supporting software for data acquisition, analysis, and visualization is included as well. The library has been tested on IRIX, Linux, and Win32 platforms.

Releases 4.0 and 5.0 of VRCO tracker daemon software (VRCO, Inc., www.vrco.com) have some provisions for integrating calibration routines with the tracker daemon server. Tracker daemon provides a very flexible framework for integration various input devices with VR applications including a variety of position tracking systems. A third-party shared object containing tracker calibration routines can be loaded by the tracker daemon server during the initialization and executed each time new tracker data is sampled. NCSA trackdCalibrator software is an example of such a third-party object. Its source code and binaries are available at http://www.ncsa.uiuc.edu/VEG/VPS/emtc/trackd.html.

7 Conclusions

Table 1 contains a short summary of all the tracker calibration articles surveyed. Table 2 contains a summary of the different error correction techniques.

Analysis of the mathematical models and the numerical results of existing methods suggests that global position error correction techniques, such as high-order polynomial fit or Hardy's multi-quadric method, are expected to perform best allowing smooth continuous mapping from the tracked space to the true space. Local interpolation methods may perform almost as good as global methods, but they do not necessarily provide smooth continuous mapping because the gradient of the interpolation function may change discontinuously. Also, orientation error correction is not always possible with local methods.

The most practical data acquisition technique is the one based on using a wooden multiple sensor holder, moving it on the rectangular grid in the true space, and recording the corresponding tracked values. Although the precision of this tracker holding device is not very high, it is certainly sufficient for the tracker calibration purpose. Re-sampling of the acquired data before it is used for the calibration is undesirable because it introduces additional errors.

VRCO trackd in combination with NCSA trackdCalibrator and libTrCalibr ver. 2.00 is the only software that provides a complete solution including data acquisition, calibration and analysis. Any application that uses VRCO trackd's services to access tracking data can effortlessly benefit from the calibrated tracking system.

Bibliography

[1] K. Meyer, H. Applewhite, and F. Biocca, A survey of position trackers, Presence, Vol. 1, No. 2, 1992, pp. 173-200.
[2] C. Cruz-Neira, D. Sandin, and T. DeFanti, Virtual Reality: The Design and Implementation of the CAVE, in Proceedings of SIGGRAPH '93 Computer Graphics Conference, ACM SIGGRAPH, August 1993, pp. 135-142.
[3] F. Raab, Remote Object Position Locater, US Patent 4054881, 1977.
[4] F. Raab, E. Blood, T. Steioner, and H. Jones, Magnetic position and orientation tracking system, IEEE Transactions on Aerospace and Electronic Systems, Vol. 15, No. 5, 1979, pp. 709-718.
[5] E. Blood, Device for quantitatively measuring the relative position and orientation of the bodies in the presence of metals utilizing direct current magnetic fields, US Patent

4945305, 1990.
[6] 3SPACE FASTRAK Manual, Polhemus Inc.
[7] The Flock of Birds Installation and Operation Guide, Ascension technology Corp.
[8] M. A. Nixon, B. C. McCallum, W. R. Fright, and N. B. Price, The Effects of Metals and Interfering Fields on Electromagnetic Trackers, Presence, Vol. 7, No. 2,· April 1998, pp. 204-218.
[9] G. Zachmann, Distortion Correction of Magnetic Fields for Position Tracking, in Proc. Computer Graphics International (CGI'97), Belgium, June 23-27 1997.
[10] M. Czernuszenko, D. Sandin, T. DeFanti, Line of Sight Method for Tracker Calibration in Projection-Based VR Systems, in Proceedings of 2nd International Immersive Projection Technology Workshop, Ames, Iowa, May 11-12, 1998.
[11] M. Ghazisaedy, D. Adamczyk, D. Sandin, Kenyon, and T. DeFanti, Ultrasonic Calibration of a Magnetic Tracker in a Virtual Reality Space, in Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS '95), Research Triangle Park, NC, March 1995.
[12] M. A. Livingston, and A. State, Magnetic Tracker Calibration for Improved Augmented Reality Registration, Presence, Vol. 6, No. 5,· October 1997, pp. 532-546.
[13] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in C, 2nd edition, Cambridge University Press, 1997.
[14] G. Zachmann, Virtual Reality in Assembly Simulation - Collision Detection, Simulation Algorithms, and Interaction Techniques. PhD Dissertation, TU Darmstadt, 2000.
[15] S. Ellis, Adelstein B, Baumeler S, Jense G, Jacoby R. Sensor spatial distortion, visual latency, and update rate effects on 3D tracking in virtual environments. In: Proceedings IEEE Virtual Reality (IEEE VR '99) Conference. Houston TX, March 1999; 218-221.
[16] D. Kenwright, A. Lane, Interactive time-dependent particle tracing using tetrahedral decomposition. IEEE Transactions on Visualization and Computer Graphics 1996; 2(2): 120-129.
[17] S. Bryson, Measurement and Calibration of Static Distortion of Position Data from 3D Trackers, in Proceedings of SPIE Conference on Stereoscopic Displays and Applications III, San Jose, CA, February 1992, pp. 244-255.
[18] W. Briggs, Magnetic Calibration by Tetrahedral Interpolation, in Proceedings of NIST-ASME Industrial Virtual Reality Symposium, Chicago, IL, November 1999, MH-Vol. 5/MED-Vol. 9, pp. 27-32.
[19] V. Kindratenko, and A. Bennett, Evaluation of Rotation Correction Techniques for Electromagnetic Position Tracking Systems, in Proc. Virtual Environments 2000 Eurographics Workshop.
[20] V. Kindratenko, Calibration of electromagnetic tracking devices, Virtual Reality: Research, Development, and Applications, Vol. 4, 1999, pp. 139-150.
[21] M. Ikits, J.D. Brederson, C. Hansen, and J. Hollerbach An Improved Calibration Framework for Electromagnetic Tracking Devices, in IEEE Virtual Reality 2001.
[22] Tammer Saleh, Volodymyr V. Kindratenko , and William R. Sherman, On Using Neural Networks to Calibrate Electromagnetic Tracking Systems, submitted to Virtual Reality: Research, Development, and Applications.

Author(s)	Calibration table topology	Calibration table measuring device	Location error correction approach	Orientation error correction approach	Form of rotation representation	Public domain software
S. Bryson [17]	rectilinear 3D grid in the true space	pegboard, 2D grid on the floor	4th order polynomial fit, local interpolation	N/A	N/A	no
M. Ghazisaedy et al. [11]	rectilinear 3D grid in the tracked space	ultrasound tracking device	tri-linear interpolation	N/A	N/A	no
M. Livingston et al. [12]	re-sampled rectilinear 3D grid in the tracked space	Faro arm	tri-linear interpolation	local interpolation	quaternions	source code
G. Zachmann [9, 14]	rectilinear 3D grid in the true space	wooden bar, 2D grid on the floor	Hardy's multi-quadric interpolation	Hardy's multi-quadric interpolation	rotation matrix	no
M. Czernuszenko et al. [10]	re-sampled rectilinear 3D grid in the tracked space	software-based visual displacement analysis	tri-linear interpolation	N/A	N/A	no
S. Ellis et al. [15]	rectilinear 3D grid in the true space	unknown	tetrahedral interpolation	local interpolation	quaternions	no
W. Briggs [18]	re-sampled rectilinear 3D grid in the tracked space	jig, 2D grid on the floor	local interpolation	N/A	N/A	no
V. Kindratenko [20]	rectilinear 3D grid in the true space	varying height sensor holder, 2D grid on the floor	4th order polynomial fit	4th order polynomial fit	Euler angles	source code, binaries
V. Kindratenko et al. [19]	rectilinear 3D grid in the true space	varying height sensor holder, 2D grid on the floor	local interpolation	local interpolation	quaternions	source code, binaries
T. Saleh et al. [22]	rectilinear 3D grid in the true space	varying height sensor holder, 2D grid on the floor	neural network	N/A	N/A	no
M. Ikits et al. [21]	rectilinear 3D grid in the true space	NDI Optotrack	4th order polynomial fit	4th order polynomial fit	quaternions	no

Table 1. Summary of electromagnetic tracker calibration literature surveyed.

Technique	Overall location error correction quality	Overall orientation error correction quality	Major advantages	Major disadvantages
Tri-linear interpolation	average	N/A		- topology limitations - discontinuity in gradient
Tetrahedral shape function	average	N/A	- unstructured grid can be used	- discontinuity in gradient
Local interpolation	average	average		- topology limitations - discontinuity in gradient
High-order polynomial fit	best	best	- smooth continuous mapping - unstructured grid can be used	- occasional increase in error value when the initial error is small
Neural network	fair	N/A	- smooth continuous mapping - unstructured grid can be used	- network parameters are not well defined - time-consuming training is required
Hardy's multi-quadric interpolation	best	best	- smooth continuous mapping - unstructured grid can be used	

Table 2. Summary of tracker calibration techniques surveyed.