

Automatic Calibration of Magnetic Motion Trackers using a Bayesian-Neural Cascade

Piperakis Romanos, Saito Suguru, Nakajima Masayuki

Graduate School of Information Science & Engineering, Tokyo Institute of Technology

E-mail: {romax, nakajima}@img.cs.titech.ac.jp, suguru@pi.titech.ac.jp

1 Introduction

Motion capture data are being used to address an increasingly broadening repertoire of animation related tasks, ranging from film production and game products to research on realistic human motion and even medical applications. Because of their vast utility, many technologies have been developed to acquire such data but magnetic trackers remain the dominant motion capturing solution for computer graphics applications. But even though magnetic trackers have many advantages versus their mechanical, optical and sonic counterparts, they are amenable to electromagnetic interference. Conventional calibration procedures are tedious and very time consuming, since they require the manual collection of measurements on a relatively dense three dimensional grid inside the tracking area. Despite this labor-intensive procedure, this type of calibration cannot compensate for every type of distortion resulting in limited accuracy and even motion discontinuities in noise contaminated environments. We propose a Bayesian-neural approach that allows the automatic calibration of magnetic motion capturing systems, using motion tracking data from a short, free-form tracking of a calibration pole.

1.1 Background

Motion capture data are continuously gaining popularity among researchers and animation industry professionals, as the recorded motion sequences inherently capture the subtleties of human bodily expressions without the need for tedious keyframing by expert animators or extensive rule based editing. By carefully piecing together appropriate motion clips it is thus possible to create custom length animations with a very high degree of realism, in a fraction of the time and effort needed by keyframe or dynamics based methods.

In the last 20 years, magnetic tracking has emerged as the preferred way of measuring motions in computer graphics applications, because it offers a simple, elegant way of measuring position and ori-

entation of heads, hands, the entire body and other objects in free space. Other tracking technologies may offer faster performance for a single sensor but the update rate and accordingly the dynamic performance, degrades inverse proportionally each time another sensor is tracked.

The main problem associated with magnetic motion capture systems is their sensitivity to electromagnetic interference and the presence of metal objects in the vicinity of the motion tracking area. Even though pulsed DC systems have been developed to limit the sensitivity of the conventional AC trackers to nearby conductive metals, this technology is unable to compensate for the distortion effects of ferrous metals and external electromagnetic interference (EMI).

In real-world settings it is often the case that the floors and ceilings of the motion capture studios contain steel bar reinforcement, which inevitably interferes with the motion capturing process, inducing measurement errors. Another source of interference comes from the magnetic signatures of electric and electronic devices. In most cases even if precautions are taken to remove large metallic objects away from the immediate tracking area, the interference coming from the studio's construction materials and its power lines, cannot be so easily counteracted.

Another problem when working with magnetic motion capturing systems is that of the field coverage region. In order for the sensors to be able to detect reliably their position and orientation, the field in the vicinity of the sensor must be stronger than a certain threshold. This follows from the nature of the motion tracker systems and is the parameter which in effect limits the motion capture range. The closer the field strength is to the threshold value at a given point, the less accurate and more susceptible to EMI are the sensor readings. Depending on the specific implementation of the capturing system, if during a motion capturing session the sensor exits the region where the field is strong enough for reliable motion tracking, then the sensor measurement will likely become stuck to the last reliable measurement; that of the exit point from the operational tracking region. Upon re-entry of the sen-

⁰This research was supported by the JSPS post-doctoral fellowship program.

sors in the operational tracking region, the sensors will regain the ability to detect their coordinates and consequently the sensor reading will jump from the previously stuck position to the new point of re-entry. The intermediate 'stuck' sensor positions will be dubbed *outliers*.

This effect could also be regarded as a type of distortion; one for which we would require a correction procedure or at the very least a detection algorithm but for which existing calibration procedures are unable to address.

2 Related Work

To control the metallic or electric distortion problem, especially for the AC tracking technologies that are more susceptible to it, special calibration techniques have been developed. The most frequently applied is called "mapping and compensation" and includes the manual collection of hundreds of data points (mapping) to determine the amount of distortion in the operational area. This collection of points is usually performed with the help of a vertical pole with the sensors placed on it at fixed intervals and orientation. This calibration pole is then consecutively positioned on each point of a dense rectangular grid drawn on the tracking area's floor. The data points collected in this manner, span a three dimensional, regular rectangular grid. These data points are then used to form a correction (compensation) map that is applied to the sensed signals.

While this procedure is relatively effective, it is time consuming and costly. Further it assumes that the field transmitter remains stationary and that no metallic objects move or are introduced into the area at any time after the mapping. Also, changes in the power distribution or changes in the operation of electric devices cannot be taken into account. In this sense, the corrections achieved with this type of calibration are ineffectual, for instance, if there is a need to periodically relocate the field transmitter, if a metal object is introduced into the motion-capture area or even if a monitor in the vicinity of the tracking area is turned on.

Approaches towards a more efficient or automated calibration have also been developed for clinical and medical applications, to accommodate the mapping or tracking body parts or organs using magnetic tracking technologies [2] [3] [5] However, all the current approaches currently involve the use of hybrid technologies. In other words, the only way to circumvent the majority of the motion tracking deficiencies of electromagnetic technologies is by attempting to combine optical, mechanical (inertial), sonic and magnetic motion tracking systems into a single framework [1] [11] [8] [9].

While this remains a valid solution to a lot of problems, it does so at the expense of the added cost and complexity of attaining, setting up and managing the hybrid systems. It also fails to provide a universal solution to one of the most important motion tracking technologies used in computer graphics today. In that sense, an approach aimed specifically to the magnetic motion capturing problems is necessary.

3 Bayesian Framework

If we are to get the most out of a calibration procedure, then it follows that we must take into account all the available domain knowledge we have about motion capturing. This fact is a fundamental axiom of plausible reasoning and is inherently contained in the Bayesian approach to probability theory. In the case of motion capturing of a human actor, this domain knowledge concerns the physical properties of motion itself, the specifications and nature of the motion tracking system, as well as the inherent attributes of the actor's kinematics. Ideally we would like to incorporate

- Motion continuity. Consecutively detected coordinates have a larger probability of being closer to each other than further.
- Distortion continuity. The electromagnetic field can be taken to be continuous if there are no metal surfaces of appropriate dimensions.
- Tracker specifications. The initial conditions of the calibration can be taken from the ideal performance of the magnetic tracker in the absence of noise, thereby optimizing the parameter search.
- Skeleton constraints. The fixed structure of the human body constraints the possible coordinates that the sensor position can be at any given time.
- Human body kinematic freedom constraints. The human body's joints have limitations as to the range of rotations they can perform as well as the acceleration of individual limbs. This further constraints the parameter space and can be used to detect and correct calibration inconsistencies.
- Human initial posture. The initial posture of the motion capturing process could provide an approximate starting point for the parameter estimation.

Using as much of this prior knowledge as possible, we could view the problem of calibration as the

parameter estimation of a mapping from sensor values to body coordinates. This parameter estimation should by no means be static. Since we don't know how the field will change due to unpredictable interference or how the sensors will move relative to the actor's joints, we cannot solve the problem of calibration before analyzing the motion tracking data itself.

This leads us to consider a different paradigm to the calibration problem. If we base our calibration procedure on the entire range of possible motion capture sessions, all the possible combinations of field distortions and different actor bodies, we are losing focus of the problem at hand, which is to acquire the best possible quality of motion capture data *given the already acquired mocap session's measurements*. The calibration mapping should thus be defined as a property of an individual session rather than a preprocessing step to the general problem of motion tracking.

As a result this approach aims to use the prior, domain knowledge in conjunction with the specific motion tracked data to produce a clean motion and at the same time detect the average distortion field for the duration of that particular capturing session. Furthermore, such a calibration procedure could be completely automated, boosting efficiency and filtering out inconsistencies in a single framework.

3.1 Neural Network Distortion Representation

A key component of any calibration system that attempts to compensate for distortions in the electromagnetic field, is the way it will map the noise contaminated sensor measurements to correct or consistent motion tracked coordinates. This mapping could be represented as a vector field which given sensor reading would yield its calibrated coordinates. To represent this vector field we can use a generalization of the 3D object representation approach of [10] with three outputs, one for each distortion coordinate. The new neural network architecture can be seen in figure 1, where the size of every layer is shown and again the R inputs can be feature functions of the input space. These feature functions can be chosen according to the shape characteristics of the field we wish to represent. This choice could thus include some of our prior knowledge about the form of the magnetic field and the distortion we expect given the system's configuration, inside the motion tracking studio.

The advantages of using a neural network representation for the distortion field are numerous. For one, we make a tremendous economy in the size of the representation. Instead of storing hundreds of sampled points and interpolating for other points

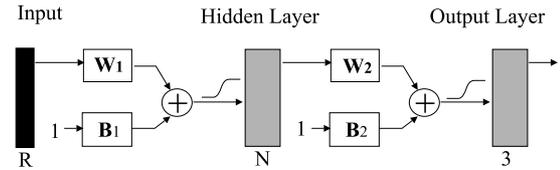


Figure 1: A three layer neural network with R inputs, N hidden nodes and 3 outputs. This type of architecture can represent any continuous, analytic vector field.

based on the local neighborhood of samples, we have a compact analytic representation, which can be evaluated directly, for any point in the tracking region. This also reveals the computational advantages of such an approach. Instead of having to find the nearest sampled points to each sensor position in order to interpolate for each sensor measurement, the distortion can be found with an evaluation of the neural network directly for that point.

Another point where the neural network representation yields superior results is the expressive power. By using custom feature functions as inputs to the network we can generalize to a better approximation of the distortion field, given the same sampled points. To see how that is possible it is enough to point out that the generalization will be a non-linear interpolation of the sampled points. The exact function used for the interpolation will be specified by the combination of the feature functions and the sigmoid activation functions of the hidden layer. That combination will in turn be the result of the training procedure and thus will have emerged as the locally optimum way of interpolating through the given sampled points.

4 Pole Calibration Algorithm

Turning our attention to the problem of calibration, in the scope of this paper and given the vast extend of this topic and the need for concrete results to verify the correctness of our proposed approach, the problem of human motion calibration had to be simplified. The developed algorithm, consequently, attempts to address the problem of automatic calibration using, not the motion tracking data of a human actor, but a free hand movement of a calibration pole. The concepts and problem formulation for that simplified problem will then be used to identify ways of addressing the general problem.

The procedure for obtaining the data to be used for calibration, uses a pole mounted with sensors in fixed length intervals. The calibration pole is then placed in a fixed position inside the motion tracking



Figure 2: The calibration procedure starts with the user picking up the calibration pole and moving it around the motion tracking region. As is apparent from the sequence, the trajectory of the pole is free although the speed should be kept relatively slow to minimize interpolation errors.

area and the motion tracking is initiated. The user picks up the calibration pole and moves it around the motion tracking region freely, as shown in figure 2. After most parts of the region have been swept by the calibration pole, a procedure which should not take more than a minute, the motion tracking is stopped. The data collected are then analyzed to determine the distortion field and the calibrated motion capture of the pole at the same time.

The aim of this section is to find a way of combining domain knowledge about the structure and the motion of the calibration pole together with the known distortion and noise characteristics, in order to learn a mapping from the measured coordinates to the calibrated ones. This task requires that we redefine our goal. Since we are primarily interested in motion capturing human animation, it is apparent that the coordinate frame of the calibrated coordinates need not be the same as that of the absolute measurement frame. Instead any orthogonal transform that consequently preserves the angles between vectors would be irrelevant to the successful utilization of the calibrated data.

Formally stated, if we signify the sensor coordinates and measurements as x and $z = f(x)$ respectively, a sufficient condition for the calibration to be successful in a region R is for the function f to be invertible in R . By reading the sensor values $z_{1,k} = f(x_{1,k})$, $z_{2,k} = f(x_{2,k})$, ... $z_{n,k} = f(x_{n,k})$ for $k = 1..m$ consecutive frames we acquire m sets of n sensor values. If we proceed to filter these data, taking into account the collinearity and spacing constraints imposed by the structure of the calibration pole, together with the prior knowledge we have about the motion continuity and velocity constraints, we can get m sets of the best possible estimates of those sensor coordinates which we will symbolize with $\hat{x}_{1,k}$, $\hat{x}_{2,k}$, ... $\hat{x}_{n,k}$.

If we were interested for the best estimates of the

motion of the calibration pole, this is perhaps where we would terminate this algorithm. However, since we want to use the pole data to get the best estimate of the correct sensor coordinates of a human motion capture session, we need to combine these mappings to a unified representation of the distortion field. The major reason for doing so is that the Bayesian filtering does not take into account the spatial characteristics of the distortion and thus might map two approximately equal measurements to different estimates of the sensor coordinate, depending on their respective trajectory history. In order to approximate the inverse of the distortion $f^{-1}(z)$, the mapping must be one to one. This is precisely where the neural network representation comes into play.

If we train the neural network to represent a function $\phi(z)$ from the measurements z to our estimate of the actual sensor coordinates, using the correspondence of $z_{i,k}$ to $\hat{x}_{i,k}$, we can effectively cross-validate or integrate all the similar measurements into a single mapping function. This mapping, by the very definition of the training method, has the property of a minimized mean square error over the entire sampled region. In other words, the neural networks serves here to combine all the measurement-estimate pairs in order to yield a consistent, one to one mapping of the sensor measurements to the calibrated coordinates.

Another point worth noting is that since the neural network function $\phi(z)$ approximates the cross-validation of the filtered sensor's coordinate estimates \hat{x} , it effectively implies that $\phi(z)$ is trained to be a linear space and that we can furthermore write:

$$|x_1 - x_2| = |\phi(z_1) - \phi(z_2)| \quad (1)$$

for any x_1 , x_2 and their corresponding measurements z_1 , z_2 . Without loss of generality we can set $\phi(0) = 0$ and use the fact that from equation 1 we get $|x| = |\phi(z)|$ to readily prove that the angle between any two vectors is preserved. Since $\phi(z)$ is trained to minimize the error between $|x_1 - x_2|$ and $|\phi(z_1) - \phi(z_2)|$ it is apparent that the resulting function will try to preserve the angles between vectors and thus is a valid candidate for a calibration mapping.

4.1 Tracking Range

During the motion tracking of the calibration pole, as the pole gets moved around the tracking area, the sensors sample a certain volume of the capturing region. Since the bounds of the tracking region are not visible to the user or performing actor, there is a relatively high chance of passing them during all

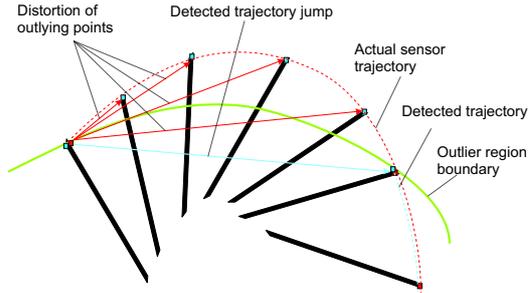


Figure 3: The effect of a calibration pole trajectory across the outlier boundary. The sensor actual position is denoted by a blue square, the actual trajectory by a dashed red line, the sensor detected position by a red square and the detected trajectory by a blue dashed line. Notice the jump from the point of exit to the point of re-entry, denoted by a blue arrow.

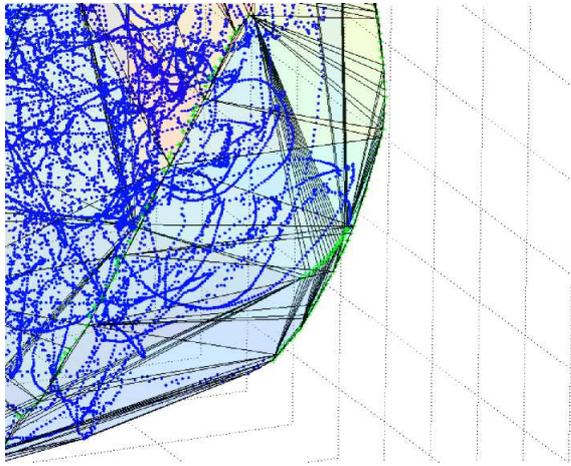


Figure 4: A zoomed portion of the convex hull of the sensor measurements. Points as blue crosses are the filtered estimates of the sensor coordinates that reside within that convex hull and points as green stars are the ones lying outside (outliers)

but the most static motion capture sessions. The points residing outside the tracking region V_t but inside the sampled volume V_s will yield outliers as discussed in previous paragraphs. The points sampled there will consequently report the last point on the boundary of V_s and V_t where the sensor exited V_t . This phenomenon is graphically shown in figure 3, where the red dashed line denotes the actual sensor trajectory, the blue points correspond to the placement of the sensor on the calibration pole, the red points signify the reported coordinates from the sensors in the absence of any other type of interference, and the red arrows give the difference between the sampled points and the reported points or else the distortion of the field in the sampled points. That figure also explains the jump effect from the

points of exit from the tracking region, to the points of re-entry. In the figure this instantaneous jump effect is shown as a blue arrow and clearly shows the type of inconsistencies and discontinuities that the outlier effect imposes. Figure 4 shows the actual filtered data that reside within the initial measurement's convex hull with blue color and the ones that reside outside (outliers) with green.

There is a number of problems associated with the inclusion of the outlier points in a neural network training procedure for the distortion field. These are associated with the fact that the training set will contain multiple conflicting associations between points in the outlying area $V_s - V_t$ and the various points of exit from V_t , as well as the discontinuities from the later to the points of re-entry into the V_t . The result of these bad examples are a compromised training procedure and a bad generalization at least in the vicinity of the outlier boundary with the V_t . Therefore it is instrumental for the efficiency and performance of the algorithm to discard these points prior to the training process. In a later paragraph we will discuss how that can be achieved.

4.2 Bayesian Filtering

Before it is possible to discard the outlying sensor measurements or train the neural network representation with the distortion field, we need to decide on an appropriate filtering architecture. The approach taken must explicitly take into account the already identified prior knowledge, the simplified structure of the calibration pole and the entire sequence of the motion tracking data, to find what is the most probable estimate of the sensor's position at each frame. The obvious choice here is Kalman filtering since it enables us to propagate our state of knowledge (probability distribution) about the sensor's correspondence to real points, throughout the motion tracking sequence while at the same time specify the restrictions imposed by the sensor setup. In other words given our initial uncertainty, the uncertainty of the starting position and the errors associated with the process and the subsequent measurements, find the most probable trajectory of the sensor's coordinates through the motion tracking area.

Unfortunately, this problem can quickly become untractable for nonlinear dynamic processes and restrictions, since even if we begin with a conjugate probability distribution function for the initial and process parameters, the distribution will soon cease to be conjugate. This in turn means that we will no more have a simple closed form for the probability distributions and the complexity of propagating that distribution forward in time will soon become

overwhelming.

Since the calibration pole is translated and rotated within the bounds of the tracking area, the most appropriate system model should accommodate this type of motion. However, rotation in Euclidean space is not a linear transform and thus the dynamics of our system model are forced to be non-linear as well. A simple linearization of the dynamics and measurement equation about the state estimate at each pose, is not sufficient to yield correct results due to the strong non-linearity of rotational transforms. Instead a numeric Gauss-Newton method was employed, that linearizes the dynamics and measurement equation iteratively about consecutive state estimates, until the state update converges.

In this approach the system model was taken to consist of the position of the sensor on one end of the calibration pole, its rotational transform in quaternion form, the linear velocity and the rotational velocity in each axis. Given the initial orientation of the calibration pole axis \mathbf{e}_0 which is readily found by the first principal component of the sensor measurements at the first frame, we can project all the measurements on that direction, sort the order of the sensors in case they have not been placed contiguously, and use the following system model equations:

$$\mathbf{t}_{k+1} = \mathbf{t}_k + \tau \mathbf{v}_k \quad (2)$$

$$\mathbf{q}_{k+1} = Q_{tran} \mathbf{q}_k \quad (3)$$

$$\mathbf{v}_{k+1} = \mathbf{v}_k \quad (4)$$

$$\omega_{k+1} = \omega_k \quad (5)$$

where \mathbf{t}_k , \mathbf{q}_k , \mathbf{v}_k and ω_k are the translation, quaternion rotation, linear velocity and angular velocity respectively and Q_{tran} , Ω and $\bar{\Omega}$ are the quaternion extrapolation matrix and the angular velocity in quaternion and matrix format respectively. The later are analytically given by:

$$Q_{tran} = \cos\left(\frac{\|\Omega\|\tau}{2}\right)I + \frac{2}{\|\Omega\|} \sin\left(\frac{\|\Omega\|\tau}{2}\right)\bar{\Omega}$$

$$\bar{\Omega} = \begin{pmatrix} 0 & \omega_x & \omega_y & \omega_z \\ -\omega_x & 0 & -\omega_z & \omega_y \\ \omega_y & \omega_z & 0 & -\omega_x \\ -\omega_z & -\omega_y & \omega_x & 0 \end{pmatrix}^T$$

The interested reader is referred to [6] for details on the linearization of the above equations.

Using the fact that the distance between the sensors is constant and that all the sensors reside on the axis of the calibration pole, we can write the following measurement update equations:

$$\mathbf{z}_{i,k} = \mathbf{t}_k + id\mathbf{q}_k\mathbf{e}_0\mathbf{q}_k^* \quad (6)$$

where $i = 1..n$ is the sensor index, k is the frame index and d is the distance between two consecutive sensors. The later can be either given by the calibration pole configuration or approximated by the average distance of consecutive sensor measurements projected on the \mathbf{e}_0 direction.

If the process noise and the measurement noise are taken to be gaussian and white, a reasonable approximation in most engineering applications, then the equations for the Kalman filter are given in [7] [4].

4.3 Cascade Training & Outliers

If we apply the Kalman filter described above to the collected motion captured data, we acquire the calibrated sensor coordinates of those measurements. This association of points can be used to train the neural network distortion field discussed previously to give a representation of the calibration errors. Having defined the way the distortion errors are derived, we can now reconsider the problem of discarding the outlying points in order to arrive to a more robust training procedure.

Since it is clear that the volume spanned by the sensor measurements is always a sampling of V_t , where the field is stronger than the detection threshold value, these sensor measurements, in a sense, define the bounding volume of $V_t \cap V_s$.

One easy way to discard "stuck" points would therefore be to find the sensor measurements that reside closest to the surface of the minimum sized polyhedron that includes all the sensor measurements. These measurements would at least include all the points on the boundary surface between V_s and V_t . Thus to remove the outliers it would suffice to discard the surface points from the training procedure. After discarding these outlying points, the training procedure can continue without the problems already mentioned.

To increase performance, in practice, the convex hull of the sensor measurements was used instead of the exact bounding polyhedron. This enabled a much faster implementation of the test conditions at the expense of some minor inaccuracies that it is believed that the neural network can generalize over.

It is possible to refine the calibration process if we simulate the neural network distortion field with the sensor measurements and refilter the resulting

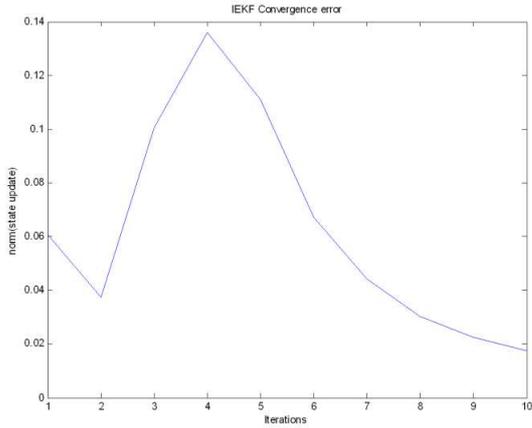


Figure 5: The mean convergence graph of the iterated Kalman filter. Iterations correspond to the application of the Gauss-Newton method around the current state estimate.

calibrated coordinates with a much smaller process and measurement error. In practice the errors of the iterated Kalman filtering can be taken from the mean square error of the neural network training session. If there are no outliers in the motion tracking data then this process will quickly converge, while if there are unremoved outliers the filtering error will reach and fluctuate around a minimum.

5 Experimental Results

The algorithm described was tested with the use of a “motion star” magnetic pulse motion tracking system by *Ascension*. One out of the two available field generators was utilized for training, while the other, unconnected field generator was used to provide additional distortion. Furthermore, the active field generator was placed at floor level to provide enough vertical space in order to test for outlying sensor measurements. The calibration pole was mounted with six sensors at constant fixed intervals of approximately 30 cm and placed in front of the active field generator.

Three different motion tracking sessions were taken, where three different users picked up the calibration pole and moved it around the room trying to sweep as much volume as possible. The duration of these sessions was approximately one minute and the motion capturing system had a sampling rate of 30 frames per second. Thus the datasets collected contained ten to twenty thousand points with six sensor readings for each frame.

The data were then preprocessed to convert from inches to meters and filtered using the iterated extended Kalman filter (IEKF) described in previous

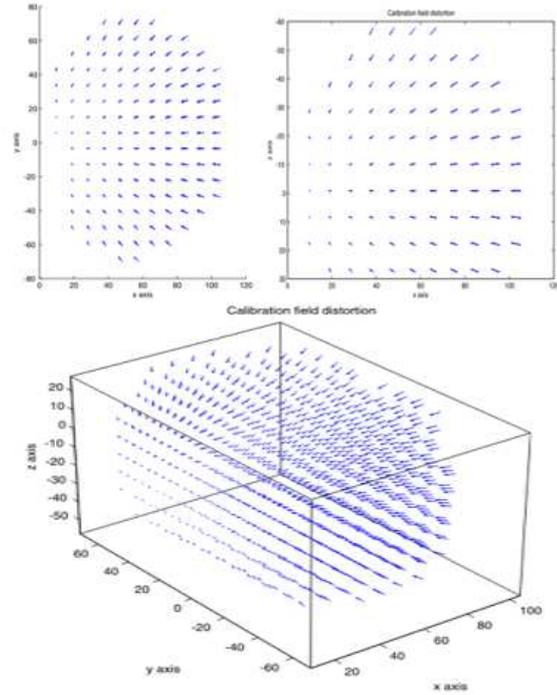


Figure 6: The distortion field learned by the neural network. The distortion grows with the radius from the field generator, as expected.

sections. Measurements were taken to find an upper bound for the linear and angular velocity of the pole, as well as the average error from the field distortion and they were used to define the initial system and measurement error covariance matrices for the operation of the filter. Figure 5 shows the average convergence of the IEKF for a typical calibration trial.

In practice, to enhance training robustness and efficiency, we can generate a 3D grid inside the convex hull of the sensor measurements and use the IEKF filtered data to interpolate calibrated values for those points. We then can use the grid based dataset itself for training the neural network. The interpolation which can be linear or of higher order will serve to remove redundant points from very densely sampled regions of space therefore speeding up training and at the same time even out minor filtered inconsistencies. While the latter could be handled by the neural network itself, interpolation has the advantage that it only has to be performed once before the training occurs.

The neural network employed had six inputs with x , y , z , x^2 , y^2 and z^2 feature functions and five nodes in the hidden layer. Also the *Levenberg Marquardt* (LM) training algorithm was used that typically converged in no more than ten epochs, to an average mean square error in the

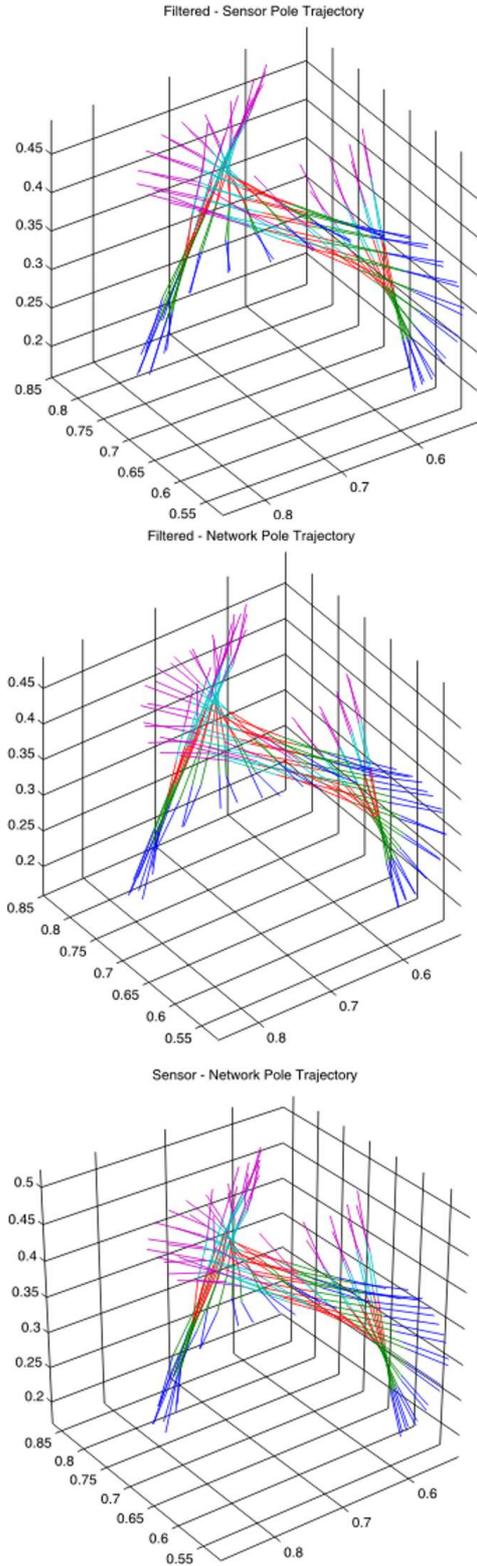


Figure 7: The trajectory of the original sensor coordinates of the calibration pole, the Kalman filtered estimates of those coordinates and the calibrated sensor coordinates using the trained neural network.

order of 10^{-3} meters. Each epoch consisting of approximately 60 thousand points, took 15 to 20 seconds on an Athlon 1Ghz PC.

The data was checked for outliers by discarding all the filtered points residing outside the sensor measurement’s convex-hull, and the remaining points were used to train the neural network distortion field. The resulting distortion field, is shown in figure 6 while figure 7 displays the sensor measurements, Kalman filtered estimates and neural network simulated coordinates in pairs for evaluation purposes.

5.1 Evaluation

From these two figures we can clearly see that the distortion is very small close to the field generator and increases almost proportionally with the distance from it. This is indeed the form of the distortion that is expected and documented for most magnetic motion capture systems with a single field generator. As another criterion of evaluating the correctness of our proposed method it must be noted that the resulting simulated sensor measurements have an average mean square error of about 1mm compared to the filtered estimates which are by definition constrained to the geometrical properties of the calibration pole.

In the above experiments there was no excessive distortion or interference factors present, which leads to the question of the algorithm’s applicability under more severe conditions. To address this question it has to be noted that an implicit assumption of any static calibration algorithm is that there is a constant one to one mapping between the measurements and the actual values and that the sample density is adequate to approximate the distortion function within reasonable accuracy inside our region of interest. Should these conditions hold an appropriate neural network will be able to represent that distortion function.

However, in real world situations there is always an amount of interference and metal objects inside the active tracking area could introduce regions of irreversible distortion. Furthermore, the sensor readings are also dependant on the way they are processed by the device driver in order to yield position and orientation measurements. This processing differentiates the current approach from self calibration methods appropriate to field measurement science, since it introduces further field independent non-linearities, especially near the active tracking range boundaries. Consequently the applicability of the algorithm must be evaluated by considering practical scenarios but with realistic expectations for this class of problem

6 Conclusions & Future Work

With the current implementation of the algorithm, it is clear that it deals with off-line motion capturing. The motion data have to be first collected before they can be used. Nevertheless, if the algorithm is used just to provide the distortion field calibration, then subsequent measurements can be corrected with respect to static interference factors online. In this scenario, the new points collected will have to be simulated (forward pass) from the neural network distortion field in order to provide the calibrated points. The only error that might result from this type of calibration procedure is a small global translation or rotation error, which is irrelevant to the application of human character motion tracking since the angles between body segments are preserved. This form of calibration however, is not customized to each specific motion tracking session any more, nor can it correct for detected outlying data and transient interference. Therefore it sacrifices some of the merits of the described approach and retains only the automatic and more accurate distortion field representation.

The probabilistic approach to calibration for individual motion captured datasets has been presented as a generalized framework, only a subset of which has been currently implemented. As such, there is a lot of extensions and enhancements that could be included in this section.

The main motivation behind the initial calibration framework approach was to automate the acquisition of clean, consistent, human animation data from magnetic motion tracking systems. As a first step, the calibration process was greatly simplified and automated by the use of a calibration pole. Nevertheless, the final target of the developed framework is the human body structure and not a calibration pole. The later was used to display the merits of the approach to a simple use case and to provide evidence for the accuracy and ease of the method. Consequently, a natural way forward is to extend the approach to more than one rigid segment, in order to represent the human skeleton articulation structure. While doing so does not impose any theoretical difficulties, it would be preferable to research ways of dealing with the limitations of the proposed method in the simple case of a calibration pole before generalizing to the more complex articulated case.

References

[1] Azuma, Ronald, Jong Weon Lee, Bolan Jiang, Jun Park, Suya You, and Ulrich Neumann. Tracking in unprepared environments for aug-

- mented reality systems. *Computers & Graphics*, 23(6):787–793, December 1999.
- [2] John P. Brennan and Jais A. Brennan. Clinical application of a high-performance computing system for visualizing and monitoring human labor and birth. In K. Morgan, H. B. Satava, R. Sieburg, Mattheus, and J. P. Christensen, editors, *Interactive Technology and The New Paradigm for Healthcare*, pages 48–52, Brooklyn, NY, 1995. IOS Press and Ohmsha.
- [3] A. M. J. Bull, F. H. Berkshire, and A. A. Amis. Accuracy of an electromagnetic measurement device and application to the measurement and description of knee joint motion. Technical Report 212, Mechanical Engineering Department, Imperial College of Science Technology and Medicine, London, May 1998. Part H.
- [4] A. Gelb. *Applied Optimal Estimation*. Cambridge, MA. MIT Press, 1974.
- [5] O. H. Gilja, P. Detmer, and J. M. Jong. Intra-gastric distribution and gastric emptying assessed by three-dimensional ultrasonography. Technical report, Medical Department A, Haukeland Hospital, University of Bergen, Department of Surgery and Anesthesiology, The Center for Bioengineering, University of Washington, Norway.
- [6] M. A. Abidi J.S. Goddard. Pose and motion estimation using dual quaternion-based extended kalman filtering. In *Int. Soc. for Optical Engineering Photonics West Symposium on Electronic Imaging*, 1998.
- [7] Emil Kalman, Rudolph. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [8] D. F. Leotta, P. Detmer, and O. H. Gilja. Three dimensional ultrasound imaging using multiple magnetic tracking systems and miniature magnetic sensors. In *Proceedings IEEE Ultrasonics Symposium*, 1995.
- [9] T. Molet, R. Boulic, and D. Thalman. Human motion capture driven by orientation measurements. *Presence*, 8(2):187–203, April 1999.
- [10] E. Piperakis. *3D Graphics with Neural Networks*. PhD thesis, Tokyo Institute of Technology, 2002.
- [11] A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Proc. SIGGRAPH*, 1996.