

Electromagnetic Tracker Measurement Error Simulation and Tool Design

Anca Berariu, Marco Feuerstein, Nassir Navab
berariu@in.tum.de

Abstract. *Electromagnetic tracking is prone to errors due to the distortions caused by the metallic objects situated in the electromagnetic field's proximity. Hence, a tool with electromagnetic sensors has first to be tested in different environments before it could be safely used in a surgical intervention. Such tests are often succeeded by design iterations meant to minimize the effect of such distortions. In order to accelerate this process, a simulation software was proposed to virtually test the tools before the physical building. The simulated tests are based on apriori measurements which yield a characterization of the error filed for the specific environments.*

1 Introduction

Through the advances in electromagnetic (EM) tracking system, the surgical interventions could benefit from the possibilities given by the electromagnetic sensors. Attached to surgical tools, these can offer a better coordination where optical tracking systems fail to work. An example constitutes the bronchoscopy intervention, when the medical tools inserted in patient's body are no longer in the line-of-sight needed by the optical tracking system to perform. But there is a payback for exceeding the limitations of the OT systems: accuracy and strict conditions regarding the environment. Not only that the data collected from EM tools is about one order of magnitude less accurate than the one from OT systems, but a ... by means of using a robot arm to control the tool more accurately seems almost impossible in practice because of the restriction for non-metallic components[CHU04]. In these conditions, the careful design of the tools and the accurate characterization of the environment are of great importance.

A tool's design refers not only on its shape and size, but also on the number of electromagnetic sensors attached to it, their relative position and their orientation. Further more, its performance depends on the choice of the coordinate reference frame (CRF). Although one could choose a theoretically optimal configuration, its true behavior cannot be entirely predicted, but has to be tested in specific environments. Hence, the design of a tool has to go through the iteration of several configurations, their physical building, measurements collection and analysis. An optimization of this process, in the sense of reducing the time needed for all the iterations, could be achieved in that the physical construction and the measurements would be replaced by simulations [FIS05].

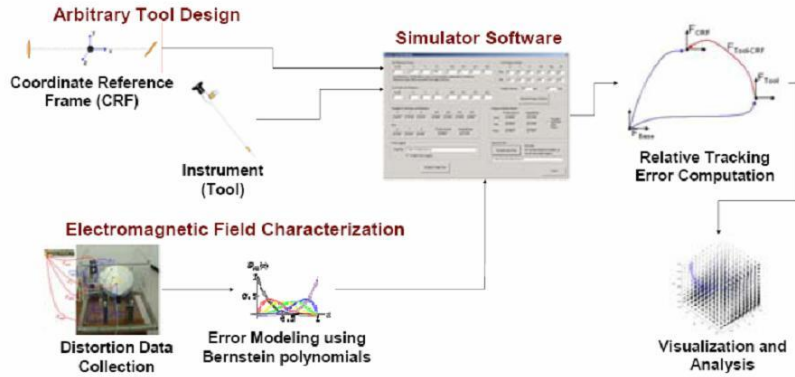


Figure 1: The simulation pipeline components (adopted from [FIS05b])

2 Related Work

The only existing software for simulating electromagnetic tracked tools is the one built by Fischer for his Master Thesis [FIS05a]. The main part of his work concentrates on how to model the error filed for a given environment. He bases his research on the work of Wu [WUX03] adding a method to deal with orientation based errors. Another work on this subject is the one of Chung [CHU04] who adopts another way of modeling the orientation error using splines instead of the linear interpolation between the angular error, as in [FIS05].

Other related work is on the area of tool design and optimization for optical tracked tools [WES04], which results were extended for the EM case [FIS05a].

3 Simulation Pipeline

The main goal of a simulation in the current case is that of replacing the real world, the lab where the experiments should be carried out. The user introduces the configuration of the tool, describes the coordinates where the measurements have to be simulated and starts the simulation. As a result, they will receive information like the error in position and orientation for each "collected" measurement, or the overall accuracy. But how can the simulation software asses the value that the EM system would have registered for the given coordinates in a specific environment, like OR for example? A way to deduce this information, a method to characterize that environment and its influence on the EM measurements is needed. The solution appears in several papers and was adopted by Fischer too: error modeling. Given a formula that describes the error introduced in the measurements by the distortions, one can now simulate this measurements by adding to the given coordinates some computed value. From here on, the simulation flows like for real tests do. The "measurements" are compensated using the same information about the environment as for introducing the distortions. Then the algorithm for frame fitting is applied to determine the position of the tool according to the measurements and the result is compared with the initial values from the configuration of the simulation (see figure 1).



Figure 2: Hybrid tools with both electromagnetic sensors and optical markers (adopted from [FIS05b]) and [CHU04]

4 Measurement Error Modeling

The first step in modeling the error is to map the error filed to the EM tracking system. The common procedure is to perform "hybrid" measurements by using a tool which has attached both electromagnetic sensors and optical markers (figure 2). The Optical Tracking (OT) System will track the tool and (having the EM system beforehand registered) will deliver the reference position and orientation of the EM sensors [FIS05a]. This is also called the ground truth reference because of OT systems's higher accuracy and their immunity to distortions.

The difference between the coordinates measured with the EM system and the reference given by the OT system represents the position error:

$$\vec{e}_{pos} = \vec{p}_{measured} - \vec{p}_{ideal} \in \mathbb{R}^3$$

The orientation error is given by a Rodriguez vector:

$$\vec{e}_{ori} = \vec{\omega}\theta \in \mathbb{R}^3$$

where $\vec{\omega}$ is the axis of rotation and θ is the magnitude required to align the reference vector to the EM measurement.

4.1 Measurement collection

can be done in different ways, from the handling of the object: freehand or robot-based, to the type of the movement: static or dynamic acquisition. In any case, there is a minimum number of data points that have to be collected in order to obtain an accurate description of the environment. It is on one's goal to keep this number as low as possible because otherwise it would imply a tremendous investment of time.

The statement of the problem to be solved now is: given a set of points and the corresponding values of the error function, find the optimal interpolation formula. The error function is defined by the position error and the orientation error. Concerning the set of points that the error depends on, there have been some discussions and some rejected statements. The first research treated the error as only being dependent on the position of the tool ([WUX03], [KIN00]). Livingston and State [LIV97] made measurements suggesting that this assumption was wrong, that the error also depends on the orientation of the tool, but the method

they developed did not correct it. Among the first solutions was given by Wu and Taylor in [WUX03b], where they present a method to interpolate between both position and orientation data. Later, Fischer [FIS05a] extended this approach and presented an algorithm to implement it.

In the following, the basics for the interpolation of the error are going to be presented followed by the description of the algorithm for the orientation interpolation, as in [FIS05].

4.2 Bernstein Polynomials

Bernstein polynomials are defined in general by:

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}$$

where

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

Figure 3 presents these basis functions up to the order of 5.

For a 1D case, the error can be modeled as:

$$e(x) = \sum_{i=0}^n c_i B_i^n(x)$$

where $e(x)$ is the error at a position x , (x is normalized between 0 and 1); c_i is the i^{th} coefficient of the polynomial; $B_i^n(x)$ is the i^{th} basis term of an n^{th} order polynomial

Extending further to a l dimensional space, the error is given by:

$$e(x_1, x_2, \dots, x_l) = \sum_{i_1=0}^n \sum_{i_2=0}^n \dots \sum_{i_l=0}^n c_{i_1, i_2, \dots, i_l} B_{i_1}^n(x_1) B_{i_2}^n(x_2) \dots B_{i_l}^n(x_l) \quad (1)$$

Using the matrix notation, the equation 1 becomes:

$$\begin{pmatrix} e(x_1, \dots, x_l) \\ \vdots \end{pmatrix}_{i,l} = \begin{pmatrix} B_0^n(x_1) \dots B_0^n(x_l) & \dots & B_n^n(x_1) \dots B_n^n(x_l) \\ \vdots & & \vdots \\ \vdots & & \vdots \end{pmatrix}_{i,(n+1)^l} \begin{pmatrix} c_{0,\dots,0} \\ \vdots \\ c_{n,\dots,n} \end{pmatrix}_{(n+1)^l,1} \quad (2)$$

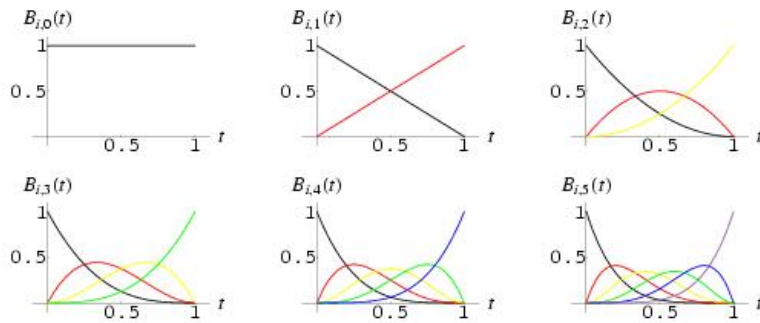


Figure 3: Bernstein basis functions up to the 5th order (adopted from [FIS05a])

However, expression 2 is still the interpolation of a scalar value. To make it applicable for the current problem, one has to extend it again to m dimensions for the interpolated value:

$$\begin{pmatrix} e_1(x_1, \dots, x_l) & \dots & e_m(x_1, \dots, x_l) \\ \vdots & \dots & \vdots \end{pmatrix}_{i,m} = \begin{pmatrix} B_0^n(x_1) \dots B_0^n(x_l) & \dots & B_n^n(x_1) \dots B_n^n(x_l) \\ \vdots & \dots & \vdots \end{pmatrix}_{i,(n+1)^l} \begin{pmatrix} c_{0,\dots,0}^1 & \dots & c_{0,\dots,0}^m \\ \vdots & \dots & \vdots \\ c_{n,\dots,n}^1 & \dots & c_{n,\dots,n}^m \end{pmatrix}_{(n+1)^l,m} \quad (3)$$

The important observation to be made in equation 3 is that l , the dimensionality of the space, and m , the dimension of the measure being interpolated, do not have to be equal.

4.3 Position interpolation

Like in the early articles, one can start by estimating the 6DoF error ($m=6$) from only the 3 dimensional space defined by the measured positions ($l=3$). This works perfectly fine using the formula 3. To determine all the coefficients, $(n+1)^l$ data points are needed, where n is the order of the polynomial. Inserting the measurements in the equation will yield three systems that can be solved with a simple least-squares algorithm.

4.4 Orientation interpolation

To include the orientation interpolation, the straightforward extension would be to increase the dimensionality of the space to 6 ($l=6$). But in doing so, one has to consider the increase of the number of coefficients that have to be determined in the equation 3 and with that, of course, the increase of data points that have to be collected. In other words, one will have to collect $(n+1)^6$ instead of only $(n+1)^3$ data points. A simple computation for $n=3$ shows that, if for the interpolation of position only one needs at least 64 measurements, for interpolating the orientation too, one would need no less than 4096 points. The realization of such requirements in practice would be possible only with very high efforts.

The solution is to use the information regarding the orientation in another way. Instead of performing an interpolation with Bernstein polynomials, an interpolation between the angular errors is chosen. In one equation, this is expressed as follows:

$$\begin{pmatrix} w_{1,b}e_x^1 & w_{1,b}e_y^1 & w_{1,b}e_z^1 & w_{1,b}e_{Rx}^1 & w_{1,b}e_{Ry}^1 & w_{1,b}e_{Rz}^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{i,b}e_x^i & w_{i,b}e_y^i & w_{i,b}e_z^i & w_{i,b}e_{Rx}^i & w_{i,b}e_{Ry}^i & w_{i,b}e_{Rz}^i \end{pmatrix} \approx \beta \begin{pmatrix} w_{1,b}c_{x_0,0,0} & w_{1,b}c_{y_0,0,0} & w_{1,b}c_{z_0,0,0} & w_{1,b}c_{Rx_0,0,0} & w_{1,b}c_{Ry_0,0,0} & w_{1,b}c_{Rz_0,0,0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{i,b}c_{x_n,n,n} & w_{i,b}c_{y_n,n,n} & w_{i,b}c_{z_n,n,n} & w_{i,b}c_{Rx_n,n,n} & w_{i,b}c_{Ry_n,n,n} & w_{i,b}c_{Rz_n,n,n} \end{pmatrix} \quad (4)$$

$$\beta = \begin{pmatrix} B_0^n(x_1)B_0^n(y_1)B_0^n(z_1) & \dots & B_n^n(x_1)B_n^n(y_1)B_n^n(z_1) \\ \vdots & \dots & \vdots \\ B_0^n(x_i)B_0^n(y_i)B_0^n(z_i) & \dots & B_n^n(x_i)B_n^n(y_i)B_n^n(z_i) \end{pmatrix}_{i,(n+1)^3}$$

The w represents some weights that are computed from the orientation measurements and b is the number of chosen basis vectors. These constructions are going to be explained in detail in the next subsection.

Regarding the number of the data points that need to be collected, one can see that now only $b * (n + 1)^3$ measurements are needed. For the previous computed example, this would mean 896 points compared to the 4096 imposed by the first approach.

4.5 Algorithm

As described by Fischer in [FIS05a], the algorithm that leads to a model of the error which considers both the position and the orientation dependency, is as follows:

- Choose a set of base orientations / basis vectors for which polynomials are generated. They should be evenly distributed as shown in figure 4, but the actual number depends on the particular field.

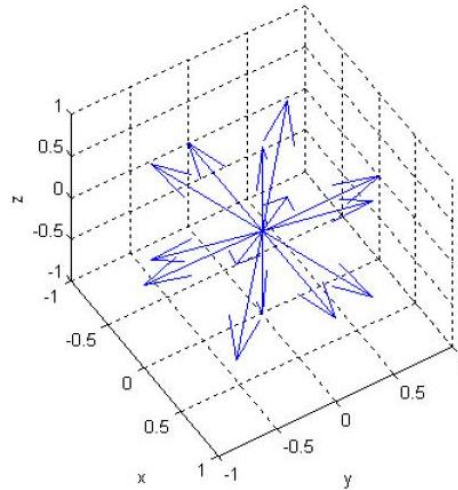
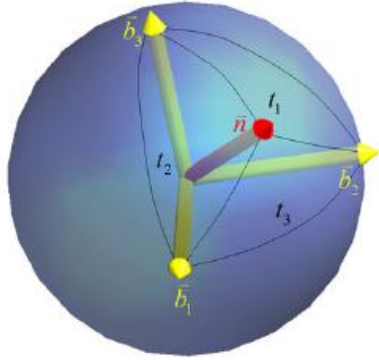


Figure 4: Example of orientation of basis vectors(adopted from [FIS05a])

An example of a base with 14 vectors is:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.57735 & 0.57735 & 0.57735 \\ -0.57735 & 0.57735 & 0.57735 \\ 0.57735 & -0.57735 & 0.57735 \\ -0.57735 & -0.57735 & 0.57735 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ -0.57735 & -0.57735 & -0.57735 \\ 0.57735 & -0.57735 & -0.57735 \\ -0.57735 & 0.57735 & -0.57735 \\ 0.57735 & 0.57735 & -0.57735 \end{pmatrix}$$



\vec{n} Measured orientation
(z axis of sensor)

$\vec{b}_1, \vec{b}_2, \vec{b}_3$ Enclosing three base
orientation vectors

t_1, t_2, t_3 Areas of corresponding
spherical triangles

Figure 5: Definition of spherical triangle used for interpolation(adopted from [FIS05a])

- For each data point, determine the closest three base orientation vectors that enclose the z-axis of the measured sensor reading, \vec{n} , inside of a spherical triangle created by the tips of the unit vectors as shown in figure 5.
- Determine the corresponding area of each of the three spherical triangles generated when subdividing the surface using Barycentric coordinates. The weighting of a particular base vectors contribution to the error is directly proportional to the area of the corresponding, opposing spherical triangle.
- Normalize the areas (t_1, t_2, t_3) of each of the three spherical triangles such that they sum to one; these numbers are the three non-zero entries of the weighting coefficient matrix to be used when generating the polynomial for each of the orientations.
- The results are put into a matrix describing the orientation weights corresponding to each data point. Table 6 contains a printout of the first 25 data points of this matrix for the base orientation vectors given before.
- Calculate the Bernstein coefficients. For each base orientation, there are six sets of coefficients to solve for: X, Y, Z, Rx, Ry, Rz. Each of these systems can be solved using the Singular Value Decomposition method.

Calculating Coefficient matrix: 16437															
0	-0.425351	-0.459193	-0.775888	0	0	0	0	0	0	0.0224916	0.260434	0.717074	0	0	0
1	-0.79435	0.06261	-0.601014	0	0	0	0	0	0	0.51439	0	0.353373	0	0	0.131616
2	-0.279788	-0.982284	-0.382933	0	0	0	0	0	0	0.468567	0.0718821	0.459451	0	0	0
3	0.529382	-0.495309	0.477056	0.107591	0	0.16884	0	0	0	0.72357	0	0	0	0	0
4	-0.585119	0.488236	-2.461895	0	0.608828	0	0	0	0	0	0	0.258237	0	0	0.138919
5	0.782203	-0.405976	-0.144779	0.455952	0	0	0	0	0	0	0	0.312768	0	0	0.23128
6	-0.081257	-0.191826	-0.02212	0	0	0	0	0	0	0.84588	0.11524	0	0.03888	0	0
7	0.115046	-0.593325	0.006886	0.0749028	0	0	0	0.0119318	0	0	0.913455	0	0	0	0
8	-0.254755	0.961597	0.102126	0	0.722149	0	0	0.175882	0	0	0.101969	0	0	0	0
9	-0.964233	-0.023132	0.263456	0	0	0.159075	0	0	0	0.0499485	0.791977	0	0	0	0
10	0.875615	0.476322	-0.080079	0.603997	0.266001	0	0	0	0	0	0	0	0	0	0.130002
11	-0.153444	0.982096	0.109259	-0.17502	0	0	0	0.195255	0	0	0.0297243	0	0	0	0
12	0.521548	-0.305331	-0.047334	0.694512	0	0	0	0	0	0	0.226792	0	0	0.0786956	0
13	0.028898	0.595146	-0.081021	0	0.329907	0	0	0	0	0	0	0.0519978	0	0	0.0162956
14	0.16039	-0.874834	0.154854	0.00359024	0	0	0	0.274515	0	0	0	0.719762	0	0	0
15	-0.486053	0.600304	0.87184	0	0	0.515559	0	0.097659	0	0	0.266772	0	0	0	0
16	-0.301759	-0.949596	0.080534	0	0	0	0	0.135554	0.147949	0.715497	0	0	0	0	0
17	-0.14581	-0.932703	0.329839	0	0	0.122084	0	0	0.245823	0	0.592093	0	0	0	0
18	-0.454741	0.875727	0.162192	0	0.54013	0	0	0.265411	0	0.19466	0	0	0	0	0
19	-0.579128	0.572398	-0.580516	0	0	0	0	0	0.00470034	0	0.00567201	0	0	0.589623	0
20	0.668096	0.099302	0.743962	0.396689	0	0.45547	0.157641	0	0	0	0	0	0	0	0
21	-0.902941	0.421655	0.083111	0	0.226544	0	0.136342	0	0	0.637115	0	0	0	0	0
22	-0.598655	0.218639	-0.117729	0	0.0574388	0	0	0	0.72717	0	0	0	0	0.205331	0
23	-0.847659	-0.108239	-0.512173	0	0	0	0	0	0.526205	0	0.250595	0.2235	0	0	0
24	-0.81512	-0.546705	0.191593	0	0	0	0	0.10913	0.453045	0.237775	0	0	0	0	0
25	-0.504126	0.815571	-0.283597	0	0.388661	0	0	0	0	0.146332	0	0	0	0.654802	0

Figure 6: The weights matrix (adopted from [FIS05a])

5 Tool Design

Having now modeled the error for a specific environment, we can turn back to the simulation problem. As stated in the section 3, an input for the simulation process has to be the description of the tool to be tested. In the system proposed by Fischer, one has the possibility to introduce the coordinates and orientation of the tool's frame and the ones of the sensors attached to it.

A limitation common to all systems dealing with tool design is the one of not being able yet to automatically generate an optimal configuration. Even if the output of the software includes details about the error and the behavior of each sensor and tool, there is no possibility yet to directly interpret this information and extract the best design.

There exist only general statements, which have been proved in experiments too. One is that the more sensors are attached to the tool, the smaller the measured error is going to be. Another states that these sensors should be evenly distributed and the third is that the best results are obtained if the sensors are closer to the tool tip.

6 Simulation - Data distortion

After the all the input is provided to the system, the simulation of the electromagnetic tracking of the tool can begin. For each point in space where the coordinates would be measured in the real case, the system has now the exact values. In order to simulated the real behavior of the tool tracking it will add a position and orientation error computed with the model determined beforehand, as in section 4.5. This is also called data distortion because it is the same as the distortion of the electromagnetic field in the real environment.

7 Data correction

From this step further, there should be no difference between a simulated test and a real experiment. They are included here though for completion.

The disturbed data set has to be corrected back, in order to guess what was the actual position/orientation of the sensors. This is done in the same way as the distortion of the data, just that in the opposite direction. The error is estimated again with the model defined before and it is extracted from the disturbed values. This provides a new set of values, which in the ideal case should be identical with the initial one, before distortion. But because the error model cannot be perfect, there will still be a slight difference in the measurements.

There has to be said that there are many solutions provided for this step. Some of them are just software solutions, like the one presented by Fischer too, others are more hardware oriented and usually come together with the toolkit for a specific system. As an example, Fischer used to test his solution by comparing it with the built-in implementation for the Aurora Tracking System.

8 Frame Fitting

This is the last step of the process. It consists in fitting (translate + rotate) the frame described by the "measured" data to the frame of the tool, as described in the input. The solution to this problem, but only for 3DoF points was early given by Arun in '87 [ARU87].

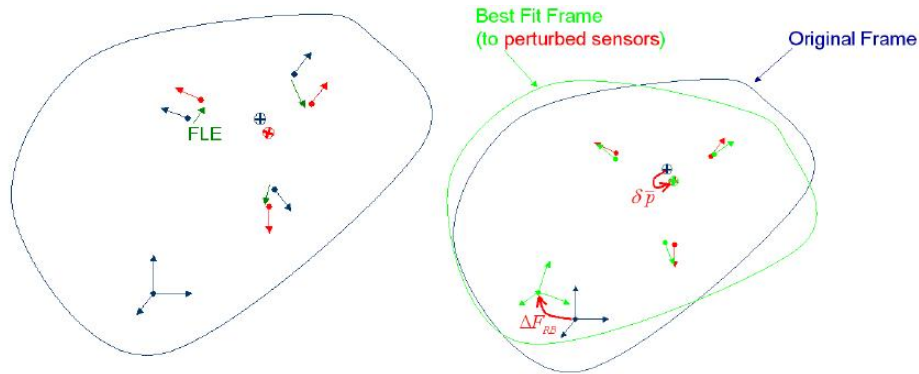


Figure 7: Frame fitting steps.(left)blue: actual sensor configuration,red: perturbed measurements; (right) best fit frame -green- to perturbed measurements -red-(adopted from [FIS05a])

The main idea is to determine the center of gravity (CG) of each of the frames. The first transformation will be the translation to align this two points. This is followed by the rotation around the CG that best aligns all the other points describing the system, in the sense of least squares. A depiction of this steps is given in figure 7.

The extension of this algorithm also considers the information about orientation, included in the collected/simulated data ([FIS05a]). The change appears only for the second transformation, namely the rotation. A weighting factor is introduced when defining the matrices to be used on the least-square solution, hence balancing the influence of the position values over the orientation ones.

9 Conclusion

Faster design of higher quality tracked instruments could be achieved if simulations would be introduced instead of lab experiments. The first system able to produce such simulations was introduced by Fischer and it yield results at least as good as or near to the results achieved by real tests. Although a stable and optimized system, it still lacks in the Graphic User Interface component. An improvement in this direction might also attract the interest of more researchers to start using this tool.

The possibility to carry out meaningful simulations relies very much on a good characterization of the environment to be used for tests. A rapid way to collect measurements data in different environments would enhance the advantage of simulations by reducing the used time.

References

- [ARU87] Arun, K.S., Huang, T.S., Blostein, S.D. "Least-Squares Fitting of Two 3-D Point Sets" IEEE PAMI. Vol. 9. No. 5. Sept. 1987: 698-700.
- [CHU04] Chung, A., Edwards, P., Deligianni, F., Yang, G.-Z. "Freehand Cocalibration of

- an Optical and Electromagnetic Tracker for Navigated Bronchoscopy” MIAR2004 pp. 311-319
- [FIS05] Fischer, G., Taylor, R. (2005) ”Electromagnetic Tracker Measurement Error Simulation and Tool Design”, MICCAI2005
- [FIS05a] Fischer, G. ”Electromagnetic Tracker Characterization and Optimal Tool Design - With Applications to ENT Surgery” Masters thesis, Johns Hopkins University, Baltimore, MD, USA. May 2005.
- [FIS05b] Fischer, G., Taylor, R. (2005) ”Electromagnetic Tracker Measurement Error Simulation and Tool Design”, Poster MICCAI2005
- [KIN00] Kindratenko, V. (2000) ”A Survey of Electromagnetic position tracker calibration techniques”
- [LIV97] Livingston, M.A., State, A. (1997) ”Magnetic tracker calibration for improved augmented reality registration”
- [WES04] West, J. Maurer, C. Jr. ”Designing Optically Tracked Instruments for Image-Guided Surgery” IEEE Trans. on Med. Imaging, Vol. 23. No. 5. May 2004: 533-545.
- [WUX03] Wu, X. Taylor, R. (2003) ”A Direction Space Interpolation Technique for Calibration of Electromagnetic Surgical Navigation Systems” MICCAI2003. Nov. 2003. Springer LNCS 2879: 215-222.
- [WUX03a] Wu, X. Taylor, R ”A Framework for Calibration of Electromagnetic Surgical Navigation Systems” Proceedings of the 2003 IEEE/RSJ
- [WUX03b] Wu, X. Taylor, R ”A Direction Space Interpolation Technique for Calibration of Electromagnetic Surgical Navigation Systems”