Methods for Designing Head-Tracking Probes

Larry Davis Jannick P. Rolland Rebecca Parsons
School of Electrical Engineering and Computer Science
University of Central Florida

Eric Clarkson
Optical Sciences Center
University of Arizona

Abstract

Augmented reality requires real and virtual objects to be registered in three dimensions from any viewing direction. Therefore, accurate, large field of regard head tracking is needed. As a part of a research effort to design probes to track the position and orientation of the head of a user in a virtual environment, an algorithm is provided for the uniform distribution of an arbitrary number of beacons on a spherical probe using simulated annealing. The validity of the algorithm is tested by comparison to the tetrahedron, octahedron, and icosahedron, which are spherical equivalents. Additionally, variations upon the cooling schedule implemented in the algorithm and the effects upon the resulting point distributions are examined. Finally, a successfully constructed head-tracking probe is presented and the generalization of the algorithm to probes of other shapes is discussed.

Keywords: Augmented Reality, Virtual Environments, Head Tracking, Probe Design, Simulated Annealing.

Introduction

One of the most promising and challenging future uses of head-mounted displays (HMDs) is in applications where virtual environments enhance rather than replace real environments [1]. This is referred to as augmented reality. To obtain an enhanced view of the real environment, users wear see-through HMDs to observe three-dimensional computer-generated objects superimposed on their real-world view [2][3]. The position and orientation of each user's head must be obtained to render the computer-generated objects at the correct depth in the field of view of the user and from the correct viewpoint [4][5]. Because virtual and real objects must be placed *into register*, that is spatial coincidence, the need for accurate tracking of head motion is predominant in augmented reality applications [6].

As a driving application for the advancement of such technology, we are developing a Virtual Reality Dynamic Anatomy (VRDA) Tool, illustrated in Fig. 1 [7]. The tool allows medical practitioners to visualize anatomical structures superimposed on their real counterparts. To this end, the medical practitioner wears a HMD that superimposes a graphical model of the knee-joint anatomy on the real knee of a model patient [8][9].

To correctly visualize knee-joint anatomy, the head of the medical practitioner must be accurately tracked. Tracking technologies employed to obtain head position and orientation include time-frequency measurement, spatial scan, inertial sensing, mechanical linkages, and direct-field sensing [10]. We employ a spatial scan, videometric method to maximize tracking accuracy and resolution. The videometric approach consists of using several cameras to acquire different views of a pattern of features, then determining position and orientation based upon the 2D projection of the pattern at each camera.

Fig. 1: (a) The VRDA tool allows superimposition of virtual anatomy on a model patient. The inset photograph contains a rendered frame of the knee-joint bone structures animated based on a kinematic model of motion. (b) An actual view from the VRDA Tool from the viewpoint of the user.

In the videometric approach to tracking, we use the OPTOTRAK 3020, a commercially available optical tracking system. The OPTOTRAK utilizes beacons that emit infrared light and fixed, infrared sensing cameras to determine the position and orientation of an object (e.g. the head of a user). The beacons are mounted in a rigid, fixed arrangement, called a probe, and are activated sequentially within the field of view of the cameras. As long as the cameras detect three beacons from the probe, the position and orientation of the probe

University of Central Florida
Orlando, FL 32816-2700
(407) 823-6830 (407) 823-6880 FAX
Email: jannick@odalab.ucf.edu

can be determined. Thus, if a probe is mounted on the head of the user, an image corresponding to the position and orientation of the probe, which coincides with the user's position and orientation, can be generated by a computer and shown using a HMD.

When designing a head-tracking probe, the specifications may include field of regard, size, accuracy and precision, speed of tracking, and cost. The probe geometry chosen will directly affect these considerations. As an example, consider a planar head-tracking probe with 100 beacons largely spread apart. This probe would provide high accuracy and precision and a large field of regard, but with a reduced tracking speed due to the large number of beacons. Also, the size and cost of this probe would be prohibitive. Instead, by placing the beacons on a toroid above the viewing optics on the HMD or on two halves of a sphere on either side of the HMD, we can "wrap" the probe around the head of the user. This way, we can still obtain a large field of regard and maintain the desired accuracy with fewer beacons, minimizing cost and maximizing tracking speed. Therefore, the trade-off between the desired accuracy and precision, field of regard, and the number of beacons used for a given application makes the design of probes subject to case-by-case requirements and necessitates a model to predict performance based on specifications and parameter choices.

We chose to design a probe with a large field of regard and selected, as a first implementation, a spherical head-tracking probe. To maximize the field of regard for a spherical probe, the beacons must be uniformly distributed, which is analogous to uniformly placing points on the surface of a sphere. A group of points can be uniformly distributed on a sphere for a number of points that is a perfect square (eg. 4, 16, 36 points, etc…) by dividing the sphere into symmetric regions of uniform solid angle. Also, one may use spherical equivalents, such as the Platonic solids, to place 4, 6, 8, 12 or 20 points directly. Additionally, to solve the specific problem of uniformly distributing points upon a sphere for an arbitrary number of beacons, the Gaussian Quadrature method has been previously proposed and implemented. A Gaussian quadrature formula on the two-dimensional sphere is a numerical integration formula, for functions on the sphere, that is exact for all spherical harmonics $Y_l^m(\theta,\phi)$ with $l \leq L$ [11]. The points on the sphere where the function to be integrated is sampled are therefore uniformly distributed. The number of sample points is determined by L, and increases as L increases. The Gaussian quadrature formulae are specific to spheres and make heavy use of the symmetries that spheres possess. However, they are difficult to generalize to other shapes.

Therefore, in developing a method that will be applicable to non-spherical probes with an arbitrary number of beacons, we approach the problem using optimization techniques. Modeling the beacons as charged particles, the cost function we seek to minimize is the potential energy of the distribution of beacons. Given that the potential energy can be expressed as an inverse relationship of the distance between all pairs of beacons on the probe, minimum potential energy implies a symmetric distribution of beacons. Thus, minimize the cost function and uniformly distribute beacons on a probe, we choose to apply the method of simulated annealing.

Published by Metropolis et al. [12], the method of simulated annealing was first used in an algorithm designed to simulate the cooling of material in a heat bath. The Metropolis algorithm simulates the change in energy of a system when subjected to a cooling process, until it converges to a steady "frozen" state. In 1983, Kirkpatrick suggested that this type of simulation could be used more generally to search the feasible solutions of an optimization problem, with the objective of converging to an optimal solution [13][14].

Simulated annealing lends itself well to our approach because of the well-specified criteria for point movement to an optimal solution during iterations of the algorithm. Furthermore, during optimization, simulated annealing includes methods for more easily escaping local minima. It is important to note, however, that other optimization methods can be used to solve this problem. As such, it is not our intention to explore the advantages of using one optimization algorithm versus another.

In the following pages, we describe our variation upon the "classical" simulated annealing algorithm. We then describe the generalization of the algorithm to arbitrary, convex shapes. Next, we provide results of our algorithm on a sphere, compared to the tetrahedron and octahedron. Finally, we present results of the annealing algorithm in the construction of a spherical head-tracking probe.

The Algorithm

The algorithm employed is based closely on the algorithm used in [12]. To start, an initial temperature is set and the number of points to be distributed is chosen. The points are placed randomly on a sphere of unit radius.

The annealing process begins with the computation of the starting value of the potential energy, *E*, of the system, which varies inversely with the distances between all the points on the sphere. The formula used for computing the energy is:

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 / r_{ij} \qquad (1)$$

where r_{ij} is the Cartesian distance between the i^{th} and j^{th} point. After computing the starting energy of the system, an iterative process begins. First, the system temperature is decreased in a systematic manner referred to as a cooling schedule. In our case, the temperature is reduced by 0.5% percent of the temperature at the previous iteration. During the next step, one point is chosen randomly and moved. The point is moved between 0 and 0.005 radians in a random direction with respect to the center of the sphere. The current energy of the system is then calculated using (1). If this current energy is less than the previous system energy, then the move is accepted. If the current energy is more than the previous system energy, a probability of accepting the move is generated based upon the current temperature. The equation used for computing the probability of accepting a point movement is:

$$P = \frac{e^{\frac{-\Delta}{T}}}{1 + e^{\frac{-\Delta}{T}}}, \qquad (2)$$

where Δ is the change in energy of the distribution due to the point movement, and T is the current temperature. A random number between zero and one is generated, determining whether the point is moved. If the random number is less than or equal to the probability in (2), the select point moves. Otherwise, the selected point remains stationary. In either case, another point is selected and moved following the same criteria until 100 iterations have been performed at the current temperature.

After 100 point movements, the temperature is decreased and the process is repeated at the next (lower) temperature. When the system temperature is less than 0.1, the iterative process stops and the system freezes. The final energy value and the distribution of the points are then displayed.

Differences between the algorithm presented here and that employed by Metropolis appear in the implementation of the cooling schedule. Instead of using an exponentially decreasing cooling schedule, we chose a geometric temperature reduction, decreasing the system temperature by 0.5% after each 100 point movements. At higher temperatures, the temperature change is more dramatic in both approaches. As the system cools, the temperature is decreased at a slower pace. Such cooling approaches allow the system greater fluctuation at higher temperatures while allowing it to “settle down” as it reaches lower temperatures. We proposed this geometric temperature reduction because of its simplicity. Also, we employed a simplification of the traditional probability expression, as shown in (2).

Generalization of the Algorithm to Other Shapes

The algorithm is generalized by recognizing that a symmetric distribution of beacons on a probe corresponds to having uniform arc length between any beacon on the probe and each of its immediate neighbors. Because we are designing probes for head tracking, the algorithm is generalized to regular geometries, meaning probes without irregular protrusions.

The generalized energy function we seek to minimize is:

$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} 1 / A_{ij}, \qquad (3)$$

where A_{ij} is the arc length between the ith and jth beacon. Depending upon the distance of the probe from the tracker, the maximum arc length between a pair of beacons can be determined, given that three beacons must remain in the tracker field of view at any time. This arc length requirement constrains the minimization process.

In generalizing the algorithm, the simplest case is a planar probe or a probe with planar surfaces. Because a plane has infinite curvature, placing the beacons at an equal arc length corresponds to placing the beacons with a uniform Cartesian distance between them. Thus, the beacons may be placed on the probe surface immediately with a Cartesian distance equivalent to the arc length requirement as previously discussed.

The other case we consider is that of a curved surface. Of the curved surfaces, the simplest case is a sphere. Because a sphere is a surface with uniform radius of curvature, we are able to approximate the arc length criterion using Cartesian distances. The more complex case is a surface with a different radius of curvature in two directions. This includes cylindrical and toroidal shapes. In this instance, we parameterize the desired surface and determine the arc lengths along the directions of principal curvature. The minimization process is then initiated. This procedure further applies to other, general probe shapes because of the fact that curved surfaces may be approximated using toroidal sections.

Results and Discussion

The simulated annealing algorithm was implemented in C++. The elapsed time for distributing the particles was determined by using the time() function. All file access activities were performed so as not to be a factor in measuring the time of execution.

The tetrahedron, octahedron, cube, icosahedron, and dodecahedron, the Platonic solids, have 4, 6, 8, 12, and 20 vertices, respectively. These polygons are spherical equivalents because for each representation, a sphere can be specified such that all the vertices of the solid satisfy its equation. Moreover, the Platonic solids are the maximally symmetric for a point distribution with 4, 6, 8, 12, or 20 points. Of the Platonic solids, the tetrahedron (4 vertices), the octahedron (6 vertices) and the icosahedron (12 vertices) are also point distributions that posses minimum potential energy.

Because the Platonic solids provide known, minimum energy outcomes, the results of the distribution algorithm for 4, 6, and 12 points were compared to the platonic solids for validation of the simulated annealing algorithm. Results show that the point distributions matched the platonic solids closely. The difference in the potential energy, ΔE, of the reference tetrahedron and the tetrahedron obtained via simulated annealing was 0.00001 (this result is unit-less because normalized distances were used to compute the potential energies). The ΔE between the reference and annealed octahedron was 0.000001 and the ΔE for the icosahedron was 0.0001. The final annealed point distributions compared to the reference tetrahedron and octahedron are shown in Fig. 2.

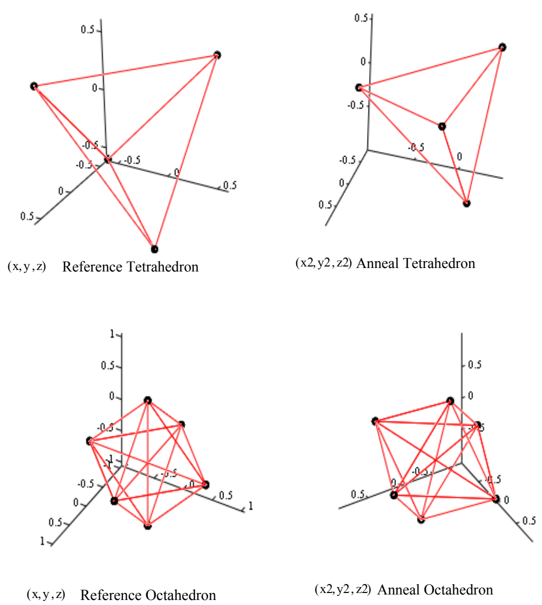

Fig 2: A validation of the algorithm with Platonic solids is demonstrated. Theoretical and annealed point distributions for (a) the tetrahedron and (b) the octahedron are shown.

It was observed that the factors that most influenced the performance of the algorithm were the cooling schedule and the number of iterations performed at each temperature. It was observed that the cooling schedule could be changed to a 1.5% decrease to improve the algorithm speed. The improvement in speed is almost 66%, without a significant loss of effectiveness in the distribution. The improvement is more dramatic for larger numbers of points. Future versions of the algorithm will be implemented with a 1.5% cooling schedule. In addition, as the initial starting temperature was lowered, the time of execution of the algorithm improved. However, the point distribution was not as even as implementations using a higher initial starting temperature.

The annealing results were further validated by the construction of a spherical head-tracking probe, shown in Fig. 3. The probe consists of 24 infrared LEDs distributed on a spherical shell 7cm in diameter and was constructed via rapid prototyping. The head probe is capable of tracking user head motion 360 degrees in the transverse plane (i.e, the user shaking her head “no”). Also, the head probe can track user head motion 180 head “yes”). The head probe reports the user position with an accuracy of 0.1 mm and a σ of 0.03 mm. The head probe reports user rotation with an accuracy of 1.38 degrees. Further optimization of the probe should improve rotational accuracy.

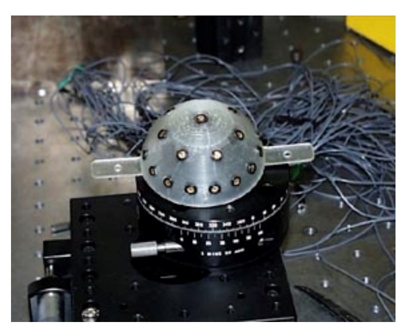

Fig. 3: A 24-LED spherical head tracking probe

In the future, we shall use these simulation results within a general framework for the design of tracking probes for virtual environments. The results achieved here may also be applied to other applications, such as the positioning of satellites orbiting the earth, information theory, or the placement of dimples on a golf ball.

Acknowledgements

The authors would like to thank Boris Zel'dovich for his discussions concerning Platonic solids and Alexei Goon for his suggestions concerning energy functions. This work was supported by the National Library of Medicine, Grant 1-R29-LM06322-01A1 and by the Florida Education Fund

References

[1] Rolland, J.P., A. Yoshida, L. Davis, and J.H. Reif, "High resolution inset head-mounted display", *Applied Optics*, 19, pp. 4183-4193, 1998.

[2] Bajura, M., H. Fuchs, and R. Ohbuchi, Merging virtual objects with the real world, *Computer Graphics*, 26, pp. 203-210, 1992.

[3] Rolland, J.P., and H. Fuchs, "A comparison of optical and video see-through head-mounted displays", in Wearable Computers and Augmented Reality, T. Caudell and W. Barfield. (Eds.), Erlbaum, 2000.

[4] Rolland, J.P. and W. Gibson, "Towards quantifying depth and size perception in virtual environments", *Presence: Teleoperators and Virtual Environments*, 4, pp. 24-49, 1995.

[5] Rolland, J.P., A. Quinn, and K. Arthur, "Quantifying accuracy and precision of rendered depth in head-mounted displays", Technical Report TR99-003, University of Central Florida, 1999. (Also submitted to J. Opt. Soc. Am. A.)

[6] Holloway, R., An analysis of registration errors in a see-through head-mounted display system for craniofacial surgery planning, Ph.D. dissertation, University of North Carolina at Chapel Hill, 1995.

[7] Wright, D.L., J.P. Rolland, and A.R. Kancherla, "Using virtual reality to teach radiographic positioning", *Radiologic Technology*, 4, pp. 167-172, 1995.

[8] Baillot, Y., J.P. Rolland, K. Lin, and D.L. Wright, "Automatic modeling of knee-joint motion for the virtual reality dynamic anatomy (VRDA) tool," *Presence: Teleoperators and Virtual Environments*, 9(3), pp. 223-235, 2000.

[9] Argotti, Y., L. Davis, V. Outters, and J.P. Rolland, "Dynamic superimposition of synthetic objects on rigid and simple-deformable real objects", *Proceedings of ISAR '01*, pp. 3-11, 2001.

[10] J.P. Rolland, L Davis, and Y. Baillot, "A survey of tracking technology for virtual environments", in Wearable Computers and Augmented Reality, T. Caudell and W. Barfield. (Eds.), Erlbaum, 2001.

[11] S.L. Sobolev, "The formulas of mechanical quadrature on the surface of a sphere", *Sibersk. Mat. \u Z.*, 3, pp. 769-796, 1962.

[12] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculation by fast computing machines", *Journal of Chemical Physics*, 21, pp. 1087-92, 1953.

[13] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing", *Science*, 220, pp. 671-680, 1983.

[14] Dowsland, K., "Simulated annealing", in Modern Heuristic Techniques for Combinatorial Problems, C.R. Reeves (Ed.), Wiley & Sons, 1993.