Marker Mapping Techniques for Augmented Reality Visualization

Felix G. Hamza-Lup[1], Larry Davis[1], Charles Hughes[1], and Jannick P. Rolland[1,2]
[1]*School of Electrical Engineering and Computer Science*
[2]*School of Optics-CREOL*
University of Central Florida
Orlando, FL 32816
{fhamza, ceh}@cs.ucf.edu, {davis, jannick}@odalab.ucf.edu

Abstract

The requirements for tracking in augmented reality environments are stringent because of the need to register real and computer-generated virtual objects. Driven by the need to track real objects within these environments, we propose two algorithms to distribute markers on complex rigid objects. The proposed algorithms employ an optimization technique with a spherical or cylindrical intermediary surface. The validity and effectiveness of the algorithms are tested heuristically by simulation.

1. Introduction

Placing real and computer-generated objects into register is a challenging problem in augmented reality (AR). Because virtual and real objects must be placed into register, i.e. spatial coincidence, the need for accurate tracking not only for the head of the user but also for other objects is predominant [1][2].

The research presented in this paper is a component in the development of a comprehensive framework for the Distributed Augmented Reality Collaborative Environment referred as DARE. Applied to medical visualization, DARE allows human anatomical 3D models to be overlaid on real patient models or human patient simulators, and the data to be shared across remote locations. At remotely dispersed locations users obtain an enhanced view of the real environment, by wearing see-through head-mounted displays (HMDs) to observe three-dimensional computer-generated objects superimposed on their real-world view [3]. The position and orientation of the real objects must be computed to render the computer-generated objects from the correct viewpoint at the correct depth.

Thus, the requirements for tracking in DARE as in other AR environments are extremely stringent whether the specific application is a training tool, a diagnosis tool or an aid to guided surgery [4]. An approach to tracking real objects is marker-based, optical tracking technology where markers are distributed on the object's surface. However, in the case of complex objects, *ad hoc* methods for marker distribution may waste resources and/or restrict tracking performance.

We propose two algorithms for marker mapping on complex rigid objects. The first algorithm, referred to as the *quiescent algorithm,* approximates a uniform distribution for a specified number of markers on the surface of the object. An iterative optimization process determines the number of markers.

The second algorithm, referred to as the *viewpoint algorithm,* minimizes the number of markers while keeping the constraint that at least k markers are seen (detected) from different viewpoints. The number of markers required to determine an object's position and orientation from each viewpoint is dependent on the tracking system used.

2. The quiescent algorithm

In this section, we describe a three-step algorithm for placing sets of markers onto complex rigid objects. The name of the algorithm is derived from the second step. Quiescence refers to the minimum potential energy (resting) state of a set of points that can be achieved via optimization.

The algorithm requires the availability of a three-dimensional triangular mesh model of the object. The problem of mapping from 2D surfaces onto complex 3D objects is encountered in texture mapping [5]. Conceptually, the texture mapping process is simple. A small area of the texture pattern maps onto the area of a geometric surface. However, in the case of a complex 3D object, its surface equations can be difficult to approximate. To solve this problem, a two-part, texture-based mapping technique is used. The technique uses intermediary three-dimensional regular surfaces.

The first step determines whether to use a cylinder or a sphere as an intermediary surface. The decision is based upon the application tracking requirements and the elongation of the object. If the object is elongated, the principle axis of symmetry is determined by eigenvector analysis and a cylinder is used as the intermediary surface. Otherwise, a sphere is used as the intermediary surface.

The extent of elongation of a complex object can be quantified by an assessment of the eigenvalues of the dispersion matrix computed from the vertices of its 3D triangular mesh. For a complex object, the ratio between the largest and the smallest eigenvalue is considered. If this ratio is greater than or equal to 10, we choose a cylinder over a sphere for the mapping.

Once the object elongation has been quantified, the main axis of symmetry is determined by the eigenvector corresponding to the strongest eigenvalue.

Let p be the centroid of the triangular mesh that represents the complex object. We compute the Cartesian coordinates of the centroid in the global coordinate system. Let n be the total number of vertices (points) in the three-dimensional triangular mesh that approximates the complex object. Let p_i be the position of the i^{th} vertex in the mesh. Let $d_i = d_i - p,\ i \in [1,n]$ the vectors between the centroid and each vertices in the mesh. The 3x3 symmetrical dispersion matrix is:

$$A = \sum_{i=1}^{n} d_i d_i^T \quad \text{,where } d_i^T \text{ is the transpose of } d_i.$$

To determine the eigenvectors of this matrix, we diagonalize it, given that A is real and symmetric. The diagonalized A can be written as:

$$D = V^{-1} A V$$

where $D_{ij} = \lambda^{[j]},\ i=j$ is the matrix of eigenvalues and V is the matrix of eigenvectors.

In the second step of the quiescent algorithm, we use an optimization procedure to uniformly distribute the markers on the cylinder or on the sphere. Using optimization, an initial number of markers are distributed on the intermediary surface. Then, the minimum number of markers that meet the tracking system detection requirements is computed in an iterative process [6]. Simulated annealing lends itself well to this approach because of the well-specified criteria for point movement to an optimal solution during iterations of the algorithm [7]. Furthermore, during optimization, simulated annealing includes methods for escaping some 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 third step, the markers are mapped from the intermediary surface to the desired complex 3D object in an approach similar to two-way texture mapping. This can be done using the normal from the intermediary surface, the normal from the object's surface, or the center or principal axis of the object. The algorithm we developed uses the center or principal axis of symmetry of the object in the mapping process. A description of the geometrical formulae for cylindrical and spherical intermediary surfaces is given in Appendix A and B, respectively.

The inputs of the quiescent algorithm are:

1. A 3D Model: the set of n points defined in Cartesian coordinates (X_i, Y_i, Z_i), where $i \in [1,n]$, n being the number of vertices in the three dimensional triangular mesh approximating the complex object onto which the markers are mapped.
2. A set of intermediary marker positions: the set of k points defined in Cartesian coordinates (X_{Mi}, Y_{Mi}, Z_{Mi}), where $i \in [1,k]$, k being the number of points that describe the initial positions of the marker centers on the intermediary surface after the annealing procedure from the second step.

The output is a set of k points describing the final positions of the marker centers on the object's surface. Each point is defined by the Cartesian coordinates (X_{Fi}, Y_{Fi}, Z_{Fi}), where $i \in [1,k]$, k being the number of markers.

3. The viewpoint algorithm

Similar to the quiescent algorithm, the purpose of this algorithm is to distribute a finite number of markers on a complex 3D object. The improvements over the previous algorithm are twofold. First, this algorithm further minimizes the number of markers. Second, it guarantees that at least k markers are visible from each viewpoint, where k is the minimum number of markers that are necessary for a given tracking system.

The viewpoint algorithm can be described in four steps:

1. A triangular mesh is generated for the complex 3D object. Each triangle of the mesh is assigned a different number.
2. The number and the positions of the viewpoints arround the complex object are selected. A higher number of viewpoints will give better results because this is equivalent to analyzing the object from more angles. Hence, more viewpoints will give better marker positions on the complex object. Moreover, the probability that at least k markers are visible will increase when the object changes

position and orientation in the tracking frame of reference. To distribute the viewpoints around the object, we place them uniformly on the sphere that surrounds the object and has a radius two times the maximum distance between any two points in the triangular mesh. All the viewpoints are added to the ViewpointsList.

3. For each triangle, the number of viewpoints from which it can be seen is computed. We create the TrianglesList. If two triangles have the same viewpoint count we sort them by the number assigned in step 1. We create a list having in each node the triangle number and the number of viewpoints from which this triangle can be seen.
4. As long as the ViewpointsList is not empty:
   - 4.1 Select the triangle with the highest viewpoint count that is next in the TrianglesList and add a marker on its surface.
   - 4.2 If fewer than k markers from that viewpoint are seen, step 4.1 is repeated.
   - 4.3 Else the viewpoint is removed from the ViewpointsList and the TrianglesList is updated.

The algorithm assures that from each viewpoint at least k markers are visible and minimizes the number of markers.

4. Experimental Results

To heuristically test the validity of the algorithms, software simulations were performed. Computer-generated, random, three-dimensional triangular meshes representing the complex objects were used.

4.1 The quiescent algorithm

The markers on the intermediary surface after annealing are represented as small spheres on the transparent surface. The small spheres on the object represent the final positions of the markers on the surface of the complex object.

The first set of simulations was performed with a sphere as an intermediary surface, shown as a transparent surface in Figure 1.

The 3D scene contains:

- a randomly generated 3D triangular mesh consisting of 10 triangles
- 30 markers
- a sphere of radius R as an intermediary surface, where R is greater then the maximum distance between any two points in the triangular mesh.

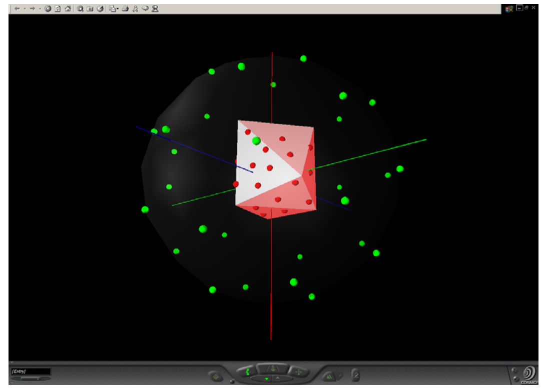

Figure 1: Sphere as intermediary surface

The second set of simulations was performed with a cylinder as an intermediary surface, shown as a transparent surface in Figure 2.

The 3D scene contains:

- a randomly generated 3D triangular mesh consisting of 10 triangles
- 24 markers
- a cylinder of radius R and height H as an intermediary surface, where $2R$ and H are greater then the maximum distance between any two points in the triangular mesh.

The validity of the algorithm on complex objects was tested heuristically. We aligned the tracking system camera with the scene camera and rotated the object. We observed that at least 3 markers per viewpoint were seen and that there was a fairly uniform distribution of markers on the surface of the object.

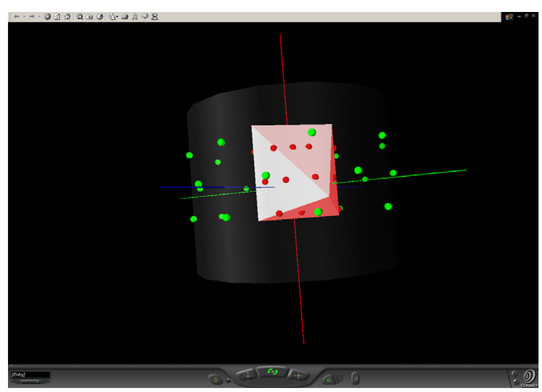

Figure 2: Cylinder as intermediary surface

4.2 The viewpoint algorithm

The uniformly distributed viewpoints on the surface of the sphere surrounding the 3D object are represented as small cubes in Figure 3. The small spheres on the object represent the final position of the markers.

The 3D scene contains:

- a randomly generated 3D triangular mesh consisting of 10 triangles
- 30 viewpoints uniformly distributed on the bounding sphere that surrounds the complex object.
- a sphere of radius *R*. *R* is greater than double the maximum distance between any two points in the triangular mesh.

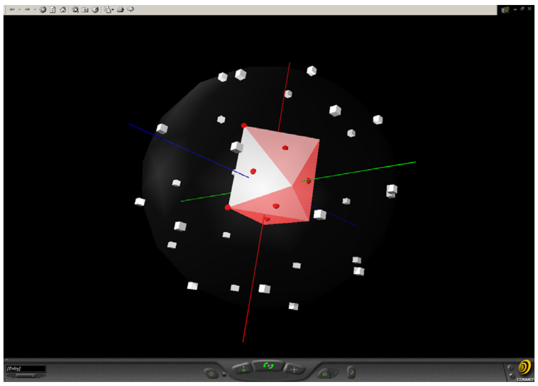

Figure 3: The viewpoint approach

The current implementation places the markers at the center of mass (centroid) of the triangles. Most optical tracking systems require at least 3 markers visible from each viewpoint to correctly determine the position and orientation of the object. If only one triangle is seen from a viewpoint, the other markers are distributed on the vertices that form the triangle.

5. Conclusions

Two algorithms are proposed for marker distribution on complex rigid objects. The experiments demonstrate the success of the algorithms applied on randomly generated complex rigid objects.

In addition to further verification, there are several issues that still need to be addressed. One issue is the type of markers: active versus passive. Another issue is accounting for the cones of emission for different type of active markers and investigating their impact on the marker distribution and orientation.

6. Acknowledgements

This work was supported in part by the US ARMY Simulation, Training, and Instrumentation Command (STRICOM) and by the National Science Foundation under grant EIA 9986051.

7. References

[1] Welch G., and G. Bishop "SCAAT: incremental tacking with incomplete information," Proceedings of SIGGRAPH '97, in Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 333-344, 1997.

[2] Argotti, Y, L. Davis, V. Outters, and J.P. Rolland "Dynamic superimposition of synthetic objects on rigid and simple-deformable real objects," *Computers and Graphics*, 26/6/2002.

[3] Rolland, J.P., and H. Fuchs, "A comparison of optical and video see-through head-mounted displays", Chap. 4 in Wearable Computers and Augmented Reality, T. Caudell and W. Barfield. (Eds.), Lawrence Erlbaum Associates, Mahwah, 113-156, 2001.

[4] 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.

[5] Blinn, J.F, M.E. Newell, "Texture and reflection in Computer Generated Images", CACM, 19(10), 542-547, 1976.

[6] Davis L, J.P. Rolland, R. Parsons, E. Clarkson, "Methods for designing head-tracking probes", Proceedings of the Joint Conference on Information Sciences (JCIS), 498-502, 2002.

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

Appendix A
Cylinder as intermediary surface

1. We align the principal symmetry axis of the object with the cylinder axis, e.g. the OX axis.
2. Let $(X_{Marker}, Y_{Marker}, Z_{Marker})$ be the Cartesian coordinates of a point in the markers intermediary position file (i.e. after the annealing algorithm has been applied on the cylinder). For each point, we find the equation of the section plane (SP) that is parallel to the OZY plane and passes through the point. This plane is unique.
3. From the triangular mesh that represents the 3D complex object we isolate the set of triangles (TS) that are intersected by SP using a range factor (RF) based on the granularity of the triangular mesh.
4. For each triangle we isolate its section segment by computing the intersection point between a line segment determined by two vertices of the triangle and the SP.

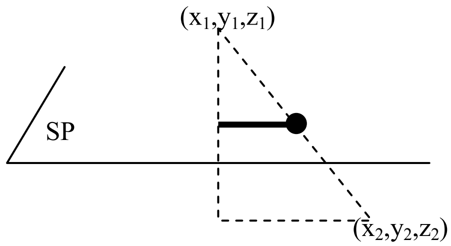

Figure A1: Section segment

Using the plane equation, we identify the coefficients. In this case SP is parallel with the OZY axis and passes through the point (X_{Marker},Y_{Marker},Z_{Marker}) hence we have: A=1, B=0, C=0 and D= (-)X_{Marker}. The SP equation is given by:

$$x - x_{Marker} = 0 \quad (A1)$$

The parametric equations of the line that passes through two points (x_1,y_1,z_1), (x_2,y_2,z_2) are:

$$x = x_1 + (x_2 - x_1)t$$

$$y = y_1 + (y_2 - y_1)t$$

$$z = z_1 + (z_2 - z_1)t$$

Substituting equation A1 into the first parametric equation for the line yields:

$$x_{Marker} = x_1 + (x_2 - x_1)t$$

$$y = y_1 + (y_2 - y_1)t$$

$$z = z_1 + (z_2 - z_1)t \text{ , unknowns: t, y, z.}$$

The solution of this system is given by:

$$t = \frac{x_{Marker} - x_1}{x_2 - x_1}$$

$$y = y_1 + (y_2 - y_1)\frac{x_{Marker} - x_1}{x_2 - x_1}$$

$$z = z_1 + (z_2 - z_1)\frac{x_{Marker} - x_1}{x_2 - x_1}$$

5. We repeat step 4 for each triangle in TS until we obtain all the section segments.
6. We repeat step 2 for each point in the markers intermediary positions file.

Appendix B
Sphere as intermediary surface

1. The centroid of the general shape is computed by applying the arithmetic mean on each dimension:

$$x_c = (x_1 + x_2 + \ldots + x_n)/n$$

$$y_c = (y_1 + y_2 + \ldots + y_n)/n$$

$$z_c = (z_1 + z_2 + \ldots + z_n)/n$$

2. Let (X_{Marker},Y_{Marker},Z_{Marker}) be the Cartesian coordinates of a point in the markers intermediary position file (i.e. after the annealing algorithm has been applied on the sphere). The parametric equations of the line that passes through the points (X_c,Y_c,Z_c), (X_{Marker},Y_{Marker},Z_{Marker}) are given by:

$$x = x_c + (x_{Marker} - x_c)t$$

$$y = y_c + (y_{Marker} - y_c)t$$

$$z = z_c + (z_{Marker} - z_c)t$$

3. For each triangle, the intersection between the plane generated by the triangle and this line is computed. Having three points: (X_1, Y_1, Z_1), (X_2, Y_2, Z_2), (X_3, Y_3, Z_3), the equation of the plane that passes through them is given by: Ax+By+Cz+D=0, where:

$$A = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$

$$B = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$

$$C = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$

$$D = x_1(y_3z_2 - y_2z_3) + x_2(y_1z_3 - y_3z_1) + x_3(y_2z_1 - y_1z_2)$$

From the plane equation and the parametric equation of the lines we can find the intersection point (X_{Sol},Y_{Sol},Z_{Sol}). Then we check whether each point is inside the triangle. While this operation can be done in several ways, we check if the point is inside 2 angles of the triangle.

4. Step 3 is repeated for each marker in the input file.