Technologies for Augmented Reality: Calibration for Real-Time Superimposition on Rigid and Simple-Deformable Real Objects

Yann Argotti[1], Valerie Outters[2], Larry Davis[1],
Ami Sun[2], and Jannick P. Rolland[1,2]

[1] School of Electrical Engineering and Computer Science
[2] School of Optics
University of Central Florida
4000 Central Florida Boulevard
Orlando, Florida 32816-2700, USA
email: jannick@odalab.ucf.edu

Abstract. A current challenge in augmented reality applications is the ability to superimpose synthetic objects on real objects within the environment. This challenge is heightened when the real objects are in motion and/or are non-rigid. Yet even more challenging is the case when the moving real objects involved are deformable. In this article, we present a robust method for calibrating marker-based augmented reality applications to allow real-time, optical superimposition of synthetic objects on dynamic rigid and simple-deformable real objects. Moreover, we illustrate this general method with the VRDA Tool, a medical education application related to the visualization of internal human knee joint anatomy on a real human knee.

1 Introduction

In a large range of fields, the ability to enhance reality with synthetic information is an exciting alternative to traditional methods of acquiring information. Applications where computer-generated objects are employed to augment user perception of the real environment are referred to as augmented reality (AR) applications.

A current challenge in AR applications is the ability to superimpose synthetic objects on real objects within the environment. To overcome this challenge, objects in the environment must be accurately tracked and the relationships between real and synthetic objects must be precisely determined. When dealing with medical AR applications, the real and synthetic objects in the environment are often human anatomical structures.

Accurately tracking human motion is a difficult task. However, there have been attempts to understand and quantify human motion. Spoor and Veldpaus published a method for calculating rigid body motion from the spatial coordinates of markers that has been adapted to tracking skeletal motion [13]. In addition, techniques have been devised that address the problems associated with accurately tracking anatomical motion [1][6][12]. In fact, much is known about the motion of anatomical structures, but they still pose a significant challenge to inclusion within AR systems. Aside from the fact that markers cannot be directly positioned on bones in daily settings, anatomical structures are not rigid. Moreover, attempting to track anatomical structures at interactive speed while maintaining registration of synthetic objects is especially challenging.

Thus, as a contribution to the body of virtual environment research, we examine the problem of tracking simple-deformable bodies within an AR application and present a general method for calibrating marker-based AR systems, which may result in accurate dynamic superimposition of synthetic objects on real objects at interactive speeds. In the following paragraphs, we present a marker-based calibration method for AR applications. In conjunction with a dynamic superimposition method found in [2], the method is applied to the Virtual Reality Dynamic Anatomy (VRDA) Tool, a visualization system developed for the study of complex anatomical joint motions [14].

2 Calibration Process

In the calibration process, the AR system is prepared for accurate superimposition. This includes computing the local coordinate system, or local frame of reference, for each real object in the environment, determining the transformations between real objects and their respective synthetic counterparts, and characterizing the stereoscopic display device. We treat the tracker coordinate system as the global frame of reference, or frame for short. In addition, we define simple-deformable objects as objects that are slightly changing in shape compared to an equivalent rigid object, which we show can be quantified by the change in the eigenvalues of the dispersion matrix associated with a cluster of markers.

2.1 Computing Local Frames for Real Objects

To determine the location and orientation of a real object, we choose to adopt a marker-based method because it offers a more accurate way to determine the position and orientation of real objects. Each real object is defined by a set of markers placed on its surface. Whatever type of marker-based tracking system is selected, we assume the system can provide the three-dimensional (3D) location of each marker in its frame of reference. However, this information is not sufficient to link real and synthetic objects; each marker location must be computed in a local frame defined for each real object.

For a rigid object, we compute the fixed marker locations in the local frame. Similarly, in the case of a simple-deformable object, we first make the assumption that the markers do not move significantly during the calibration phase. The relative motion of each marker in its local frame will be assessed in a later step and accounted for. In both scenarios, we can establish a local coordinate frame using eigenvalues and eigenvectors obtained from the locations of the markers on the object [1][6]. For rigid objects, the eigenvalues of a matrix characterizing the marker distribution are invariant, even though its eigenvectors vary according to the position and orientation of the object in the global frame or any other frame. This extends to simple-deformable objects under the previous assumption.

Thus, we take advantage of this invariance property to build a local marker frame, defined with the eigenvectors of a matrix that characterizes the marker set. A matrix that meets this criterion is the dispersion matrix, which is a form of the spatial covariance matrix over the marker distribution. We denote the coordinates of the centroid of the marker distribution y, the total number of markers n, and the coordinates of the i^{th} marker in the global frame y_i. We also define $y'_i = y_i - y, i \in [1; n]$. Thus, the 3 x 3 symmetrical dispersion matrix, K, is given by

$$K = \sum_{i=1}^{n} y'_i {y'_i}^T \quad . \tag{1}$$

We define the local frame as the eigenvectors of K, which will allow direct computation of the required transformation for the local frame to the global frame as now detailed. To determine the eigenvectors, we first diagonalize K, given that K is real and symmetric. The diagonalized K can be written as

$$diag\,(k_1, k_2, k_3) = V^{-1} K V \tag{2}$$

where k_1, k_2, and k_3 are the eigenvalues of K and V is the matrix of eigenvectors. Provided that V has an inverse, the relationship described in (2) is a similarity transform. An important property of similarity transforms is that eigenvalues are preserved, meaning K and its diagonalized version can be related to one another using an eigenvector basis. In this case, V is the eigenvector basis between K and its diagonalized version. The frame where K is diagonal is unique. We then sort the eigenvalues such that $k_1 > k_2 > k_3 > 0$ and recompute K using (2). V is the orthonormal matrix that transforms the frame where K is diagonal to the frame where K was first computed, meaning that V is the transformation from the local frame to the global frame. In addition, the inverse of V is equal to its transpose because V is an orthonormal matrix. Thus, we can express the local coordinates, $x_i, i \in [1; n]$, of the markers in a frame whose origin is the centroid, y, of the marker set and whose axes are given by the eigenvectors of the matrix K, with the relationship expressed as

$$x_i = V^T y'_i,\; i \in [i; n] \quad . \tag{3}$$

The process previously described is valid when the tracking system detects all the markers on an object simultaneously. However, all the object markers may not be visible simultaneously, a situation that occurs when tracking systems are subject to occlusion.

We now present an extension of the method to compute the local coordinates of all markers, whether simultaneously visible or not. First, a local frame is built from the coordinates of the visible markers. At least three markers must be first detected. Then, the transformation from the local frame to the global frame, V, is determined. Once V is computed with a limited number of visible markers, we determine the coordinates of all the visible markers in the local frame, which we refer to as a partial local frame.

Next, the real object is slowly rotated. This allows previously undetected makers to be detected with at least three markers that were previously detected. We call these previously undetected markers, new markers, and the markers that were detected previously, old markers. When the new markers are detected, the rotation matrix, R_{ro_o}, and translation vector, T_{ro_o}, from a partial local frame to the global frame are calculated. A scaling transformation is unnecessary at this stage because we are using normalized coordinate systems.

The technique used for finding the rotation matrix and translation vector is Singular-Value Decomposition (SVD) [4]. We chose this method because it is a robust optimization method that always gives a solution, and, moreover, it is the best solution by projection on the solution space. The computation of the rotation matrix and translation vector, as well as a modification to the SVD method to account for hidden markers and simple-deformable objects, can be found in [2]. The inputs to the SVD are the computed marker coordinates x_i in the local frame and the measured marker coordinates y_i in the global frame. After the rotation matrix R_{ro_o} and the translation vector T_{ro_o}, are determined from the optimization procedure, we compute the local coordinates of the new markers in the current local frame as

$$x_i = {R_{ro\_o}}^T \left(y_i' - T_{ro\_o}\right) \quad . \tag{4}$$

Finally, with all the new and old markers expressed in the current local coordinate system, we again compute the dispersion matrix and eigenvalues (see equations 1–3) to find an estimate of the new local frame and then compute the new local coordinates of the markers. We repeat this process as the real object is rotated until all the markers have been visible. This technique allows us to build a local frame for any kind of marker set and quickly calculate the local coordinates of the markers.

2.2 Determining Transformations Between Synthetic and Real Objects

After computing the local frames of real objects, we then associate the synthetic representation of an object to its corresponding real object. To link a real object and its synthetic representation, the transformation matrix from the synthetic object frame to the real object frame is computed. Because the definition of a synthetic object frame can be arbitrary, extra data must be provided to build the transformation matrix. To supply the additional information needed, landmarks are defined. These landmarks are corresponding points of significance on the real and synthetic objects. By utilizing landmarks and decomposing the transformation matrix to a rotation, scaling, and translation component, we can specify a relationship between the real and synthetic worlds as $y = RSx + T$.

We shall describe how we first solve for the scaling, S, due to the possibility of system noise and/or measurement errors distorting the size of the objects, and apply it to the entire synthetic object. Then, the rotation, R, and translation, T, are computed using a modified SVD method, with the real and synthetic landmark coordinates as input. The solution of R and T is detailed in [2].

Solving for S consists of computing the mean scaling, S_{mean}, for all the landmarks and deciding if uniform scaling by S_{mean} is appropriate. The mean scaling is determined by calculating the ratio of the distance between two landmarks in the real object frame and the distance between two landmarks in the synthetic object frame for all two- landmark combinations. The ratios are summed for each corresponding pair of landmarks and averaged as

$$S_{mean} = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\frac{d(y_i, y_j)}{d(x_i, x_j)} \right)}{C_2^n}, i \neq j \qquad (5)$$

where, y_i is the i^{th} landmark in the real object frame, x_i is the corresponding i^{th} landmark in the synthetic object frame, and n is the total number of landmarks. It is important to note that the equation for finding the mean scaling is independent of any rotation and translation transformation of the landmarks. We also compute the standard deviation of the mean scaling, S_{sd}. If the value of S_{sd} is under a particular threshold level, the variations in scale are small enough to conclude that the scaling is uniform in three dimensions. The calculation of the threshold level is explained in [3]. If uniform scaling is sufficient, then $S = S_{mean} \cdot I$, where I is the identity matrix. If uniform scaling is insufficient, then scale parameters must be computed for the x, y, and z directions, respectively.

2.3 Characterizing the Display Device

The goal in characterizing the stereoscopic display device is to determine the correct viewpoint locations for each eye, the field of view of the optics, as well as account for optical distortions in order to correctly display synthetic objects. The stereo display device for AR applications is typically a Head-Mounted Display (HMD), of which there are three types: optical see-through, video see-through, and projective [9][7]. The display configuration is fixed and markers are placed upon the HMD to compute the changing location and orientation of the head of the user.

The choice of the local frame for the head of the user is not arbitrary. We set the head frame origin, O_h, at the middle of the line segment between the eyes of the user. The x-axis of the head frame, X_h, is along the segment, with +x oriented toward the right eye of the user. The y-axis, Y_h, is oriented along the line segment perpendicular to X_h , with +y oriented in the direction of the view of the user. Finally, the z- axis, Z_h, is obtained from X_h and Y_h with a cross product: $Z_h = X_h \times Y_h$.

To create a stereoscopic view, we must also define a transformation to the points from which the scene is rendered, referred to as the eyepoints. The choice of eyepoint location is important in minimizing depth perception errors [10]. The location of the eyepoints can be calculated as a translation from O_h of half of the inter-pupilary distance (IPD) along the X_h axis in the +x and -x directions (right and left eye, respectively). Thus, the transformations from the head frame to the left and right eyepoints are a translation of $-IPD/2$ and $IPD/2$, respectively. In addition to transformation matrices, we also define a perspective transform for each eye based upon the display device field of view [8].

Display devices with large fields of view are subject to optical distortion. This distortion is a warping of the image that can be calculated and then corrected optically, electronically, or within the rendering software. Optical correction and electronic correction may not be feasible for a given display device. Thus, we quantify the optical distortion within a system using either a theoretical model or a metrology approach and correct for it within the rendering process [8].

3 An Application: Dynamic Superimposition of a Knee Joint on a Patient Leg

When combined with the dynamic superimposition method found in [2], the calibration method described is well suited for implementation in complex AR systems. The Virtual Reality Dynamic Anatomy (VRDA) Tool is a system that allows medical practitioners to visualize anatomical structures superimposed on their real counterparts. To realize this effect, the medical practitioner wears a HMD to view a computer graphics model of the knee superimposed on the real leg of a model patient. In the following paragraphs, we demonstrate how the method is integrated within the VRDA Tool.

We treat the leg as two separate objects; the first object is associated with the thigh and the second object is associated with the shank. Each part of the leg is tracked independently. To find the best location of the markers, we considered the shape of the leg and chose the marker locations where they would probably move the least [6]. We defined the landmarks in places where there is less flesh, allowing the landmarks to be closer to the bones to reduce scaling or location errors. The landmark locations are shown in Fig 1. To avoid collisions between the synthetic objects, we refer to a precomputed look-up table that encodes the correct location of the femur relative to the tibia [5]. The entry to the table is the transformation between the two synthetic objects. The table returns the real-time location of the objects, allowing smooth, realistic motion without collisions.

We also evaluated the scaling threshold to determine if uniform or non-uniform scaling between real objects and synthetic objects is appropriate. To determine the relative motion of the markers on the leg, we made 1000 measurements of the global 3D location of the markers over a 10 second interval of standard motion for the leg. We found that the maximum standard deviation of the motion of markers is less than 15 mm. The tracking system is accurate to within 0.1 mm.

For the eyepoints, we chose the center of rotation of the eye because rendering is applied in near field visualization [11]. The field of view of the HMD is 26.11 degrees and the display resolution is 640 x 480 pixels. We also applied a coating to the LCD displays to minimize the pixelization of our synthetic objects.

The 3D models that represent the knee joint anatomy are high-resolution models from Viewpoint Corporation. The tracking system we employ is a NDI OPTOTRAK 3020, which uses active, infrared LEDs as markers. The choice of this system is based upon its resolution, robustness against common perturbations, and speed. The display device is a prototype see-through head mounted display. We perform both computations and stereoscopic rendering a SGI Onyx2. The complete implementation of this method allows superimposition at interactive-speed. We are currently able to achieve stereo frame rates of up to 26.6 Hz. Furthermore, because of the choice of the SVD method and the enhancement of noise attenuation, the superimposition process is robust and accurate.

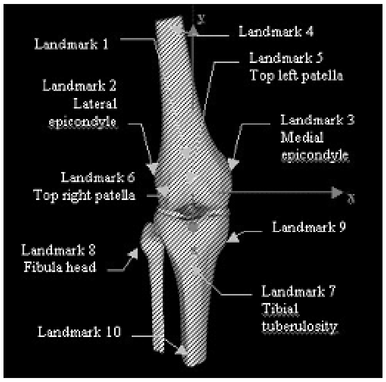

Fig. 1. Landmark locations

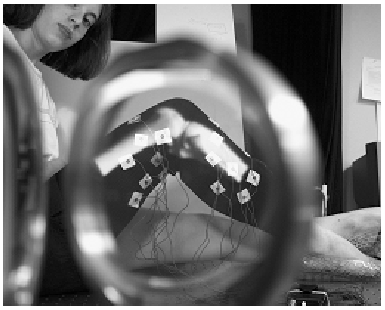

Fig. 2. The dynamic superimposition

4 Future Work

Future work with the VRDA tool will demonstrate deformable structures such as ligaments and muscles with respect to the bones as well. However, such demonstration is not required in quantifying the methods presented here. Furthermore, methods of non-uniform scaling of synthetic objects will be implemented in future developments. This is especially important in working with generic models that must be registered with specific real objects. Applications of augmented reality methods presented here will further be extended to perform full body motion capture.

5 Acknowledgements

The work is supported by the National Institute of Health under grant 1-R29-LM06322-01A1, the National Science Foundation under grant EIA-99-86051, the Florida Education Fund, and the Lockheed-Martin Aeronautics Company. Videotapes may be requested at jannick@odalab.ucf.edu .

References

[1] T.P. Andriacchi, G. Alexander, M.K. Toney, C. Dyrby, and J. Sum. A Point Cluster Method for In Vivo Motion Analysis: Applied to a Study of Knee Kinematics. *Journal of Biomechanics*, 1999.

[2] Y. Argotti, L. Davis, V. Outters, and J.P. Rolland. Dynamic Superimposition of Synthetic Objects on Rigid and Simple-Deformable Real Objects. In *The Second IEEE and ACM International Symposium on Augmented Reality (ISAR '01), New York, NY.* IEEE Computer Society, IEEE Press, October 2001.

[3] Y. Argotti, V. Outters, L. Davis, A. Sun, and J.P. Rolland. Technologies for Augmented Reality: Real-time Superimposition of Synthetic Objects on Dynamic Rigid and Simple-deformable Real Objects. Technical Report TR01-004, University of Central Florida, 2001.

[4] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987.

[5] Y. Baillot, 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):223–235, 2000.

[6] A. Cappello, A. Cappozzo, P.F. La Palombara, L. Lucchetti, and A. Leardini. Multiple Anatomical Landmark Calibration for Optimal Bone Pose Estimation. *Human Movement Science*, 16:259–274, 1997.

[7] H. Hua, A. Girardot, C.Y. Gao, and J.P. Rolland. Engineering of Head-Mounted Projective Displays. *Applied Optics*, 39(22):3814–3824, 2000.

[8] W. Robinett and R. Holloway. The Visual Display Transformation for Virtual Reality. *Presence: Teleoperators and Virtual Environments*, 4(1):1–23, 1995.

[9] J.P. Rolland and H. Fuchs. Optical Versus Video See-Through Head-Mounted Displays in Medical Visualization. *Presence: Teleoperators and Virtual Environments*, 9(3):287–309, 2000.

[10] J.P. Rolland, W. Gibson, and D. Ariely. Towards Quantifying Depth and Size Perception in Virtual Environments. *Presence: Teleoperators and Virtual Environments*, 4(1):24–49, 1995.

[11] J.P. Rolland and L. Vaissie. Albertian Errors in Head-Mounted Displays: Choice of Eyepoint Location. Technical Report TR01-001, University of Central Florida, 2001.

[12] I. Söedrkvist and P.Å. Wedin. Determining the Movement of the Skeleton Using Well-Configured Markers. *Journal of Biomechanics*, 26(12):1473–1477, 1993.

[13] C.W. Spoor and F.E. Veldpaus. Technical Note: Rigid Body Motion Calculated from Spatial Coordinates of Markers. *Journal of Biomechanics*, 13:391–393, 1980.

[14] D.L. Wright, J.P. Rolland, and A.R. Kancherla. Using Virtual Reality to Teach Radiographic Positioning. *Radiologic Technology*, 66(4):167–172, 1995.