

Modeling head and hand orientation during motion using quaternions

Su Bang Choe and Julian J. Faraway
University of Michigan

Copyright © 2004 Society of Automotive Engineers, Inc.

ABSTRACT

Some body parts, such as the head and the hand, change their orientation during motion. Orientation can be conveniently and elegantly represented using quaternions. The method has several advantages over Euler angles in that the problem of gimbal lock is avoided and that the orientation is represented by a single mathematical object rather than a collection of angles that can be redefined in various arbitrary ways. The use of quaternions has been popular in animation applications for some time, especially for interpolating motions. We will introduce some new applications involving statistical methods for quaternions that will allow us to present meaningful averages of repeated motions involving orientations and make regression predictions of orientation. For example, we can model how the glancing behavior of the head changes according to the target of the reach and other factors.

We will give a brief introduction to the mathematics of quaternions and how they can be used to represent orientations. We will compare it to other methods of representation. Due to the quite different mathematics involved, existing statistical methods cannot be directly applied and even simple concepts such as the average, need to be defined. We will introduce a method to analyze orientation over time via functional regression analysis with the help of quaternion splines.

We will demonstrate the utility of these methods for analyzing orientation data. We will show an application to the prediction of head and hand orientation during motion.

Keywords: Rotation Matrix, Euler Angles, Exponential Map, Functional Regression.

INTRODUCTION

Motion can be characterized as a change in both position(translation) and orientation(rotation). Human motion modelling involves analysis of both position and orientation of the various body segments. Position is represented as a three-dimensional vector of the x, y, z co-ordinates. Orientation, however, is much more complex and there are several different ways we can parameterize it.

Euler's displacement theorem states that however a rigid body tumbles, every orientation can be achieved by a single rotation from a reference frame. The classical way of representing rotations is by way of rotation matrices (sometimes called Direction Cosine Matrices), but there exist alternatives. Many use Euler angles as they are somewhat easier to picture and mathematically simpler to handle, but such simplicity comes at a price and this form of parametrization suffers from a few problems. Another method involves the use of quaternions first introduced by Hamilton as a form of hyper-complex number. There are a few other methods, but these three are the main ones. We will describe these in detail and describe why the quaternion parametrization is better suited for our purposes.

The mathematics of quaternions are somewhat different and as such new modelling techniques need to be developed if we are to use this form of representation. Quaternion averages and splines will be described in this article which we will use as building blocks to more complex modelling techniques such as regression models and functional regression models for analyzing orientation over time. We will outline the difficulties and issues involved.

ROTATION PARAMETERIZATIONS

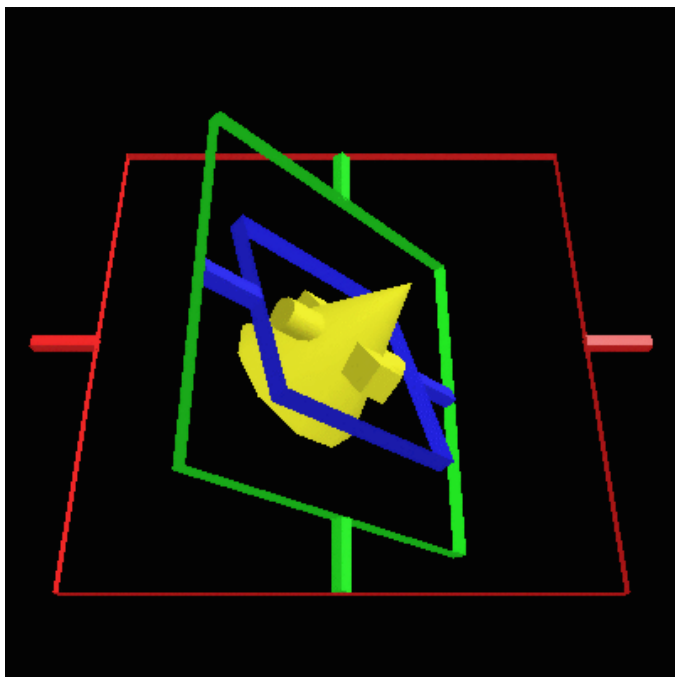


Figure 1: A gimbal with three rings

ROTATION MATRICES A 3×3 matrix M is considered a rotation matrix if and only if:

$$\begin{aligned} M^T M &= I, \\ \det M &= 1. \end{aligned}$$

This means that the three column vectors of M are orthonormal and indicate the directions of the co-ordinate axes in the new rotated space. The set of all 3×3 rotation matrices forms the group $SO(3)$

The fact that we have to use nine numbers to describe something with three degrees of freedom (Euler's rotation theorem states that an arbitrary rotation may be described by only three parameters) is a disadvantage and leads to complicated calculations involving six linear constraints (three constraints to maintain the unit length of the columns and three to enforce mutual orthogonality). Still, it does have an advantage over the other methods of representation in that each rotation corresponds to a unique rotation matrix, something the other forms do not have.

EULER ANGLES Euler angles represent rotations as a series of three angles where each angle represents the amount of rotation about a single predetermined axis. By concatenating these single-axis rotations in sequence we obtain the final rotation. Due to the fact that rotations do not commute this sequence is important. There exist as many as twelve different such sequences and as such there are many different forms Euler angles can take. It has the advantage of being intuitive and interpretable thus allowing for easy visualization of the orientation.

One major disadvantage of Euler angles is that they suffer from *gimbal lock*. A gimbal is a gyroscopic device used

on aircraft to keep track of its orientation. It consists of three interlocking rotating rings (see fig. 1). Gimbal lock occurs when two of the three rotation axes align resulting in the loss of a rotational degree of freedom. In terms of the figure this happens when the middle ring turns 90 degrees and causes the entire gimbal to become flat like a plate. This causes the outer and inner rings to rotate in the same direction.

In fact any representation that is embedded in \mathbb{R}^3 will suffer from this as \mathbb{R}^3 cannot be mapped into $SO(3)$ without singularities (Mathematically, a singularity is a continuous subspace of the parameter space, all of whose elements correspond to the same rotation - thus movement within the subspace produces no change in rotation), *i.e.* gimbal lock.

One can express the Euler angles as a series of three rotation matrices. Using the *roll* (rotation about the x -axis), *pitch* (y -axis) and *yaw* (z -axis) definition (also known as Cardan angles), we have the following rotation matrices:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix},$$

$$\begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix},$$

$$\begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

where α is the roll angle, β is the pitch angle and γ is the yaw angle. Multiplication of the above three matrices in order will give the final rotation matrix.

QUATERNIONS Quaternions were first conceived in 1843 by William Hamilton as a sort of hyper-complex number. They consist of a real component and three imaginary components. They can be written in several different ways and it is helpful to know them all as each form is useful.

$$\begin{aligned} \mathbf{q} &= [\mathbf{v}, w] \\ &= [(x, y, z), w] \\ &= [x, y, z, w] \\ &= ix + jy + kz + w. \end{aligned}$$

where $x, y, z, w \in \mathbb{R}$, $\mathbf{v} \in \mathbb{R}^3$, $i^2 = j^2 = k^2 = ijk = -1$.

The last form shows the classical complex number form with i, j and k being the three imaginary numbers. For simplicity, we can ignore the fact that they are complex numbers and write them as a vector in \mathbb{R}^4 as in the third

form or we can group the x, y and z into a vector \mathbf{v} separate from the scalar w as in the first form.

Even though we sometimes write a quaternion as a four dimensional vector, the usual vector algebra cannot be used. Addition is the same, but multiplication is different. There are two well known methods for vector multiplication: the cross product and the dot product, each yielding a different type of result. The quaternion product can be described in terms of these.

Addition

$$\begin{aligned}\mathbf{q}_1 + \mathbf{q}_2 &= [\mathbf{v}_1, w_1] + [\mathbf{v}_2, w_2] \\ &= [\mathbf{v}_1 + \mathbf{v}_2, w_1 + w_2].\end{aligned}$$

Multiplication

$$\begin{aligned}\mathbf{q}_1 * \mathbf{q}_2 &= [\mathbf{v}_1, w_1] * [\mathbf{v}_2, w_2] \\ &= [\mathbf{v}_1 \times \mathbf{v}_2 + w_1 \mathbf{v}_2 + w_2 \mathbf{v}_1, w_1 w_2 - \mathbf{v}_1 \cdot \mathbf{v}_2].\end{aligned}$$

Other important operations, reminiscent of those found in complex number theory, such as the conjugate, norm and inverse are given below.

Conjugate

$$\begin{aligned}\mathbf{q}^* &= [\mathbf{v}, w]^* \\ &= [-\mathbf{v}, w]\end{aligned}$$

Norm

$$\begin{aligned}\|\mathbf{q}\| &= \mathbf{q}\mathbf{q}^* \\ &= w^2 + \mathbf{v} \cdot \mathbf{v} \\ &= w^2 + x^2 + y^2 + z^2\end{aligned}$$

Inverse

$$\mathbf{q}^{-1} = \frac{\mathbf{q}^*}{\|\mathbf{q}\|}$$

We use unit(norm 1) quaternions to represent rotations and thus the set of all rotations in quaternion space is S^3 , the unit hypersphere on \mathbb{R}^4 . For interpretation, a rotation around an axis \mathbf{u} by an angle 2θ can be written in quaternion form as $[\mathbf{u} \sin \theta, \cos \theta]$. If we wish to rotate a vector \mathbf{a} by a rotation parameterized by a quaternion q we can obtain the rotated vector \mathbf{a}' using the following formula:

$$[\mathbf{a}', 0] = \mathbf{q} * [\mathbf{a}, 0] * \mathbf{q}^{-1}$$

where the multiplication is via the quaternion product. One should note that we would get the same answer if we replaced \mathbf{q} by $-\mathbf{q}$ in the above formula. Thus every rotation in \mathbb{R}^3 may be expressed uniquely up to a sign change. Concatenating rotations are handled seamlessly by quaternions. If \mathbf{q}_1 is the first rotation and \mathbf{q}_2 is the second then:

$$\begin{aligned}[\mathbf{a}', 0] &= \mathbf{q}_2 * (\mathbf{q}_1 * [\mathbf{a}, 0] * \mathbf{q}_1^{-1}) * \mathbf{q}_2^{-1} \\ &= (\mathbf{q}_2 * \mathbf{q}_1) * [\mathbf{a}, 0] * (\mathbf{q}_2 * \mathbf{q}_1)^{-1}\end{aligned}$$

One can convert a quaternion to a rotation matrix using the following formula:

$$\mathbf{Q} = \mu(\mathbf{q})$$

$$= \begin{pmatrix} w^2 + x^2 - y^2 - z^2 & 2(xy - wz) \\ 2(wz + xy) & w^2 + y^2 - x^2 - z^2 \\ 2(xz - wy) & 2(yz + wx) \\ 2(wx + xz) & \\ 2(yz - wx) & \\ w^2 + z^2 - x^2 - y^2 & \end{pmatrix}$$

where $\mathbf{q} = [(x, y, z), w]$. Again note that $\mu(\mathbf{q}) = \mu(-\mathbf{q})$ reiterating the point that only rotation matrices uniquely express rotations.

Quaternions like rotation matrices make visualization of the orientation difficult and for such purposes Euler angles are superior. Quaternions are mathematically more stable than Euler angles as they do not suffer from gimbal lock, thus making them better suited for statistical analysis. One can always convert quaternions to Euler angles when it comes time to interpret results.

QUATERNION STATISTICS

QUATERNION AVERAGE Suppose that we have a sample of repeated orientations $\mathbf{q}_1, \dots, \mathbf{q}_n$ from S^3 . We would like to find the cluster point or average orientation. The usual arithmetic mean $\bar{q} = \frac{1}{n} \sum_{i=1}^n q_i$ would not be a unit quaternion and thus would not be an orientation and subsequent normalization would be required. One method by [1] derives a *spherical average* which respects *spherical distance* and another method given by [5] involves eigendecomposition of the sample moment of inertia matrix.

In normal Euclidean space the average can be defined as the point \mathbf{q} that minimizes the sum of squared distances

$$\mathbf{q}^* = \arg \min_{\mathbf{q}} \left(\frac{1}{n} \sum_{i=1}^n \|\mathbf{q} - \mathbf{q}_i\|^2 \right).$$

In S^3 we can no longer use normal Euclidean distance as it would result in the average lying outside of S^3 . We require a new distance measure that applies to points on S^3 which as stated before is the surface of the four dimensional unit hypersphere. We shall call this distance measure as *spherical distance* and it is defined as

$$\text{dist}_S(\mathbf{q}, \mathbf{p}) = \arccos(\mathbf{q} \cdot \mathbf{p})$$

where $\mathbf{p}, \mathbf{q} \in S^3$. This is the length of the shortest geodesic segment from \mathbf{q} to \mathbf{p} ; the geodesic in this case is the two dimensional great circle that passes through \mathbf{p} , \mathbf{q} and the origin. From this we can define the *spherical average* as the point \mathbf{q}^* that minimizes the sum of squared spherical distances

$$\mathbf{q}^* = \arg \min_{\mathbf{q}} \left(\frac{1}{n} \sum_{i=1}^n \text{dist}_S(\mathbf{q}, \mathbf{q}_i)^2 \right).$$

[1] goes on to provide conditions for existence and uniqueness of the spherical average and provides an iterative algorithm to obtain it.

A key component of the algorithm involves mapping to and from the tangent plane of S^3 using what is called the *exponential map*. The exponential map maps a vector in \mathbb{R}^3 describing the axis and magnitude of a three dimensional rotation to S^3 , the quaternion space. The map is given by

$$\text{expmap}(\mathbf{v}) = \begin{cases} [0, 1] & \text{if } \mathbf{v}=\mathbf{0} \\ [\hat{\mathbf{v}}\sin \theta, \cos \theta] & \text{if } \mathbf{v}\neq\mathbf{0} \end{cases}$$

where $\mathbf{v}\in \mathbb{R}^3$, $2\theta = \|\mathbf{v}\|$ and $\hat{\mathbf{v}} = \mathbf{v}/2\theta$. Thus a vector \mathbf{v} is mapped to a quaternion representing a rotation of 2θ (i.e. $\|\mathbf{v}\|$) about \mathbf{v} .

In the statistics literature, [5] describes a simple way to obtain the average quaternion. He shows that the eigenvector corresponding to the largest eigenvalue of the sample moment of inertia matrix $\mathbf{T} = \frac{1}{n} \sum_{i=1}^n \mathbf{q}_i \mathbf{q}_i'$ gives us the quaternion average. This method gives us the average immediately without having to go through a slow iterative algorithm.

QUATERNION INTERPOLATION A lot of interest exists in ways to smoothly interpolate between successive observations. Much of the existing work has focused on finding different ways of constructing sphere-based analogues of cubic splines in the plane.

[8], who first introduced quaternions to the computer animation world, created the *slerp*(spherical /linear interpolation) algorithm for interpolating between two quaternions. This slerp algorithm produces interpolants which lie on the shortest geodesic between the two quaternions and thus produces a smooth and natural interpolation. For interpolating more than two points he described a method to calculate Bèzier curves to define curves on a sphere. Intuitively one can see that there are many possible ways to interpolate these points and various methods have been put forward involving minimization of different critical functions.

[1] introduces a spherical spline method based on the spherical average. Suppose we are given points $\mathbf{q}_1, \dots, \mathbf{q}_n$ on S^3 observed at times $t_1 < t_2 < \dots < t_n$. We wish to find a smooth curve s lying on the hypersphere parameterized by t such that $s(t_i) = \mathbf{q}_i$ for all i . The problem becomes finding control points p_i that define a spline curve that satisfies the conditions.

Using a standard B-spline implementation we have basis functions which are piecewise cubic and have continuous second derivatives. The spline curve is then given by

$$s(t) = {}_{sph} \sum_{i=1}^n f_i(t) \cdot p_i$$

where ${}_{sph} \sum$ represents a weighted spherical average and p_i are the control points. Obtaining these control points then involves solving an iterative algorithm, details of which are given in [1].



Figure 2:
Initial Hand
Position



Figure 3:
Quaternion



Figure 4:
Euler



Figure 5:
Final Hand
Position

[6] produced a way of interpolating points on $SO(3)$ by first mapping to the tangent space using the exponential map and using normal Euclidean splines on each of the three components. [4] shows however that such a method is far from optimal and a straight line linking two orientations in exponentially mapped \mathbb{R}^3 is not, in general, equivalent to the geodesic between the two orientations in S^3 .

Euler angles have been widely used for interpolation purposes due to reasons of simplicity. Each angle is interpolated independently from the others and this leads problems resulting in unnatural interpolation. This is due to the fact that this method ignores the inherent order involved. Problems with gimbal lock exist as well which reinforces the unsuitability of Euler angles for this task. Figures 2-5 show us an illustration of this problem. Here we have two key frame positions (figs. 2 and 5) close together where one is just the other flipped over. We then show the interpolation halfway between the two key frames. The quaternion interpolation is where it should be; halfway between flipping over (fig. 4). The Euler interpolation seems to be completely off in another direction (fig. 5).

QUATERNION REGRESSION AND FUNCTIONAL REGRESSION Let us suppose that we are interested in modelling the orientation of the hand(parameterized as

quaternions) of the final posture in a reaching task. We wish to use predictors such as age, gender and stature of the person performing the reach as well as the location of the target he is reaching to.

Suppose we have a data set with n observations and p covariates. A straight multivariate regression model could be fit as follows:

$$Y_{n \times 4} = X_{n \times p} \beta_{p \times 4} + \varepsilon_{n \times 4}.$$

Unfortunately such a model would not in general produce predictions that would be unit quaternions. One possible way around this is to simply normalize the predicted values.

Another way would be to transform the problem using the exponential map. Since the tangent space to S^3 is \mathbb{R}^3 we no longer have a restriction, allowing us to use a normal multivariate regression model as follows:

$$Y_{n \times 3}^* = X_{n \times p} \beta_{p \times 3} + \varepsilon_{n \times 3}.$$

We can then transform back to S^3 to obtain our quaternion results.

The best method would be to somehow figure out how to solve the regression with this non-linear constraint and we are currently working on this.

If we are now interested in modelling the head orientation during the entire motion we will have to use a functional regression [7]; [2] where the response is a quaternion function over time. Such a model may be written as:

$$Y(t) = X\beta(t) + \varepsilon(t)$$

where $Y(t) = (y_1(t), \dots, y_n(t))$ We can estimate the regression coefficients $\beta(t)$, which are a vector of functions, using the familiar looking estimator:

$$\hat{\beta}(t) = (X^T X)^{-1} X^T Y(t)$$

To approximate the functions we will use the quaternion splines outlined in the preceding section. Thus using the methodology of [1] we obtain:

$$y_i(t) =_{sph} \sum_{j=1}^m f_j(t) \cdot y_{ij}$$

where y_{ij} is the j th control point for the B-spline of $y_i(t)$. Note that the number of control points m is dependent on the number of data points observed in $y_i(t)$. There is no guarantee that we will have the same number of observed points for all i , as subjects move at different speeds and distances to reaching targets will vary. Additional to this is the fact that we might observe upwards of one hundred data points during a motion making data reduction an important issue. An approximating or smoothing spline would therefore be a better choice [3].

If this was simply a univariate problem we can now rewrite the model in the following form:

$$Y_{n \times m} F_{m \times 1}(t) = X_{n \times p} B_{p \times m} F_{m \times 1}(t) + \varepsilon_{n \times 1}(t).$$

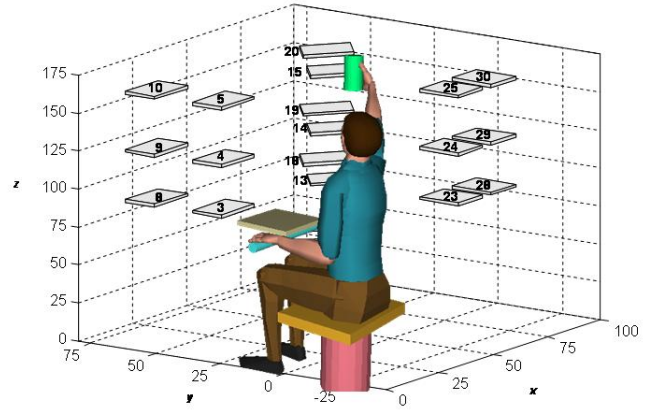


Figure 6: Target shelf layout in SCI/LBP experiment conducted at HUMOSIM

If we factor out the vector of basis functions $F(t)$ we get a normal multivariate regression model:

$$Y_{n \times m} = X_{n \times p} B_{p \times m} + \varepsilon_{n \times m}$$

which can be solved in the usual way.

However in quaternion splines the y_{ij} 's are quaternions, which means that for each observation we no longer have an m -dimensional vector of control points as in the univariate case but rather an $m \times 4$ matrix where the row vectors are the quaternion control points. This means each observation is matrix-variate and this is a problem as we do not yet know how to do a matrix-variate regression. To escape from this predicament we propose stacking the quaternion control points on top of one another to create a $4m$ dimensional vector thus leading us to the multivariate regression model:

$$Y_{n \times 4m} = X_{n \times p} B_{p \times 4m} + \varepsilon_{n \times 4m}.$$

We will then have to use a more complicated version of the quaternion regression techniques outlined in the beginning of this section (such as mapping onto the tangent plane) to handle the fact that we require each control point (i.e. each block of four in the predicted response vector) to be a unit quaternion.

DATA EXAMPLE

Here is a sample analysis to illustrate the use of the quaternion functional regression technique. The data is from an experiment conducted at the HUMOSIM laboratory at the University of Michigan. The purpose of the experiment was to assess the difference in motion caused by spinal cord injury and lower back pain. Three groups of ten subjects each: a control group with no injury, a group with spinal cord injury and a group with lower back pain, were instructed to deliver objects to shelves placed within reach while seated (See figure 5). Their motions were recorded and analyzed.

For the purposes of this example, we looked at the orientation of the right hand with respect to the wrist for delivery motions to a target shelf high and to the right of the subject (target no.25 in figure) where the object delivered was a vertical cylinder. Using the tangential mapping technique, we mapped the quaternion data in S^3 to the tangent space, \mathbb{R}^3 . Regression splines were fit to each observation using eight basis functions thus giving us eight trivariate spline control points. Stacking these control points on top of one another we obtain a vector of length 24 which becomes our response. A multivariate regression was then fit with gender, stature and status(control/spinal cord injury(SCI)/lower back pain(LBP)) as predictors. Status was found to be significant with the SCI and LBP group having a different orientation motion as compared to the control group.

Figures 7-21 show five freeze frames of the predicted orientation motion of the right hand with respect to the wrist for a male of the same stature from each group. We can see a difference between the control group and the SCI or LBP groups.

CONCLUSIONS AND FUTURE WORK

It is well known that Euler angles suffer from many problems which make it ill suited for the task of modelling human motion. Quaternions provide an elegant alternative that is free of such problems but because of the particular mathematics involved requires new methodology for statistical modelling.

We have introduced a list of potential strategies one can use but more work must be done for us to be able to compare and rate which method is the best. As stated before we would like to find a way to fit a regression model with the restriction that we must have unit quaternions. The proposed strategy of mapping to a tangent space is not ideal and it would be much better if we did not have to leave quaternion space during our calculations. For functional quaternion regression we require an elegant way of handling the matrix-variate response produced by the B-spline approximation.

ACKNOWLEDGEMENTS

We would like to thank Dr. Don Chaffin and the HUMOSIM laboratory at the University of Michigan. This research was made possible by funding from Daimler-Chrysler, Ford, GM, EDS, International Truck and Engine, Lockheed-Martin, US Postal Service, US Army-TACOM and the Automotive Research Center at the University of Michigan.

REFERENCES

- [1] S.R. Buss and J.P. Fillmore. *Spherical Averages and Applications to Spherical Splines and Interpolation*. ACM Transactions on Graphics, Vol. 20, No. 2, 95-126, 2001.
- [2] J. Faraway. *Regression Analysis for a Functional Response*. Technometrics, 39, 254-261, 1997.
- [3] J. Faraway. *Modelling Reach Motions Using Functional Regression Analysis*. SAE Technical Paper 2000-01-2175, 2000.
- [4] F. Grassia. *Practical Parameterization of Rotations Using the Exponential Map*. The Journal of Graphics Tools, Vol. 3.3, 1998.
- [5] M.J. Prentice. *Orientation Statistics Without Parametric Assumptions*. Journal of the Royal Statistical Society. Series B(Methodological), Vol. 48, Issue 2, 214-222, 1986.
- [6] M.J. Prentice. *Fitting Smooth Paths to Rotation Data*. Applied Statistics, Vol. 36, Issue 3, 325-331, 1987.
- [7] J. Ramsay and B. Silverman. *Functional Data Analysis*. Springer, New York, 1997.
- [8] K. Shoemake. *Animating Rotation with Quaternion Curves*. ACM SIGGRAPH, Vol. 19, No. 3, 1985.

CONTACT INFORMATION

Su Bang Choe: 437 West Hall 550 East University Ann Arbor MI 48109-1092. (734)764-2646. choes@umich.edu.

Julian J. Faraway: 460 West Hall 550 East University Ann Arbor MI 48109-1092. (734)763-5238. faraway@umich.edu.

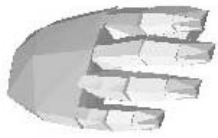


Figure 7: CTL no.1



Figure 8: SCI no.1



Figure 9: LBP no.1



Figure 10: CTL no.2



Figure 11: SCI no.2



Figure 12: LBP no.2



Figure 13: CTL no.3

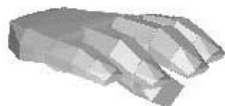


Figure 14: SCI no.3



Figure 15: LBP no.3



Figure 16: CTL no.4



Figure 17: SCI no.4



Figure 18: LBP no.4



Figure 19: CTL no.5

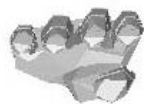


Figure 20: SCI no.5



Figure 21: LBP no.5