Mathematical Models of Binary Spherical-Motion Encoders

David Stein, Edward R. Scheinerman, and Gregory S. Chirikjian, *Member, IEEE*

***Abstract*—This paper presents several algorithms that solve the problem of determining the orientation of a freely rotating ball that is partially enclosed in a housing. The ball is painted in two colors (black and white) and the housing has a number of sensors that detect these colors. The question we answer is: Knowing how the ball is painted, knowing the location of the sensors, and given a complete set of sensor measurements, how does one determine the orientation of the ball to within an acceptable error threshold? The algorithms we present to solve this problem are based on methods and terminology from geometric control theory. Essentially, we generate dynamical systems that evolve on the group $SO(3)$. These dynamical systems are constructed so as to attract the computed orientation of the ball to the actual one being detected by the sensors. Solving this spherical decoding problem is important in applications where spherical motion must be detected. One such application is the feedback control of spherical motors.**

***Index Terms*—Gradient descent, nonsmooth optimization, optical encoder, rotation group, spherical motion.**

I. Introduction

CONSIDER the following problem. A sphere is held in a cradle, but is free to rotate arbitrarily. How can the orientation of the sphere be determined? This paper explores the development of reliable decoding algorithms for a spherical encoder that does not use a mechanical link to the rotating sphere. The encoder itself is constructed by painting the surface of the sphere with two colors (black and white). Fixed point sensors are located in the cradle and the orientation of the sphere is determined from the binary feedback of each of these sensors.

Sensing devices such as this spherical encoder have applications in the control of spherical motors, such as those developed in [1]–[15]. We have also developed a spherical motor [16], and our original motivation to study spherical encoders was in the context of control of this motor [17], [18]. Potential applications of spherical motors include spacecraft attitude control, omni-directional vehicle propulsion, and actuation of haptic interfaces, just to name a few. In all of these applications, the feedback loop can only be closed if knowledge about the current orientation of the rotating ball can be sensed.

In contrast to the relatively large literature on spherical motors, very little work has been reported on encoders for spherical motion. The few spherical orientation systems used in industry today rely on mechanical links connected to the rotating sphere to determine its orientation. They are greatly limited by their inaccuracy as well as the need to be in contact with the rotating sphere. However, there has been some previous work on sensors that do not rely on a mechanical coupling to the sphere. In 1993, Pettypiece [19] patented a spherical optical encoder for motion about three mutual orthogonal axes. This system is based on three orthogonal gradient encoders that enable the system to detect the orientation of the sphere over a small range. Lee [20] has developed a machine-vision-based orientation system which is very accurate but needs a great amount of computing power and utilizes more complicated components than the techniques developed in this paper.

The spherical encoders investigated in this paper are based on the following operating principle first proposed and developed in [17]. The spherical body to be encoded is free to move in a cradle that overlaps the sphere. A two-color pattern (painting) is fixed on the sphere's surface (hence the name "binary encoder"). Discrete state sensors, which are in one of two states determined by the color they detect, are placed in known locations within the cradle structure. The state of these sensors is constantly updating depending on the color they detect on the surface of the spherical body. For any orientation of the sphere there will be a corresponding binary string constructed from the sensor output. If there are n sensors placed in the cradle then this binary string has up to 2^n possible combinations. The resolution of this encoder is bounded by the fact that there are an infinite number of possible orientations of the sphere and only a finite number of binary strings with which to represent them. An ideal painting of the sphere will result in the output of all the 2^n possible combinations of sensor values when the sphere is moved through all of its possible orientations. The larger the number of unique strings that can be output by the sensors, the higher the potential resolution of the encoder.

The remainder of this paper is broken down into several sections. Section II explores various paintings of the sphere; Section III investigates how the characteristics of specific sphere paintings and the number and location of sensors in the housing relate to the resolution of the encoder; Section IV presents two algorithms for decoding the orientation of the sphere from point sensor data. Section V presents two techniques for initializing the encoder.

Manuscript received August 1, 2002; revised January 20, 2003. This work was supported by the National Science Foundation under Grants IIS-0098382, IRI-9731720, and DMS-9971696. Recommended by Guest Editors C. Mavroidis, E. Papadopoulos, and N. Sarkar.

D. Stein was with the Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA. He is now with Diagnostic Products Corporation (DPC), Instrument Systems Division, Flanders, NJ 07836.

E. R. Scheinerman is with the Department of Mathematical Sciences, The Johns Hopkins University, Baltimore, MD 21218 USA.

G. S. Chirikjian is with the Department of Mechanical Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: gregc@jhu.edu).

Digital Object Identifier 10.1109/TMECH.2003.812824

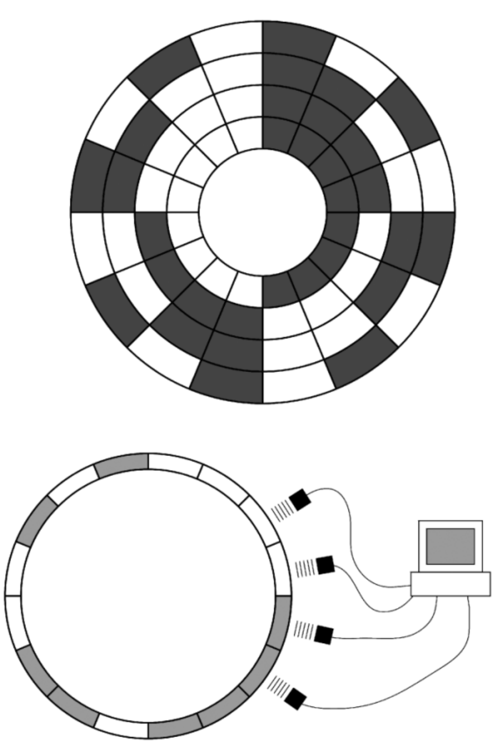

Fig. 1. Grey code rotary encoder.

II. Sphere Painting

A concept which is fundamental to all forms of encoders is the marking of the moving component to create reference points for the sensors to detect. The marking can be as simple as parallel lines placed on a moving slide, or can be as complex as the highly irregular sphere paintings that will be outlined later in this paper. Unfortunately, a simple, regular painting of the sphere is unacceptable; such paintings often have symmetries and consequently, different orientations of the sphere produce identical sensor readings. On the other hand, a highly irregular coloring might be difficult to describe and be inefficient to handle computationally. We examine these and other issues in the subsections that follow.

A. Geometric Painting

A classical method used to encode rotary motion about a fixed axis is to use a Grey Code/deBruijn encoder. The top of Fig. 1 shows a standard encoding wheel for optical encoding. For this encoder the sensors must be able to see the interior of the disk. An alternative method, shown on the bottom of Fig. 1, is based on a *de Bruijn sequence*. It allows the determination of the angle of the disk using colors and sensors along the edge of the disk.

Let n be a positive integer. A *de Bruijn sequence* is a sequence of 2^n zeros and ones arranged so that each of the 2^n possible n-bit binary numbers appears (cyclically) exactly once in the sequence. For example, the sequence

0000111101100101

contains all 2^4 length-four binary numbers. The binary number 0011 appears starting at the third element of the sequence, and the binary number 0100 appears starting at the next-to-last element of the sequence (and wrapping around to the start). By painting the edge of the disk according to a de Bruijn sequence as in the lower portion of Fig. 1, all 2^n bit patterns appear exactly once and 2^n different rotations of the disk can be resolved. See [21], [22].

Now, the idea of a nonrepeating pattern can be applied to the sphere. To reiterate, the chief requirement of a sphere painting is for it to result in the highest possible resolution for the number of discrete sensors used in the spherical encoder. If a nonrepeating simple geometric pattern is placed on each hemisphere with the borders between the two colors parallel to lines of constant longitude, can an acceptable sphere painting be created? The answer is yes and a simple example is shown in Fig. 2. An initial prototype of our encoder is shown in Fig. 3.

The problem with this type of painting is that since it has large areas of solid color, small movements of the spherical body have a very limited effect on the bit pattern received from the sensors. This limited change greatly reduces the resolution of the encoder. Methods to optimize the resolution of the encoder are presented in Section III. For the prototype presented in [18] this simple geometric painting can be made finer for better resolution.

B. Random Voronoi

In general, the more random the coloring the better, so a technique called *random Voronoi coloring* has been developed as follows. Let m be a positive integer, choose m *anchor points* independently and uniformly at random on the surface of the sphere; call these points $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m$. Assume these points lie on a unit sphere, so all $\mathbf{a}_i \in \mathbb{R}^3$ and $\|\mathbf{a}_i\| = 1$. Next assign a color to each of these points; independently, color point $\mathbf{a}_i$ either black or white, each with probability 1/2. Let $c(\mathbf{a}_i)$ be the color of $\mathbf{a}_i$. It is convenient to take $c(\mathbf{a}_i) = 1$ if $\mathbf{a}_i$ is black and $c(\mathbf{a}_i) = -1$ if the point is white.

Now, let $\mathbf{x}$ be an arbitrary point on the surface of the sphere. Color $\mathbf{x}$ to match the color of its nearest anchor point. That is, choose i so that $d(\mathbf{x}, \mathbf{a}_i) \leq d(\mathbf{x}, \mathbf{a}_j)$ for all j, where $d(\mathbf{x}, \mathbf{y}) = \cos^{-1}(\mathbf{x} \cdot \mathbf{y})$ is the length of the shortest geodesic connecting $\mathbf{x}$ and $\mathbf{y}$. Let $c(\mathbf{x}) = c(\mathbf{a}_i)$. Such a coloring is mildly ambiguous. If $\mathbf{x}$ is nearest to two (or more) points of different colors, we assign $c(\mathbf{x})$ arbitrarily (e.g., to match the nearest anchor point of lowest index). Such a coloring of the sphere is shown in Fig. 4.

The anchor points $\mathbf{a}_1, \ldots, \mathbf{a}_m$ partition the surface of the sphere into regions based on nearest neighbor. That is, each point is assigned to a region based on which of the anchor points is nearest to it. The boundaries of these regions (points that are nearest to two or more anchor points) are arcs of great circles. Such a partitioning of the sphere is known as a *Voronoi* decomposition. Thus, given random anchor points, independently color each region of the Voronoi decomposition black or white (each with probability 1/2).

III. Design Optimization of a Noncontact Spherical Encoder

The optimization of any design boils down to tweaking the variables most important to the performance of the system. This

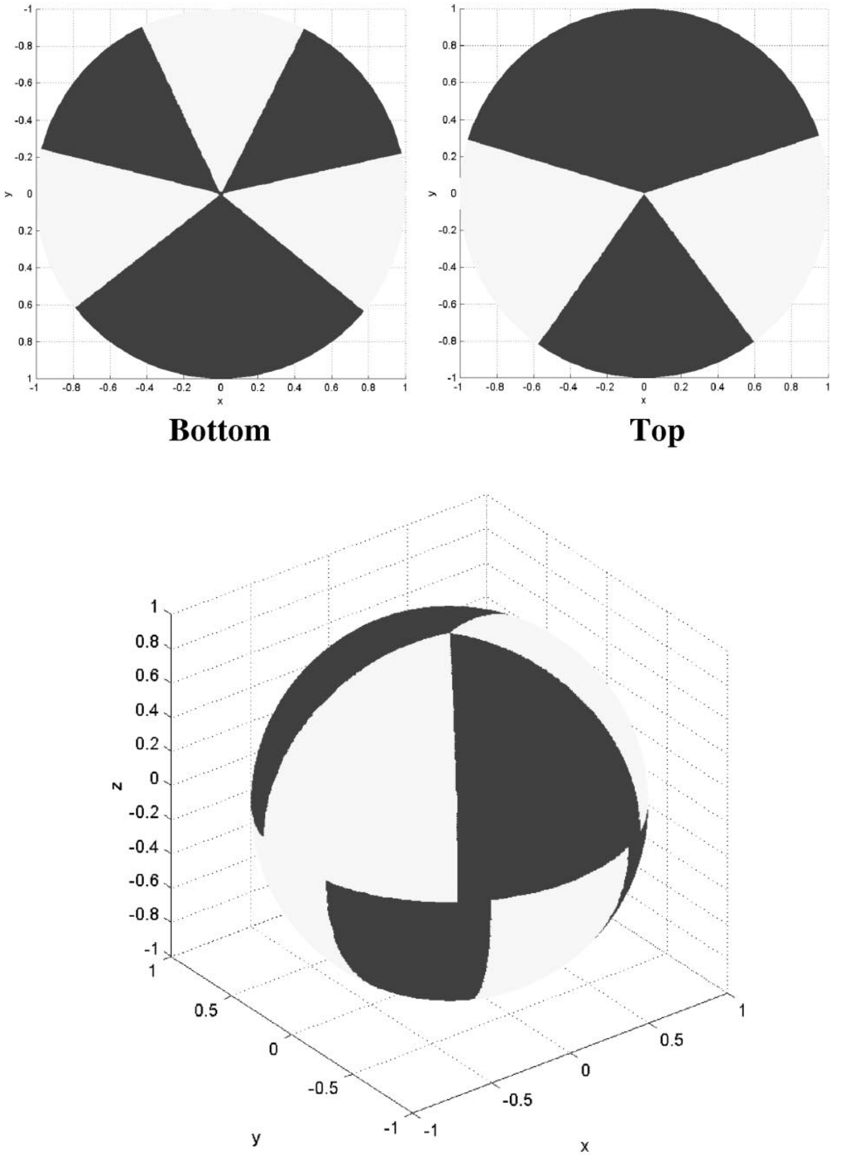

Fig. 2. Simple geometric sphere painting.

section explores how the three basic design variables used in the construction of a noncontact spherical encoder using discrete point sensors affect the encoders resolution. The three design variables are: 1) pattern used to paint the sphere; 2) location of these sensors; and 3) number of sensors used. The effect of these three variables will be explored by looking at "resolution slices" enumerated for individual encoder designs.

The resolution of this type of encoder is measured by the magnitude of rotation in any direction the sphere can undergo before the output string from the sensors change. To completely characterize the resolution of a spherical encoder, the resolution has to be determined for all orientations of the sphere, $A \in SO(3)$. A convenient way to parameterize orientations $A \in SO(3)$ is to use Euler Angles. Euler angles are generated by three successive rotations about independent axes. While the singularities of the Euler angles are well known to cause problems in kinematic and dynamic simulations (see, e.g., [23, Ch. 5]), they suffice in the context of the current problem; we can think of A as being parameterized with ZXZ Euler angles as $A = R(\mathbf{z}, \alpha)R(\mathbf{x}, \beta)R(\mathbf{z}, \gamma)$. However, any method to parameterize orientation of the sphere could be used.

To begin, lets first review the operating principles of the encoder system. The orientation of a sphere free to rotate in a cradle is determined by reading the color at various points on its surface. That is, an array of sensors is placed in the device's cradle. Each sensor is capable of detecting the color (black or white) at a point on the surface of the sphere. The locations of the sensors correspond to points on the sphere. If there are n sensors, let $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$ be the points on the sphere (so $\mathbf{s}_i \in \mathbb{R}^3$ and $\|\mathbf{s}_i\| = 1$) corresponding to the sensors.

Fix a particular orientation of the sphere as a home orientation. Every orientation of the sphere can then be specified by a rotation matrix $A \in SO(3)$. If the sphere has been rotated from the home position via a rotation A, then the color detected by

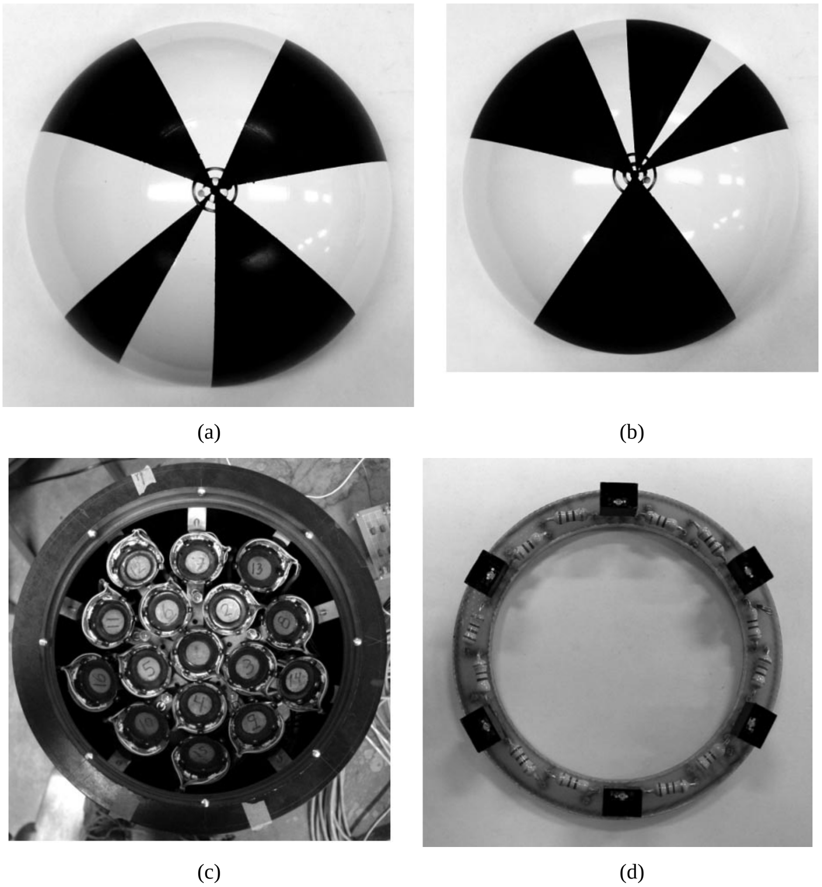

Fig. 3. Prototype geometric sphere painting. (a) Bottom of sphere. (b) Top of sphere. (c) Stator housing with embedded rings of light sensors. (d) One of the sensor rings.

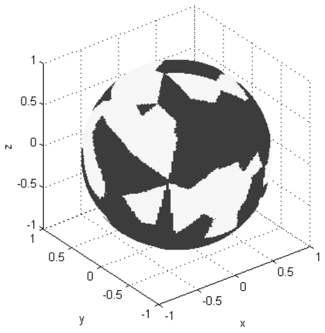

Fig. 4. Random Voronoi painting of the sphere with 200 anchor points/regions.

sensor i is $c(A^T\mathbf{s}_i)$. The *color vector* of a rotation A is defined to be the following vector whose entries are 1 for black and -1 for white:

$$\mathbf{c}(A) = \begin{bmatrix} c(A^T\mathbf{s}_1) & c(A^T\mathbf{s}_2) & \cdots & c(A^T\mathbf{s}_n) \end{bmatrix}^T. \quad (1)$$

Of course, the color vector depends not only on the rotation A, but also on the painting of the sphere $\mathbf{c}$ and the placement of the sensors $\mathbf{s}_i$. However, only the orientation A changes; the painting and sensor placement are fixed in each system.

The resolution of a spherical encoder of this type is the largest angular distance between two orientations that produces identical sensor outputs. In an encoder with n sensors the resolution can be calculated for the sphere in a specific orientation, A, by using the following method.

1) Calculate $\mathbf{c}(A)$ for the current orientation of the sphere.

2) The sphere is now rotated from its current orientation, A, to a new orientation AR, where $R = R(\mathbf{x}, \theta)R(\mathbf{y}, \phi)R(\mathbf{z}, \psi)$. $R(\mathbf{x}, \theta)$ is a rotation about the global $\mathbf{x}$ direction by a magnitude of θ, etc. For increments of 0.1° in all three variables $\mathbf{c}(AR)$ is calculated for θ, ϕ, ψ ranging from -10° to 10°. We denote this discrete subset of $SO(3)$ as S. The values of θ, ϕ, ψ can be increased if the encoder has a very low resolution. At each orientation the dot product between $\mathbf{c}(A)$ and $\mathbf{c}(AR)$ is calculated. If the dot product between the two-color vectors equals n then the two orientations produce the same sensor output and the angular distance between A and AR is calculated by

$$d(A, AR) = \cos^{-1}\left(\frac{\mathrm{tr}(A(AR)^T) - 1}{2}\right). \quad (2)$$

This distance measure can be further simplified using the fact that $\mathrm{tr}(A(AR)^T) = \mathrm{tr}(AR^TA^T) = \mathrm{tr}(R)$, so

$$d(A, AR) = d(I, R). \quad (3)$$

where I is the identity matrix. The final simplified relation is

$$d(A, AR) = \cos^{-1}\left(\frac{\mathrm{tr}(R) - 1}{2}\right). \quad (4)$$

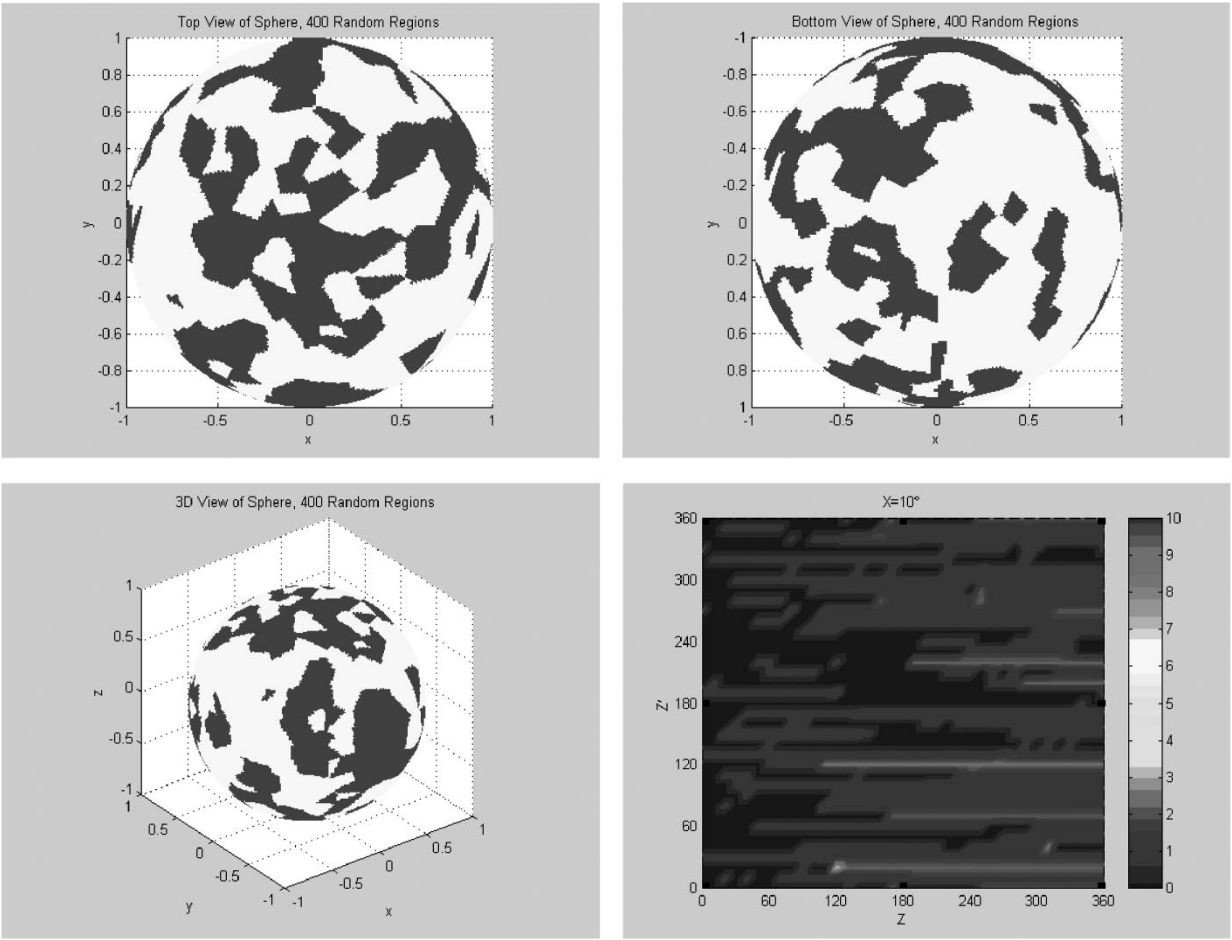

Fig. 5. Example of a random Voronoi painting with randomly placed anchor points. The resulting encoder has a resolution of 1° or better for any orientation when 192 sensors are placed uniformly in a hemisphere. Also shown is one slice of a resolution plot when only 96 sensors are used.

3) The largest angular distance calculated for which $f(A) = \max_{R \in S \subset SO(3)} d(I, R) | \mathbf{c}(A) \cdot \mathbf{c}(AR) = n$ is the resolution of the encoder in orientation A.

Using this formalism, we varied the number, shape, and size of the black and white regions on the sphere. We also varied the number and location of the sensors. As one would expect, the increase in the number of sensors increases the resolution of the encoder. We found that with 192 sensors packed into an area smaller than a hemisphere, we were able to obtain a resolution of 1° or better when the ball was at any orientation when using paintings such as those in Figs. 5 and 6.

IV. Encoder Algorithms

A. Gradient Descent Encoder

To resolve the orientation of the sphere from the sensor data, the problem that needs to be solved is: Given a color vector $\mathbf{c}(A)$, determine A [17]. The problem does not have a unique solution. Because there are only n sensors, there are at most 2^n possible color vectors, but there are infinitely many different rotation matrices A. Therefore, the problem that is actually solved is, given a color vector $\mathbf{c}(A)$, find a rotation matrix B so that $\mathbf{c}(A) = \mathbf{c}(B)$, and B is close to A. Exactly how close will depend on the resolution dictated by the painting and sensor layout.

The first algorithm presented to resolve the orientation of the sphere will be called the Gradient Descent Algorithm. This algorithm was introduced in [17]. The problem it solves is: Given $\mathbf{c}(A)$, find a $B \in SO(3)$ so that $\mathbf{c}(B) = \mathbf{c}(A)$. To this end, a function is defined $f\colon SO(3) \to \mathbb{R}$ by

$$f(B) = \frac{1}{\sqrt{n}} \|\mathbf{c}(B) - \mathbf{c}(A)\|. \tag{5}$$

The factor $1/\sqrt{n}$ is not strictly necessary, but simply rescales the image of the function so that it always lies in $[0,\, 2]$ regardless of the size of n [since the entries of $\mathbf{c}(A)$ and $\mathbf{c}(B)$ are ± 1, the value of $\|\mathbf{c}(B) - \mathbf{c}(A)\|$ is at most $2\sqrt{n}$].

The problem can then be restated: find $B \in SO(3)$ so that $f(B) = 0$. Equivalently, since f is nonnegative, this problem can be thought of as the search for a (global) minimizer of f.

This section presents an iterative method that leads to a solution provided it is given an initial B_0 that is close (within 15 degrees) to the desired solution. This assumption is justifiable for two reasons. First, as the sphere turns in its cradle, the sensors continuously track its progress and the previously known orientation of the sphere can be used as an initial guess for the current orientation. Second, methods are described in Section V that provide initial guesses B_0 reasonably close to the solution value.

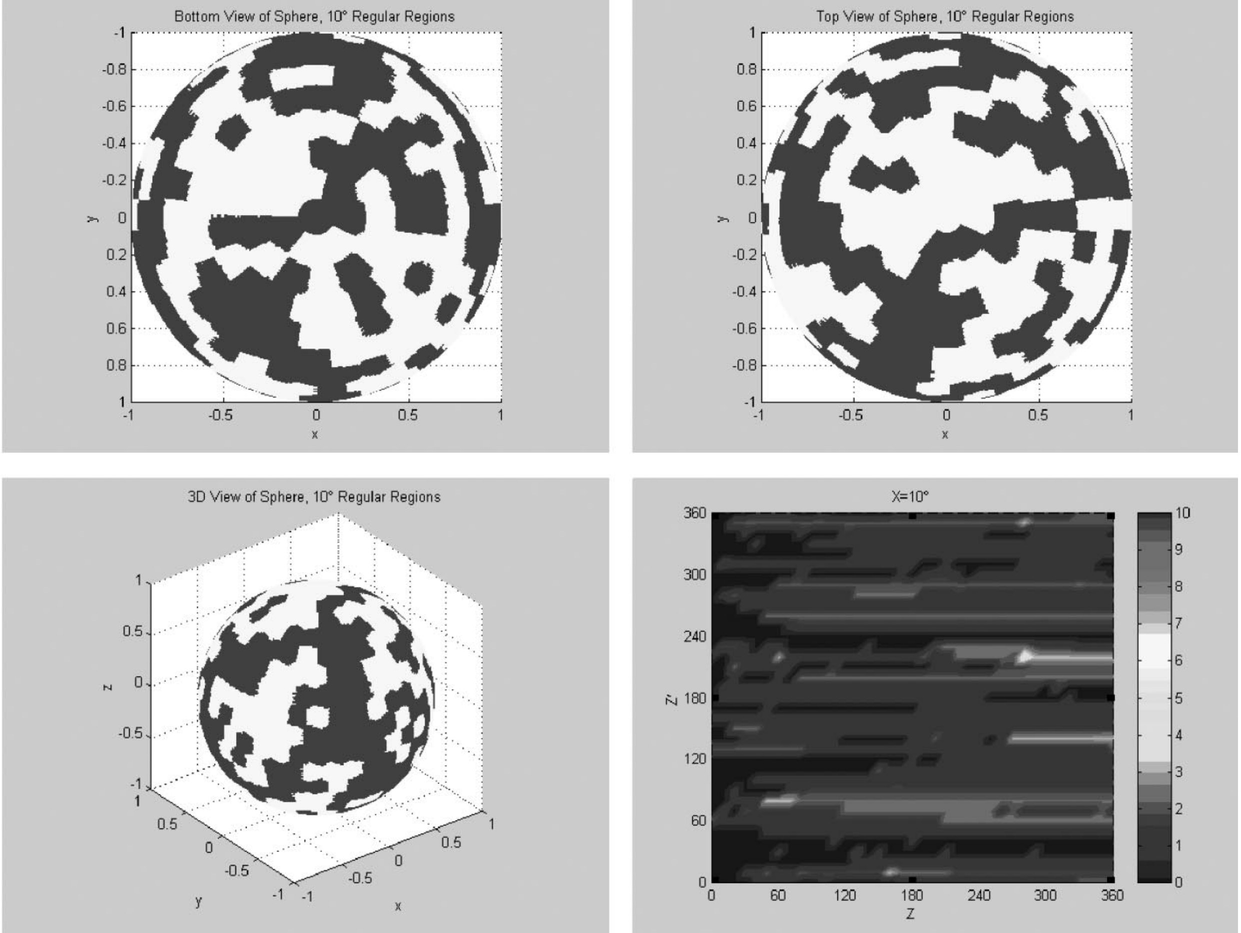

Fig. 6. Example of a random Voronoi painting with regularly spaced anchor points. The resulting encoder has 1° resolution for any orientation when 192 sensors are placed uniformly in a hemisphere. Also shown is one slice of a resolution plot when only 96 sensors are used.

Begin by examining a plot of $f(B)$ where B is restricted to a single axis of rotation; such a plot is presented in Fig. 7. To create this figure, a sphere painting with only 75 black and white regions and 44 sensors was used.

Notice that the graph of f is piecewise flat. This happens because, as the sphere rotates, sensors cross the boundaries between the colored regions and suddenly change state. This causes a step-jump in the value of f. Also notice that within 15° of the actual position, the graph of f decreases (stepwise) to the minimum value. Thus, a sensible approach to minimizing $f(B)$ would be to follow a steepest-descent trajectory. However, to make this precise, one needs to be clear on what is meant by the gradient of f, and then deal with the fact that wherever the gradient is defined, its value is zero.

Because the domain of f is $SO(3)$, concepts from Lie groups/algebras [23]–[26] are used to speak carefully of the gradient of f; see [27]. Let X be a 3×3 skew-symmetric matrix. Define the differential operator

$$(X^R f)(B) = \left. \frac{df\left(Be^{tX}\right)}{dt} \right|_{t=0}. \tag{6}$$

X^R can be thought of as a (right) directional derivative of f in the direction X. The superscript R is to distinguish this operator from the matrix X and the "left" differential operator defined in [23] (which is not used here).

Define the (right) gradient of f at B to be the three-dimensional vector (with ith scalar entry $E_i^R f$) as

$$\left[E_1^R f \quad E_2^R f \quad E_3^R f\right]^T \tag{7}$$

where

$$E_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$
$$E_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$
$$E_3 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{8}$$

The gradient points in the direction of steepest ascent, so following a path in the opposite direction leads to a (local) minimum of the function. However, for the function under consideration, the gradient of f at B is **0** for almost all $B \in SO(3)$, and is undefined otherwise.

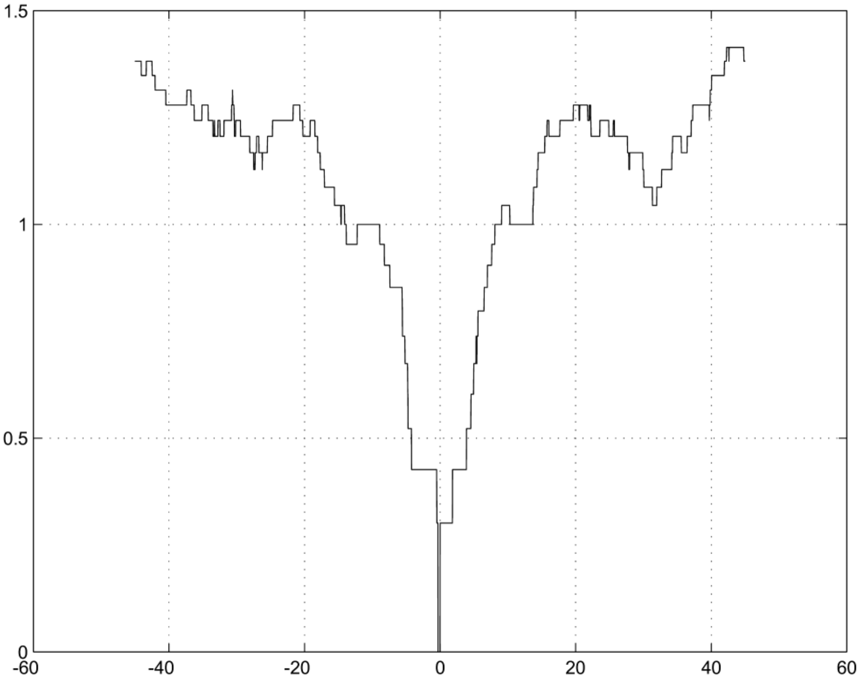

Fig. 7. Plot of $f(B)$ as B varies through a single axis of rotation. On the horizontal axis, 0° corresponds to the home orientation.

One solution is to use approximate gradients, replace the right derivatives $E_j^R f$ (with $j = 1, 2, 3$) with finite difference approximations

$$\frac{f(Be^{tE_j}) - f(Be^{-tE_j})}{2t}. \tag{9}$$

Begin with a specific step size for t (e.g., $t = 0.01$). Knowing the encoder resolution as a function of sphere orientation could also be used here to choose the initial step size. If the approximate gradient calculated is $\mathbf{0}$, increase t by a constant factor. Do this until the approximate gradient is some nonzero vector $\mathbf{x}$, then update B by a step of size t in the negative gradient direction, i.e.,

$$B \leftarrow B \exp\left\{ -t \begin{bmatrix} 0 & -x_1 & x_2 \\ x_1 & 0 & -x_3 \\ -x_2 & x_3 & 0 \end{bmatrix} \right\} \tag{10}$$

where x_j is the finite difference approximation to $(E_j^R f)(B)$.

This is an iterative process. At each successive step, decrease t by a constant factor so as not to overrun the minimum.

Ideally, this procedure is continued until a B is found for which $f(B) = 0$. Of course, this process could get trapped at a local minimum. In this case, if $f(B)$ is fairly small (say $f(B) < 0.2$) it is actually quite close to the minimum and the algorithm can terminate.

B. Jacobian-Based Determination of Orientation

This technique for decoding the sphere orientation was inspired by the resolved-rate numerical inverse kinematics technique commonly used for robot arms. Namely, we iteratively solve a system of equations similar to $\dot{\mathbf{x}} = J\dot{\mathbf{q}}$ for $\dot{\mathbf{q}}$ where $J = J(\mathbf{q})$ is Jacobian matrix, then update $\mathbf{q}(t)$. Only now, $\mathbf{q}(t)$ is a set of parameters describing rotational motion, and $\dot{\mathbf{x}}$ is a vector of differences of sensor measurements at two different times. Since our Jacobian matrix is $n \times 3$ due to $n \gg 3$ sensors and three parameters to describe rotations, it is an overdetermined system, and hence we use the appropriate psuedoinverse to isolate the best possible $\dot{\mathbf{q}}$. In the following subsections, the details of this procedure are reviewed.

1) Algorithm Applied to a Spherical Encoder: The analogy of the forward kinematics for this system is the sensor output that is produced for specific orientations of the sphere. This output is a discontinuous function of the sphere's orientation, hence a derivative cannot be taken directly. Instead, a finite-difference approximation is used as in the gradient descent algorithm. This is accomplished by first defining a function $f_i(A) = c(A^T \mathbf{s}_i)$ which returns the color viewed by sensor $\mathbf{s}_i$ with the painted sphere in orientation A. Following the technique presented in Section IV-A the right gradient of $f_i(A)$ is the vector

$$\left[E_1^R f_i(A) \quad E_2^R f_i(A) \quad E_3^R f_i(A) \right]^T \tag{11}$$

as defined in (6)–(9).

Since the painting is represented by a value of -1 for points on the sphere that are white and 1 for black points the values returned for $E_j^R f_i(A)$ are either 0 or $\pm 1/t$. Since for small changes in orientation, there is no change in the sensor values, the most common value returned is zero. This produces a Jacobian for the sphere painting that is very sparse with only $\pm 1/1t$ and 0s occupying it. A Jacobian of this structure is of limited use in this numerical technique.

2) Rounded Painting Model: What can be done to construct a useful Jacobian? Can the actual painting of the sphere be modeled to capture its discrete nature, while also having derivatives that give information about the painting? This leads to a new way to model the painting of the sphere. This painting

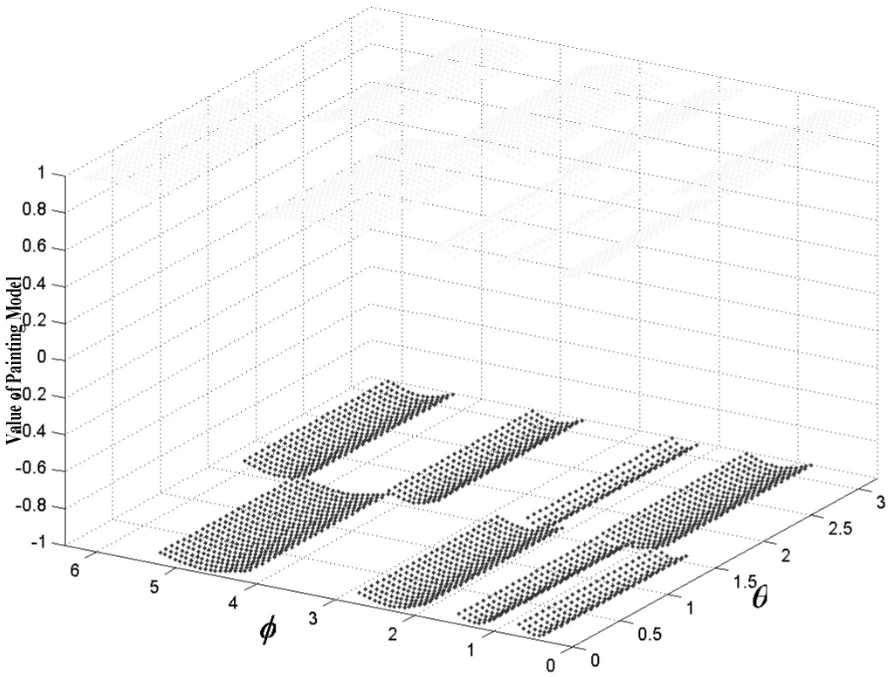

Fig. 8. Rounded painting model based on prototype painting.

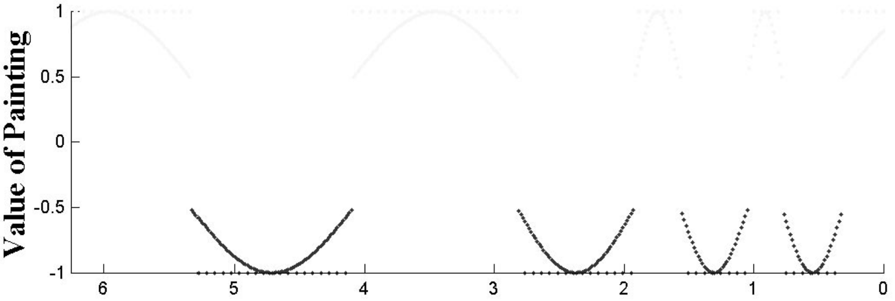

Fig. 9. Slice of the rounded prototype painting (not to scale).

model is called the *rounded painting model* and is constructed by rounding the edges of each solid painted region. In this rounded model the locations in the center of the painted regions are still given the full value of ±1, but the value of the painting is increased/decreased smoothly toward 0 by a small amount as the evaluation point moves toward the edge of the painted region. A rounded painting model based on the prototype sphere painting is shown in Fig. 8 with elevation on the θ axis and azimuth on the ϕ axis, and the value of the painting on the z axis. In this painting, the regions are only rounded in the ϕ direction. A slice of the northern hemisphere of the rounded painting model is shown in Fig. 9. This figure shows both a slice of the actual discrete painting and an exaggerated representation of the rounding used in the model.

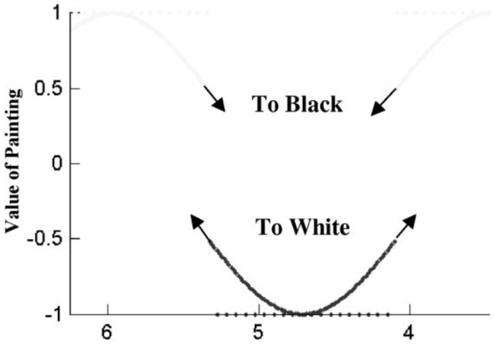

Fig. 10. Derivative "pointing" of rounded model (not to scale).

Now the derivative of the rounded painting model gives feedback on how specific rotations of the painted sphere affect the value of a fixed observation location. Since the edges of the painting are rounded toward zero, they "point" in the direction of the next painted region as shown in Fig. 10. The derivative can be said to "point" because it returns values of the derivative that tell whether rotation about a specific axis will bring the value of the point under observation toward a region of a different color or deeper into the same region. Only a very slight rounding of

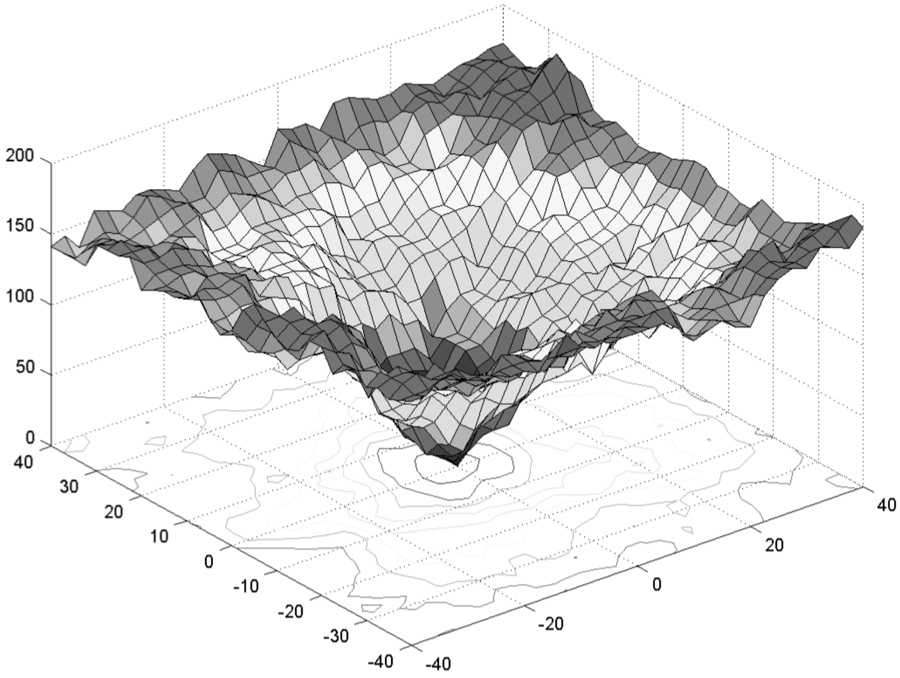

Fig. 11. $f(A)$ as A varies along two orthogonal axes.

the regions is used because excessive rounding would affect the ability of the model to converge by adding artificial forcing to the system.

Note that in this rounding model, the actual physical painting of the ball and sensor measurements are the same as before. The rounding only exists in the computer model to condition the Jacobian appropriately.

3) Numerical Jacobian-Based Orientation Decoding Algorithm: The numerical Jacobian-based orientation decoding algorithm starts by initializing the spherical encoder system at an orientation B. $A \in SO(3)$ is the unknown orientation that the sphere assumes an instant later. $\mathbf{c}(A)$ is the vector of sensor data with entries ± 1. It is analogous to the end-effector position $\mathbf{x}$ for a manipulator arm. $\mathbf{c}_{\text{model}}(B)$ is the vector of sensor values calculated from the rounded model evaluated in the initial orientation. The difference between the measured and modeled color vectors is set equal to the product of a Jacobian and a vector of changes in orientation parameters

$$\mathbf{c}(A) - \mathbf{c}_{\text{model}}(B) = J(B)\Delta\boldsymbol{\theta} \tag{12}$$

where

$$J(B) = \begin{bmatrix} E_1^R f_1(B) & E_2^R f_1(B) & E_3^R f_1(B) \\ E_1^R f_2(B) & E_2^R f_2(B) & E_3^R f_2(B) \\ \cdot & \cdot & \cdot \\ E_1^R f_i(B) & E_2^R f_i(B) & E_3^R f_i(B) \\ \cdot & \cdot & \cdot \\ E_1^R f_n(B) & E_2^R f_n(B) & E_3^R f_n(B) \end{bmatrix}$$

$$\Delta\boldsymbol{\theta} = \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \Delta\theta_3 \end{bmatrix}.$$

Here, $\Delta\theta_i$ is the magnitude of rotation in the E_i direction. The system has more inputs than variables so the Jacobian is not square. The Jacobian is inverted using a pseudo-inverse as follows.

First, both sides of the equation are multiplied on the left by $J^T(B)$

$$J^T(B)(\mathbf{c}(A) - \mathbf{c}_{\text{model}}(B)) = J^T(B)J(B)\Delta\boldsymbol{\theta}. \tag{13}$$

Next, both sides of the equation are multiplied on the left by $(J^T(B)J(B))^{-1}$ to yield:

$$\Delta\boldsymbol{\theta} = ((J^T(B)J(B))^{-1})J^T(B)(\mathbf{c}(A) - \mathbf{c}_{\text{model}}(B)). \tag{14}$$

The last know orientation B is updated with a rotation matrix constructed from $\Delta\boldsymbol{\theta}$. The orientation estimate is updated as $B \leftarrow B\exp(\sum_{i=1}^{3} E_i \cdot \Delta\theta_i)$. The algorithm is repeated until the magnitude of the error is reduced to a threshold value that depends on the amount of rounding added to the painting model. If the ball continues to move, the value of $\mathbf{c}(A)$ is updated also.

V. Global Encoder

The previous section presented techniques to find a $B \in SO(3)$ that solves the equation $\mathbf{c}(B) = \mathbf{c}(A)$ provided that a starting B sufficiently near A (within 15°) is provided. The previously presented methods fail if they are started too far from the solution because of the presence of many local minima and the highly nonsmooth nature of the functions $f(A)$ and $f_i(A)$. For example, Fig. 11 illustrates $f(A)$ as A varies along two orthogonal axes.

This creates a need to come up with a "rough" global spherical encoder technique to produce a close approximation to the orientation of the sphere. A global encoder is also needed if the sphere doesn't have a "home" configuration that can be used to initialize the encoder. The global orientation algorithms can then be used as the starting point for the models outlined in the previous sections. The problem with the global encoder techniques is that they require a lot of memory/storage. The following subsections outline two techniques for constructing global encoders. Detailed numerical and experimental studies performed on these techniques can be found in [29].

A. Matrix Decomposition

Since $SO(3)$ is compact and since the techniques outlined in the previous sections just need a starting value that is reasonably close to the correct value, $f(B)$ can be evaluated over a discrete subset of $SO(3)$. That is, choose a threshold (say 10°) and find a finite subset $Z \subset SO(3)$ so that for all $A \in SO(3)$ there is a $B \in Z$ such that the rotational difference between A and B is below the threshold.

The technique starts by precomputing the set $Z \subset SO(3)$ [17]. Record in a table the vector $\mathbf{c}(B)$ for all $B \in Z$. Suppose $Z = \{B_1, B_2, \ldots, B_k\}$. Create a $k \times n$ matrix M whose i, jth-entry is $c(B_i^T \mathbf{s}_j)$; i.e., this entry is the color observed by the jth sensor if the sphere is rotated by B_i from its home position.

Let $A \in SO(3)$ be arbitrary. Then choose B_i in $SO(3)$ so that $\mathbf{c}(A) \cdot \mathbf{c}(B_i)$ is as large as possible (best possible match). This can be done by a single matrix-vector multiply, $M\mathbf{c}(A)$, and finding the coordinate with largest index.

This calculation is reasonably fast, but the matrix M is quite large (for 100 sensors and 10 000 saved orientations, it has one million entries). One idea that saves some memory and speeds up the calculation is to replace M by a reduced-rank approximation.

Let $M = U\Sigma V^T$ be M's singular value decomposition [28]. Here, U is a $k \times k$ real orthogonal matrix, V is an $n \times n$ real orthogonal matrix, and Σ is a $k \times n$ diagonal matrix whose diagonal entries are real, nonnegative, and in decreasing order. Let r be a modest positive integer (e.g., $r = 10$) and let

$\hat{\Sigma}$ $r \times r$ upper left corner of Σ;
$\hat{U}$ $k \times r$ matrix formed by choosing just the first r columns of U;
$\hat{V}$ $n \times r$ matrix formed by choosing just the first r columns of V.

Then, $M \approx \hat{U}\hat{\Sigma}\hat{V}^T$. This approximate decomposition can be computed without finding the full singular value decomposition. It is a one-time computation that is reasonably fast.

Notice that these three matrices consume a total of $rk + r^2 + rn$ storage. For $k = 10\,000$, $n = 100$ and $r = 10$, this is about 100 000 which is significantly less than one million. We then approximate $M\mathbf{c}(A)$ by $\hat{U}\hat{\Sigma}\hat{V}^T\mathbf{c}(A)$, and select the largest component(s) to give reasonable starting values to the orientation decoding algorithms presented in the previous section. Our experience has been that we need to try a few of the largest starting values of the approximation to $M\mathbf{c}(A)$ to get a good starting value.

B. Sorted Tree Lookup

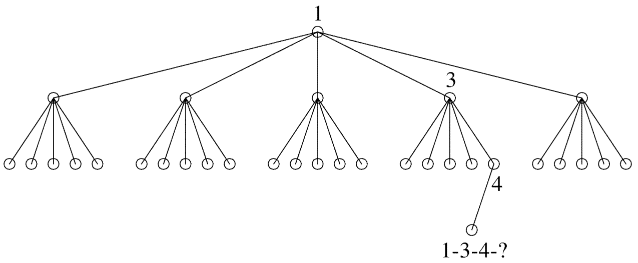

Fig. 12. Lookup tree.

This technique divides the set of unique sensor outputs for different orientations within the resolution of the encoder into several groupings. To illustrate this technique, consider an encoder with 100 sensors. The raw sensor data contains up to 2^{100} possible combinations. Can grouping the sensors help in searching such a large space? To start answering this question, the sensor output is encoded as 1 for black and -1 for white and the outputs of any ten sensors are summeed to produce a number between -10 and 10. The sensors are divided into 10 sets of 10 sensors. Each individual set of 10 sensor outputs are summed so that the possible combinations of sensor outputs is dramatically less than 2^{100}. This is used to re-sort the original M so that it becomes 13 by 10 000 (10 000 saved orientations), with each row containing the sum of each group of ten sensors and three parameters defining the orientation of the sphere.

If M is smaller then it is sorted using a tree sort. This is done by first rearranging M so that the entries are placed in ascending order. All the entries that do not have a first sum matching the sensor input are removed. The remaining entries which do not have a second entry matching the sensor input are removed. This is repeated for the ten groups until M is reduced to a very small number of entries. There will now be a small number of remaining entries in M so they can now be sorted using direct comparison. A tree sort is illustrated in Fig. 12 and shows a 4 level sort where the first sensor group sum is 1, the second is 3, etc.

Even with this regrouping of M, the problem is still too big to be sorted efficiently. Further sorting has to be pre-performed on M so that orientations matching the grouped sensor data can quickly be retrieved. The technique used to further sort M grows out of the tree sort. For a system with 96 sensors broken down into 8 sets of 12 (as is done in our prototype), 13^8 folders are created. The folders are labeled with 8 numbers. These numbers correspond to the sums of the sensors groupings; for example if a folder is labeled 1-4-6-13-0-6-8-8, the first sensor group sum would be 1, the second sum is 4, etc., M is now examined row by row and the three parameters defining the orientation of the sphere (Euler angles) are put in the appropriate folder. By putting only the orientation parameters in their appropriate folders, the amount of data is reduced further from 13 × 10 000 to 3 × 10 000. The real elegance of this technique comes from the fact that M is fully sorted. When the sensor inputs are polled, the appropriate folder just has to be retrieved and the possible orientations can then be sorted directly to find which one results in a bit pattern closest to the encoder input. The best match is then plugged into one of the algorithms previously presented to decode the current orientation of the sphere.

VI. Conclusion

Encoding spherical motion is essential in the feedback control of spherical motor systems. This paper presents mathematical models of spherical encoders based on a finite number of binary sensors and a two-color painting of a ball rotating within a housing. These mathematical models form the basis for decoding algorithms developed here for determining the orientation of the ball from sensor measurements. It is shown that these algorithms, which are for absolute encoders, perform well in terms of speed and accuracy. Speed is increased when these absolute encoders are used in an incremental mode in which the orientation of the ball at a previous instant is assumed to be known. The accuracy of the encoder depends on the distribution of sensor locations and the pattern with which the ball is painted. A methodology for evaluating the quality of a particular painting is also presented.

References

[1] F. Williams and E. R. Laithwaite, "A brushless variable-speed induction motor," *Proc. Inst. Elect. Eng.*, vol. 101, no. 11, pp. 203–213, 1954.

[2] F. Williams, E. R. Laithwaite, and L. S. Piggot, "Brushless variable-speed induction motor," *Proc. Inst. Elect. Eng.*, vol. 103, no. 6, pp. 102–122, 1956.

[3] F. Williams, E. R. Laithwaite, and J. F. Eastham, "Development and design of spherical induction motors," *Proc. Inst. Elect. Eng.*, vol. 106, pp. 471–484, 1959.

[4] E. R. Laithwaite, "Design of spherical motors," *Elect. Times*, pp. 921–925, 1960.

[5] K. Davey and G. Vachtsevanos, "The analysis of fields and torques in a spherical induction motor," *IEEE Trans. Magn.*, vol. MAG-23, pp. 273–282, Mar. 1987.

[6] ——, "Spherical motor particularly adapted for robotics," U.S. Patent no. 4 739 241, Apr. 19, 1988.

[7] Y. Kaneko, I. Yamada, and K. Itao, "A spherical DC servo motor with three degrees-of-freedom," *ASME Dyn. Syst. Contr. Div.*, vol. 11, pp. 398–402, 1988.

[8] K.-M. Lee, G. Vachtsevanos, and C. Kwan, "Development of a spherical stepper wrist motor," in *Proc. IEEE Conf. Robotics and Automation*, Philadelphia, PA, Apr. 15–29, 1988, pp. 267–272.

[9] J. Pei, "Methodology of design and analysis of variable-reluctance spherical motors," Ph.D. dissertation, Dept. of Mech. Eng., Georgia Inst. of Technol., Atlanta, GA, 1990.

[10] K.-M. Lee and C.-K. Kwan, "Design concept development of a spherical stepper for robotic applications," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 175–181, Feb. 1991.

[11] R. B. Roth and K.-M. Lee, "Design optimization of a three degrees-of-freedom variable-reluctance spherical wrist motor," *ASME J. Eng. Ind.*, vol. 117, pp. 378–388, 1995.

[12] Z. Zhou and K.-M. Lee, "Real-time motion control of a multi-degree-of-freedom variable reluctance spherical motor," in *Proc. IEEE Int. Conf. Robotics and Automation*, Minneapolis, MN, Apr. 1996, pp. 2859–2864.

[13] T. He, "Effects of rotor configuration on the characteristic torque of a variable-reluctance spherical motor," Ph.D, dissertation, Dept. of Mech. Eng., Georgia Inst. of Technol., Atlanta, GA, 2000.

[14] S. Toyama, S. Sugitani, G. Zhang, Y. Miyatani, and K. Nakamura, "Multi degree of freedom spherical ultrasonic motor," in *Proc. IEEE Int. Conf. Robotics and Automation*, Nagoya, Japan, 1995, pp. 2935–2940.

[15] S. Toyama, G. Zhang, and O. Miyoshi, "Development of new generation spherical ultrasonic motor," in *Proc. IEEE Int. Conf. Robotics and Automation*, Minneapolis, MN, Apr. 1996, pp. 2871–2876.

[16] G. S. Chirikjian and D. Stein, "Kinematic design and commutation of a spherical stepper motor," *IEEE/ASME Trans. Mechatron.*, vol. 4, pp. 342–353, Dec. 1999.

[17] E. R. Scheinerman, G. S. Chirikjian, and D. Stein, "Encoders for spherical motion using discrete sensors," in *Proc. Workshop Algorithmic Foundations of Robotics*, Hanover, NH, Mar. 2000.

[18] D. Stein, G. S. Chirikjian, and E. R. Scheinerman, "Theory, design, and implementation of a spherical encoder," in *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, May 2001, Available: CDROM.

[19] R. P. Pettypiece, Jr., "Spherical optical encoder for detecting the position and motion about three mutual orthogonal axes," U.S. Patent no. 5 223 709, 1993.

[20] K. M. Lee, "Orientation sensing system and method for a spherical body," U.S. Patent no. 5 319 577, 1994.

[21] J. A. Bondy and U. S. R. Murty, *Graph Theory With Applications*. New York: North-Holland, 1976.

[22] D. B. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[23] G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis*. Boca Raton, FL: CRC Press, Oct. 2000.

[24] I. M. Gel'fand, R. A. Minlos, and Z. Ya. Shapiro, *Representations of the Rotation and Lorenz Groups and Their Applications*. New York: MacMillian, 1963.

[25] S. Helgason, *Differential Geometry, Lie Groups, and Symmetric Spaces*. New York: Academic, 1978.

[26] W. Miller, Jr., *Lie Theory and Special Functions*. New York: Academic, 1968.

[27] C. J. Taylor and D. J. Kriegman, "Minimizations on the Lie group SO(3) and related manifolds," Yale Center for Systems Science, New Haven, CT, Tech. Rep. 9405, Apr. 1994.

[28] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[29] D. Stein, "Design of a spherical stepper motor system," Ph.D. dissertation, Dep. of Mech. Eng., Johns Hopkins Univ., Baltimore, MD, Aug. 2001.

David Stein received the B.S.E. degree in biomedical engineering and the M.S.E. and Ph.D. degrees in mechanical engineering, all from the Johns Hopkins University, Baltimore, MD.

He is currently Manager of the Systems Engineering Group of the Instrument Systems Division of Diagnostic Products Corporation (DPC), Flanders, NJ.

Edward R. Scheinerman received the Bachelor's degree from Brown University, Providence, RI, and the doctoral degree from Princeton University, Princeton, NJ.

He is a Professor and Chair of Mathematical Sciences at Johns Hopkins University, Baltimore, MD. He serves on the editorial boards of the Journal of Graph Theory and the American Mathematical Monthly.

Gregory S. Chirikjian (M'93) received the Ph.D. degree from the California Institute of Technology, Pasadena, CA, in 1992.

Since the summer of 1992, he has been with the Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD, where he is now a Professor. His research interests include the kinematic analysis, motion planning, design, and implementation of biologically inspired robots. In particular, "hyper-redundant," "metamorphic," and "binary" manipulators, and most recently self-replicating robots. In recent years he has also been applying methods from robotics to model conformational transitions in biological macromolecules.

Dr. Chirikjian is a 1993 NSF Young investigator, a 1994 Presidential Faculty Fellow, and a 1996 recipient of the ASME Pi Tau Sigma Gold Medal.