REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 05-05-2006	**2. REPORT TYPE** Paper PREPRINT	**3. DATES COVERED** *(From - To)* 2006

4. TITLE AND SUBTITLE Rapid, Robust, Optimal Pose Estimation from a Single Affine Image (PREPRINT)	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) John E. McInroy[1], R. Scott Erwin[2], Lawrence M. Robertson[2]	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) [1]Department of Electrical and Computer Engineering University of Wyoming Dept 3295 1000 E. University Ave Laramie, WY 82071	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) [2]Air Force Research Laboratory Space Vehicles 3550 Aberdeen Ave SE Kirtland AFB, NM 87117-5776	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFRL/VSSV
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-VS-PS-JA-2006-1011

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited. (Clearance # VS06-0180)

13. SUPPLEMENTARY NOTES

Submitted to IEEE Transactions on Robotics
Government Purpose Rights

14. ABSTRACT

Determining the rigid transformation relating a 2d image to known geometry is a classical problem in computer vision. To date, the most accurate methods require performing an unknown number of iterations until a numerical algorithm converges to the desired tolerance. For the case of affine imaging, this paper replaces these nonlinear numerical iterations with solving the standard 3d-3d optimal orientation problem 2^n times, where n is the number of data points. The 2^n successive optimal orientation calculations are speeded through use of Gray code, and have the dual advantages of speed and predictable execution time. Angular errors caused by scaling imperfections are quantified, and a least upper bound estimate of the scaling is proposed. It is shown that the worst case viewpoints depend only on the data points chosen, and a new convex linear matrix inequality optimization is derived for determining the worst viewpoint. This new analysis tool is useful for evaluating a particular set of data and suggests methods of designing the data for high performance.

15. SUBJECT TERMS

Affine Cameras, Pose Estimation, Visual Servoing, Template Matching, Object Recognition

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Lawrence Robertson
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	Unclassified	Unlimited	11	**19b. TELEPHONE NUMBER** *(include area code)* 505-846-7687

Rapid, Robust, Optimal Pose Estimation from a Single Affine Image

John E. McInroy, *Senior Member, IEEE,* R. Scott Erwin and Lawrence M. Robertson

***Abstract*— Determining the rigid transformation relating a 2d image to known geometry is a classical problem in computer vision. To date, the most accurate methods require performing an unknown number of iterations until a numerical algorithm converges to the desired tolerance. For the case of affine imaging, this paper replaces these nonlinear numerical iterations with solving the standard 3d-3d optimal orientation problem 2^n times, where n is the number of data points. The 2^n successive optimal orientation calculations are speeded through use of Gray code, and have the dual advantages of speed and predictable execution time. Angular errors caused by scaling imperfections are quantified, and a least upper bound estimate of the scaling is proposed. It is shown that the worst case viewpoints depend only on the data points chosen, and a new convex linear matrix inequality optimization is derived for determining the worst viewpoint. This new analysis tool is useful for evaluating a particular set of data and suggests methods of designing the data for high performance.**

***Index Terms*— affine cameras, pose estimation, visual servoing, template matching, object recognition**

I. Introduction

Determining the rigid transformation relating an image to known geometry, i.e. pose estimation, is a central problem in computer graphics, computer vision, robotics, and photogrammetry. In computer graphics, it is important in tasks which combine computer generated objects with natural photos. In computer vision, it is central to object recognition. In robotics, it is useful for coordinating the motion of the hand and the eye. In photogrammetry, it is key to detailed inspection from grainy images.

Iterative, nonlinear numerical optimizations can provide fully optimal solutions which are therefore of the highest possible accuracy. There are many examples of such methods. Since this paper is concerned with rapid techniques, [10] is a fast iterative technique based on object, rather than image, iterations. Global convergence is found. Both [3] and [5] use iteratively re-weighted least squares techniques for tracking rapidly. These approaches are based on linearization using the Lie group's infinitestimal generator, thus they are suitable for tracking small changes in pose between images. Let SO(3) denote the three dimensional Special Orthogonal group which models rotation. Cyclic coordinate descent, which alternately finds the optimal rotation, $R \in SO(3)$, then Euclidean terms such as the translation, then recalculates R and so forth is used for camera calibration [12], [8].

Closed form solutions are available for $n = 3$ or $n = 4$ correspondence points. The roots of a fourth or fifth order polynomial contain the solution ([4] provides one example algorithm and references for other algorithms). For a small set of points when $n > 4$, non-iterative techniques methods are found in [14], [6], and [1]. The solution in [1] solves a quadratic problem through an over-parameterization, then multiple linear singular value decompositions. The number of variables generated can be high. For instance, the constraint $R^T R = I$ requires 45 variables to represent this three dimensional element in SO(3).

Despite the availability of many algorithms, there remains a need for a rapid, optimal, dependable algorithm suitable for a small number of point correspondences (but more than $n = 4$). For instance, [15] recently added high bandwidth inertial sensing to complement the available pose estimation because it alone had too low of a bandwidth. For the case of affine imaging, this paper develops a new method that is closed form, yet provides optimal estimates even when $n > 4$. It does so by solving the standard 3d-3d optimal orientation problem 2^n times, where n is the number of data points. Since the optimal orientation problem can be very quickly solved as a 4 x 4 matrix eigenvalue problem, the overall computational burden is light. However, the method clearly becomes less attractive for large n, as 2^n becomes extremely large. To help offset this problem, the 2^n successive optimal orientation calculations are speeded through use of Gray code, thus they have the dual advantages of speed and predictable execution time. Angular errors caused by scaling imperfections are quantified, and a least upper bound estimate of the scaling is proposed. It is shown that the worst case viewpoints depend only on the data points chosen, and a new convex linear matrix inequality optimization is derived for determining the worst viewpoint. This new analysis tool is useful for evaluating a particular set of data. In addition, it suggests methods of designing the data for high performance.

II. Optimal Estimation of Pose from 3-D Data

First, classical techniques [17] for solving the optimal orientation problem will be briefly reviewed. Quaternion based methods will be used here–see [10] for a description of singular value decomposition based methods. A complete proof of the optimal quaternion algorithm is presented to emphasize the structure of the solution. This bilinear structure will be heavily exploited later in this paper.

From an object's geometry, direction vectors $\vec{z}_i \in \mathbb{R}^3,\ i = 1, \ldots, n$ are known in an object's coordinate frame. Those same vectors, $\vec{v}_i \in \mathbb{R}^3$, are measured in a sensor coordinate frame. Theorem 1 provides optimal estimates of the rotation matrix from the object to the sensor frame, R. First, define

This work was supported by the National Research Council and Air Force Office of Scientific Research.

J. McInroy is with the Department of Electrical and Computer Engineering E-mail:mcinroy@uwyo.edu, Phone: (307) 766-6137 , fax: (307)766-2248.

L. Robertson and S. Erwin are with the Air Force Research Laboratory, Space Vehicles Directorate, AFRL/VSSV, Bldg. 472 Kirtland AFB, NM, 87117-5776, E-mail:lawrence.robertson@kirtland.af.mil, richard.erwin@kirtland.af.mil..

the hat function $(\hat{\cdot})$ as the cross product matrix, i.e.

$$\hat{\vec{z}} = \left[\begin{array}{ccc} 0 & -z_3 & z_2 \\ z_3 & 0 & -z_1 \\ -z_2 & z_1 & 0 \end{array} \right]$$

Theorem *1:* Given $\vec{z}_i$, $\vec{v}_i \in \mathbb{R}^3$ and $w_i \in \mathbb{R}_+$, $i = 1 \dots n$, the minimum of

$$J = \sum_{i=1}^{n} w_i ||R\vec{z}_i - \alpha\vec{v}_i||^2 \tag{1}$$

over $R \in$ SO(3), $\alpha \in \mathbb{R}_+$ is found by first calculating

$$D = \sum_{i=1}^{n} w_i K(\vec{v}_i, \vec{z}_i) \tag{2}$$

where K is the bilinear, symmetric matrix function

$$K(\vec{v}, \vec{z}) = \left[\begin{array}{cc} \vec{v}\vec{z}^T + \vec{z}\vec{v}^T - 2\vec{v}^T\vec{z}I & \hat{\vec{z}}\vec{v} \\ (\hat{\vec{z}}\vec{v})^T & 0 \end{array} \right] \tag{3}$$

The eigenvector, $\vec{\epsilon}$, of D corresponding to the maximum eigenvalue is the unit quaternion representation of the optimal R. The optimal scaling is

$$\alpha = \frac{\sum_{i=1}^{n} w_i \vec{v}_i^T R \vec{z}_i}{\sum_{i=1}^{n} w_i ||\vec{v}_i||^2} \tag{4}$$

Proof: Any $R \in$ SO(3) can be written as the following quadratic function of a unit quaternion:

$$R = 2[(0.5 - \vec{e}^T\vec{e})I + \vec{e}\vec{e}^T + \epsilon_4\hat{\vec{e}}] \tag{5}$$

where $\vec{\epsilon}^T = [\epsilon_1\ \epsilon_2\ \epsilon_2\ \epsilon_4] \in \mathbb{R}^4$ is the unit quaternion. The first three elements of $\vec{\epsilon}$ form the $\vec{e}$ vector, $\vec{e}^T = [\epsilon_1\ \epsilon_2\ \epsilon_3]$. The optimization problem with objective (1) can be rewritten in terms of unit quaternions as

$$\min_{\vec{\epsilon}, \alpha} \sum_{i=1}^{n} w_i ||2[(0.5 - \vec{e}^T\vec{e})I + \vec{e}\vec{e}^T + \epsilon_4\hat{\vec{e}}]\vec{z}_i - \alpha\vec{v}_i||^2 \tag{6}$$

subject to $\vec{\epsilon}^T\vec{\epsilon} = 1$. Multiplying out this equation, and discarding terms that are not a function of either $\vec{\epsilon}$ or α, the Lagrangian can be formed:

$$L = -2\alpha \sum_{i=1}^{n} w_i \vec{v}_i^T 2[(0.5 - e^Te)I + ee^T + \epsilon_4\hat{\vec{e}}]z_i$$

$$+\alpha^2 \sum_{i=1}^{n} w_i ||\vec{v}_i||^2 + \lambda(1 - \vec{\epsilon}^T\vec{\epsilon})$$

where λ is a Lagrange multiplier. Before calculating the necessary conditions for an optimum, first consider the simpler term $\vec{v}^T R\vec{z} = 2[0.5\vec{v}^T\vec{z} - \vec{v}^T\vec{e}^T\vec{e}\vec{z} + \vec{v}^T\vec{e}\vec{e}^T\vec{z} + \epsilon_4\vec{v}^T\hat{\vec{e}}\vec{z}]$. Since $\vec{x}^T\vec{y} = \vec{y}^T\vec{x}$, $\hat{\vec{x}}\vec{y} = -\hat{\vec{y}}\vec{x}$ and $\hat{\vec{x}}^T = -\hat{\vec{x}}$ for any vectors $\vec{x}$ and $\vec{y}$, $\vec{v}^T R\vec{z} = 2[0.5\vec{v}^T\vec{z} - \vec{v}^T\vec{z}\vec{e}^T\vec{e} + \vec{e}^T\vec{z}\vec{v}^T\vec{e} - \epsilon_4\vec{v}^T\hat{\vec{z}}\vec{e}]$. Then

$$\frac{\partial\vec{v}^T R\vec{z}}{\partial\vec{e}} = 2[-2\vec{v}^T\vec{z}\vec{e} + \vec{z}\vec{v}^T\vec{e} + \vec{v}\vec{z}^T\vec{e} + \epsilon_4\hat{\vec{z}}\vec{v}]$$

and

$$\frac{\partial\vec{v}^T R\vec{z}}{\partial\epsilon_4} = 2[\hat{\vec{z}}\vec{v}]^T\vec{e}$$

These can be combined into a single vector derivative

$$\frac{\partial\vec{v}^T R\vec{z}}{\partial\vec{\epsilon}} = 2 \left[\begin{array}{cc} -2\vec{v}^T\vec{z}I + \vec{z}\vec{v}^T + \vec{v}\vec{z}^T & \hat{\vec{z}}\vec{v} \\ (\hat{\vec{z}}\vec{v})^T & 0 \end{array} \right] \vec{\epsilon} = 2K(\vec{v}, \vec{z})\vec{\epsilon}$$

Note that the matrix function $K(\vec{v}, \vec{z})$ is symmetric and linear in each of its arguments (bilinear overall).

Applying these formulas to each term of the Lagrangian, the necessary conditions are found to be

$$\frac{\partial L}{\partial\vec{e}} = 0 = -2\alpha \sum_{i=1}^{n} w_i[-2\vec{v}_i^T\vec{z}_i + \vec{z}_i\vec{v}_i^T + \vec{v}_i\vec{z}_i^T]\vec{e}$$

$$-2\alpha \sum_{i=1}^{n} w_i\hat{\vec{z}}\vec{v}\epsilon_4 - 2\lambda\vec{e} \tag{7}$$

and

$$\frac{\partial L}{\partial\epsilon_4} = 0 = -2\alpha \sum_{i=1}^{n} w_i 2[\hat{\vec{z}}\vec{v}]^T\vec{e} - 2\lambda\epsilon_4 \tag{8}$$

or

$$\frac{\partial L}{\partial\vec{\epsilon}} = 0 = -4\alpha \sum_{i=1}^{n} w_i K(\vec{v}_i, \vec{z}_i)\vec{\epsilon} - 2\lambda\vec{\epsilon} \tag{9}$$

The matrix $D = \sum_{i=1}^{n} w_i K(\vec{v}_i, \vec{z}_i)$ is a 3 × 3 symmetric matrix that can be calculated directly from the given data. The optimal quaternion is the solution to

$$-2\alpha D\vec{\epsilon} = \lambda\vec{\epsilon}$$

Thus $\vec{\epsilon}$ is an eigenvector of D. Note that the partial of the objective is

$$\frac{\partial J}{\partial\vec{\epsilon}} = -4\alpha D\vec{\epsilon} = 2\lambda\vec{\epsilon}$$

Thus $J = 2\lambda\vec{\epsilon}^T\vec{\epsilon} = 2\lambda$ plus terms constant in $\vec{\epsilon}$. The minimal value of J is thus given by the minimal eigenvalue λ of $-2\alpha D$. Since α scales all the eigenvalues equally, the minimum J is given by the maximum eigenvalue of D. Hence the optimal quaternion is that eigenvector of D corresponding to the maximum eigenvalue. Once $\vec{\epsilon}$ is found, the optimal rotation matrix can be found from (5).

Note that the optimal R is independent of the scaling. Thus scaling all sensor measurements, $\vec{v}_i$, by the same amount will not affect the orientation estimate. Once R is found, it can be used to calculate the scaling term. The necessary condition for optimality is

$$\frac{\partial L}{\partial\alpha} = \frac{\partial J}{\partial\alpha} = 0 = -2\sum_{i=1}^{n} w_i\vec{v}_i^T R\vec{z}_i + 2\alpha\sum_{i=1}^{n} w_i||\vec{v}_i||^2$$

Rearranging gives (4).

□

Remarks: Theorem 1 is a classical result [17]. It requires solution of a 4 x 4 matrix eigenvalue problem. This paper utilizes it because it is a standard technique, but it should be noted that other faster techniques may be fruitful for integration with the this paper's main new contributions. For instance, [9] suggests an optimal orientation algorithm requiring solution of only a 3 x 3 eigenvalue problem.

As Theorem 1 shows, the solution is invariant with respect to scaling. This effect will be of crucial importance when this algorithm is extended further to use 2-dimensional (2-d), rather than 3-dimensional (3-d) data. In addition, this formulation of the solution highlights a powerful property (bilinearity of K) that will be exploited for pose estimation from both 3-d and 2-d data. Proposition 2 will use bilinearity to quantify the effect of misclassification.

Proposition *2:* Suppose vectors $\vec{v}_j$ and $\vec{v}_k$ are misclassified, i.e. $\vec{v}_j$ is matched incorrectly with $\vec{z}_k$, and $\vec{v}_k$ is matched incorrectly with $\vec{z}_j$. Then D changes by

$$\Delta D = K(\vec{v}_j - \vec{v}_k, w_k\vec{z}_k - w_j\vec{z}_j) \qquad (10)$$

Proof: Most terms in the incorrect summation, D', are the same as the correct summation, D. Only those involving $\vec{v}_j$ and $\vec{v}_k$ differ. Thus from (2), $D' = D - w_jK(\vec{v}_j, \vec{z}_j) - w_kK(\vec{v}_k, \vec{z}_k) + w_jK(\vec{v}_k, \vec{z}_j) + w_kK(\vec{v}_j, \vec{z}_k)$. Since K is linear in each of its arguments, $\Delta D = D' - D = w_jK(\vec{v}_k - \vec{v}_j, \vec{z}_j) + w_kK(\vec{v}_j - \vec{v}_k, \vec{z}_k)$. Moving the weights inside K, $\Delta D = K(\vec{v}_k - \vec{v}_j, w_j\vec{z}_j - w_k\vec{z}_k)$.

□

Remarks: Vectors are most likely to be misclassified if they are close together, i.e. $\vec{v}_j - \vec{v}_k$ is small and $w_k\vec{z}_k - w_j\vec{z}_j$ is small. Prop. 2 shows that these two (typically small) vectors are multiplied together to change D. Hence ΔD is also small for most misclassifications. D totally determines the solution, hence the solution also has small errors when close vectors are misclassified.

The next proposition shows that, if the data points are chosen to satisfy a mild symmetry condition, then the non-linearly coupled estimation of pose (combined position and orientation) gives way to an analytic, decoupled solution.

Proposition *3:* Given $\vec{z}_i$, $\vec{q}_i \in \mathbb{R}^3$, $w_i \in \mathbb{R}_+$, $i = 1 \dots n$, $\alpha \in \mathbb{R}_+$, such that

$$\sum_{i=1}^{n} w_i\vec{z}_i = 0 \qquad (11)$$

, the minimum of

$$J = \sum_{i=1}^{n} w_i||R\vec{z}_i + \vec{p} - \alpha\vec{q}_i||^2 \qquad (12)$$

over $g = (R, \vec{p}) \in$ SE(3) ($R \in$ SO(3), $\vec{p} \in \mathbb{R}^3$), is found by first calculating

$$D = \sum_{i=1}^{n} w_iK(\vec{q}_i, \vec{z}_i)$$

where K is the bilinear, symmetric matrix function given in (3). The eigenvector, $\vec{\epsilon}$, of D corresponding to the maximum eigenvalue is the unit quaternion representation of the optimal R. The optimal $\vec{p}$ is

$$\vec{p} = \frac{\alpha\sum_{i=1}^{n} w_i\vec{q}_i - \sum_{i=1}^{n} w_iR\vec{z}_i}{\sum_{i=1}^{n} w_i} \qquad (13)$$

Proof: Multiplying out J yields

$$J = \sum_{i=1}^{n} w_i[\vec{z}_i^T\vec{z}_i + 2(\vec{p} - \alpha\vec{q}_i)^TRz_i + (\vec{p} - \alpha\vec{q}_i)^T(\vec{p} - \alpha\vec{q}_i)]$$

As in Theorem 1, the objective will be rewritten in terms of quaternions and a Lagrangian will be formed, $L = J + \lambda(1 - \vec{\epsilon}^T\vec{\epsilon})$. Using the same basic steps as those of Theorem 1, it can be shown that

$$\frac{\partial J}{\partial\vec{\epsilon}} = 4\sum_{i=1}^{n} w_iK(\vec{p} - \alpha\vec{q}_i, \vec{z}_i)\vec{\epsilon}$$

Thus

$$\frac{\partial L}{\partial\vec{\epsilon}} = 0 = \frac{\partial J}{\partial\vec{\epsilon}} - 2\lambda\vec{\epsilon} = 4\sum_{i=1}^{n} w_iK(\vec{p} - \alpha\vec{q}_i, \vec{z}_i)\vec{\epsilon} - 2\lambda\vec{\epsilon}$$

This equation can be significantly simplified by use of the bilinearity of K

$$\begin{aligned} 0 &= [4\sum_{i=1}^{n} w_iK(\vec{p}, \vec{z}_i) - 4\alpha\sum_{i=1}^{n} w_iK(\vec{q}_i, \vec{z}_i)]\vec{\epsilon} - 2\lambda\vec{\epsilon} \\ &= 4K(\vec{p}, \sum_{i=1}^{n} w_i\vec{z}_i)\vec{\epsilon} - 4\alpha D\vec{\epsilon} - 2\lambda\vec{\epsilon} \\ &= 4K(\vec{p}, 0)\vec{\epsilon} - 4\alpha D\vec{\epsilon} - 2\lambda\vec{\epsilon} = 0 \end{aligned}$$

where D is calculated from (2) by replacing $\vec{v}_i$ with $\vec{q}_i$. A zero argument in any linear function returns zero, thus $K(\cdot, 0) = K(0, \cdot) = 0$. Finally,

$$-4\alpha D\vec{\epsilon} = 2\lambda\vec{\epsilon}$$

Thus the optimal unit quaternion, $\vec{\epsilon}$, is the eigenvector of D corresponding to the maximum eigenvalue. The optimal R can be found directly from $\vec{\epsilon}$ using (5). Once R is found, the translational vector $\vec{p}$ can be easily calculated:

$$\frac{\partial L}{\partial\vec{p}} = 0 = \frac{\partial J}{\partial\vec{p}} = \sum_{i=1}^{n} w_i[2R\vec{z}_i + 2\vec{p} - 2\alpha\vec{q}_i]$$

Solving for $\vec{p}$ yields (13).

□

Remarks: Without the symmetry condition (11), direct estimation of the optimal pose requires a highly coupled, non-linear numerical optimization. In contrast, the new solution is very simple and fast to calculate, requiring only an eigenvalue operation on a 4 × 4 matrix for R and a summation operation for $\vec{p}$. The condition $\sum_{i=1}^{n} w_i\vec{z}_i = 0$ is very mild and can often be realized in practice. One method is by selecting the points, $\vec{z}_i$, so they satisfy (11). For instance, one solution is to always choose the $\vec{z}$'s in symmetrical pairs so both $\vec{z}$ and $-\vec{z}$ are members of $\vec{z}_i$, $i = 1 \dots n$, and both $\vec{z}$ and $-\vec{z}$ have the same weighting. This can often be accomplished in practical applications, although occlusions may limit its use at times.

A second, fundamentally separate method for satisfying (11) is by choosing the weights. Letting $Z = [\vec{z}_1\ \vec{z}_2 \dots\ \vec{z}_n]$ and $\vec{w}^T = [w_1\ w_2\ \dots w_n]$, the condition $\sum_{i=1}^{n} w_i\vec{z}_i = 0$ can also be written as $Z\vec{w} = 0$. Given Z, $\vec{w}$ must be chosen in the null space of Z. Generation of the null space is very straightforward. However, to maintain a convex objective function, $\vec{w}$ must have all non-negative elements. This can also often be easily accomplished, see [7] for further details.

Simultaneous selection of both the data points, $\vec{z}_i$, and the weights, w_i, can also performed to increase the space satisfying (11).

Also note that when calculating the optimal pose, the data scaling (α) should be known, as it is needed when calculating $\vec{p}$.

III. Optimal Estimation of Pose from 2-D Data

Affine imaging systems [16] are a limiting case of perspective cameras for the situation wherein the depth of the correspondence points is much larger than the size of the object. All correspondence points are then scaled by approximately the same value, and the data loss becomes a linear loss of one dimension (along the sensing axis).

Without loss of generality, assume the standard camera model with camera axis along the "z" dimension, so that the missing component of data is the third (or "z") component. When one dimension of the data is missing, the pose estimation problem can be phrased as:

> Given $\vec{z}_i \in \mathbb{R}^3$, $\vec{d}_i \in \mathbb{R}^2$, $w_i \in \mathbb{R}_+$, $i = 1 \ldots n$, $P = [I_2\ 0]$, find the minimum of
>
> $$J = \sum_{i=1}^{n} w_i ||P[R\vec{z}_i + \vec{p}] - \alpha \vec{d}_i||^2 \qquad (14)$$
>
> over $g = (R, \vec{p}) \in$ SE(3) ($R \in$ SO(3), $\vec{p} \in \mathbb{R}^3$), and $\alpha \in \mathbb{R}_+$.

Direct minimization of (14) involves a nonlinear numerical optimization. This paper will derive a much faster method via a sequence of analytic solutions.

A. Orientation Estimation: Scaling Known

First, the estimation of orientation alone will be treated. As will soon be explained, orientation estimation from 2-d data does not enjoy the invariance to scaling found in Theorem 1, so the case where scaling is known will be solved, then scaling will be estimated separately.

Proposition *4:* Given $\vec{z}_i \in \mathbb{R}^3$, $\vec{d}_i \in \mathbb{R}^2$, and $w_i \in \mathbb{R}_+$, $i = 1 \ldots n$, $P = [I_2\ 0]$ the minimum of

$$J_{2d} = \sum_{i=1}^{n} w_i ||PR\vec{z}_i - \vec{d}_i||^2 \qquad (15)$$

over $R \in$ SO(3), is found by first calculating

$$\vec{v}_i = \left[\begin{array}{c} \vec{d}_i \\ \cdots \\ \pm\sqrt{||\vec{z}_i||^2 - ||\vec{d}_i||^2} \end{array} \right], \; i = 1 \ldots n \qquad (16)$$

Next, the 2^n 3-d orientation problems corresponding to all possible v_i, $i = 1 \ldots n$ ± sign choices are solved using Theorem 1:

$$\min_{R \in SO(3)} \sum_{i=1}^{n} w_i ||R\vec{z}_i - \vec{v}_i||^2$$

From these 2^n solutions, the optimum is that minimizing J_{2d}.

Proof: Let $\vec{v}_i^T = [\vec{d}_i^T\ v_{z_i}]^T$ where v_{z_i} is the missing component of $\vec{v}_i$. Since rotation of a vector does not change its length, $||\vec{z}_i||^2 = ||\vec{v}_i||^2 = ||\vec{d}_i||^2 + v_{z_i}^2$. Therefore $v_{z_i} = \pm\sqrt{||\vec{z}_i||^2 - ||\vec{d}_i||^2}$. The knowledge of the vector's length in one frame has been exploited to find the missing component of each data vector, up to a sign. That choice of signs minimizing the objective is the optimal solution.

□

Remarks: The re-calculation of $D = \sum_{i=1}^{n} w_i K(\vec{v}_i, \vec{z}_i)$ necessary for each of the 2^n solutions of the 3-d optimal orientation algorithm (Theorem 1) can be expedited in two ways. First, recall that Gray code is a binary code wherein each successive element toggles only one sign. Thus, by searching through the signs in an order specified by Gray code, only one sign will change between each successive solution. Suppose D has already been calculated, but for the next step the sign of v_{z_i} is toggled:

$$\vec{v}_{j+} = \left[\begin{array}{c} \vec{d}_j \\ v_{z_j} \end{array} \right] \rightarrow \vec{v}_{j-} = \left[\begin{array}{c} \vec{d}_j \\ -v_{z_j} \end{array} \right]$$

The new D, D', is $D' = D + w_j[K(\vec{v}_{j-}, \vec{z}_j) - K(\vec{v}_{j+}, \vec{z}_j)]$. Bilinearity of K reduces this to $D' = D + w_j K(\vec{v}_{j-} - \vec{v}_{j+}, \vec{z}_j) = D + w_j v_{z_j} K(\vec{z}, \vec{z}_j)$ where $\vec{z}^T = [0\ 0\ 1]$. Inserting the constant $\vec{z}$ into (3) yields

$$\Delta D = D' - D = w_j \left[\begin{array}{cccc} -2z_{3_j} & 0 & z_{1_j} & z_{2_j} \\ 0 & -2z_{3_j} & z_{2_j} & -z_{1_j} \\ z_{1_j} & z_{2_j} & 0 & 0 \\ z_{2_j} & -z_{1_j} & 0 & 0 \end{array} \right] \qquad (17)$$

where $\vec{z}_j^T = [z_{1_j}\ z_{2_j}\ z_{3_j}]$ This implies that each successive solution can be quickly calculated–simply insert $\vec{z}_j$ and w_j into (17), to get the new $D = D + \Delta D$. The optimal quaternion is then the eigenvector of D with maximal eigenvalue.

B. Errors Due to Scaling

Although estimation of orientation from 3d data is invariant to scaling (Theorem 1), estimation of orientation from 2d data using Prop. 4 is *not* invariant to scaling. Thus, before using Prop. 4, the data should be scaled. However, knowing the scaling is tantamount to knowing the depth. For instance, under the perspective transformation common to most optical systems, image measurements are scaled by the depth. It is, of course, possible to use another separate depth sensor. However, this subsection will show that scaling can be easily estimated from 2d data and target geometry for *any* orientation. Independence from a particular orientation is novel, as prior algorithms have required coarse knowledge of the orientation to get coarse scaling estimates. Estimating scaling is closely related to estimating depth. See [1] for a more complex, yet linear method for recovering the different depths of several points, when the affine assumption of nearly equal depths does not hold.

The effects of scaling estimation errors are studied, and are found to depend upon the geometry of the data vectors, $\vec{z}_i$, as well as the viewpoint. Convex methods for finding the worst case viewpoint and estimation error for a given set of data vectors are developed. A solution to the problem via linear matrix inequalities is derived which efficiently finds the global optimum.

Proposition *5:* If the true scaling, α, and estimated scaling, α_e, differ by the factor $\beta = \alpha_e / \alpha$, then the angular error between the actual measurement vector, $\vec{v}_i$, and its estimate, $\vec{v}_{e_i}$, is

$$\theta = \text{acos}\left[\beta p_i^2 + \sqrt{(1-p_i^2)(1-\beta^2 p_i^2)}\right] \tag{18}$$

where

$$p_i = \frac{\alpha||\vec{d}_i||}{||\vec{z}_i||} \tag{19}$$

is the percentage of the measurement's norm passing through the projection onto 2d.

Proof: Let $\vec{v}_i = [\vec{d}_i^T \; v_{z_i}]^T$ as in Prop. 4. Because rotation matrices preserve lengths, $||\vec{z}_i||^2 = \alpha^2||\vec{v}_i||^2 = \alpha^2(||\vec{d}_i||^2 + v_{z_i}^2)$. Dividing by $||\vec{z}_i||^2$ yields

$$1 = p_i^2 + \frac{\alpha^2 v_{z_i}^2}{||\vec{z}_i||^2} \tag{20}$$

Also,

$$\vec{v}_i = \left[\begin{array}{c} \vec{d}_i \\ \cdots \\ \pm\sqrt{\frac{||\vec{z}_i||^2}{\alpha^2} - ||\vec{d}_i||^2} \end{array}\right] \tag{21}$$

Performing the same calculation using an estimated scaling, α_e, rather than the true scaling, α gives an estimate of the vector $\vec{v}_i$

$$\vec{v}_{e_i} = \left[\begin{array}{c} \vec{d}_i \\ \cdots \\ \pm\sqrt{\frac{||\vec{z}_i||^2}{\alpha_e^2} - ||\vec{d}_i||^2} \end{array}\right] \tag{22}$$

Defining the multiplicative scaling error as $\beta = \alpha_e/\alpha$ and taking dot products between normalized versions of (21) and (22) gives the cosine of the angle between the true and estimated measurement. Simplification yields (18).

□

C. Estimation of Scaling Using the LUB

From (20), it is clear that $\vec{p}_i = \frac{\alpha||\vec{d}_i||}{||\vec{z}_i||} \leq 1$. Solving for α gives

$$\alpha \leq \frac{1}{||\vec{d}_i||/||\vec{z}_i||}$$

This equation must be satisfied for all of the data points, $i = 1 \ldots n$. Consequently, one estimate for the scaling is to use the least upper bound (LUB), i.e. let

$$\alpha_e = \min_{i=1 \ldots n} \frac{1}{||\vec{d}_i||/||\vec{z}_i||} = \frac{1}{\max_{i=1 \ldots n} ||\vec{d}_i||/||\vec{z}_i||} \tag{23}$$

The maximum percentage of norm passing through the 2d sensor is then from (19)

$$\bar{p} = \alpha \max_{i=1 \ldots n} \frac{||\vec{d}_i||}{||\vec{z}_i||} \tag{24}$$

From (23) and (24),

$$\beta = \frac{\alpha_e}{\alpha} = \frac{1}{\bar{p}} \tag{25}$$

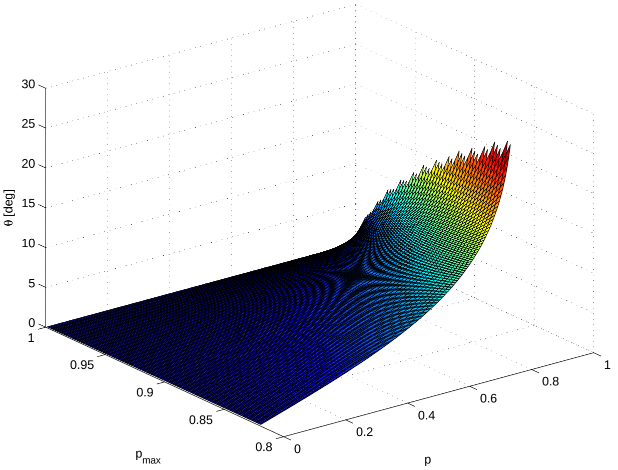

Fig. 1. The angular error due to errors in estimating the scaling.

Inserting (25) into (18) yields the angular estimation error corresponding to the least upper bound estimation of α:

$$\theta = \text{acos}\left[\frac{p_i^2}{\bar{p}} + \sqrt{(1-p_i^2)(1-\frac{p_i^2}{\bar{p}^2})}\right] \tag{26}$$

Figure 1 plots θ vs. p_{max} ($\bar{p}$) and p (p_i). Note that, if $\bar{p} = 1$, then $\theta = 0$ (no estimation error). In this case, at least one $\vec{v}_i$ lies in the x-y plane, so the full norm passes through the compression to 2d. Note also that θ decreases as $\bar{p}$ increases, thus designing the data so that the minimum $\bar{p}$ is large will reduce errors.

D. Worst Case Maximum p

Since $\vec{z}_i$, $i = 1 \ldots n$ are known apriori', the angles between them are also. Using this information, it is possible to find the worst case viewpoint, i.e. that which minimizes, over all possible viewpoints, $\bar{p}$. Let $\vec{a}$ be the worst case viewpoint. For a camera, $\vec{a}$ is physically the unit vector pointing along the camera's optical axis–commonly this is made the "z" vector in the camera coordinate system, which is equivalently the third row of R. Let $\vec{n}_i = \vec{z}_i/||\vec{z}_i||$, i.e. normalized data vectors. The following theorem finds the worst case viewpoint.

Theorem *6:* The worst case viewpoint, $\vec{a} \in \mathbb{R}^3$, i.e. the unit vector such that

$$\min_{\vec{a}^T\vec{a}=1} \bar{p} = \bar{p}_{min} \tag{27}$$

Can be found by solving the following convex problem subject to linear matrix inequality constraints:

$$\max_{t \in \mathbb{R}, \; \vec{a} \in \mathbb{R}^3} \; t$$

$$\left[\begin{array}{cc} \vec{a}^T\vec{n}_i & t \\ t & \vec{a}^T\vec{n}_i \end{array}\right] > 0, \; i = 1, \ldots, n \tag{28}$$

$$\left[\begin{array}{cc} I_3 & \vec{a} \\ \vec{a}^T & 1 \end{array}\right] > 0$$

Moreover, the minimum maximum percentage norm is

$$\bar{p}_{min} = 1 - t^2 \tag{29}$$

The worst case angular estimation errors are given by

$$\text{acos}\left[\frac{p_i^2}{\bar{p}_{min}} + \sqrt{(1-p_i^2)(1-\frac{p_i^2}{\bar{p}_{min}^2})}\right] \tag{30}$$

Proof: Direct solution of (27) is nonlinear and non-convex, therefore the optimization is slow and globality of the minimum is not guaranteed. However, a complementary problem can be equivalently solved in a convex setting. Since data along the $\vec{a}$ direction is lost in the projection to 2d, bad viewpoints have a large component of the data along $\vec{a}$, thus $\vec{a}^T n_i$ is large. However, if even one data vector has a large projection into the space perpendicular to $\vec{a}$, then that vector will have a large percentage of its norm sensed, so $\bar{p}$ will in turn be large. Consequently, finding the worst viewpoint can be formulated as:

$$\max_{\vec{a}^T\vec{a}=1} \min_{i=1,\ldots,n} |\vec{a}^T\vec{n}_i| \tag{31}$$

This can be equivalently stated as

$$\max_{\vec{a},t} t$$

subject to

$$(\vec{a}^T\vec{n}_i)^2 > t^2, \text{ i=1}, \ldots, n \tag{32}$$

$$\vec{a}^T\vec{a} = 1$$

The nonlinear, quadratic constraints can be replaced with equivalent *linear* matrix inequalities (LMI's) by using the Schur complement. If

$$X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

is a partitioned matrix, then $X > 0$ (X positive definite) is equivalent to $A > 0$ and $S = C - B^T A^{-1} B > 0$, where S is the Schur complement [2]. Thus $\vec{a}^T\vec{n}_i > 0$, $(\vec{a}^T\vec{n}_i)^2 - t^2 > 0$ is equivalent to $X > 0$, where X is the 2×2 matrix

$$\begin{bmatrix} \vec{a}^T\vec{n}_i & t \\ t & \vec{a}^T\vec{n}_i \end{bmatrix} \tag{33}$$

Moreover, the objective increases as the norm of $\vec{a}$ increases, thus the equality constraint $\vec{a}^T\vec{a} = 1$ can be replaced with the inequality constraint $\vec{a}^T\vec{a} \leq 1$. This nonlinear inequality can then be written as a linear matrix inequality by using the Schur complement:

$$\begin{bmatrix} I_3 & \vec{a} \\ \vec{a}^T & 1 \end{bmatrix} > 0$$

Thus the optimization problem (32) can be expressed as minimization of a linear objective subject to linear matrix inequalities as in (28). The variable t is the component of $\vec{n}$ along $\vec{a}$, where $\vec{n}$ represents the data vector with smallest projection along $\vec{a}$. A unit vector perpendicular to $\vec{a}$, $\vec{a}^\perp$, can then be found such that $\vec{n} = t\vec{a} + \bar{p}_{min}\vec{a}^\perp$. Calculating the norm squared of both sides then yields $1 = t^2 + \bar{p}_{min}^2$. Rearranging yields (29). Finally, (30) is obtained by substituting (29) into (26).

□

Note that the linear matrix inequality solution adds an extra condition, $\vec{a}^T\vec{n}_i > 0$. This condition cannot be satisfied unless all of the data vectors are in the same hemisphere. This condition poses no problem since the magnitude of the projection of a vector is the same as the magnitude of the projection of the negated vector. Thus $\vec{n}_i$ not in the upper hemisphere are negated, without changing the worst viewpoint.

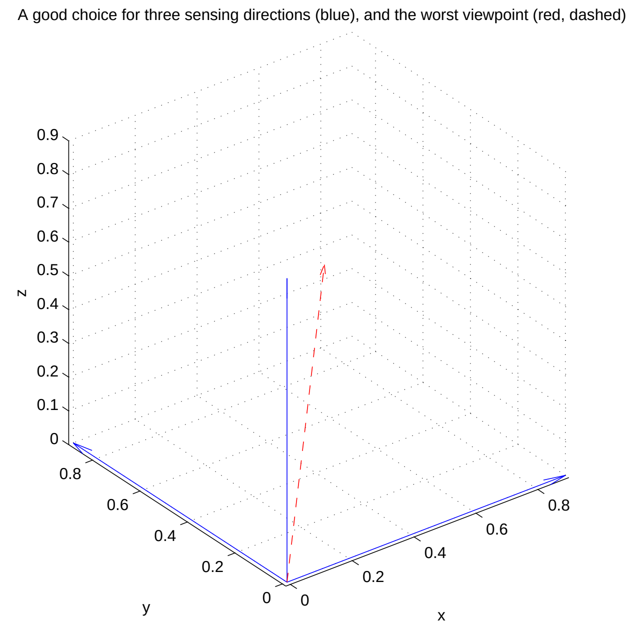
A good choice for three sensing directions (blue), and the worst viewpoint (red, dashed)

Fig. 2. Three orthogonal data vectors and their worst case viewpoint.

E. LMI Solution and Data Design

The global optimum of the convex optimization problem in Theorem 6 can be quickly found with a number of software packages. Figures 2 and 3 illustrate some solutions obtained using the "mincx" function of Matlab's Robust Control Toolbox. Figure 2 illustrates $Z = I$, i.e. three data vectors are arranged in a mutually orthogonal fashion. The worst case viewpoint minimizes the maximum projection orthogonal to itself, thus in this case $\vec{a} = [1\ 1\ 1]^T/\sqrt{3}$ and $\bar{p}_{min} = 0.82$. Substitution into (30) quantifies the angular errors. This application of Theorem 6 illustrates how it can be used to assess a set of data vectors, Z.

Theorem 6 also immediately suggests methods for designing the set of data vectors, Z, so they inherently provide accurate estimates. For instance, since $\vec{a}$ represents the worst case viewpoint, a data vector added in the space perpendicular to $\vec{a}$ converts this worst case viewpoint into an ideal viewpoint, thus eliminating this poor viewpoint. If the worst case viewpoint for the new, augmented set of data vectors is found, then the data vectors can again be augmented with a vector in the new $\vec{a}^\perp$, and so on. Thus a good set of n data vectors can be found inductively by repeating applying Theorem 6 and adding data points in $\vec{a}^\perp$.

Figure 3 illustrates this process when applied to the data vectors from Fig. 2. This process raises $\bar{p}_{min}$ from $\bar{p}_{min} = 0.82$ to $\bar{p}_{min} = 0.95$. This large value for the worst case $\bar{p}$ results in very small angular input errors (see Fig. 1 and Eqn. (26).

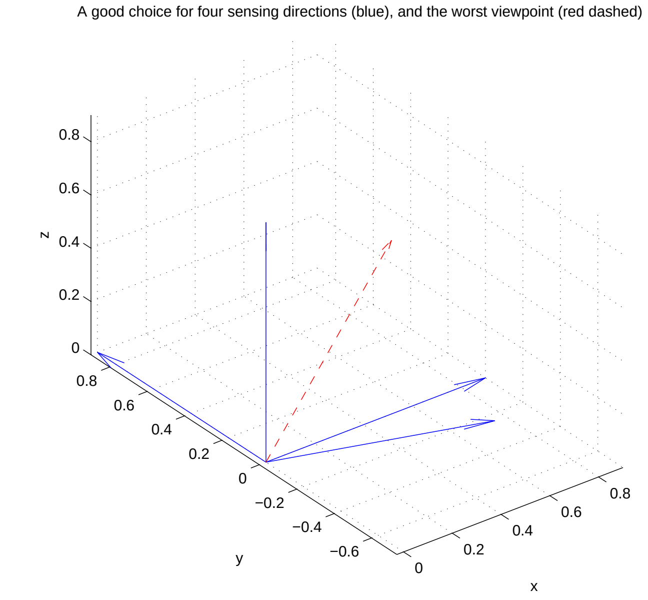

Fig. 3. Four data vectors and their worst case viewpoint.

F. Orientation Estimation: Scaling Unknown

Lemma 7 will now show how these results can be brought together to estimate orientation from 2-d measurements with unknown scaling.

Lemma *7:* Given $\vec{z}_i \in \mathbb{R}^3$, $\vec{d}_i \in \mathbb{R}^2$, and $w_i \in \mathbb{R}_+$, $i = 1 \ldots n$, $P = [I_2\ 0]$ the minimizer of

$$J = \sum_{i=1}^{n} w_i ||PR\vec{z}_i - \alpha \vec{d}_i||^2 \tag{34}$$

over $R \in \text{SO}(3)$, $\alpha \in \mathbb{R}_+$, is approximated by first estimating α by $\alpha_e = \frac{1}{\max_{i=1\ldots n} ||\vec{d}_i||/||\vec{z}_i||}$. R can then be estimated by replacing all measurement vectors $\vec{d}_i$ with scaled versions $\vec{d}_i \mapsto \alpha_e \vec{d}_i$, $i = 1 \ldots n$, and applying Prop. 4. The worst case errors can then be found from Theorem 6.

Proof: Lemma 7 extends Prop. 4 to handle differences in scaling between the data vectors, $\vec{z}_i$, and the measurements, $\vec{d}_i$. The LUB estimate of scaling given by (23) will contain error unless at least one measured vector truly does lie in the sensor (x-y) plane. Theorem 6 quantifies the worst case effects of this estimation error.

□

G. Pose Estimation from 2-d Measurements

Prop. 3 derives a decoupled method of calculating the pose from 3-d measurements. Unfortunately, the effects of scaling preclude a similar result from 2-d measurements. This section develops techniques of extending the 2-d orientation methods developed herein to estimate both orientation and position.

Proposition *8:* Given data points $\vec{z}_i \in \mathbb{R}^3$ and corresponding sensed points $\vec{q}_i \in \mathbb{R}^2$, along with weightings $w_i \in \mathbb{R}_+$, $i = 1 \ldots n$, $P = [I_2\ 0]$, the minimizer of

$$J = \sum_{i=1}^{n} w_i ||P[R\vec{z}_i + \vec{p}] - \alpha \vec{q}_i||^2 \tag{35}$$

over $g = (R, \vec{p}) \in \text{SE}(3)$ ($R \in \text{SO}(3)$, $\vec{p} \in \mathbb{R}^3$), and $\alpha \in \mathbb{R}_+$ can be estimated by forming the free vectors $\vec{z}_{ij} = \vec{z}_i - \vec{z}_j$, $\vec{d}_{ij} = \vec{q}_i - \vec{q}_j$, $i, j = 1 \ldots n$. R and α can then be estimated using Lemma 7 by replacing $\vec{z}_i$ by $\vec{z}_{ij}$ and $\vec{d}_i$ by $\vec{d}_{ij}$. The first two components of $\vec{p}$ are estimated by

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \vec{p}_a = P\vec{p} = \frac{\alpha \sum_{i=1}^{n} w_i \vec{q}_i - R_a \sum_{i=1}^{n} w_i \vec{z}_i}{\sum_{i=1}^{n} w_i} \tag{36}$$

where R_a denotes the 2×3 matrix consisting of the first two rows of R. When the measurements arise from an optical system with focal length f, and p_z denotes the third component of $\vec{p}$, and if the imaging is affine ($||\vec{z}_i|| \ll p_z$ for all points $i = 1 \ldots n$), then

$$p_z \approx \alpha f \tag{37}$$

Proof: Lemma 7 requires *free* direction vectors $\vec{z}_i$ and $\vec{d}_i$ as inputs. However, given position vectors $\vec{z}_i$, free direction vectors can be constructed by subtracting positions to find $\vec{z}_{ij} = \vec{z}_i - \vec{z}_j$. Their corresponding sensed free direction vectors are $\vec{d}_{ij}$. Weighting of these free vectors can be performed in accordance with their known error characteristics, or their geometric or arithmetic means ($w_{ij} = \sqrt{w_i w_j}$ or $w_{ij} = (w_i + w_j)/2$). The data and measurements are then ready for application of Lemma 7 to estimate R and α. With these estimates in hand, the first two translational elements can then be estimated by the unconstrained least squares problem:

$$LS = \min_{\vec{p}_a} \sum_{i=1}^{n} w_i ||R_A \vec{z}_i + p_2 - \alpha \vec{q}_i||^2$$

Setting $\frac{\partial LS}{\partial \vec{p}_a} = 0$ yields (36). To find the depth, p_z, consider a point at the origin of the data frame. Its representation in the sensor frame is $\vec{p}$. Orienting the optical system along the z axis, the perspective transformation caused by a focusing lens gives the image plane position, $\vec{q}$:

$$\vec{q} = \frac{f}{p_z} \vec{p}_a$$

But $\alpha \vec{q} = \vec{p}_a$, thus $p_z = \alpha f$. If $||\vec{z}_i|| \ll p_z$ for all points $i = 1 \ldots n$ (affine imaging), then the perspective transformation scales all image points by very close to the same factor, thus one α suffices for the entire set of data, and it is proportional to depth.

□

IV. Simulation Results

Figures 4–5 compare the performance of the new orientation estimates found using Theorem 4 to estimates obtained from the standard approach–use of Theorem 1, with the unknown "z" component of $\vec{v}_i$ assumed to be zero. Figure 4 plots the measured image plane direction vectors, $\vec{v}_i$, (solid). R and α are then estimated by the techniques of both Theorem 1 (with $\vec{v}_{z_i} = 0$) and Theorem 4. From the data vectors Z, the image plane direction vectors are then reconstructed: $\vec{v}_{e_i} = R\vec{z}_i/\alpha$. R and Z are intentionally chosen to produce large enough errors to be visible. Fig. 4 illustrates that the new method does produce better reconstructions of the image directions. Indeed, the angle required to rotate the true R to its estimate is 33^o for the standard approach vs. only 11^o using the new

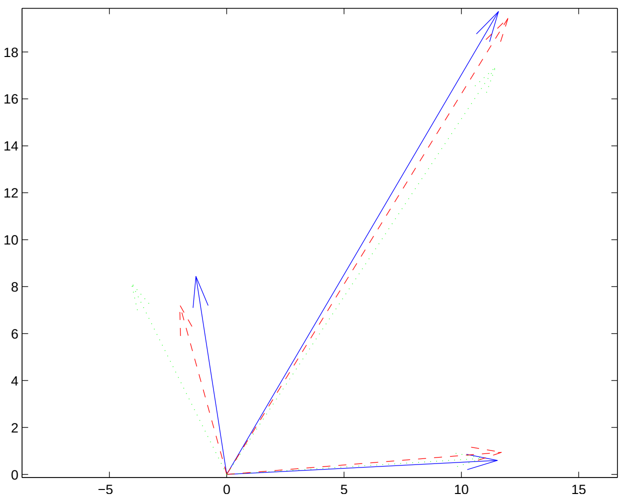

Fig. 4. Three image plane directions (solid), their reconstruction using the standard approach (dotted), and their reconstruction using the new method (dashed).

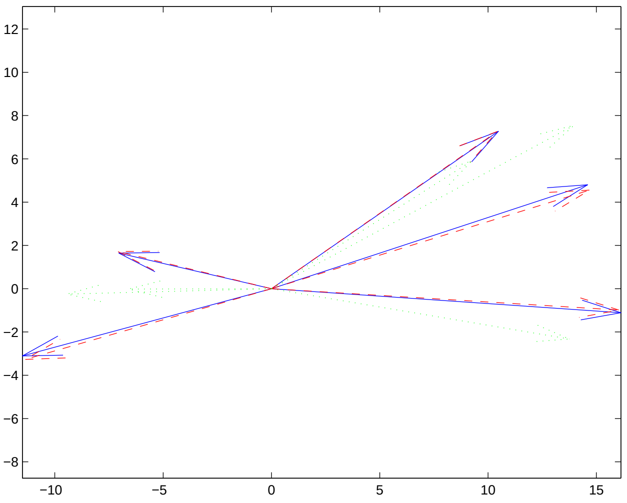

Fig. 5. Five image plane directions (solid), their reconstruction using the standard approach (dotted), and their reconstruction using the new method (dashed).

techniques developed in Theorem 4. Fig. 5 displays the same results when $n = 5$, rather than $n = 3$, measurements are taken. In this case, the angle required to rotate the true R to its estimate is 21^o for the standard approach vs. only 1.4^o using the new method. These results are typical comparisons.

Fig. 6 compares three pose estimation techniques. From four given image points (denoted by +), the pose $(R, \vec{p})$ and scaling α are estimated, then used to attempt a reconstruction of the given image points. This gives a visual indication of the pose estimation's accuracy, which will also be complemented with the actual estimation errors. The actual rotation matrix is chosen so it has the worst case viewpoint calculated via Theorem 6. The position vector is $\vec{p} = [0.369\ 0.179\ 10^5]^T$. Its "z" component is 100km because our current application involves ground based images of satellites.

For comparison purposes, the pose is first estimated using

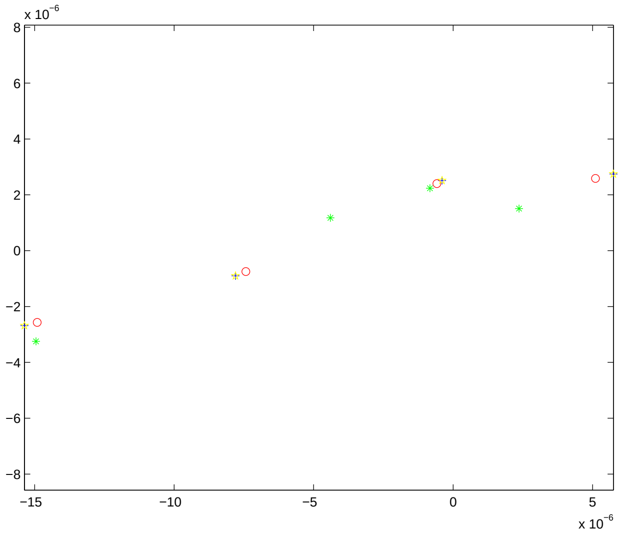

Fig. 6. Four image plane points (+), their reconstruction using the standard approach (*), their reconstruction using the new method (0), and their reconstruction using full 3-d sensed data (pentagram).

3-d measurements. This method requires another sensor such as a depth sensor or a stereo camera, and estimation of pose from such data sets is a mature science. One technique is to find free vectors by subtracting the points in the same manner used by Prop. 8. Theorem 1 can then use these 3-d free vectors to estimate R and α. Finally, the optimal $\vec{p}$ can be found using (13). Since this method has access to the full 3-d data, its reconstructed image points (pentagrams) closely match the given image points (+). A simple application of these same ideas using 2-d data is to assume that all measurement points $\vec{q}_i$ have "z" components equalling zero, then apply the above 3-d algorithm after augmenting each $\vec{q}_i$ with a zero "z" component. This is termed the standard approach, and reconstructions based on it are depicted by (*) in Fig. 6. The standard approach performs erratically, usually providing very poor pose estimates. Reconstructed image points found using the new technique derived in Lemma 8 are illustrated by (0). Like the full 3-d algorithms, they closely match the given image points, but unlike the full 3-d algorithm, the estimates can be obtained from a single monocular image.

The table below documents the angle required to rotate the estimated R to the true R, e_θ. Letting the translational error be $\vec{e}_p = \vec{p} - \vec{p}_{estimated}$, the percentage of normed translation error for each case is also available, $||\vec{e}_p||/||\vec{p}||$. While the algorithm that has access to the full 3-d measurements performs best, its accuracy is rivaled by the new algorithm, which uses only 2-d sensor readings. The standard approach has very erratic behavior. It this case, its estimates are poor, although other simulations exhibit far better and far worse accuracy.

	Full 3-d	Standard 2-d	New 2-d
$\vert\vert\vec{e}_p\vert\vert/\vert\vert\vec{p}\vert\vert$	8×10^{-6}	0.44	0.07
e_θ	0.27^0	30^0	1.4^0

The new method has been developed, in part, to aid in the automatic inspection of satellites. From a noisy, diffraction limited image of a known satellite in an unknown pose, the

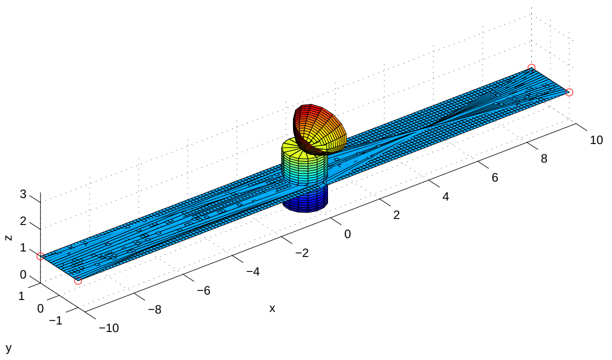

Fig. 7. CAD model of the imaged satellite, with O's denoting the point correspondences.

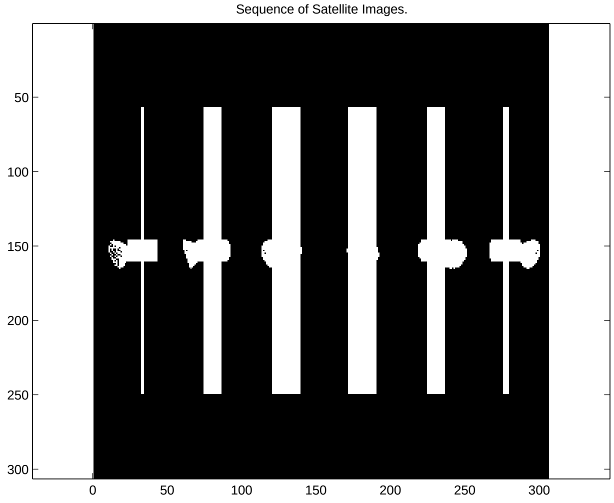

Fig. 8. As the satellite sweeps across the sky, several images are captured.

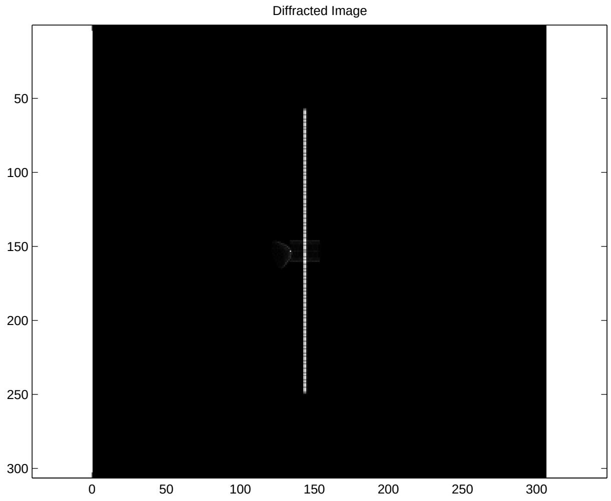

Fig. 9. The first raw, diffracted image.

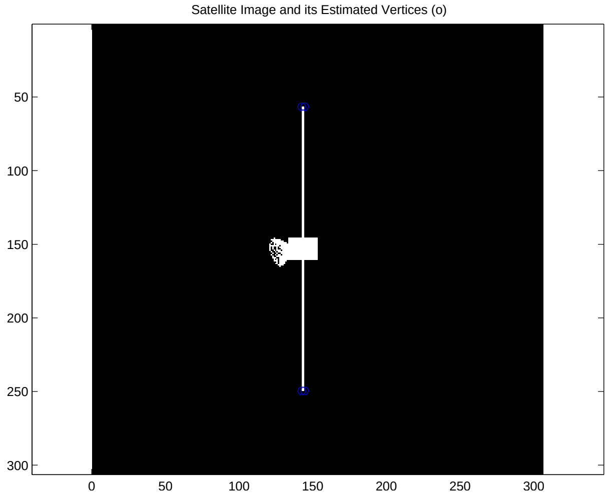

Fig. 10. The first image with the sensed points, $\vec{q}_i$, denoted by (0).

pose will be estimated. The CAD model can then be placed in the same pose and projected onto two dimensions to create a synthetic prediction of what the satellite's image should look like. Subtraction of the actual image from the synthetic image produces a signal useful for automatically checking any satellite anomalies.

Fig. 7 depicts a hypothetical satellite. The far corners of the solar arrays are chosen as the data points ($\vec{z}_i$, $i = 1, \ldots, 4$, depicted as 0's), as they produce an easily identifiable signature. As the satellite moves, several images are captured (Fig. 8). The pose will be estimated and the satellite inspected for images 1 and 4 (1 is the leftmost).

Because satellites orbit at significant distances from the Earth, the actual image is degraded appreciably by diffraction effects (see Fig. 9). Using a binary thresholding technique, it is enhanced, and the solar array corners in the image ($\vec{q}_i$) are found (Fig. 10).

Despite the effects of diffraction, thresholding, and pixel round-off, the pose estimate is quite accurate, producing an angular estimation error of only $e_\theta = 5.3^o$. A thresholded version of the predicted image is subtracted from the thresholded measured image (Fig. 10) to produce the inspection image (Fig. 11). Although it suffices for our present purposes, an improved match can certainly be obtained from this subtracted image by employing, for example, the methods in [5] to fine tune the estimate.

Automatically determining the point correspondences $\vec{q}_i$ can be difficult. These simulations first find a point inside the satellite by a very coarse template matching strategy. Then, a statistical snake [13] grown from that initial point robustly identifies the perimeter. Finally, vertices along that perimeter are extracted (see [11] for further details). Figures 12 and 13 illustrate the results on image 4. In this case, $e_\theta = 25^o$.

V. Conclusions

For the case of affine imaging, this paper replaces nonlinear numerical iterations or extremely high dimension linear solutions with solving the standard 3d-3d optimal orientation problem 2^n times, where n is the number of data points. The 2^n successive absolute orientation calculations are speeded through use of Gray code, and have the dual advantages of speed and predictable execution time. Angular errors caused

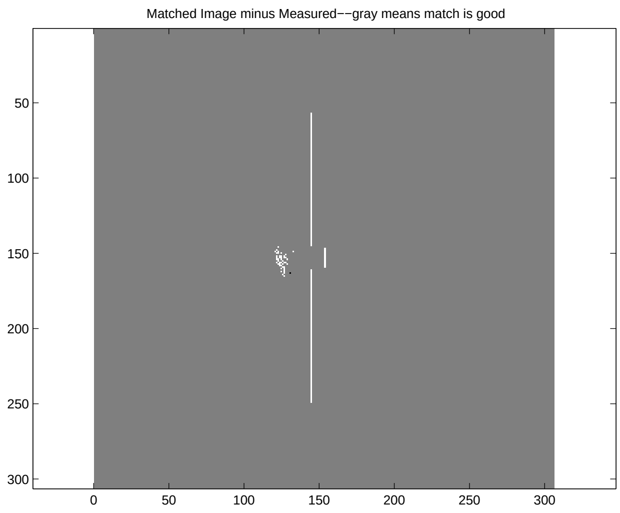

Fig. 11. Combining pose estimates with the CAD model, a predicted image is calculated and subtracted from the measured image.

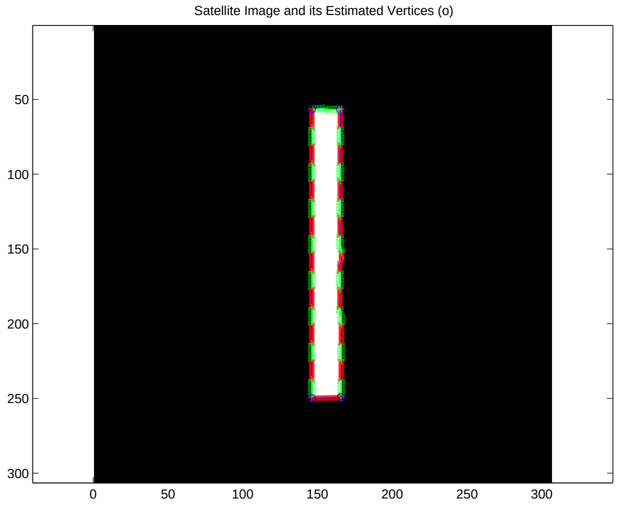

Fig. 12. The fourth image, with its points $\vec{q}_i$ denoted by (0). The outside perimeter is identified by statistical snakes.

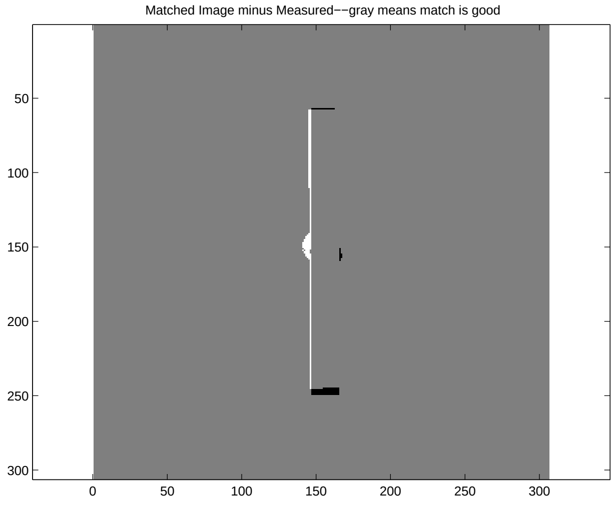

Fig. 13. Combining pose estimates with the CAD model, a predicted image is calculated and subtracted from the fourth measured image.

by scaling imperfections are quantified, and a least upper bound estimate of the scaling is proposed. It is shown that the worst case viewpoints depend only on the data points chosen, and a new convex linear matrix inequality optimization is derived for determining the worst viewpoint. This new analysis tool is useful for evaluating a particular set of data and suggests methods of designing the data for high performance. Simulation results for a satellite inspection problem demonstrate the viability of the technique and its superiority over standard methods.

REFERENCES

[1] Adnan Ansar and Kostas Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):578–589, May 2003.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] A.I. Comport, D. Kragic, E. Marchand, and F. Chaumette. Robust real-time visual tracking: Comparison, theoretical analysis and performance evaluation. In *IEEE International Conference on Robotics and Automation*, pages 2852–2857, Barcelona, Spain, April 2006.

[4] D. DeMenthon and L.S. Davis. Exact and approximate solutions of the perspective three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1100–1105, November 1992.

[5] T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):932–946, July 2002.

[6] P.D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:140–148, 2001.

[7] Zhijiang Guo, J.E. McInroy, and F. Jafari. Realization of micromanipulating gough-stewart platforms with desired dynamics. In *IEEE International Conference on Robotics and Automation*, pages x–x, Orlando, FL, 2006.

[8] S. Gwak, J. Kim, and F.C. Park. Numerical optimization on the euclidean group with applications to camera calibration. *IEEE Transactions on Robotics and Automation*, 19(1):65–74, February 2003.

[9] Yonghuai Liu and Ying Wang. Evaluating 3d-2d correspondences for accurate camera pose estimation from a single image. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 703–708, Washington, DC, USA, October 2003.

[10] C. Lu, G. Hager, and E. Mjolsness. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):610–622, June 2000.

[11] J.E. McInroy, L.M. Robertson, and R.S. Erwin. Distant visual detection of satellite anomalies. In *Preparation for the IEEE International Conference on Robotics and Automation*, April 2007.

[12] F.C. Park, J. Kim, and C. Kee. Geometric descent algorithms for attitude determination using the global positioning system. *Journal of Guidance, Control, and Dynamics*, 23(1):850–863, January-February 2000.

[13] D. Perrin and C. Smith. Rethinking classical internal forces for active contour models. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

[14] L. Quan and Z. Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:774–780, 1999.

[15] H. Rehbinder and B.K. Ghosh. Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48(2):186–199, February 2003.

[16] R. Vidal. Multi-subspace methods for motion segmentation from affine, perspective, and central panoramic cameras. In *IEEE International Conference on Robotics and Automation*, pages 1228–1233, Barcelona, Spain, April 2006.

[17] J. R. Wertz. *Spacecraft Attitude Determination and Control*. D. Reidel Publishing Company, Boston, MA, 1978.