The Pyramid Star Identification Technique

Daniele Mortari*, Malak A. Samaan†, Christian Bruccoleri‡, and John L. Junkins§
Texas A&M University, College Station, Texas

Abstract

A new highly robust algorithm, called Pyramid, is presented to identify the stars observed by star trackers in the general lost-in-space case, where no a priori estimate of pointing is available. At the heart of the method is the k-vector approach for accessing the star catalog, which provides a search-less means to obtain all cataloged stars from the whole sky that could possibly correspond to a particular measured pair, given the measured interstar angle and the measurement precision. The Pyramid logic is built on the identification of a four star polygon structure - the Pyramid - which is associated with an almost certain star identification. Consequently, the Pyramid algorithm is capable of identifying and discarding even a high number of spikes (false stars). The method, which has already been tested in space, is demonstrated to be highly efficient, extremely robust and fast. All of these features are supported by simulations and by a few ground test experimental results.

INTRODUCTION

The Pyramid Lost-In-Space Algorithm has been used repeatedly in successful on-orbit applications in the High Energy Transient Explorer (HETE) satellite, which has been operating in the flight system since July 2002 [1]. Draper Laboratory is currently developing the Inertial Stellar Compass, which uses the Pyramid algorithm as the main program for star identification in the Inertial Stellar Compass (ISC), a low-power stellar inertial attitude determination system [2]. Also of significance, the method, which has been extended to

*Daniele Mortari, Associate Professor, Department of Aerospace Engineering, 701 H.R. Bright Building, Room 741A, Texas A&M University, 3141 TAMU, College Station, Texas 77843-3141 Tel. (979) 845-0734, FAX (979) 845-6051. mortari@aero.tamu.edu

†Malak A. Samaan, Post-Doc Research Associate, Spacecraft Technology Center, Room 127H, Texas A&M University, College Station, TX 77843-3141, Tel. (979) 845-8768, samaan@tamu.edu

‡Christian Bruccoleri, PhD candidate, Department of Aerospace Engineering, 701 H.R. Bright Building, Room 620C, Texas A&M University, College Station, TX 77843-3141, Tel. (979) 485-0550, FAX (979) 845-6051, bruccoleri@tamu.edu

§John L. Junkins, George J. Eppright Chair Professor, Director of the Center for Mechanics and Control, 722 Bright Bldg., Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141, Tel: (979) 845-3912, Fax: (979) 845-6051, junkins@tamu.edu

multiple-fields-of-view star trackers, has been adopted for the StarNav dual-field-of-view autonomous star tracker, which is the science attitude determination system for the NASA EO-3 Geosynchronous Imaging Fourier Transform Spectrometer (GIFTS) New Millennium mission.

The star polygon geometric structure is defined by the set of $M = \dfrac{n!}{(n-2)!\,2!}$ interstar angles associated with a spherical polygon set of n stars, such as pairs ($n = 2$), triangles ($n = 3$), as well as pyramids ($n = 4$). The spherical polygon is closely related to the usual polygon, where the straight line sides are replaced by great circle arcs (angles) on the surface of a unit sphere connecting the neighboring pairs of stars in a set of p stars. More specifically, the star pattern geometric structure for the purpose of star identification is defined by the set of M interstar angles $\{\vartheta_{ij} = \vartheta_{ji} = \cos^{-1}(\mathbf{b}_i^{\mathrm{T}}\mathbf{b}_j)\}$ measured between each distinct pair of the p line-of-sight vectors $\{(\mathbf{b}_i, \mathbf{b}_j) : (i,j) \in \{1, 2, \ldots p\}\}$ that point from the sensor toward the vertices of the star spherical polygon on the celestial sphere. Note we adopt the convention that the measured line-of-sight unit vectors with components in the sensor *body* axes are denoted $\mathbf{b}_i$, whereas the corresponding line of sight vectors based on cataloged information with components in the inertial $reference$ frame of the star catalog are denoted $\mathbf{r}_I$. The whole objective of star identification can be boiled down to finding the correspondence between indices (i) of measured stars and the indices (I) of cataloged stars.

Matching the set of M measured interstar angles $\cos^{-1}(\mathbf{b}_i^{\mathrm{T}}\mathbf{b}_j)$ with a cataloged set of interstar angles $\cos^{-1}(\mathbf{r}_I^{\mathrm{T}}\mathbf{r}_J)$, to within measurement precision provides the basis for a hypothesis that the stars at the vertices of the measured polygon of stars are indeed the cataloged stars at the corresponding vertices of the matching polygon from the star catalog. Such a match clearly does not dictate a certain star identification, however, and acceptance of this hypothesis should be informed by the expected frequency of such a match occurring at random between the measured polygon and an invalid set of p cataloged stars. For example, if we know the theoretical expected frequency of mismatching an invalid polygon with a given set of measured interstar angles, to within known measurement error statistics, is negligible - e.g., one mismatch every 10^{20} tested polygons - we can be justifiably optimistic that the star identification hypothesis is valid, especially if some of the identified stars are reidentified consistently on overlapping adjacent images. On the other hand, if the random invalid match frequency is greater than some tolerance to be decided experimentally, we should have cause for concern and should likely reject the hypothesis and/or match more stars until the prescribed tolerance in the random match frequency is passed.

To implement such a statistical decision process based upon theoretical frequencies, we need to know the frequency formulas. Reference [3] first introduced the idea of an analytical approach to estimate the frequency of false matches associated with a given star scenario, which was based on the assumption of a uniform star distribution. This analytical approach, with frequency formulas restricted to up to a pyramid star polygon, is given in the next section.

The Pyramid algorithm proposed in this paper can be considered a first step toward a star-ID algorithm that is fully based on invalid match frequency formulation. These explicit

analytical formulas provide a basis for statistical inference logical tests of a star pattern identification hypothesis that has not been available previously, although practical algorithms have been developed empirically based on Monte Carlo testing and trial-and-error tuning of available star identification algorithms. A fundamental difficulty associated with such Monte Carlo testing, is that random sampling statistical inference computational approaches become infeasible if one is pursuing frequencies on the order of 10^{-7} or smaller, because of the necessity of having a very large number (certainly $> 10^8$) of samples to establish statistically significant frequency estimates for such infrequently occurring events. Indeed, it would be desirable, if computationally feasible, to reduce the expected frequency of a random star identification to far less than once over the lifetime of a given mission - i.e., 1/(# of star identifications over the life of a multiyear mission), which for the anticipated high-frame-rate active pixel cameras corresponds to a frequency of less than $< 10^{-10}$. Of course, we must acknowledge that there is a large distinction between the lost-in-space case and the case of recursive star identification, where we can use new stars in polygons with previously identified stars whose catalog identity is near certain. Therefore, the no prior information perspective implicit in the above discussion will typically be highly conservative, except for the first lost-in-space star identification.

In any event, we anticipate that over the course of the next decade, there will be at least occasional need for star identification algorithms with an overall expected frequency of mismatches approaching 10^{-10}, and for longer missions with high star camera frame rates, we can conceive of a need for exceptionally small expected star mismatch frequencies (perhaps even $< 10^{-20}$). Obviously, validation of such frequency estimates is compounded because every sensor design change and every resetting of any variable system parameter may necessitate repeating the mismatch frequency analysis. It is evident that Monte Carlo processes will be impractical in this situation. As with the perpetual thirst for faster computers, we can never develop a star identification algorithm that fails too rarely! Moreover, even without pursuing such small frequencies of spurious star identifications, it is very obvious that having the capability to quantify and minimize the frequency of failure is fundamental to analyzing/optimizing overall mission reliability. Therefore, we expect that the formulas already developed [3] and future refinements [4] to find a very practical home in sensor design and mission analysis as a result of eliminating the reliance on slowly converging statistical simulations, such as Monte Carlo processes.

A high percentage of spurious star images (spikes) introduces a crisis in almost all existing algorithms for star pattern recognition for stars imaged by charge coupled devices (CCD) star trackers. Failures and anomalies associated with such spurious images have been experienced several times in space missions using star trackers to estimate the spacecraft attitude. For example, the Space Transportation System (STS) 101 Space Integrated Global Positioning System/Inertial Navigation System Orbital Attitude Readiness (SOAR) star tracker experiment encountered spurious sun reflections off debris released by an adjacent experiment, resulting in a large number of spikes that, in turn, caused the star pattern recognition algorithms used for SOAR to fail.

The Pyramid algorithm, better than any known approach to solve the star identification problem, offers the simultaneous advantages of being extremely efficient and robust to ran-

dom spurious images and making it possible to compute an estimate that informs the decision to accept or reject a star pattern match hypothesis. If the estimated frequency of a random pattern match is not sufficiently small, remedial hypothesis/test logic can be invoked to add more stars to the pattern or reject the entire image. Actually, the method presented herein can tolerate even more spurious images as long as there are at least four catalog stars (for the case of modern star trackers with equivalent angle star centroiding errors of a few arc-seconds); however, the computation time is obviously a function of the number of spurious images.

FREQUENCY OF STAR PATTERN MISMATCHING

This section provides the mathematical tools that establish, in a closed form, the reliability of the star identification process associated with the three most fundamental star structures used in the star-ID algorithms. These star structures consider the interstar angles associated with polygon sets of $p = 2, 3, 4$ stars such as a pair, a triangle, or a Pyramid of four stars.

Let us consider the whole sky with a uniform star distribution. This implies that the star density ρ (which depends on the given magnitude threshold m), is simply given as

$$\rho(m) = \frac{N(m)}{4\pi} \tag{1}$$

where $N(m)$ is the overall number of stars with magnitude less than m.

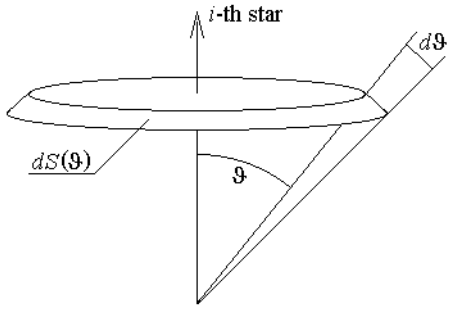

Figure 1: Infinitesimal spherical area

Let us consider the spherical surface defined by a cone of aperture ϑ, that is, the area

$$S(\vartheta) = 2\pi\,(1 - \cos\vartheta) \tag{2}$$

and let us consider that the axis of this cone is aligned with the i-th star. Referring to Figure 1, in the infinitesimal spherical area $dS(\vartheta)$, which can be evaluated as the difference between two cones of apertures $(\vartheta + d\vartheta)$ and ϑ, and which has the area

$$dS(\vartheta) = S(\vartheta + d\vartheta) - S(\vartheta) = 2\pi\,\sin\vartheta\,d\vartheta \tag{3}$$

the expected number of stars falling in $dS(\vartheta)$ will be

$$dn(\vartheta) = \rho^* \, dS(\vartheta) = \frac{(N-1)}{2} \, \sin\vartheta \, d\vartheta \tag{4}$$

where $\rho^* = (N-1)/(4\pi)$ indicates a uniform star density that differs slightly from the expression of the uniform star density ρ given in equation (1). This difference is due to the fact that, by aligning the axis of the cone with the i-th star, we force that star to be deleted from the overall number of stars available in the counting of the star density. Now, the product

$$df_{ij}(\vartheta) = \frac{1}{2} \, N \, dn(\vartheta) = \frac{N(N-1)}{4} \, \sin\vartheta \, d\vartheta \tag{5}$$

provides the number of star pairs i-j separated by an angle ranging from ϑ to $(\vartheta + d\vartheta)$. The division by 2 is due to the fact that the product $N \, dn(\vartheta)$ double counts the number of star pairs because it considers the star pair i-j and the same star pair j-i. By integrating equation (5) over the whole sky, it is possible to find the number of possible combinations N_{comb} of N objects (stars) taken two by two:

$$N_{comb} = \int_0^{\pi} df_{ij}(\vartheta) = \frac{N(N-1)}{2} \tag{6}$$

This equation can be obtained independently. Note that it is the number of combinations of N objects taken two at a time, and thus we have an independent check on equation (5). Hence, the expectation of the overall number of admissible star pairs f_{ij} displaced by an angle that varies from $(\vartheta_{ij} - k\sigma)$ to $(\vartheta_{ij} + k\sigma)$ is found by integrating equation (5) over this small region as

$$\begin{aligned} f_{ij} &= \frac{N(N-1)}{4} \int_{\vartheta_{ij}-k\sigma}^{\vartheta_{ij}+k\sigma} \sin\vartheta \, d\vartheta = \\ &= \frac{N(N-1)}{2} \, \sin(k\sigma) \, \sin\vartheta_{ij} \end{aligned} \tag{7}$$

This equation represents the expected frequency that false matches between measured "objects", to within measurement precision, are matched by random pattern combinations in the catalog, assuming a uniform star density.

We note that the actual star distribution is not uniform, however, simulations indicate that at most factors of two frequency variations occur, and thus we must take this into account conservatively in interpreting the frequency results obtained (i.e., it would be unwise to believe the frequencies exactly, but we can usually tolerate the factor of two difference between small numbers such as 1×10^{-7} or 2×10^{-7}!).

Figure 2 shows, for the StarNav I experiment (8 deg^2 field of view [FOV], magnitude threshold = 5.5, 512 × 512 pixel CCD, focal length = 50 mm, and 3σ = 10 arcsec), the residuals between the values for f_{ij} provided by equation (7) and random simulated data. After some experimentation, we found that the adopted value for k should be about 6.4, a value somewhat greater than the 3σ value of $k = 3\sqrt{2}$ (derived for an interstar angle associated with two stars whose direction precision is normally distributed). This has been found to

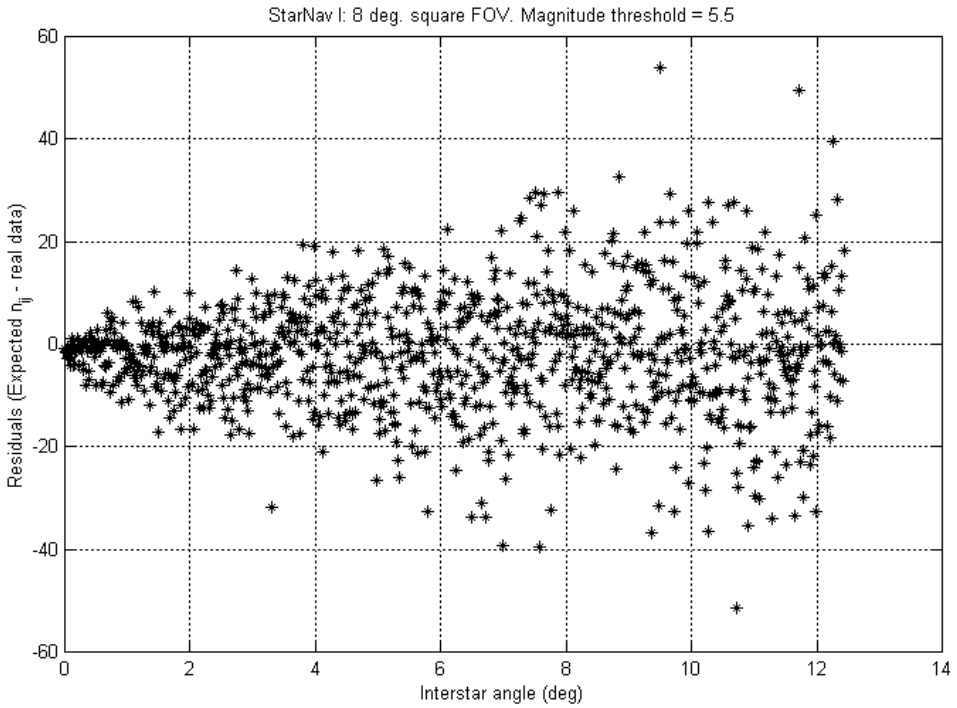

Figure 2: Residuals between equation (7) and random simulated data

essentially guarantee that the actual star pair measured is contained in the k-vector subset with f_{ij} elements. Adopting k of about 6.4 ensures that we obtain essentially all the possible measured stars as candidates, even with the actual nonuniform star density. For typical parameter settings (StarNav I), we find $f_{ij} = 200$ to be typical, so $\sigma_{f_{ij}} \cong 0.1 f_{ij}$, and the approximation of uniform density leads to moderate errors. However, this approximation is indeed adequate for order-of-magnitude analysis.

Star Pattern with two Legs

Consider the case of a three star pattern ijk. We seek to match the measured interstar angles $(\vartheta_{ij} \pm k\sigma)$ and $(\vartheta_{ik} \pm k\sigma)$. To do this, let us consider the i-th star. Now, using equation (4), it is easy to evaluate the number $\bar{f}_{ij}$ of stars j displaced from i by an angle that varies from $(\vartheta_{ij} - k\sigma)$ to $(\vartheta_{ij} + k\sigma)$. This number is

$$\bar{f}_{ij} = \int_{\vartheta_{ij}-k\sigma}^{\vartheta_{ij}+k\sigma} dn(\vartheta) = (N-1)\,\sin(k\sigma)\,\sin\vartheta_{ij} \tag{8}$$

Analogously, the number $\bar{f}_{ik}$ of stars k displaced from i by an angle that varies from $(\vartheta_{ik} - k\sigma)$ to $(\vartheta_{ik} + k\sigma)$ is

$$\bar{f}_{ik} = \int_{\vartheta_{ik}-k\sigma}^{\vartheta_{ik}+k\sigma} dn(\vartheta) = (N-1)\,\sin(k\sigma)\,\sin\vartheta_{ik} \tag{9}$$

Hence, the frequency that a star matches with both legs (star pairs ij and ik) is

$$f_{i-(j,k)} = N\ \bar{f}_{ij}\ \bar{f}_{ik} = N\left[(N-1)\ \sin(k\sigma)\right]^2\ \sin\vartheta_{ij}\ \sin\vartheta_{ik} \tag{10}$$

Star Pattern with m Legs

Equation (10) can be easily generalized to quantify the frequency that a star matches with m legs (that is, with m other stars identified by), obtaining

$$f_{i-(j_1,\ldots,j_m)} = N\ \prod_{j=j_1}^{j_m} \bar{f}_{ij} = N\left[(N-1)\ \sin(k\sigma)\right]^m\ \prod_{j=j_1}^{j_m} \sin\vartheta_{ij} \tag{11}$$

which completes the searched solution.

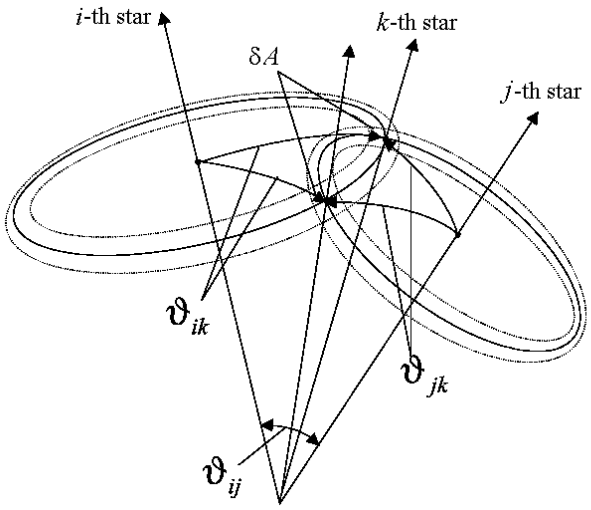

Figure 3: Differential Area Associated with Measurement Error for 3 Stars

Triangle Star Pattern

Let us now consider the frequency of random occurrence associated with matching all three angles. With reference to Figure 3, let us consider one of the f_{ij} star pairs, where f_{ij} has the expression provided by equation (7), and let us consider the intersection area δA of the two spherical surfaces associated with the angles ϑ_{ik} (centered at the i-star) and ϑ_{jk} (centered at the j-star). Note that for the given measured angles, and associated uncertainties, the k-th star must lie in one of the two small areas δA. The area δA can be approximated by considering the spherical square as planar (see Figure 4).

In this case, the area $\delta A = \ell\,(2k\sigma)$, since $\ell\,\sin\vartheta_k = 2k\sigma$, can be approximated as

$$\delta A = \frac{(2k\sigma)^2}{\sin\vartheta_k} \tag{12}$$

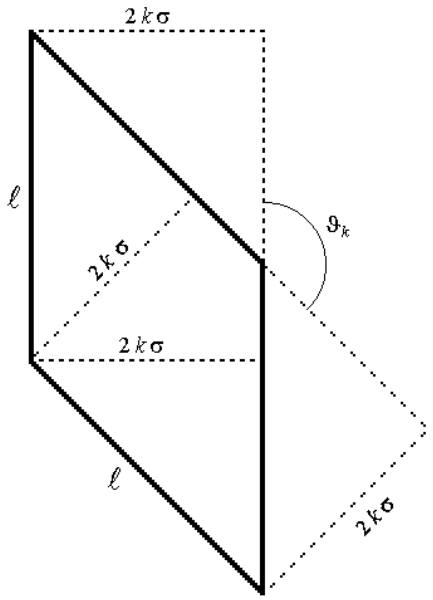

Figure 4: Intersection area

This implies that the expected frequency of random occurrence (that a given star triangle is matched within measurement precision) is

$$f_{ij-k} = f_{ij}\,(\rho^{**}\,2\delta A) \cong \frac{N(N-1)(N-2)}{\pi}\,(k\sigma)^3\,\frac{\sin\vartheta_{ij}}{\sin\vartheta_k} \tag{13}$$

where $\rho^{**} = (N-2)/(4\pi)$ is a modified star density that does not take into account the stars i and j that obviously cannot fall into the two small δA cone intersection areas. Note that equation (13) is symmetric (as expected) with respect to any considered side of the star triangle. In fact, the *sine* law for the spherical triangles establishes that

$$\frac{\sin\vartheta_{ij}}{\sin\vartheta_k} = \frac{\sin\vartheta_{ik}}{\sin\vartheta_j} = \frac{\sin\vartheta_{jk}}{\sin\vartheta_i} \tag{14}$$

Pyramid Star Pattern

The expected frequency associated with a four star pattern can easily be derived from that of a triangle star pattern. In fact, under the assumption of a uniform distributed catalog, the probability to find the fourth star in the assigned star uncertainty area (represented by a cone of aperture $k\sigma$), is simply the ratio of that area with respect to the overall 4π (entire celestial sphere), multiplied by the overall number of available stars in the catalog $(N-3)$. Therefore, for a pyramid of stars, the frequency is

$$f_{ijkr} = (N-3)\frac{1-\cos(k\sigma)}{2}\,f_{ijk} \tag{15}$$

A SMART TECHNIQUE TO SCAN TRIANGLES

The Pyramid algorithm is built starting with a basis star triangle. It is important to devise a suitable technique to scan subsequent triangles because we would like to avoid persisting in using a star that may be a spike and not a real catalog star. The right choice of the triangle sequence implies the need to maximize the changes on the index stars resulting from one selection with respect to the next. This maximization defines the optimal sequence. The original Pyramid version [3] proposed a heuristic approach indicating that although the sequence obtained by a random shuffling of all the triad combinations can never be guaranteed to be optimal, it will statistically avoid retaining spurious stars and wasting the time to perform trials to match angles to other stars. To this end, the original Pyramid version reads out a file containing the indices of the shuffled sequence triads for triangle selection. This method, however, which has been found to be much more suitable than the crude approach of three inner loops, has the disadvantage of requiring additional memory (especially when a high value of the observed stars n is adopted), since all the triangle index combinations must be memorized.

To avoid this problem, the new Pyramid version proposed herein contains a smart technique for producing the indices of subsequent star triangles that is based on the simple three inner loops concept. This technique, whose results should be compared with respect to the mathematically rigorous optimal solution to this problem (still unknown), is described below using a pseudocode language, which can easily be translated into any another existing programming language.

```
LOOP dj from 1 to (n-2),
   LOOP dk from 1 to (n-dj-1),
      LOOP i from 1 to (n-dj-dk),
         j = i+dj,
         k = j+dk,
         next combination is "[i j k]",
      END LOOP i,
   END LOOP dk,
END LOOP dj,
```

For instance, for $n = 5$ observed stars, the smart sequence of triad indices is 1-2-3, 2-3-4, 3-4-5, 1-2-4, 2-3-5, 1-2-5, 1-3-4, 2-4-5, 1-3-5, 1-4-5. The extension of this technique (devised specifically for the Pyramid algorithm) to the general case of n objects taken k by k would be of great interest.

THE PYRAMID ALGORITHM

In lieu of immediately presenting all the details of the proposed algorithm, we first summarize the major logical steps and a few new features associated with this algorithm. Subsequently, we go into selective detail.

The Pyramid algorithm contains several important new features. The first is access to the star catalog using the k-vector approach, [5]-[7] instead of the much slower binary search technique (see the appendix). The k-vector database is built a priori for some given working magnitude threshold and for the star tracker maximum angular aperture. Essentially, the k-vector table is a structural database of all cataloged star pairs that could possibly fit in the camera FOV over the whole sky. The star pairs are ordered with increasing interstar angle. The data stored are the k index, the cosine of the interstar angle, and the master catalog indices $I[k]$ and $J[k]$ of the kth star pair. The k-vector access logic is invoked in real time for a minimal set of star pairs in elementary measured star polygons (three for a triangle, six for a four-star pyramid, etc.); the fact that the vertices between adjacent measured star pairs share a common cataloged star is the key observation leading to logic for efficiently identifying the stars by simply comparing the k-vector accessed catalog indices from the several sets of candidate star pairs (which must contain the common measured pivot star, if it is in the catalog).

The second new feature is avoidance of identical (redundant) information requests. The information provided by the k-vector for an observed star pair (say, $[\mathbf{b}_i, \mathbf{b}_j]$) is stored so that any further request for such information does not require accessing the catalog again. The stored information consists primarily of the number of admissible stars pairs, along with the identifiers of the involved catalog stars - information contained in the two integer vectors of indices I and J.

Third, a "smart" technique to scan triangles. Existing algorithms that are based on star triangles and are designed to identify and discard spikes should be considered the nonnegligible possibility: if all the possible observed star triads are scanned using three inner loops and the star associated with the most external loop is a spike, most of the time spent has been wasted. To avoid such waste, the combination sequence for the considered triads is produced by a three inner loop-based technique that avoids persisting with a given index. The resulting triangle sequence is an attempt to maximize the changes in the three indices identifying the triads (as opposed to the traditional schemes, which in essence pivot exhaustively about a possibly invalid star). This is accomplished in the present version of Pyramid by the described smart three inner loop technique, which avoids both the memory required for stored optimal sequences and persistence in using a given star index (no pivot star is retained for more than two successive instances of matching logic).

Fourth, a robust four-star-based Pyramid is used instead of the more classic triangle to increase the probability of a correct star identification process. It is important that all interstar angles be matched to within measurement precision (six for a four-star-polygon, see Figure 5) to obtain the maximum confidence in the match. This also provides an easy way to identify spikes. The only limitation is that at least four catalog (good) stars are needed to build the Pyramid. For purposes of this paper, a good star is one that is included in the star catalog used by Pyramid, so it can be a star derived either by using a bigger catalog or by merging a double star when the centroiding resolution is not enough to resolve the two objects separately. Anything else, such as planets, artificial satellites, debris, and, more commonly, stars not included in the catalog, is generally considered noise. The star tracker hardware must therefore be designed to image at least four stars within the chosen integration time.

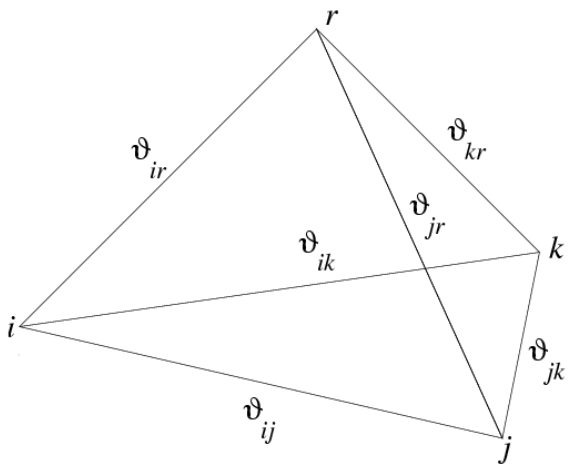

Figure 5: Basic Star Triangle and Pyramid

This is not considered an issue since the CCDs available for astronomy purposes can easily achieve the required performances with an FOV of 10 to 20 deg. At the precision limit of a few arcseconds achieved by state-of-the-art star trackers, a star identification process that matches the six measured and cataloged interstar angles for a four star pyramid is essentially a certain match.

Finally, we use the analytical means [1, 4] to compute the expected frequency of random occurrence that a cataloged polygon of stars could match, to within camera precision, the given measured polygon. This analytical means of computing the expected frequency is novel and is important to eliminate the need for expensive and slowly converging Monte Carlo estimates of star identification reliability.

The proposed Pyramid algorithm uses no information on star magnitude. Figure 5 shows the basic star structure used within the algorithm, which consists of a basic star triangle, identified by the indices $i,\ j,\ k$, together with a "confirming fourth star," identified by the index r.

The method, depicted in Figure 6, essentially accomplishes the task by the following steps (where n is the number of observed stars):

(1) if $n = 3$, then the four star pyramid $[i,\ j,\ k,\ r]$ cannot be built. Therefore, the Pyramid logic simply seeks to establish whether the triangle is unique. We note that the three-star case is dangerous, as it is associated with a higher probability of misidentification. The three-star case is introduced because sparse region of sky with low star density exists, and also because the three-star case represents the worst scenario for which a star identification process - based entirely on interstar angles can still be accomplished. However, when we are engaged in recursively applying the star identification algorithm, we will have identified some of the stars in previous images containing more than three stars, and much higher confidence than is provided by three stars alone is possible when the three-star identifications are consistent with previous images/identifications.

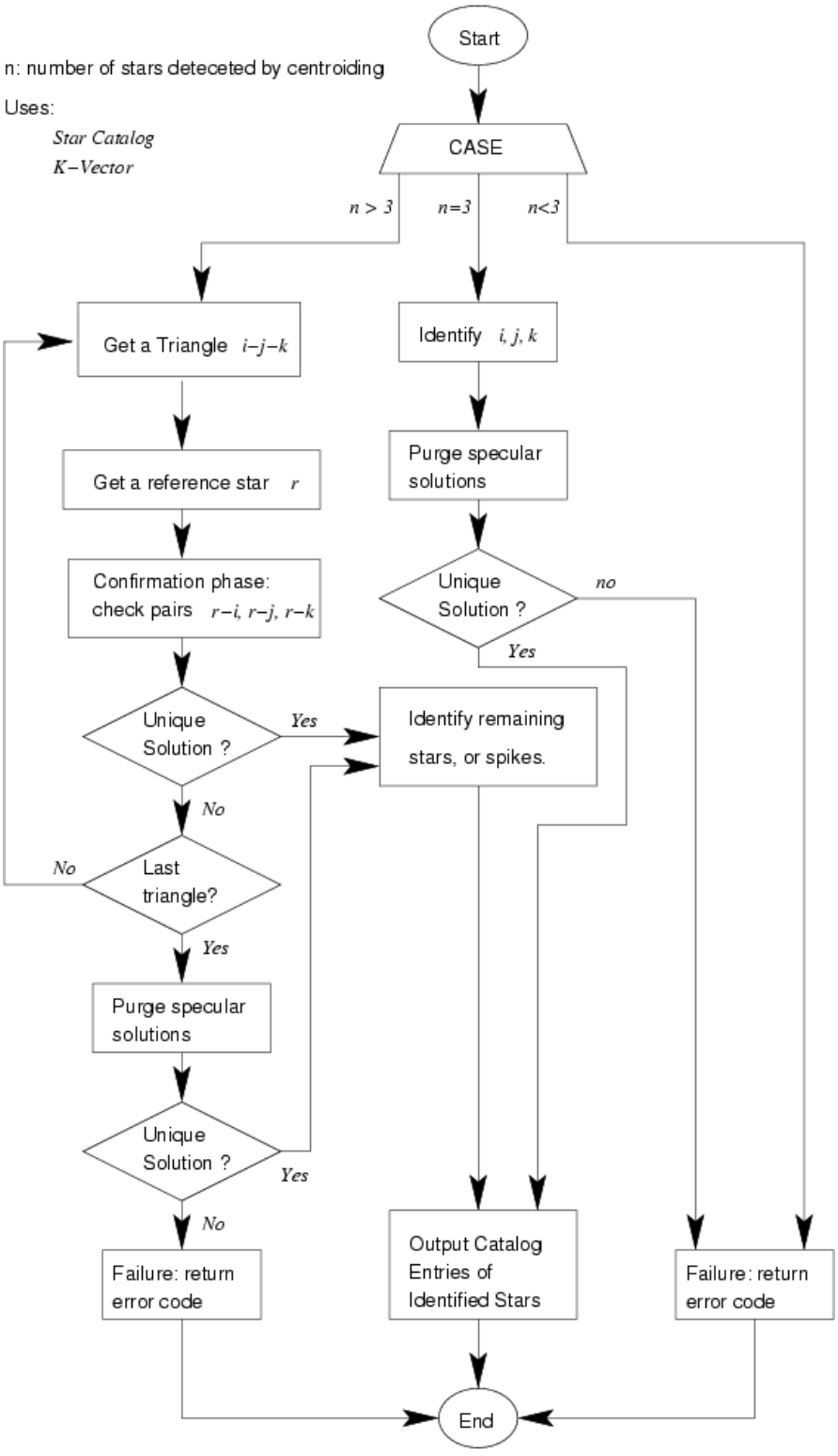

Figure 6: Pyramid Flowchart

For this mode (recursive star identification), we believe three stars to be sufficient if consistency tests are used to augment the lost-in-space algorithm. The problem of specular triangles is solved here by the following discriminating criterion. Let i-j-k be the indices of an observed star triangle and I-J-K be the indices of the corresponding catalog star triangle. Provided that the i-j-k triangle is not degenerating (coplanar or double stars), the catalog triangle is not specular if

$$\text{sign}[\mathbf{b}_i^{\mathrm{T}}(\mathbf{b}_j \times \mathbf{b}_k)] = \text{sign}[\mathbf{r}_I^{\mathrm{T}}(\mathbf{r}_J \times \mathbf{r}_K)] \tag{16}$$

is satisfied. This allows us to discard specular triangles. Note that if more than one nonspecular cataloged triangle is found to match the measured triangle to within the measurement tolerance, the star identification is not accepted as unique.

(2) If $n > 3$, the Pyramid algorithm looks for a unique triangle $[i,\ j,\ k]$ by scanning the "smart" combination indices associated with all the star triangles and checking the k-vector accessed indices to establish a hypothesis for the cataloged indices for each star. Also, using the derived formulas, the frequency with which this measured polygon match could be made with a random invalid polygon from the catalog can be computed; if this number is greater than some tolerance, the star identification is rejected.

(3) If a high-confidence triangle identification is found in step 2, Pyramid will scan the remaining stars to find one that further confirms the basic star triangle $[i,\ j,\ k]$, with the analytical frequency test employed at each stage.

(4) When step 3 is accomplished successfully, the pyramid is identified as that having the star indices $[i,\ j,\ k,\ r]$. This means that these four stars are, at this point, identified with a very high confidence. The three stars constituting the basic star triangle $[i,\ j,\ k]$ are then used to identify the remaining stars (p) as good ones (when the stars confirms the basic star triangle) or to identify the measured image as a spike. If desired, the entire set of identified stars can be used to form an n-star polygon, and a final frequency can be analytically computed to indicate the likelihood that a random match could match all of the angles to within measurement precision. Typical random frequencies for modern star trackers with four or more valid stars are smaller than 10^{-7}, so matching four or more stars usually results in near-certain star identification, especially if this occurs on successive star identifications and the identified stars overlap.

(5) If a confirming r star is not found, another basic star triangle $[i,\ j,\ k]$ is selected by choosing another "smart" combination of star indices. This means returning to step 2.

(6) If all the "smart" combinations of star indices are used, the Pyramid algorithm will provide the basic star triangle $[i,\ j,\ k]$, if unique. Otherwise the Pyramid logic will output a flag indicating a failure in the star identification process. Note our basic philosophy: we establish a level of confidence a priori, and we prefer to report a star identification failure (perhaps once in 1,000 images with four or more valid stars) rather than output a lower confidence star identification. Modern attitude estimation

algorithms can easily tolerate infrequent data dropouts, but are generally much less forgiving of invalid star identification.

PYRAMID RESULTS

The Pyramid algorithm has proven successful in Monte Carlo star image simulations, in night sky tests, and in orbit on the HETE spacecraft [1].

Comparisons of the Pyramid algorithm (in terms of speed, robustness, memory required, etc) with existing alternative algorithms are beyond the scope of this paper. These comparisons, which are much too lengthy for a paper introducing a new algorithm, would require, at the least, obtaining the original codes by the authors, which would be difficult.

MONTE CARLO SIMULATIONS

End-to-end numerical tests based on simulations of random unknown spacecraft attitude, star image centroid measurements (including measurement errors), star catalog access, Star-ID, and attitude estimation have been carried out. To determine robustness, speed and accuracy, we tested the algorithm with an increasing number of spikes. Pyramid was found to accomplish the star identification process reliably with as few as four valid stars and up to 24 random spikes. This extreme number of spurious images would be a rare occurrence in practice, but we believe that the previous-generation algorithms encounter reliability difficulties with far fewer spikes.

To simulate the centroiding errors, Gaussian noise of 50μrad (3σ) was added to the star directions. Figure 7 shows a histogram of the number of stars (good stars + spikes) for the simulated test cases, along with the number of spikes in each image.

The execution time for each image, using a MATLAB program running on 450 MHz PC with Microsoft Windows 98, is shown in Figure 8. The computation time is obviously a function of the number of spurious images; in our simulations, however, the star identification is always accomplished in less than 1s using a compiled version of the algorithm from C source code.

Using the fastest available attitude estimator (that is, ESOQ-2 [8], with the latest improvement [9]), the estimated attitude direction cosine matrix $\mathbf{C}_E(t)$ is calculated using the observed star vectors and the cataloged star vectors of the identified stars. The parameter that quantify "how far" the estimated attitude is with respect to the true is described by the maximum direction error ($\max\{\varepsilon\}$) or the expectation ($E\{\varepsilon\}$) of the direction error, which are evaluated according to

$$\max\{\varepsilon\} = \cos^{-1}\left(\frac{\text{trace}[\mathbf{C}_T\,\mathbf{C}_E^{\mathrm{T}}] - 1}{2}\right) \qquad \text{and} \qquad E\{\varepsilon\} = \frac{\pi}{4}\,\max\{\varepsilon\} \tag{17}$$

During simulation, the true attitude $\mathbf{C}_T$ is known. This allows us to conveniently describe the error of $\mathbf{C}_E$, provided by equation (17), by three different meaningful components. These

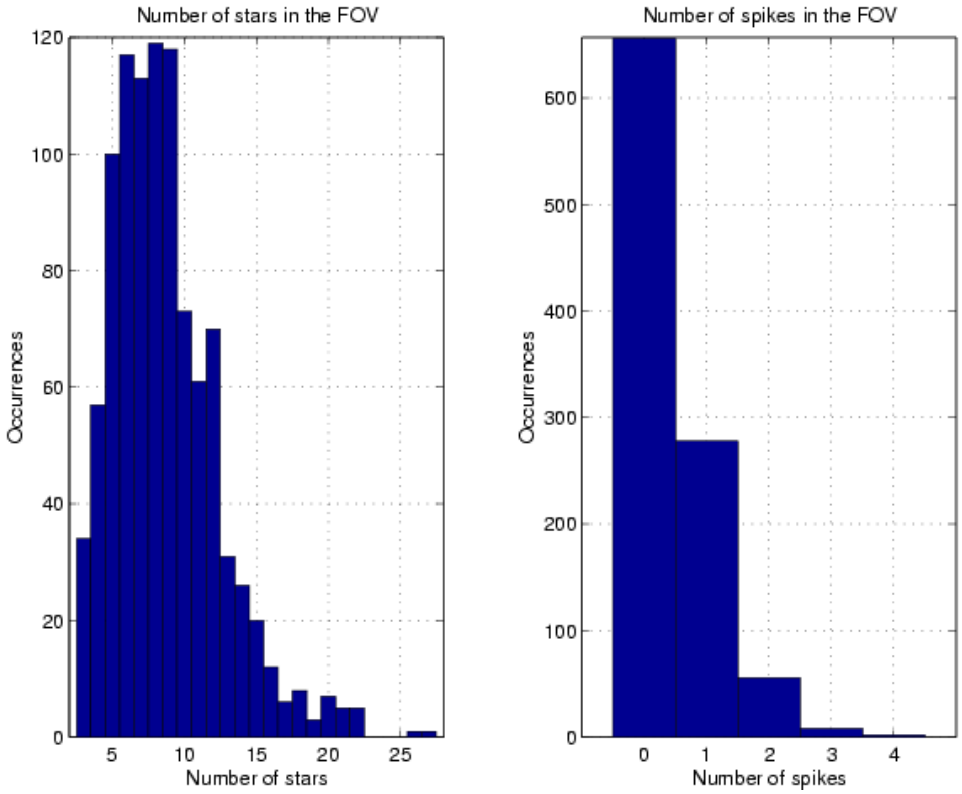

Figure 7: Histogram of the Number of Star and Spike Occurrence during Tests

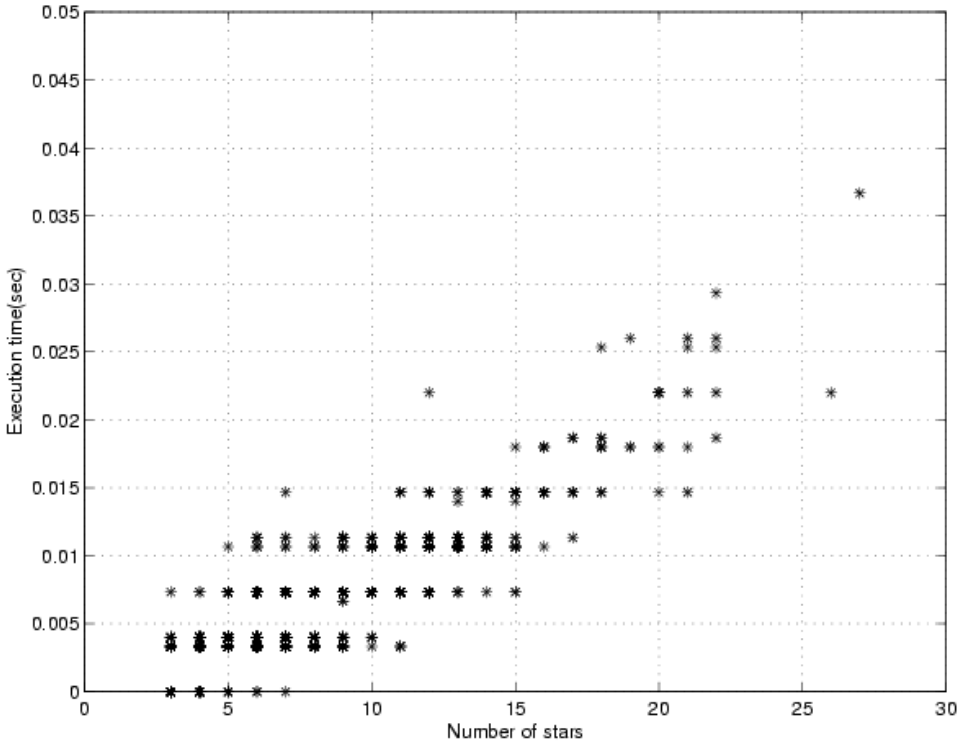

Figure 8: The Execution Time VS the Number of Stars

errors are (1) the error ε_{oa} of the sensor optical axis (OA), and (2) the error range, $\min(\varepsilon_n)$ and $\max(\varepsilon_n)$, experienced by the directions orthogonal to the OA. These errors describe the polarization of the attitude error about the OA when the attitude is estimated by single FOV star tracker, and fully justify the proposed use of multiple-FOV star trackers. From a mathematical point of view, these errors are evaluated as

$$\varepsilon_{oa} = \cos^{-1}(\mathbf{b}_{oa}^{\mathrm{T}} \mathbf{C}_T \mathbf{C}_E^{\mathrm{T}} \mathbf{b}_{oa}) \qquad \text{and} \qquad \begin{cases} \min(\varepsilon_n) = \cos^{-1}(\mathbf{b}_n^{\mathrm{T}} \mathbf{C}_T \mathbf{C}_E^{\mathrm{T}} \mathbf{b}_n) \\ \max(\varepsilon_n) \equiv \max\{\varepsilon\} \end{cases} \tag{18}$$

where

$$\mathbf{b}_n = \frac{\mathbf{b}_{oa} \times (\mathbf{b}_{oa} \times \mathbf{e})}{\|\mathbf{b}_{oa} \times \mathbf{e}\|} \tag{19}$$

In particular, $\mathbf{b}_{oa}$ identifies the on-board direction of the OA, and $\mathbf{e}$ is the principal axis of the corrective attitude matrix $\mathbf{C}_T \mathbf{C}_E^{\mathrm{T}}$. The numerical values of ε_{oa} and $[\min\{\varepsilon_n\},\ \max\{\varepsilon_n\}]$, obtained by numerical tests, are shown in Figures 9, 10, and 11, respectively.

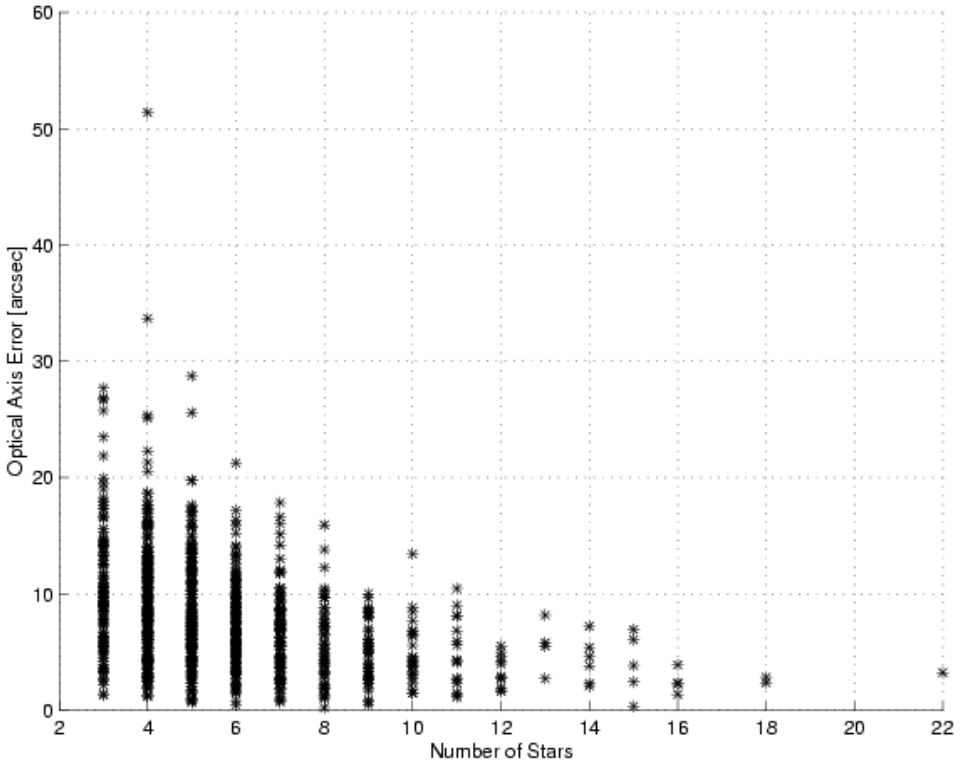

Figure 9: Mean Errors of the Optical Axis

To validate Pyramid, a 1,000-snapshot observations (images) were simulated using random attitudes. The tests simulated a VC51 camera (752×582 CCD pixels of 6.5μm$\times 6.25\mu$m pixel size) with a 35mm lens corresponding to a rectangular 3.92×2.91 deg. FOV. The magnitude threshold was set to 5.8, which implies a 3,694-star catalog. Correspondingly, the k-vector was 57,798 elements long.

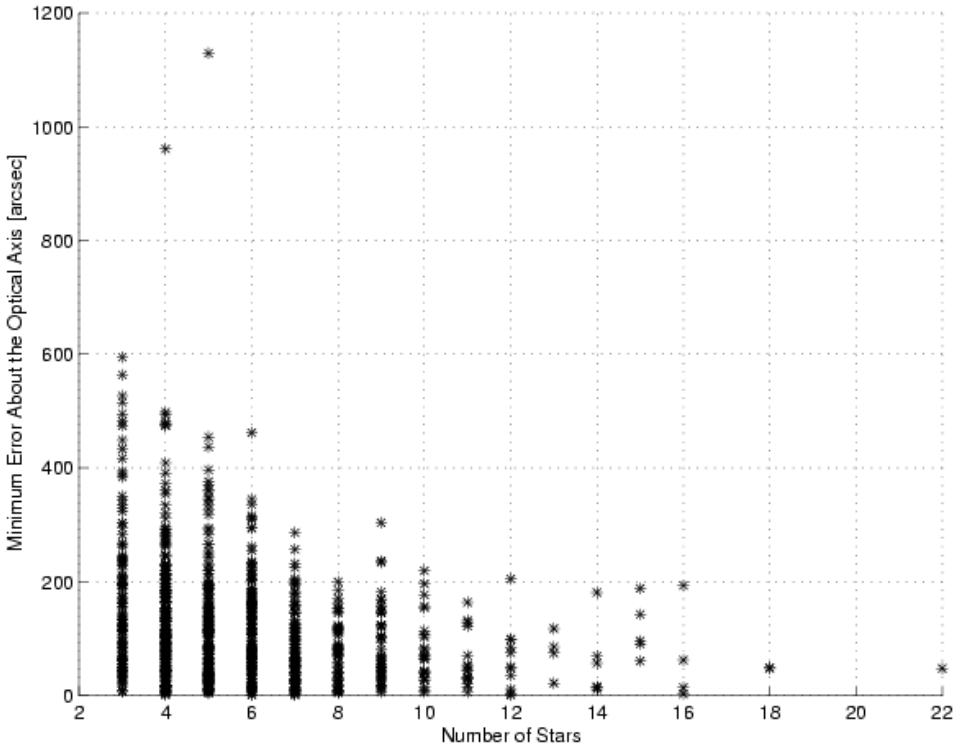

Figure 10: Minimum Errors About the Optical Axis

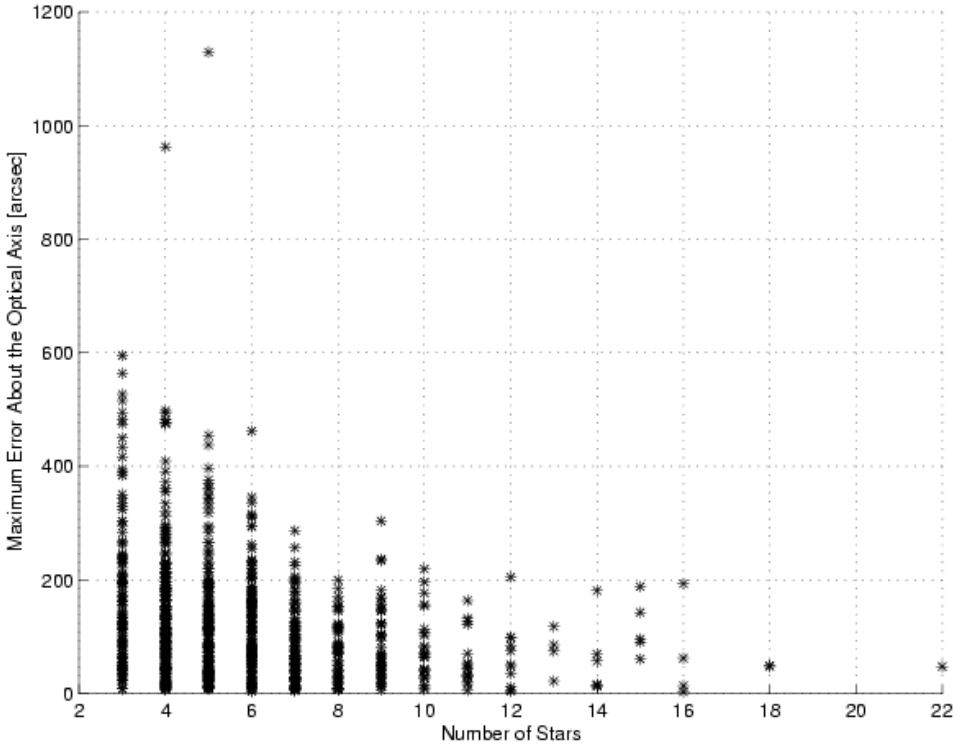

Figure 11: Maximum Errors About the Optical Axis

In the 1,000 tests, the number of successful identifications were 958, which represents a successful proportion of 95.8 percent. The 42 failures were all experienced when $n = 3$ stars were observed, and multiple catalog triangles were found that could be associated with the observed triangle. In these cases, Pyramid outputs a failure identification flag. No failure was experienced for the $n \geq 4$ scenarios. Many $n = 3$ cases (associated with a unique catalog triangle) were successfully identified. However, these tests confirmed that the $n = 3$ case is dangerous because it is associated with a greater chance of failure.

SKY NIGHT TESTS

Pyramid has been tested extensively and successfully both in orbit [1] and in night sky experiments. For the latter, a VC51 camera (752×582 CCD pixels of 6.5μm$\times 6.25\mu$m size) was used. The VC51 is equipped with an ADSP2181 32 MHz digital processor, a memory of $16K \times 16$ bits for programs and $16K \times 24$ bits of data. For the image storage and processing, the Digital Signal Processor (DSP) is connected with a bus to a Dynamic Random Access Memory (DRAM) of 8 MB. It is possible to save several images on board the camera using a 2 MB Erasable Programmable Read Only Memory (EPROM). The lenses used were 24 mm, 28 mm, and 35 mm Nikon lenses, connected through a C adapter. The DSP Pyramid code can simultaneously track up to 24 stars. Before performing the tests, the correct focal length f of the lens was evaluated using the new Non-Dimensional Star-ID algorithm [10], capable of identifying observed stars even with a lens whose focal length is completely unknown. Successful additional night sky tests of Pyramid were performed with the GIFTS prototype camera Star1000 (1024×1024 pixels). These tests validated the extension of Pyramid to multiple-FOV camera.

The most likely failures are associated with a defective image (intrusion by sun/moon/earth, etc.), sensor hardware malfunction, and sensor aging. Occasionally, sparse star fields containing fewer than four stars would be encountered. These events are anticipated in the algorithm and detected by failure modes.

CONCLUSIONS

This paper has introduced a novel method for star pattern identification based on matching interstar angles between measured vectors to those from a star catalog. We have consider the case of having no prior information, and shown that our Pyramid algorithm can solve the lost-in-space problem while being highly tolerant of spurious events such as reflections and electronic noise. The Pyramid algorithm is also believed to be the most efficient algorithm for solving the lost-in-space problem. Even with $\sim 6,000$ cataloged stars, a high percentage of spurious images, and no prior information, we can typically identify a measured star field within a small fraction of a second given the constraints of routinely available computers. Such velocity and robustness have been achieved because at the heart of Pyramid is the k-vector method for accessing the candidate catalog stars without searching for any measured pair, and because the Pyramid idea is established and supported by analytically expected

random frequencies associated with matching interstar angles from measured star polyhedra. Finally, we note that the computational and night sky experimental validations of the results of this paper will be augmented by on-orbit validation in Draper's Inertial Stellar Compass for the New Millennium Program.

APPENDIX: The k-vector range searching technique

Let y be an n-long data array (the data might be the interstar dot products of all the star pairs admissible within the star tracker FOV) and s be the same array but sorted in ascending mode, i.e. $s(i) \leq s(i+1),\ i = 1, \cdots, n-1$. Let I be the integer vector that keeps a record of the sorting $y(I(i)) = s(i)$. In particular, let us to define $y_{\min} = \min_i y(i) = s(1)$ and $y_{\max} = \max_i y(i) = s(n)$.

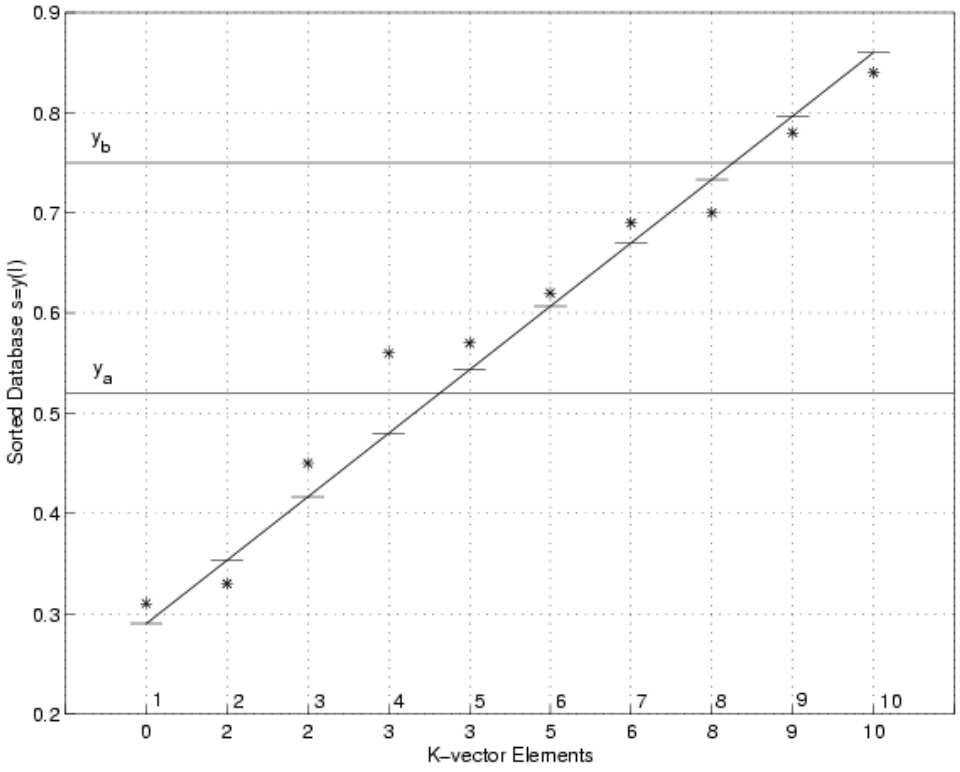

Figure 12: Example of k-vector Construction

k-vector construction

Under the hypothesis that y has uniformly distributed data, the straight line connecting the two points $[1,\ y_{\min}]$ and $[n,\ y_{\max}]$ has, on average, $E_0 = \dfrac{n}{n-1}$ elements for each $d = \dfrac{y_{\max} - y_{\min}}{n-1}$ step. With reference on the example given in Figure 12, let us consider a

slightly steeper line which connects the point $[1,\, y_{\min} - \xi]$ with the point $[n,\, y_{\max} + \xi]$, where $\xi = \varepsilon \max[\,|\, y_{\min} \,|, |\, y_{\max} \,|\,]$ and ε is the relative machine precision ($\varepsilon \simeq 2.22 \cdot 10^{-16}$ for double-precision arithmetic). The equation of this slightly steeper line line is

$$z(x) = m\, x + q \tag{20}$$

where

$$m = \frac{y_{\max} - y_{\min} + 2\xi}{n-1} \qquad \text{and} \qquad q = y_{\min} - m - \xi \tag{21}$$

Starting with $k(1) = 0$, the integer vector k is then constructed as follows

$$k(i) = j \quad \text{if the } j \text{ index satisfies} \quad s(j) \le z(i) < s(j+1) \tag{22}$$

where the index i ranges from 2 to $(n-1)$. From a practical point of view, $k(i)$ gives the number of the elements $s(j)$ below the value $z(i)$. Figure 12 shows the construction of the k-vector for a 10-element database. The small horizontal lines are equally spaced at points $z(i)$, and they give the k-vector values

$$k := \{\, 0,\ 2,\ 2,\ 3,\ 3,\ 5,\ 6,\ 8,\ 9,\ 10 \,\}$$

k-vector use

The evaluation of the two indices identifying the data falling within the range $[y_a,\ y_b]$ becomes an almost searchless task. The indices associated with these values in the s vector are simply provided as

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \qquad \text{and} \qquad j_t = \left\lceil \frac{y_b - q}{m} \right\rceil \tag{23}$$

where the function $\lfloor x \rfloor$ is the integer number immediately below x, and $\lceil x \rceil$ is the larger integer number next to x. In the example of Figure 12, $j_b = 4$ and $j_t = 9$. Once the indices j_b and j_t have been evaluated, it is possible to compute

$$k_{\text{start}} = k(j_b) + 1 \qquad \text{and} \qquad k_{\text{end}} = k(j_t) \tag{24}$$

Knowledge of k_{start} and k_{end} represents the solution of the range searching, since the searched elements $y(i) \in [y_a,\, y_b]$, are all the $y(I(k))$ elements provided by ranging k from k_{start} to k_{end}. In the example of Figure 12, however, the searched elements should be those identified by the range indices $k_{\text{start}} = 4$ and $k_{\text{end}} = 8$, while the proposed technique outputs $k_{\text{start}} = 4$ and $k_{\text{end}} = 9$. This problem can easily be solved with a linear search at the beginning and end of the retrieved data. The details can be found in [6].

References

[1] Crew, G. B., Vanderspek, R., and Doty, J., *HETE Experience with the Pyramid Algorithm*, MIT Center for Space Research, Cambridge, MA, 02139 USA, 2002.

[2] Brady, T., Tillier, C., Brown, R., Jimenez, A., and Kourepenis, A., *The Inertial Stellar Compass: A New Direction in Spacecraft Attitude Determination*, Paper SSC02-II-1, 16th Annual USU Conference on Small Satellites, 2002

[3] Mortari, D., Junkins, J.L., and Samaan, M.A., *Lost-In-Space Pyramid Algorithm for Robust Star Pattern Recognition*, Paper AAS 01-004, Guidance and Control Conference, Breckenridge, Colorado, 31 Jan. - 4 Feb. 2001.

[4] Mortari, D., Junkins, J. L., and Samaan, M. A., *An Analytical Approach to Star Identification Reliability," in preparation.*

[5] Mortari, D., *A Fast On-Board Autonomous Attitude Determination System based on a new Star-ID Technique for a Wide FOV Star Tracker*, Advances in the Astronautical Sciences, Vol. 93, Pt. II, 1996, pp. 893-903.

[6] Mortari, D., *Search-Less Algorithm for Star Pattern Recognition*, Journal of the Astronautical Sciences, Vol. 45, No. 2, April-June 1997, pp. 179-194.

[7] Mortari, D., and Neta, B., *k-vector Range Searching Technique*, Paper AAS 00-128 of the 10th Annual AIAA/AAS Space Flight Mechanics Meeting, Clearwaters, FL, Jan. 23-26, 2000.

[8] Mortari, D., *Second Estimator of the Optimal Quaternion*, Journal of Guidance, Control, and Dynamics, Vol. 23, No. 5, Sept.-Oct. 2000, pp. 885-888.

[9] Markley, L.F., and Mortari, D., *New Developments in Quaternion Estimation from Vector Observations*, Paper AAS 00-266 of the Richard H. Battin Astrodynamics Symposium Conference, Texas A&M University, College Station, TX, Vol. 106, March 20-21, 2000, pp. 373-393.

[10] Samaan, M. A., Mortari, D., and Junkins, J. L., *Non-Dimensional Star Identification for Uncalibrated Star Cameras*, Paper AAS 03-131 Space Flight Mechanics Meeting, Ponce, Puerto Rico, February 9-13, 2003.