Design and simulation of a star tracker for the Aramis nanosatellite

Diego A. Urbina[1]

Abstract

Star Trackers are devices that provide higher accuracy than other attitude sensors with the added benefits of 3-axis attitude determination. Nevertheless, Star Trackers are frequently heavy, complex and costly systems that can not be adopted by small satellites such as the ARAMIS from Politecnico di Torino, which needs high-accuracy attitude determination to cover the requirements of certain types of payload.

A preliminary design of a low-mass, low-cost, low-power and coarse accuracy Star Tracker is proposed to satisfy the requirements of the ARAMIS spacecraft. Diferent available algorithms for identifying the presence of single stars on the imager plane are analyzed, as well as those for pattern recognition necessary to ultimately measure the spacecraft attitude. One set of such image processing and pattern recognition algorithms are chosen for use on board Aramis. Subsequently, they are tested with the experimental use of the 3D open source planetarium Celestia, while a parallel test of the image processing algorithms is performed on real star field imagery to confirm their capabilities with real-world data.

A scheme is proposed to reduce the amount of false results thanks to the use of attitude approximations coming from other sensors, through the homogeneous segmentation of the celestial sphere.

Key words:
Star Tracker, attitude and orbit determination systems, image processing

Resumen

Los sensores de estrellas son sistemas que proveen una exactitud mayor que otros sensores de orientación, con el beneficio añadido de una determinación de la orientación en 3 ejes. Desafortunadamente, los sensores de estrellas son frecuentemente pesados, complejos y costosos, de manera que no pueden ser adoptados por satélites como el ARAMIS del Politécnico de Turín, que necesita una alta exactitud para cubrir los requisitos de ciertos tipos de carga útil.

Un diseño fue propuesto, de sensor de estrellas de reducidos masa, costo y potencia; y de exactitud más baja que los sensores de estrellas comerciales, pero más alta que otros tipos de sensores de orientación; que satisfaga los requisitos de ARAMIS. A continuación, diferentes algoritmos para identificar la presencia de la estrella individual en el sensor CMOS fueron analizados, junto con aquellos necesarios para medir la orientación del satélite. Un conjunto de algoritmos de procesamiento de imagen y reconocimiento de patrón fueron seleccionados. Luego, fueron probados con el uso experimental del popular planetario de fuente abierta Celestia, mientras que un test paralelo de los algoritmos de procesamiento de imagen fueron hechos en imágenes del cielo real, para confirmar su capacidad en el mundo real.

Un esquema fue propuesto para reducir la cantidad de resultados falsos gracias a aproximaciones de orientación provenientes de otros sensores, a través de la segmentación homogénea de la esfera celeste.

Palabras clave:
Sensor de estrellas, sistemas de determinación de orientación y órbita, procesamiento de imagen.

1 Electronics Engineering Department, Politecnico di Torino, Torino, Italy, diego.urbina@masters.isunet.edu

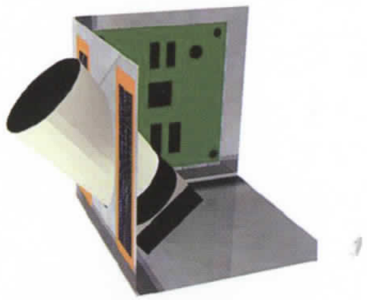


Figure 1. Mockup image of the Star Tracker Setup.

Introduction

Orientation by stars can be traced back to the earliest navigators of the seas who used the stars to know the direction of their journeys by interpreting the visible star constellations. Modern satellites use this very concept to orient themselves in space, but in a more efficient manner.

A Star Tracker takes pictures of the sky, identifies the patterns and compares them with a database of stored patterns. This process ultimately leads to a measurement of the direction in which the spacecraft is pointing, or attitude. Many different sensors able to determine the attitude of a satellite exist. Of this methods, the one that offers the highest accuracy is the Star Tracker [1].

Aramis is a new type of satellite, evolved from Politecnico di Torino's PiCPoT CUBESAT. The project aims at the creation of a highly flexible small satellite, which won't have to be redesigned almost from scratch for each different mission, taking advantage of the possibilities given by a modular architecture.

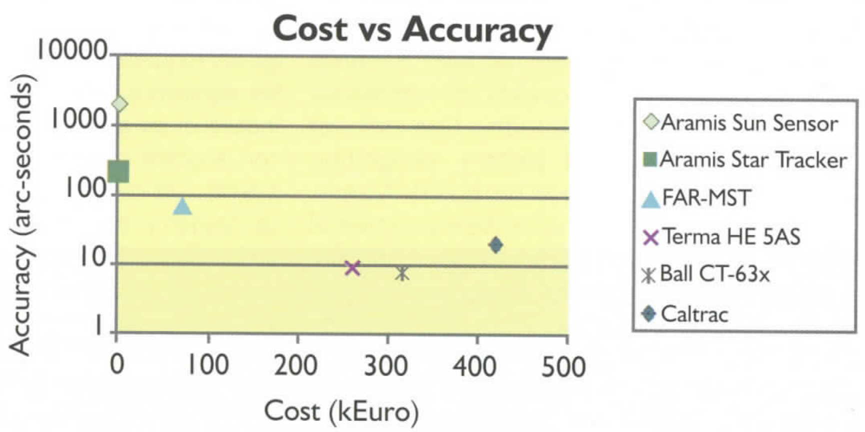


Figure 2. Comparison of the Aramis Star Tracker with other Star Trackers: Cost versus Accuracy.

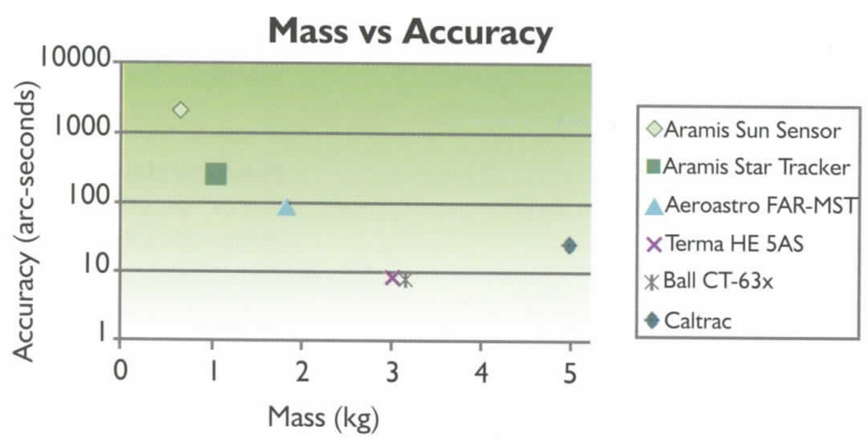


Figure 3. Comparison of the Aramis Star Tracker with other Star Trackers: Mass versus Accuracy.

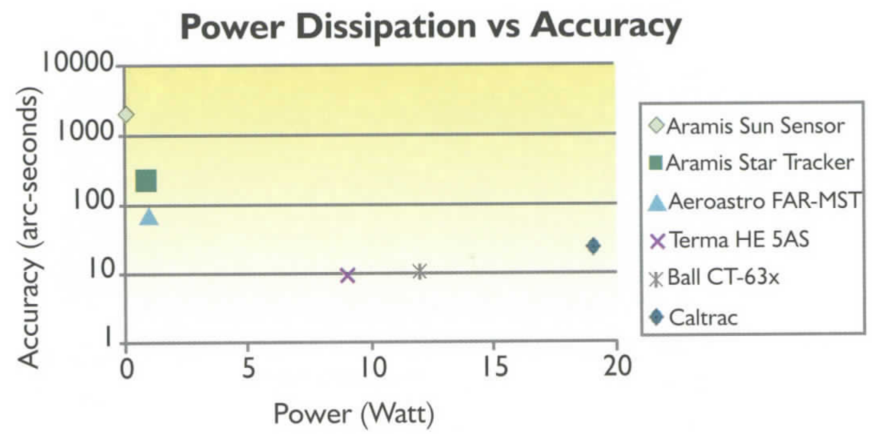


Figure 4. Comparison of the Aramis Star Tracker with other Star Trackers: Power versus Accuracy.

The Aramis Bus will be able to carry different kinds of commercial or scientific payloads that are meant to function in Low Earth Orbit. Certain kinds of payloads, such as telescopes, require high accuracy attitude determination, thus, the need for a Star Tracker on board Aramis arises.

The space environment in which the Star Tracker will work is quite harsh, and consequent Single Event Latchups and Upsets can happen [2][3]. SELs are handled by the power management subsystem, avoiding the complete loss of the Star Tracker and SEUs are mitigated by the use of components less susceptible to radiation, specially for the memories; and in the case of the Imager, by invalidating any attitude reading that comes from SEU-affected images.

Architecture overview

The Star Tracker is composed by two sections. The Star Tracker Camera Unit (SCU) which consists in the optics plus an imager, and a Star Tracker Processing Unit (SPU), the section that processes the incoming image.

The optics will be based on a CMOS Aptina MT9M001 sensor working together with a Techspec Double Gauss Lens that helps avoid the distorting effects of field curvature [4]. The pinhole solution was discarded because of the excessive necessary integration times due to their scarce light collection area.

The imager detects the impinging photons and translates them into electrons. This signal is then transmitted to the processing unit that is in charge of interpreting the image to obtain relevant attitude data.

The SPU microprocessor will be an Analog Devices Blackfin DSP, a low power consuming yet powerful device [5] that presents an relatively adequate radiation tolerance, and was found being used in various projects including Star Trackers, and Payloads such as those of PicPot and even satellites operating in Polar Orbit [6], which is a high radiation orbit. Furthermore, in order to protect the necessary databases from SEUs, a Hamming Code scheme will be implemented.

A Yale Bright Star Catalogue was modified for use with the Star Tracker. The whole celestial sphere was scanned with a window with the dimension of the FOV (entered by the user), with intervals of one degree, leaving the 6 brightest stars for each iteration. Similarly, objects that are not regarded as stars were filtered out from the catalogue.

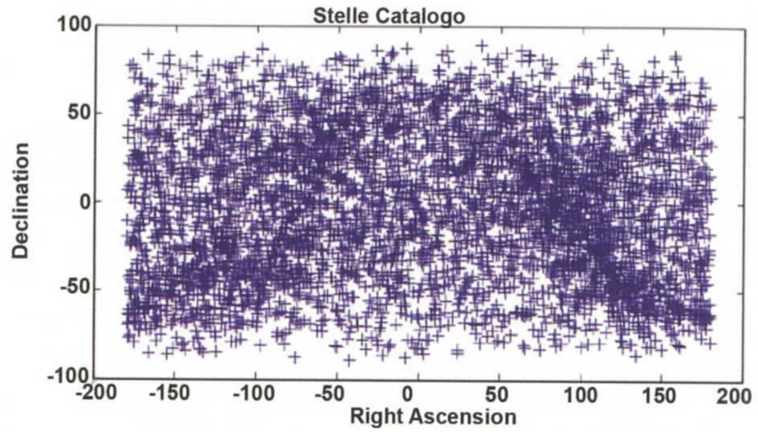


Figure 5. Full set of stars from the Yale Bright Star Catalogue

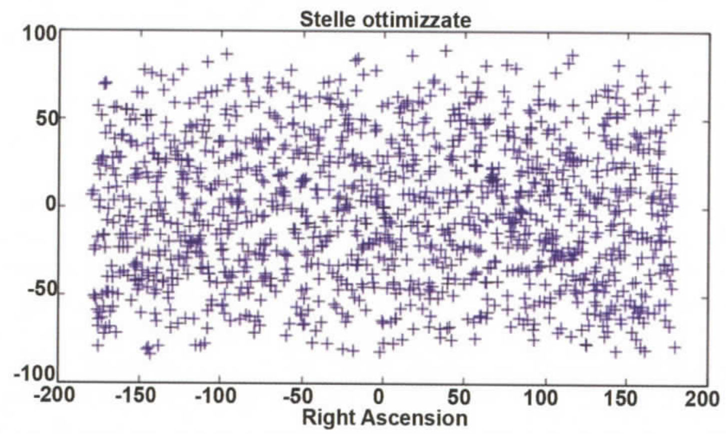


Figure 6. Filtered YBSC stars, leaving at least 6 for any given FOV.

Algorithms

Image Thresholding and Centroiding

A threshold to differentiate star pixels from the background was identified at two times the average grayscale level of the entire picture, averaged also in time, in the last 5 frames. Everything below this level is filtered out, leaving only a group of *potentially valid* pixels and residual noise peaks.

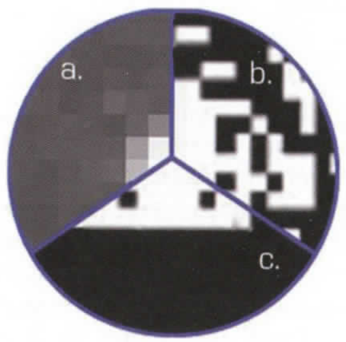


Figure 7. a. Original simulated noisy star. b. Resulting mask after thresholding of the simulated image. c. Mask after spike removal.

In order to remove these pixels, K. Huffman [7] proposes a 5-by-5 window that scans the image with 5 pixel steps to check the *potentially validity* of at least half of the pixels in the window deeming them ultimately *valid* pixels.

After the noise spike filtering is complete, a centroiding algorithm is run on the valid pixels. Using the grayscale level as the weight, it calculates the x-y position on the imager plane with equation 1

$$pos = \sum_{i=1}^{n} \frac{Ph_i \cdot position}{\sum_{i=1}^{n} Ph}$$

Equation 1

where **i** is the actual pixel, **n** is the total number of pixels of the star and **Ph** is the number of photons, or for that matter, the grayscale level of the actual pixel.

Pattern Recognition

A method was developed by C. Cole and J. Crassidis [8] to use planar triangles made up by combinations of three stars. The core idea is that more information can be obtained from a triangle than just from three angles, the latter being a common approach to pattern recognition. This will accelerate the attitude calculation time and will allow us to use less stars in average.

With the unitary vectors pointing to the three stars in the local frame of reference, it is possible to calculate the *area* and *polar moment* of the planar triangle produced by them.

A database was compiled, containing the areas of all the possible star triangles smaller than the Field Of View, along with their polar moment and Harvard Revised Numbers of the stars composing each triangle.

The software running on the Star Tracker computes these two values for three stars in the Field Of View and tries to find a match in the on board planar triangle database.

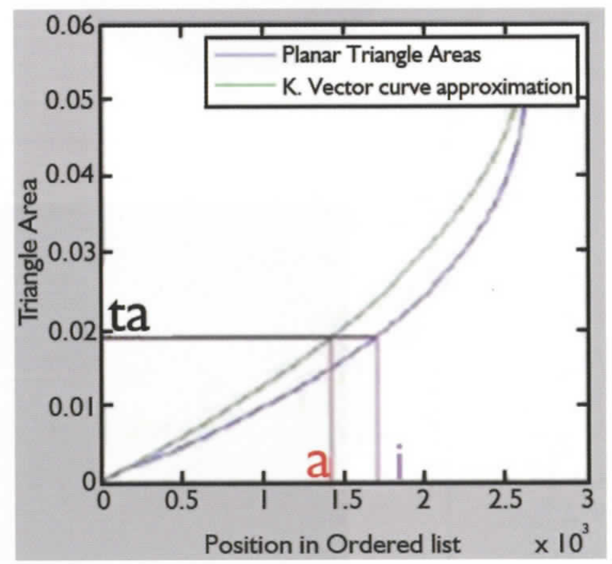
K-Vector

In the ground two lists are made, one with the original indexes of the triangle list ordered with respect to the Polar Moment, and another list of indexes **I**, ordered with respect to their Area. Then, a line is traced from each value of area to the approximating curve. The resulting list **A** will act as an index into the list of indexes *I*, producing the *i* value, which is an index into the list of triangles.

Take for instance, a measured triangle area **ta**. Its value is evaluated in the function describing the k-vector curve, which gives the **a** value. This value is the direct index into the list **I**, which gives the i value, that is, the true index in the unsorted list of areas. The same procedure is done for the Polar Moments.

Two boundaries are established based on the variances of Area and Polar Moment. As values Area and Moment obey to a normal distribution, 3σ

Figure 8. K-vector method applied to the Planar Triangle Areas.



where σ is the variance of each of them, is an interval that theoretically guarantees that 99.7% of the times the measurement will fall within these bounds.

Pivoting

In order to avoid the the ambiguous determination of the area and polar moment caused by the use of a range determined by the boundaries described earlier, a pivoting method must be adopted [8]:

An area and moment measurement is performed on the triangle. Three things can happen: no match is found, case in which measurements are restarted from the beginning with a new acquired image *(no result)*; a match is found *(conclusive result)*, case in which the process is considered successful; and a third case in which no single match is found *(partial inconclusive result)*. In the last case, the measurement of a second triangle is made, this time keeping two of the original triangle's stars, and picking the fourth brightest star, and so on. Each time that a new pivoting occurs, all the triangles that do not share two stars with the previous triangle, are eliminated from the list. At the end, when no more triangles are available, there are three possibilities, *no result*, *conclusive result* and *inconclusive result*.

Aided Mode

In order to avoid taking in consideration triangles that are clearly out of the FOV when a first approximation from a coarse sensor is available, every triangle in the database of star triangles is tagged with the number of the face of an iteratively-divided icosahedron in which its unitary vector falls. Once a first approach of attitude is obtained from the coarser satellite sensors, the system evaluates on which of the 320 faces this approximation vector falls in. It will then exclude any triangle whose center is located outside this face or its directly adjacent faces.

Algorithm test

The centroiding algorithm was tested by selecting a series of adequate night sky images from a popular picture website. These images were taken with commercial photographic equipment.

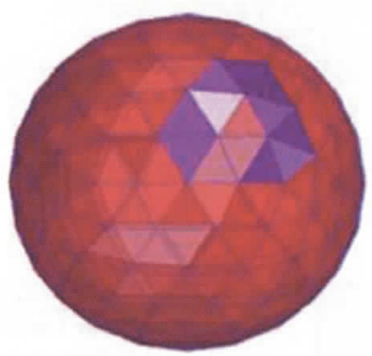


Figure 9. Geodetic sphere showing the area of interest upon reception of the coarse attitude measurement.

Figure 10. A real image with three centroided stars and detail of one of them.

The whole (thresholding and centroiding) algorithm was tested with the experimental use of the Celestia 3D planetarium, its scripting language based on LUA. The catalog was produced in C, and MATLAB, and a simulation using a video feed from Celestia runs on MATLAB. The algorithm attempts to identify the triangle formed by the 3 brightest stars at any time, using the pivoting technique.

The centroider performed flawlessly on images fairly free of compression artifacts (the image will not be compressed in the real hardware).

The simulation sequence, when the aided mode was enabled, gave only true conclusive results as opposed to when the LIS mode was active. It helped increase the reliability of the algorithm, as it automatically rejects results that are blatantly far from reality.

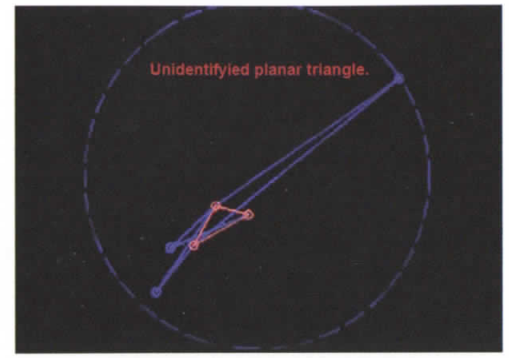


Figure 11. An inconclusive result in the simulation sequence

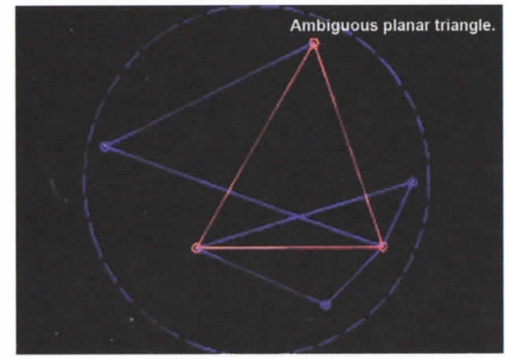


Figure 12. An ambiguous result in the simulation sequence

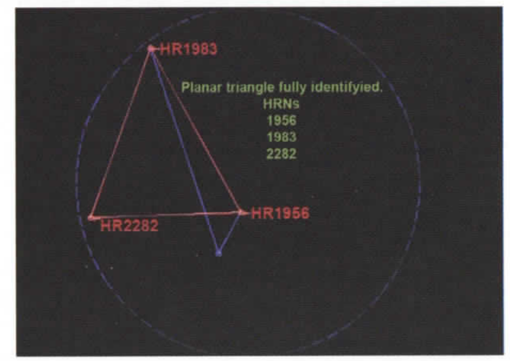


Figure 13. A conclusive result in the simulation sequence

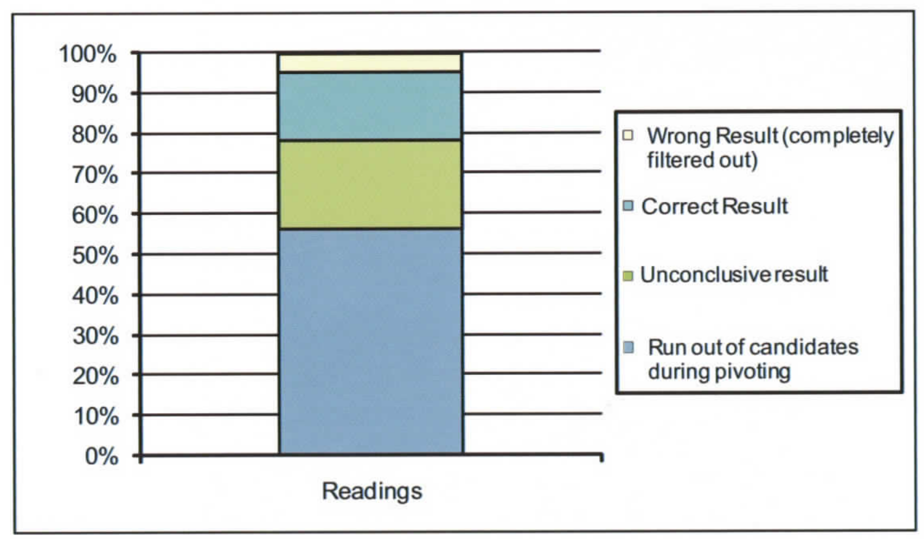


Figure 14. Distribution of the simulation results

Conclusions and future work

A low cost, low mass and low power consuming Star Tracker design was proposed and compared with the state of the art. A preliminary schematics design was produced, and components were chosen based on radiation tolerance, performance and commercial off-the-shelf availability criteria.

Radiation effects on the imager can be mitigated by data encoding and unused pattern exclusion by the means of an operative mode (aided mode) in which data from coarser Aramis sensors is used. This allows to have a smaller pool of patterns, so the likelihood of identifying a triangle that contains a false star diminishes.

Different available algorithms were analyzed and a set was picked from them. The thresholder, centroider and pattern recognition algorithms were first tested with the use of the Celestia 3D planetarium.

A set of m-files was produced and documented in order to support the future development of this project.

A star database filterer that can generate a new database with the x brightest stars per FOV was developed and used to make a 6 stars per FOV list with all the necessary data for each star. Along with it, a pattern database generator for the planar triangle was made and used to produce a database of 265292 triangles with their correspondent HRNs, areas and polar moments.

The algorithms were first tried on carefully configured screenshots and then on image sequences.

Future work shall include using the implemented algorithms with the actual optics assembly on the real star field, which should reduce the number of inconclusive and no-results, as the distortions of the star field simulator are eliminated.

Acknowledgment

Thanks to prof. Leonardo Reyneri, prof. Alberto Vallan, prof. Nikolai Tolyarenko, Dr. Angie Buckley, Dr. Jeff Hoffman, Danilo Roascio, Fabio Maggioni.

References

V. L. Pisacane. Fundamentals of Space Systems. Oxford University Press, 2005.

KENNETH A. LaBel. Single event effect criticality analysis, 15/02/1996 (Retrieved on 15/10/2008).

NASA. Single event effects specification (draft), (Retrieved on 15/10/2008). http://radhome.gsfc.nasa.gov/radhome/papers/seespec.htm.

UNIVERSITY OF BRITISH COLUMBIA MATHEMATICS DEPARTMENT. "Field curvature", (Retrieved on 15/10/2008). http://www.math.ubc.ca/ cass/courses/m309-01a/chu/Aberration/field.htm.

ANALOG DEVICES INC. ADSPBF533 data sheet, 2007.

JET PROPULSION LABORATORY M. Elghefari. Interoffice memorandum iom 5144-07-017a on ecliptic sel test, 2007.

KARA M. Huffman. Designing star trackers to meet micro-satellite requirements. Master's thesis, Massachussets Institute of Technology, 2006.

CRAIG L. Cole (1); John L. Crassidis. Fast star-pattern recognition using planar triangles. Journal of guidance, control, and dynamics, pages 64-71, 2006.