Star Field Recognition

Jachin Rupe

Department of Computer Science
Augsburg College
rupe@augsburg.edu

Abstract

This paper describes research done last summer at Oklahoma State University in the area of star field recognition. Star field recognition involves taking a photograph of stars and matching the stars in the photograph to a database of known stars. In order to do this one must first locate the stars in the photo and then set up a correspondence between them and stars in the data base. Patterns of stars in the photograph are matched to patterns in the database of stars so that individual stars can be identified. This information is used to figure out the orientation of the camera that took the photograph. The main application for star field recognition is navigation in space. On earth, technologies such as global positioning can be used for navigation, but satellites in deep space must depend on techniques such as dead reckoning to figure out where they are. When a satellite is on a deep space mission, it is critical that it keep track of what direction it is facing so it can keep an antenna pointed at earth to send back the data it is collecting. If something goes wrong, such as a piece of space debris bumping the satellite, or a rocket misfiring, or its antenna losing its line of site with earth, the current technique is to begin scanning the sky and hope that it can find the signal from earth. However if a satellite was outfitted with a camera and had the ability to recognize stars, it could orient itself precisely. The techniques used in star field recognition may have applications in machine vision as well.

1 Introduction

Star field recognition is one of the oldest techniques of navigation and has been used by humans for millennia with a great deal of success. In the past century many accurate navigation techniques were developed for navigating on earth, such as the Global Position System (GPS). However, satellites on deep space missions do not have options like these available to them. They do have the stars. The stars are convenient because the location of any particular star in the celestial sphere remains relatively constant anywhere inside of the solar system due to the great distance between the stars and the solar system. Therefore, star field recognition would be a valuable tool to allow deep space satellites to correct their orientation. This paper describes the basics of such an algorithm.

1.1 The Problem

Throughout the history of deep space exploration a lot of satellites have been lost, many for unknown reasons. One of the possible causes for losing satellites has to do with the fragile nature of the communication link between a satellite and earth. A satellite needs to have an antenna pointed directly at earth. If something happens that causes the satellite to lose its signal with earth it is forced to sweep the sky for the remote possibility of re-detecting the signal. The margin of error for the orientation of the antenna is small and it is possible that once the signal with earth is lost, a satellite might not be able to find it again. However, if a satellite was outfitted with a digital camera and had the ability to recognize stars, the satellite could determine what direction it was facing and make an educated guess about where earth was, significantly improving the chances of locating earth.

The algorithm to solve this problem should be able to do the following in a reasonable running time: Given a database of stars and a photo with an arbitrary orientation, it would match the stars in the photo to the stars in the database. Obviously the photo would have to contain a reasonable number of stars from the database and be high resolution.

Star field recognition is fairly easy for humans. Many people know how to find the big dipper or the north star. With any reasonable sized picture of the sky it would not be difficult for a knowledgeable human to match a photo of the stars with an area of the sky and from there identify individual stars. For computers though, this is a much more difficult problem because stars alone are not easily distinguishable from each other. In a digital gray-scale photo (one that captures the spectrum of light that's visible to humans), each star is just one or more neighboring pixels that are notably lighter than the sky. Some stars are brighter or larger than others but there are many stars with identical features. The only way to distinguish one star from another is for an algorithm to work with groups of stars (star fields) and their respective locations.

The primary application and goal of this project is satellite navigation. A camera mounted on the satellite would take a picture of the sky, then the image would be analyzed and the stars in the photo identified. That information can be used to determine the direction of the camera combined with the direction the camera is facing and therefore which direction earth is in.

There are some other potential applications. For instance, having a portable telescope automatically find a particular spot in the sky and then keeping it centered there despite the rotation of the earth. The algorithm may also be used in other areas for pattern recognition of a number of discrete objects.

2 The Algorithm

The solution that this paper is proposing is a software program that can match stars from a photo of the sky in a reasonable amount of time. The program needs to be fairly robust. It needs to be able to function properly despite arbitrary rotation of the image, space objects that are not stars (planets, comets, etc.) appearing in the picture and the round off error that comes from dealing with digital photos. It would also be good if the software could compensate for the scaling and size of an image.

The general idea is that first it needs a photo and a database of stars with their brightness, size and location in the celestial sphere (ascension and declination). It analyzes the star fields in the photo and attempts to match the stars in the photo to stars in the database. Once it has matched the stars in the photo, it uses information from the database about the stars in the photo to determine the orientation of the camera that took the picture.

The most complex part of this algorithm and the part that this paper is focusing on is matching the stars in the photo to stars in the database. The basic strategy for matching star fields comes from the fact that stars can be considered the corners of a polygon. Angles of a polygon stay the same over rotation and scaling operations (Figure 1). It would be possible to compare all the polygons in a picture to the polygons that can be artificially formed from the information in the database and get an accurate recognition of the photo. However, this would be very inefficient, consequently not practical.

A way to drastically reduce the amount of time it takes to match a star field is to derive all the star fields that could be in a photo. These patterns of star fields from the library of stars are used to create a library of patterns and compare patterns of star fields out of the photo which are likely to correlate with the patterns from the

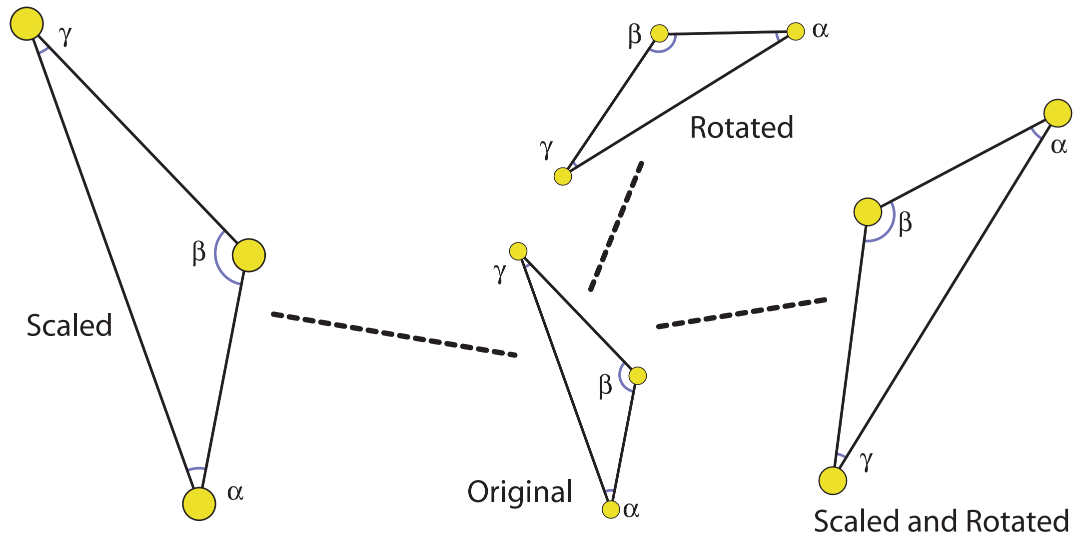

Figure 1: The angles of a polygon do not change over scaling and rotation.

same stars in the database. It requires quite a bit of computation to build the library of patterns, but once that is done the work to match each photo is much less. The number of stars in the patterns will need to large enough so the patterns are fairly unique and still be small enough so that any photo will contain enough of a pattern to identify it.

The library of patterns is generated by using the 3D graphics algorithm of an artificial camera to create images of an appropriate size. This algorithm projects points from 3D space onto 2D space much in the same way a camera in the real world projects 3D objects onto film. The same pattern-generating algorithm that is used to process the actual photos is applied to these artificial images.

The two features that are needed from a star photo is the location and relative brightness of each star. To find the stars, the first thing that has to be done is to pick a threshold. In a typical 8 bit per pixel gray scale image this is a number between 0-255. Any pixel above the threshold is considered part of a star and any pixel lower is considered part of the sky. Once that is done the connected-components algorithm processes the image, returning a list of stars and which pixels are in each star. Finally the location of each star is calculated and the values of all the pixels of each star are added together to get its relative brightness.

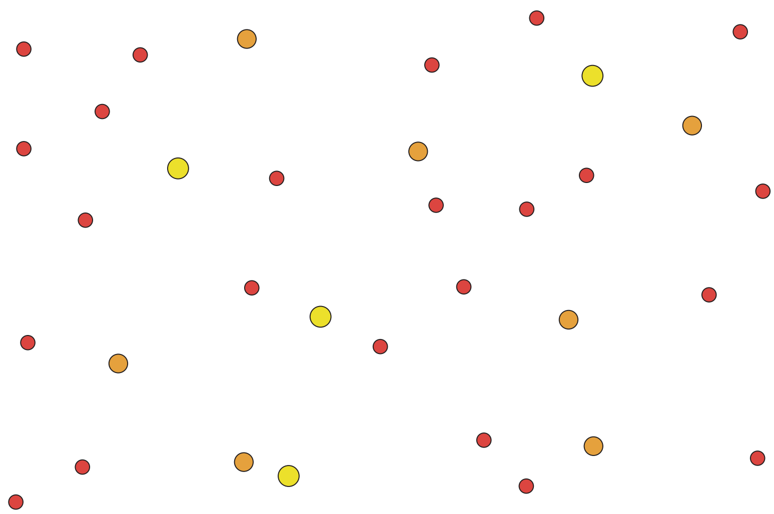

Figure 2: The is an example of a star photo that has already been processed, each circle represents a star. Large circles represent stars that will be considered for the center star, medium circles represent stars that will be considered by the algorithm as stars, and the small circles represent stars that will be ignored.

2.1 The Pattern-Generating Algorithm

The pattern-generating algorithm works by first takes a photo of the sky (Figure 2) and picking a star for the center. Next the K-Nearest algorithm determines the K nearest stars to the center. These stars will be in the pattern. The distance between the center star and the furthest star from it that will be in the pattern becomes the radius of a circle. This circle is divided into N regions. Each region represents a pie shaped piece of the circle centered around the center star. The pattern that is generated will be in the form of a bit string with one bit for each region. If there is a star in a region then the bit for that region will be a 1, otherwise a 0. Both K and N are user defined. For example, in Figure 3 there are four stars. After the center star is picked, the rest of the picture is divided up into 12 regions. There is a star in three of them, so it would generate the following bit string: 010000101010. The center star is chosen by its relative brightness. The patterns are taken at regular intervals throughout the sky.

Because of the pixilation in star photos, the location of the center of a star has a tendency to be prone to error. Consequently the error in angle measurement between stars that are very close to the center star tends to be higher, getting smaller the further the star is from the center of the pattern. In order to reduce the error, stars close to the center are ignored.

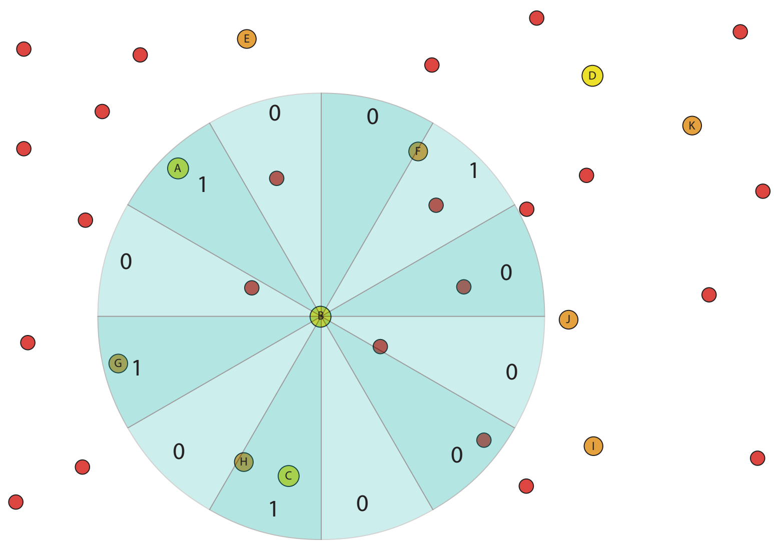

Figure 3: The same star field as in Figure 2, showing how a bit string pattern is generated from a star field.

2.2 The Matching Algorithm

The algorithm for finding matches in the library of patterns does not simply accept or reject a particular pattern. The matching algorithm takes in two bit strings and compares them to each other, returning a score that represents how similar they are. In order to match one bit string to another, rotation has to be taken into account. All the entries in the library also have to be considered. This gives the pattern matching algorithm a running time of $n^2 * m(n$ as the number of bits in the string and m as the number of patterns in the library). This is computationally expensive. However, most of the operations are bitwise operations which are very fast and experimenting with this algorithm produced some promising results.

Once the matching algorithm is done, the goal is that it will narrow the search down to a few small sections of the sky. Then particular stars can be identified. For the application of satellite navigation it is possible that this last step will not be necessary if a single location is identified from matching the pattern, but there could be some advantages for identifying particular stars that could also prove useful.

3 Future Research

The code for the artificial camera needs to be completed so the algorithm can be tested with the actual sky data. There are a lot of variables that need to be experimented with.

Geometric Hashing could also prove to be a better option than the pattern matching algorithm. So that option needs to be investigated.

The process of actually matching particular stars in the photo to stars in the database needs to be looked at more closely. It's possible that this last step is unnecessary but it might be possible to get a more accurate estimation of the orientation of the camera. It is also possible that the earth could also be identified in a photo taken from deep space which may make reorienting the satellite antenna much more precise.

Investigating other applications that this algorithm could be adapted and applied to.

4 Acknowledgments

The author would like to thank Dr. Douglas Heisterkamp and Dr. Blayne E. Mayfield from Oklahoma State University for their guidance and Amos Zoellner from the University of Minnesota for his help with the research. This research was supported by National Science Foundation grant #0097625.