Fast Spherical Search Algorithm[1,2]

Carl Christian Liebe
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr, Pasadena CA 91109-8099, USA
818 354 7837
carl.c.liebe@jpl.nasa.gov

Abstract--This paper describes an algorithm to search a number of spherical points and find the points that are within a given distance to a given point. The described algorithm is not optimal, but is easy to implement. Basically, it utilizes a number of spherical coverings to build the data structure. Results of an implementation are presented.

TABLE OF CONTENTS

1. INTRODUCTION

Many applications require identification of (guide) stars in a star catalog that are close to a given celestial coordinate. The objective of this paper is to discuss a method of organizing the star catalog and an algorithm to search the catalog structure very fast (O(logN)) for small Field Of View (FOV). For larger star catalogs (>10^7 entries) a fast spherical search algorithm is imperative.

2. SPHERICAL SEARCH STRUCTURE

There exist a number of different mathematical theories and criteria for placing N points optimally on the sphere [1]. In this application, it is desirable to minimize the maximum distance from any point on the sphere to the closest of the N points. This is called a spherical covering. The theory will not be discussed further in this paper, but two examples of spherical coverings are shown in Figure 1 for N=18 and N=1082.

A number of independent levels with homogeneous distributed points on a sphere must be constructed. From [1] precompiled constellations can be downloaded. The different levels are independent but contain an increasing number of points. The points in each level are linked to the closest points in the previous level and the closest points in the next level. Finally an entry level containing only one point is added as the entry point and the stars from a star catalog are linked to the closest elements in the last level. An example of this structure is shown in Figure 2.

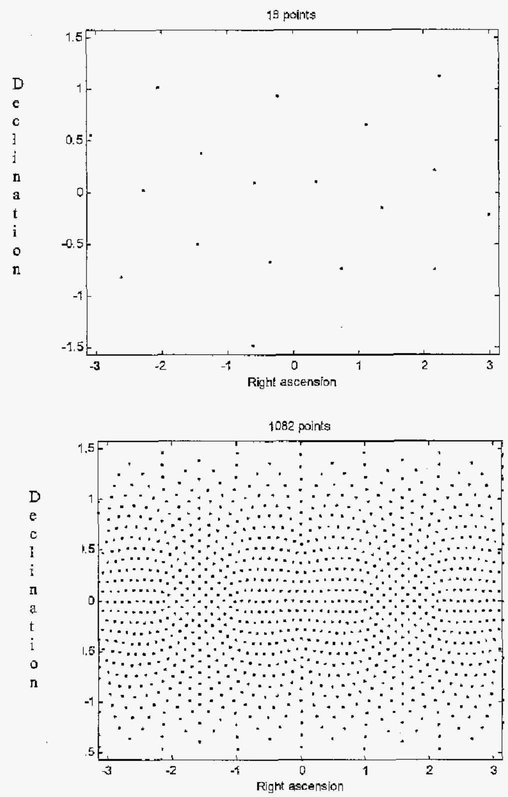

Figure 1 - Spherical coverings for N=18 and N=1082. The x-axis is the right ascension and the y-axis is the declination

There are many sources for commercial star catalogs [2] – [4]. In Figure 2 the structure contains a star catalog of 9 stars. The search structure has 2 levels excluding the entry point and the star catalog entries. The levels contain constellations of 3 and 6 elements homogeneously distributed on the sphere. A prototype element is shown on the right side of Figure 2. It contains information about the element number, a reference to the closest element in the previous level and a reference to the first element in the next level. The element also contains information about the Cartesian coordinate on the sphere. Finally, the element has information about the largest angle to any element in all levels below this element. This value will be close to 180° for the entry element and will decrease as the number of elements on the sphere increases. For the star catalog

[1] 0-7803-8155-6/04/$17.00© 2004 IEEE
[2] IEEEAC paper #1586, Version 1, Updated December 09, 2003

entries, this distance is 0°, to indicate that there are no further sub levels.

In Figure 2 it is observed that an element may refer to different numbers of elements in the next level. It is undesirable to have a variable number of pointers in a fixed size record. Therefore, the elements are numbered as shown in Figure 2. The elements are then stored in a 1 dimensional list. Elements that are referred from the same element are stored consecutive. The algorithm then determines if the numbers in the "reference to element # in previous level" field are similar in the next element. If the numbers are the same, the algorithm recursively branches to the next element.

3. SPHERICAL SEARCH ALGORITHM

The spherical search algorithm is recursive. As input the algorithm requires a Cartesian coordinate on the sphere and a radius. The angle between the point on the sphere and the element is calculated. If the angle is smaller than the pre-calculated angle to the most distant element plus the FOV, the algorithm branches out to the first element in the next level that is linked to this element. The algorithm terminates if the angle between the point on the sphere and the element is larger than the angle to the most distant element plus the FOV, since there will be no matching stars below this element. When the algorithm reaches a star catalog entry (they do not refer to a next level) they are identified, if they are within the FOV.

The algorithm is sketched in Figure 3. It is assumed that the entire search structure is stored in a global 1 dimensional structure, as shown on the right side of Figure 3. When the spherical search algorithm is initially invoked, the parameters are: the FOV, a point on the sphere and N is the entry point, N=1. In Figure 3, there is a box called detect star. In this box, the actual registration of the star takes place.

To show how the recursive function is called, an example is shown in Figure 4. In Figure 4, the same star catalog as shown in Figure 2 is used. The following stars are being detected: 13, 14 and 18.

4. IMPLEMENTATION

An implementation of a search structure will be described in this section. In this implementation the largest constellation consists of 78032 points. The different levels were chosen so that there were approximately 4 times more points in each successive level. The different constellations are shown in Table 1.

The total number of elements in the search structure is 102546 plus the number of stars in the star catalog. A demonstration of the algorithm effectiveness is shown in Figure 5. A pointing direction with a declination of 0° and a right ascension of 0° is chosen. This is in the center of the figures. The FOV is 0.5°. In Figure 5 the size of the area (solid angle) that is covered by the different levels are shown. The recursive algorithm is called a total of 144 times.

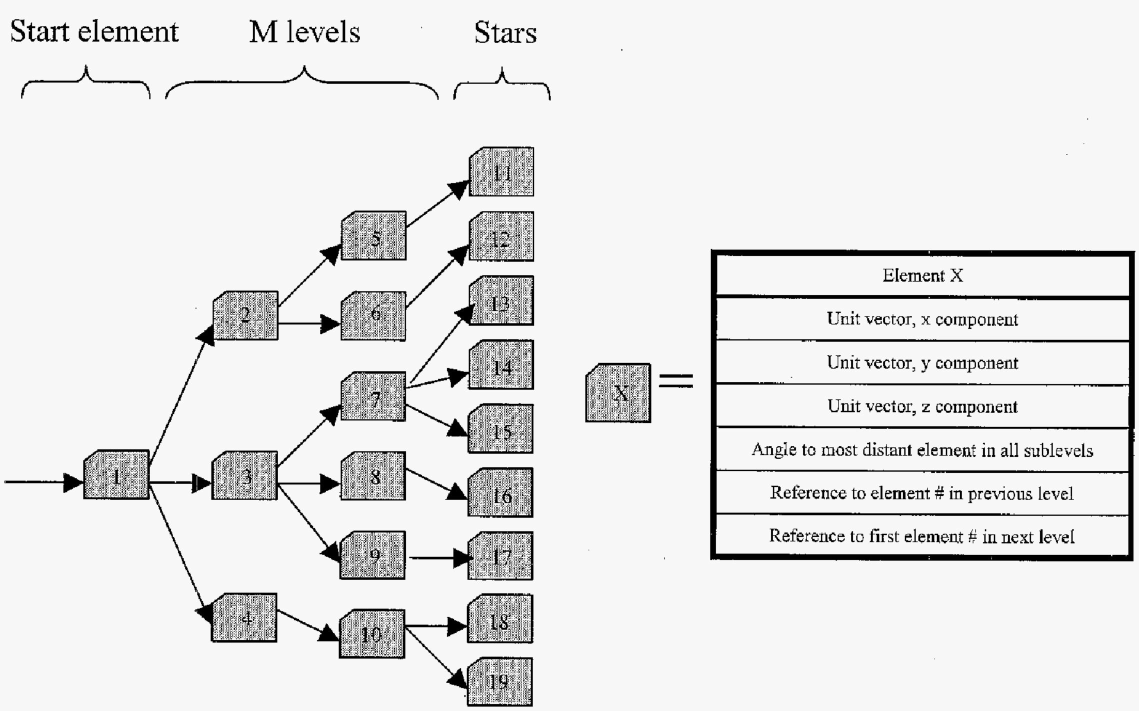

Figure 2 - The spherical search structure

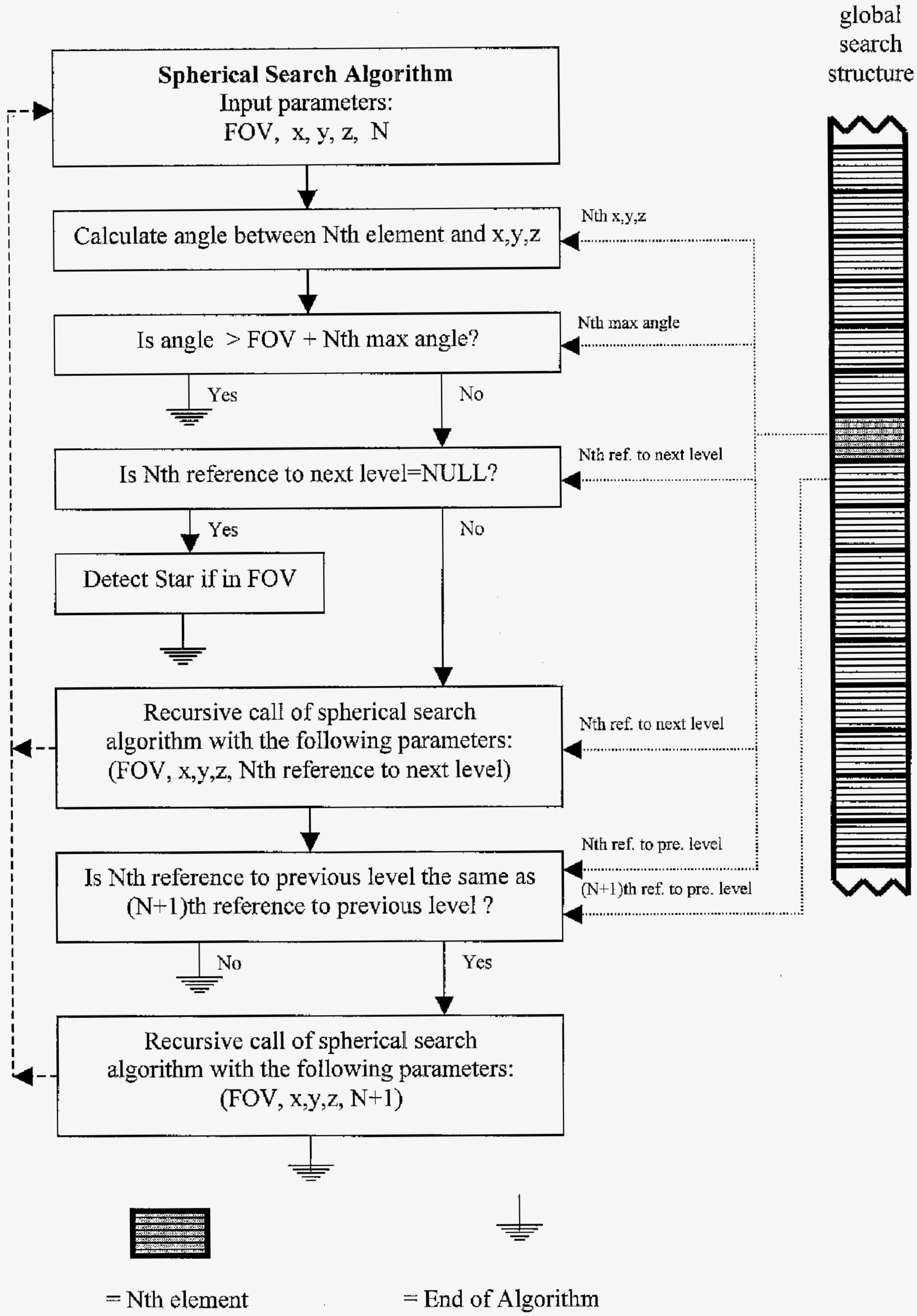

Figure 3 - The spherical search algorithm

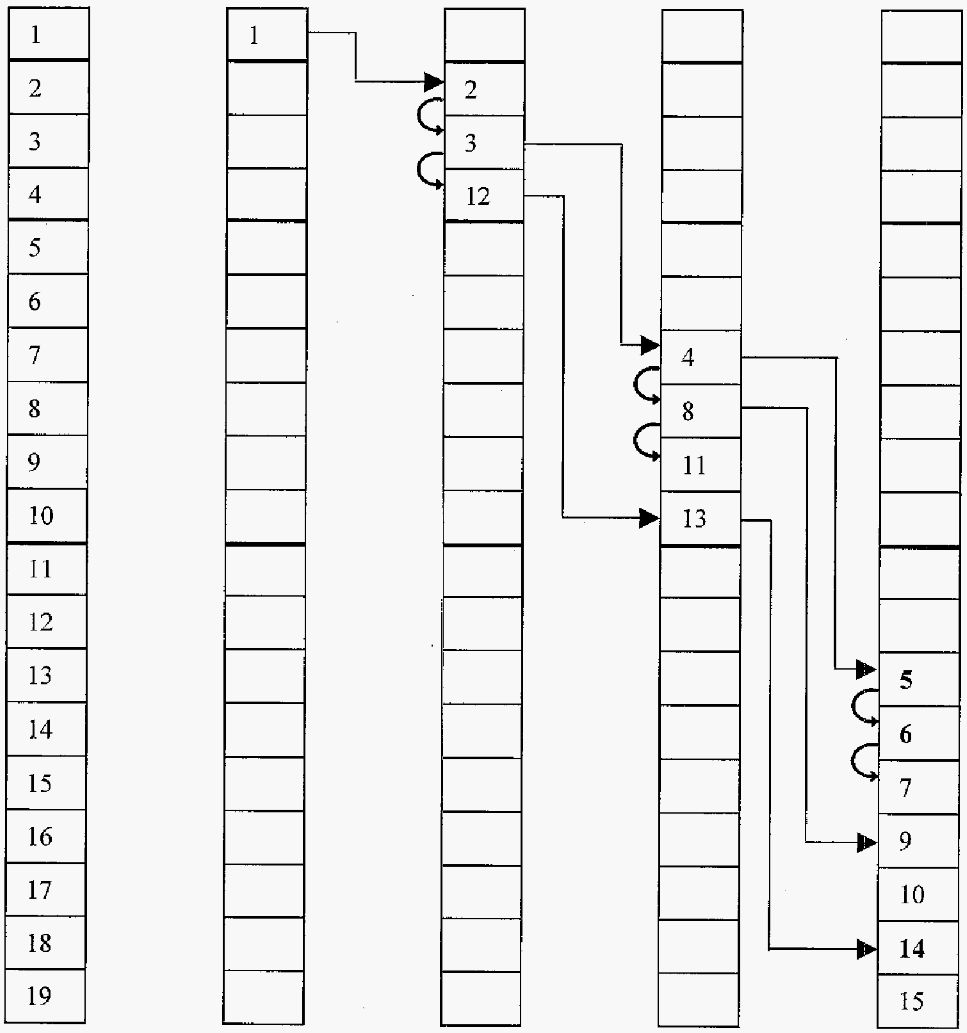

Figure 4 - An example of the recursive calls for the search algorithm. The first column is a one dimensional representation of the same data structure shown in Figure 2. The numbers in this column refers to the same numbers as in Figure 2. The next 4 columns show how the recursive function is called from the different elements. In these 4 columns the number refer to the number that the recursive function has been called. Element 13, 14 and 18 are identified.

Table 1. The different constellations in the search structure

Level	Number of Points	~ Largest angle to star in sublevel (degrees)
Start	1	180.0
1	5	124.9
2	18	61.5
3	72	30.4
4	252	15.3
5	1082	7.2
6	4332	3.3
7	18752	1.4
8	78032	0.5
9	Star catalog	0

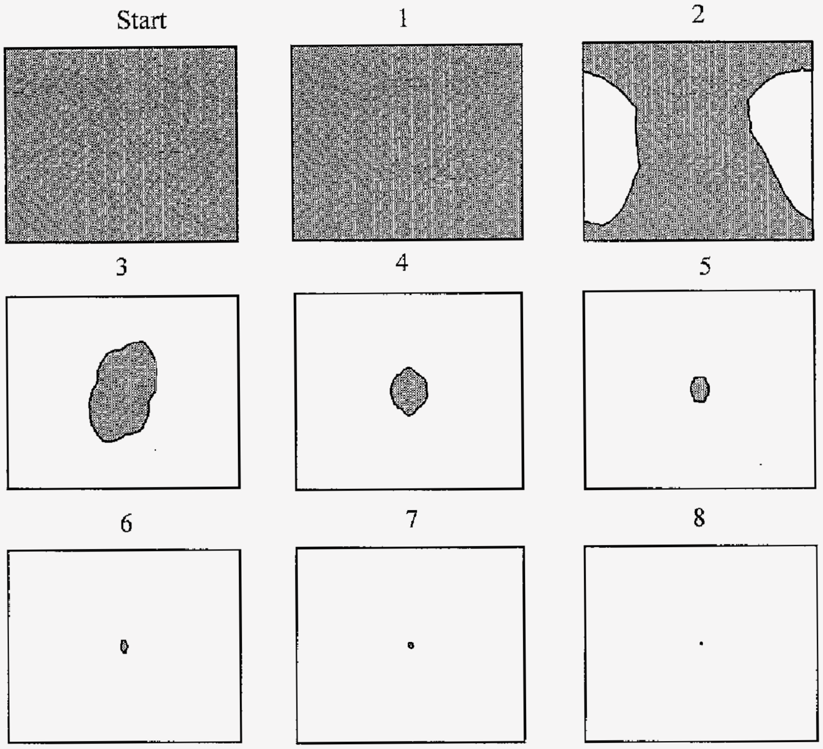

Figure 5 - The figure shows the area that is covered by the different levels in the search structure. On each figure, the x-axis is the right ascension and the y-axis is the declination.

It should be noted that the choices of constellations was not optimized for this application. This should be done for a specific size of FOV and star catalog. Also, this algorithm utilizes spherical coverings that assume the distribution of stars to be homogeneous. This is not the case since the density of stars is larger in the galactic plane. Therefore, the search tree will not be completely balanced and a few more levels should be included in the search structure.

The described algorithm is easy to implement, but it is not the optimal algorithm. First of all, it assumes homogeneity and secondly the search structure itself does take up unnecessary memory. Theoretically, it would be possible to utilize stars themselves in the search structure and choose the stars so the search structure becomes completely balanced. However, the task of generating the optimal search structure would be very difficult.

5. SUMMARY

This paper has discussed a data structure and an search algorithm that makes it possible to organize and search on a celestial sphere with an algorithm that is O(logN) for small FOV. An actual implementation with 9 levels and 78032 points were discussed.

A star catalog containing a few thousand stars might not require the described spherical search algorithm. However, when a star catalog such as the PPM star catalog with ~300.000 stars or the HST Guide Star Catalog with ~20 million stars is used, it is imperative to use an algorithm as described in this paper.

ACKNOWLEDGEMENTS

The author would like to thank Curtis Padgett, JPL for reviewing this paper.

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process or service by trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

REFERENCES

[1] R. H. Hardin, N. J. A. Sloane and W. D. Smith, Tables of spherical codes with icosahedral symmetry, [on-line] URL: http://www.research.att.com/~njas/icosahedral.codes/. [Cited March 26,2001].

[2] Simbad, France: [on-line] URL: http://cdsweb.u-strasbg.fr/Simbad.html. [Cited January 4th 2001].

[3] Goddard Space Flight Center, NASA, US: [on-line] URL: http://adc.gsfc.nasa.gov/adc/adc_cat1_holdings.html [Cited March 26th 2001].
[4] HIPPARCOS homepage: [on-line] URL: http://astro.estec.esa.nl/SA-general/Projects/Hipparcos/catalog.html. [Cited January 4th 2001].

BIOGRAPHY

Dr. Carl Christian Liebe received the M.S.E.E. degree in 1991 and the Ph.D. degree in 1994 from the Department of Electro physics, Technical University of Denmark. Since 1997, he has been an employee at the Jet Propulsion Laboratory, California Institute of Technology. Currently, he is a senior member of the technical staff in the Laser Remote Sensing Group in the Interferometry and Large Optical Systems Section. His research interests include new technologies and applications for avionics sensors.