

基于局部敏感哈希的导航星库快速搜索算法

朱海龙, 梁斌, 张涛

(清华大学 自动化系, 北京 100084)

摘要:为提高星图识别过程中导航星库的搜索速度,提出基于局部敏感哈希的导航星库快速搜索算法。通过分析星图识别原理,以角距误差限为基准,量化星角距,将有序星点集星图识别模式转换为具有局部敏感特性的整数数组。然后引用 STLport 中整数哈希函数对整数数组进行散列,得到哈希值以及对应的存储有序星点集模式中心星点编号的集合。实验结果表明:提出算法的时间复杂度为 $O(1)$,优于直接遍历搜索、二分查找搜索以及 k-vector 搜索算法。考虑实际工程应用情况,可以选择星角距误差限为 1 个像素对应角距,角距数量,此时星图识别过程中哈希表的冲突率为 0.74%,平均搜索次数为 1.007 4,星图平均识别时间 22 μs 。

关键词:星图识别;有序星点集;局部敏感哈希;星角距量化;角距误差限;仿真实验

中图分类号:V249.323

文献标志码:A

文章编号:1000-2758(2018)05-0988-07

星敏感器具有姿态测量精度高、自主工作能力强和可靠性高等优点,已经成为各类航天器首选姿态测量设备。根据工作场景不同,星敏感器的工作模式可以分为全天球自主工作模式和星跟踪模式 2 种^[1]。在初次进入工作状态或者故障恢复后,星敏感器没有任何先验姿态信息,进入全天球自主工作模式。此时星敏感器通过其光学系统获取星图,计算出星点在星图中位置,然后构造出识别模式,并且与导航星数据库中存储的识别模式进行搜索匹配,识别出星图中的导航星,进而确定航天器瞬时姿态信息。在得到先验姿态信息后,星敏感器就进入星跟踪工作模式,此时星敏感器可以利用在全天球自主工作模式下获取的先验姿态信息和航天器的运动方程,快速更新航天器姿态信息。可以看出,如何在全天球自主工作模式下准确快速地识别出导航星是星敏感器稳定可靠工作的关键。

目前,已经有许多全天球自主工作模式下的星图识别算法提出,典型的算法有三角形识别算法^[2]、Pyramid 识别算法^[3]和栅格识别算法^[4]等。在这些算法及其改进算法中,导航星数据库的搜索速度是决定星图识别算法效率的核心因素。Liebe 提出的三角形算法中选用“边角边”识别模式,直接

建立星图识别数据库,通过直接遍历搜索进行星图识别^[5]。由于三角形识别模式维度较低,在导航星数量多的情况下会导致误匹配率高和识别速度慢。为提高星图识别的鲁棒性, Mortari 等人提出 Pyramid 星图识别算法,通过增加一个角距来判断星图识别的唯一性。为了提高星图识别速度, Mortari 等人对星角距升序排列,提出 k-vector 区域搜索算法^[6],该算法的搜索速度比遍历查找算法提高了 10~50 倍。张广军等人提出基于径向和环向特征的星图识别算法^[7],该算法在星点位置误差 1 个像素的情况下识别成功率达到 97.57%,并且通过线性数据库搜索方法,提高了星图识别速度。江洁等人对张广军等人提出算法进行改进^[8],提出冗余径向和环向特征识别模式,通过冗余编码方式,降低星等和星点位置噪声对星图识别的影响,但是在识别速度方面略慢于张广军等人提出的算法。杨君等人通过 2 级压缩技术将导航星分布的二维数组压缩成一维数组,然后建立 Hash 映射实现导航星表快速搜索,实现了导航星表的无冗余存储,提高导航星的搜索速度^[9]。张磊等人在改进三角形识别算法过程中,用哈希方法散列星角距和相对星等差,提高了识别速度^[10]。熊雪等人在研究多视场星图识别技

术中,利用哈希查找的方法快速访问分块存储的星对角距,识别速度提高了 11 倍左右^[11]。Rao 等人利用星角距值在坐标系轴的投影建立哈希函数,进而提高星角距数据库的搜索速度^[12]。郭磊利用双哈希表方式存储和访问导航星数据库,利用拉链法处理冲突,降低哈希表冲突率,并提高了搜索速度^[13]。利用哈希表建立和访问导航星数据库,时间复杂度能够达到 $O(1)$,明显优于遍历查找方法、二分查找方法和 k-vector 区域搜索方法,这些方法的时间复杂度分别是 $O(n)$, $O(\log(n))$ 和 $O(k)$ 。但是哈希搜索算法是一种典型的空间换时间的方法,导航星数据库的存储量会随着访问时间的降低而快速增加。因此如何在提高导航星数据库搜索速度的同时尽可能降低导航星数据库的存储容量,是本文主要的研究方向。

由于星图成像存在各种噪声,在已有的星图识别算法中,通常是建立“一对一”的星图识别模式数据库,即在星图识别数据库中有且仅有一个识别模式与星图中构造的识别模式对应匹配。因此,这种方式建立的数据库在受到噪声影响时,不仅影响星图识别成功率,还影响星图识别速度。本文选用有序星点集作为星图识别模式,考虑角距误差对星图识别速度的影响,提出基于局部敏感哈希的快速星图识别算法,以角距误差限为基准对星图识别数据库中星角距进行量化,建立实际角距与待匹配角距之间的映射关系,利用整数哈希函数散列得到的映射值,把每个识别模式的中心星点对应的编号对应于多个哈希值,建立“一对多”的星图识别模式数据库,即在星图识别数据库中,存在多个识别模式与星图中构造的识别模式匹配,可以有效提高星图识别的速度。

1 导航星库构建

星图识别模式及其数据库是影响搜索速度的基础因素,本文选用有序星点集(ordered star points set, OSPA)作为星图识别模式,建立星图识别数据库。

1.1 有序星点集星图识别模式

星点 s_0 的有序星点集星图识别模式如图 1 所示,定义 $pat(s_0, \mathbf{S}, \mathbf{F}_i) (i = 0, \dots, k)$ 为星点 s_0 的有序星点集星图识别模式,其中 s_0 为主星点, \mathbf{S} 为主星点 s_0 的有序星点集, $\mathbf{F}_i (i = 0, \dots, k)$ 为 \mathbf{S} 中每个星点的

特征。

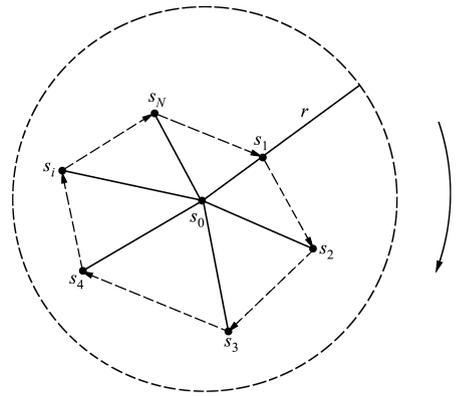


图 1 有序星点集星图识别模式

在选定主星点 s_0 后,利用 s_0 近邻算法,选取 s_0 (s_0 小于星敏感器视场角的一半) 范围内到星点 s_0 最近的 s_0 颗星,选择 s_0 颗星中距离星点 s_0 最近的星为参考星点 s_0 。以 s_0 为起点,按照顺时针顺序对 s_0 颗星排序,就可以得到星点 s_0 的有序星点集 s_0 。对于 s_0 中每个星点 s_i ,都可以用星点之间的角距表示其特征 s_0 。可以分为以下几种情况:

- 1) 主星点 s_0 的特征为 $\mathbf{F}_0 = \{e_{0,i}\} (i = 1, \dots, k)$;
- 2) 参考星点 s_1 的特征为 $\mathbf{F}_1 = \{e_{k,1}, e_{0,1}, e_{1,2}\}$;
- 3) 星点 $s_i (2 \leq i \leq k - 1)$ 的特征为 $\mathbf{F}_i = \{e_{i-1,i}, e_{0,i}, e_{i,i+1}\}$;
- 4) 星点 s_k 的特征为 $\mathbf{F}_k = \{e_{k-1,k}, e_{0,k}, e_{k,1}\}$

其中, e 表示星点之间的距离星角距,下脚标表示星点的编号。按照星点有序集 \mathbf{S} 中星点的顺序把每个星点的特征 \mathbf{F}_i 组合起来,就可以得到星点 s_0 的有序星点集星图识别模式 $pat(s_0, \mathbf{S}, \mathbf{F}_i) (i = 0, \dots, k)$ 。

1.2 星图识别数据库

在星敏感器光电系统探测范围内,按照一定原则筛选后的恒星是可用于星图识别的导航星。假设可用于星图识别的导航星的数量为 M ,按照恒星编号顺序排列后,可以得到星敏感器的星图识别数据库如图 2 所示。其中第 1 列为主星点的编号,第 2 列和第 3 列分别是主星点的赤经和赤纬,第 4 列为主星点的有序星点集模式特征 $\mathbf{F}(i) (i = 1, \dots, M)$ 。

星图识别数据库			
星号	赤经	赤纬	有序星点集特征
编号 1	α_1	δ_1	$F(1)$
⋮	⋮	⋮	⋮
编号 i	α_i	δ_i	$F(i)$
⋮	⋮	⋮	⋮
编号 M	α_M	δ_M	$F(M)$

图 2 有序星点集星图识别数据库

根据 1.1 节对有序星点集星图识别模式的定义,星角距存在重复现象。因此,为了降低星图识别数据库的存储量,每个星点的有序星点集模式特征 $F(i)$ 的存储格式如图 3 所示。其中, k 表示主星点 s 的最近邻星点数量, $D_i (i = 1, \dots, k)$ 表示每颗最近邻星点的信息, 包含星编号、到主星点的角距 e_{0i} 和到下一颗近邻星点的角距 $e_{i,i+1}$ 。

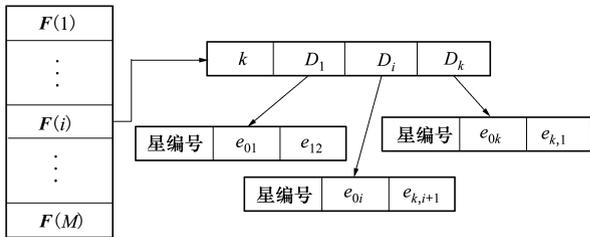


图 3 单星点有序星点集识别模式存储格式

2 局部敏感哈希快速搜索算法

星图识别的本质是在星图识别数据库中搜索匹配星图中构造的识别模式,由于导航星数量众多且每个星图识别模式中包含多个星点特征,因此需要设计快速搜索算法提高星图识别速度。此外,由于实际星图中获得的识别模式存在误差,因此可以把星图识别模式在星图识别数据库内的搜索匹配过程认为是最近邻搜索问题。本文提出适用于星图识别数据库快速搜索的局部敏感哈希算法,可以在存在误差的情况下,快速搜索匹配星图识别模式,进而识别出导航星,提高星图识别速度。

2.1 星图识别局部敏感哈希映射原理

有序星点集识别模式是由星角距组合形成的,因此实际星图中有序星点集星图识别模式的误差也是由星角距误差引起的。所以星图识别的过程本质

又可以理解为用带有误差的角距与理想角距的匹配过程。在匹配过程中,对于任意星点 s_i 和 s_j , e_{ij} 表示理想角距值, e'_{ij} 表示星图中计算得到的带有误差的角距值,则存在角距误差 Δ , 当 $e'_{ij} \in [e_{ij} - \Delta, e_{ij} + \Delta]$ 时,认为该角距匹配成功。为方便构建哈希函数,本文以角距误差 Δ 为基准,对星图识别模式中角距进行量化。定义

$$\begin{cases} n = \lfloor \frac{e'_{ij}}{\Delta} \rfloor \\ n = \lfloor \frac{e_{ij}}{\Delta} \rfloor \end{cases} \quad (1)$$

式中, n' 和 n 分别表示实际角距与理想角距对应的量化值,则当 $n' \in \{n - 1, n, n + 1\}$ 时,就可以认为该星角距匹配成功。假设有 N 个角距同时匹配成功,则在星图识别数据库中就会存在有 3^N 个星角距量化值组合判定星图识别模式匹配成功。

以 3 个角距为例,如图 4 所示。假设 3 个理想角

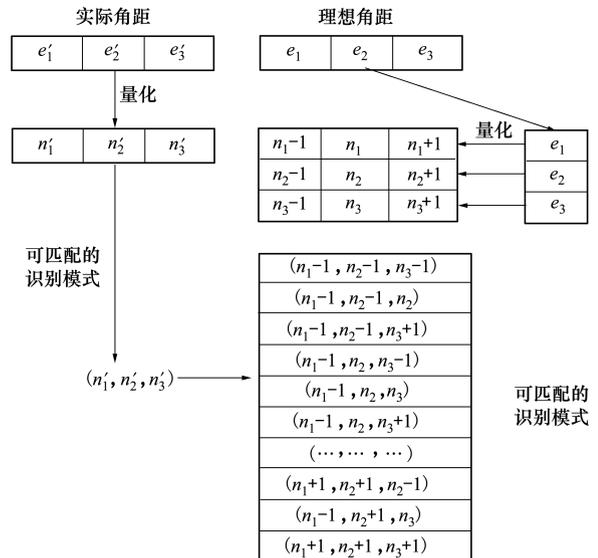


图 4 局部敏感哈希映射原理示意图

距分别是 e_1, e_2 和 e_3 , 在实际星图中计算得到的对应角距分别为 e'_1, e'_2 和 e'_3 。以角距误差 Δ 为基准,实际角距对应的量化数值为 n'_1, n'_2 和 n'_3 , 则组成的待匹配的识别模式为 (n'_1, n'_2, n'_3) 。理想角距对应的量化数值为 n_1, n_2 和 n_3 , 按照局部敏感哈希映射原理,当

$$\begin{aligned} & (n'_1 \in \{n_1 - 1, n_1, n_1 + 1\}) \\ & \cap (n'_2 \in \{n_2 - 1, n_2, n_2 + 1\}) \end{aligned}$$

$$\cap (n'_3 \in \{n_3 - 1, n_3, n_3 + 1\})$$

时,可以认为该识别模式匹配成功。因此,对于理想角距而言,一共可以产生 27 个组合形成的识别模式,只要待匹配的模式与其中任意一个匹配上,就可以识别模式匹配成功。

2.2 哈希表建立过程

根据所需要识别的角距数量的不同,可以把星图识别数据库中的星图识别模式组合成 3^N 个可以匹配的识别模式。图 5 给出了利用局部敏感哈希原理对星图识别数据库内识别模式的哈希表建立过程,具体的实现过程如下:

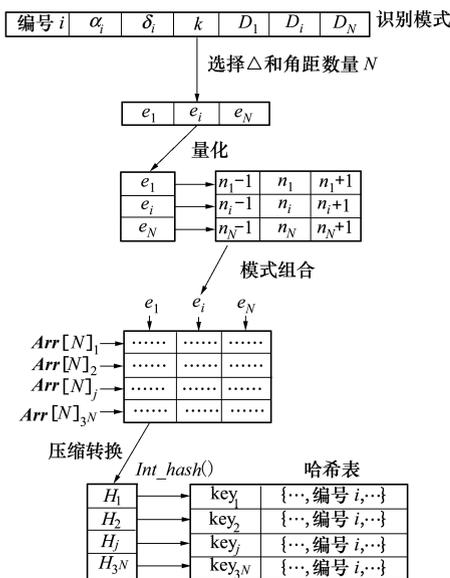


图 5 星图识别数据库哈希表建立过程

1) 选择星图识别的角距误差限 Δ 和需要识别的角距数量 N ;

2) 在星图识别模式中,选择需要量化的 N 个角距,并且按照公式(1)进行量化,得到量化后的角距值 $n_1, \dots, n_i, \dots, n_N$;

3) 把每一个角距量化值集合 $\{n_i - 1, n_i, n_i + 1\}$ 按照角距选择顺序排列组合,然后以 16 位形式存储到数组 $Arr[N]_i$ 中;

4) 为了压缩存储空间,把 $Arr[N]_i$ 中的数利用 CRC32 转换为 32 位整数 H_i

5) 采用整数哈希函数对 H_i 进行散列,每个哈希表 key 值对应的 value 是有序星点集模式的中心星点对应的星编号的集合,即

$$\begin{cases} key = Int_hash(H_i) \\ value = set(Star_Number) \end{cases} \quad (2)$$

在角距量化的过程中,每一个角距通过公式(1)转换为一个 16 位的整数,在选择 N 个量化角距的情况下,量化后的识别模式需要占用的存储空间为 $16 \times N$ 位,因此为了压缩存储空间,本文采用了 CRC32 编码的方式,把量化后的识别模式 $Arr[N]_i$ 转换成 32 位的整数 H_i 。然后,本文引用 STLport 5.2 版本中的整数哈希函数,把由识别模式转换得到的 32 位整数 H_i 散列到哈希表中。

按照上述哈希表的建立过程,对每个导航星对应的星图识别模式逐个转换,就可以建立星图识别数据库对应的哈希表。

2.3 哈希表搜索过程

建立了星图识别数据库以及对应的哈希表之后,就可以进行星图识别。在得到实际星图后,通过星点质心定位算法确定星图中恒星的位置,构造每个星点的有序星点集识别模式,然后按照局部敏感哈希映射原理,得到有序星点集识别模式对应的哈希值,在星图识别数据库内搜索匹配,完成星图识别。具体的哈希表搜索过程如图 6 所示。

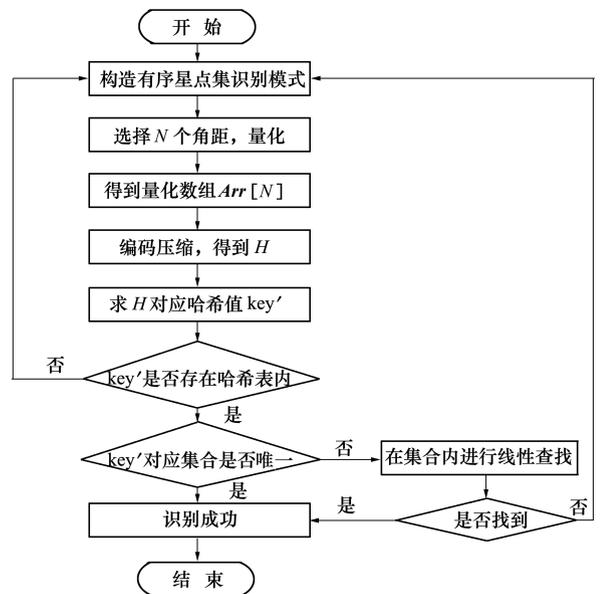


图 6 哈希表搜索过程

具体的搜索过程可以表示为:

1) 在星图中构造有序星点集模式,并选择其中 N 个角距作为搜索特征;

2) 以 Δ 为基准量化 N 个选定角距,然后按照顺序存入对应的数组 $Arr[N]'$ 中;

3) 把 $Arr[N]'$ 利用 CRC32 方式编码压缩,得到对应的 32 位整数 H' ;

4) 采用建立哈希表时使用的整数哈希函数对 H' 进行散列, 得到对应的哈希值 key' ;

5) 查找导航星数据库哈希表, 是否存在与 key' 对应的哈希值。若不存在, 则识别失败。如果存在, 检查此哈希值对应的星编号集合是否唯一, 如果唯一, 则星图识别成功; 如果不唯一, 则在对应的集合内进行线性搜索, 进行星图识别。

在建立的星图识别数据库哈希表中, 每一个星图识别模式都存在多个哈希值。通过建立这种“一对多”的星图识别方式, 可以在星图识别模式存在误差的情况下, 快速匹配星图识别模式, 识别出导航星。

3 仿真实验结果与分析

3.1 仿真条件

仿真实验中选择 Sky2000 星表中亮度高于 7.0 等星作为导航星, 共有 11 003 颗导航星, 建立有序星点集识别模式的半径为 $r=5^\circ$, 仿真计算机的 CPU 为 Intel Core i7-4770MQ, 仿真环境为 Microsoft VS2008, 其他仿真参数如表 1 所示。

表 1 仿真实验参数

仿真参数	定义值
视场角/ $^\circ$	$12^\circ \times 12^\circ$
像平面分辨率	1 024×1 024
像元尺寸/mm	0.015

通过蒙特卡罗方法随机生成 1 000 个视场指向并仿真星图。根据分析可知, 角距误差限 Δ 和需要识别的角距数量 N 是影响哈希表最主要因素。根据仿真条件可知, 单个像元对应的角距为:

$$e_{\text{pixel}} = \frac{12^\circ}{1\ 024} = 0.0117^\circ \quad (3)$$

仿真中以像素对应的角距为基准进行量化。在哈希表中, 如果哈希值对应的主星点编号不唯一, 定义此哈希值存在冲突, 定义存在冲突的哈希值数量与占总哈希值数量的比值为冲突率。

3.2 仿真结果

星角距误差限和选择匹配的星角距数量是影响本文算法的主要参数。利用本文提出的算法, 对仿真生成的 1 000 幅星图进行识别, 可以得到在不同角距误差限和选用不同角距匹配数量情况下的结果如表 2~表 4 所示。

表 2 $\Delta = e_{\text{pixel}}$ 条件下不同角距数量的仿真结果

角距数量	冲突率/%	最大冲突数	评价查找次数	占用内存/kB	平均星图识别时间/ μs
1	100	94	66.837 2	34	48
2	36.07	8	1.498 7	235	22
3	0.72	3	1.007 4	904	22
4	0.06	3	1.000 8	2 726	23

表 3 $\Delta = 2e_{\text{pixel}}$ 条件下不同角距数量的仿真结果

角距数量	冲突率/%	最大冲突数	评价查找次数	占用内存/kB	平均星图识别时间/ μs
1	100	113	66.837 2	34	57
2	56.53	12	2.186 7	193	33
3	2.72	4	1.028 1	892	28
4	0.24	3	1.002 5	2 723	32

表 4 $\Delta = 3e_{\text{pixel}}$ 条件下不同角距数量的仿真结果

角距数量	冲突率/%	最大冲突数	评价查找次数	占用内存/kB	平均星图识别时间/ μs
1	96.84	159	68.158 1	34	66
2	68.58	18	3.533 6	158	31
3	8.20	5	1.089 4	859	30
4	0.91	4	1.009 3	2 711	29

通过仿真结果可以看出:

1) 在同样星角距误差限的情况时, 选用星角距的数量为 $N=1$ 和 $N=2$ 时, 虽然哈希表占用存储空间比较低, 但是哈希表的冲突率比较高, 最大冲突数量也比较高。在 $N=3$ 和 $N=4$ 时, 哈希表的冲突比较低, 每个哈希值最大冲突数量为 3, 平均查找次数基本接近与 1, 但哈希表占用存储空间比较大;

2) 在选择同样匹配星角距数量情况下, 随着星角距误差限的增大, 哈希表冲突率、最大冲突数量、平均查找次数增多、哈希表占用的存储空间以及平均星图识别虽然都是呈现递增的趋势, 但是影响不显著。

3.3 仿真结果分析

通过仿真结果可以看出, 角距误差限 Δ 在不同

的情况下,对搜索速度结果影响不显著。因此本文提出算法可以提高星图识别对角距误差限的容忍能力和鲁棒性。在选择角距数量 $N = 1$ 和 $N = 2$ 时,由于存在比较高的冲突率和最大冲突数量,不适合选择建立哈希表。在 $N = 4$ 的情况下,哈希表的冲突率和最大冲突数量都是最低的,但对应的哈希表占用存储空间也非常大。虽然在 $N = 3$ 的情况下建立哈希表的冲突率相对于 $N = 4$ 的情况下略高,但是最大冲突数、平均查找次数以及平均星图识别时间都非常接近,而且哈希表占用的存储空间仅为 $N = 4$ 的情况下的30%,因此选择 $N = 3$ 是比较适合建立导航星快速搜索哈希表。

本文提出的局部敏感哈希搜索算法的时间复杂度为 $O(1)$,与传统哈希算法相同。但是传统算法建立的是星图识别模式中角距对应的哈希表,因此对整个星图识别模式匹配过程,至少需要完成多个角距的搜索匹配。本文提出算法则需要完成一次搜索就可以完成星图识别模式的匹配。虽然本文提出算法在时间复杂度上与传统哈希算法一样都是 $O(1)$,但是在星图识别过程中,本文提出算法的识别速度优于传统哈希算法。在星图识别模式中需要匹配角距数量 $N = 3$ 的情况下,本文搜索算法与传统哈希算法的对比结果如表5所示。

表5 哈希搜索算法性能对比分析

搜索算法	时间复杂度	评价识别时间/ μs	哈希表存储空间/kB
传统哈希	$O(1)$	74	679
本文	$O(1)$	22	904

参考文献:

- [1] 张广军. 星图识别[M]. 北京:国防工业出版社,2011
Zhang Guangjun. Star Identification[M]. Beijing, National Defense Industry Press, 2011 (in Chinese)
- [2] Junkins J L, White W, Turner D. Star Pattern Identification for Real Time Attitude Determination[J]. Journal of Guidance, Control and Dynamics, 1977, 25: 251-270
- [3] Mortari D, Samaan M A, Bruccoleri C. The Pyramid Star Identification Technique[J]. Navigation, 2004, 51(3): 171-184
- [4] Padgett C, Delgado K K. A Grid Algorithm for Star Identification[J]. IEEE Trans on Aerospace and Electronics Systems, 1997, 33(1): 202-213
- [5] Liebe C C. Pattern Recognition of Star Constellations for Spacecraft Application[J]. IEEE Aeronautics Electronic System Magazine, 1992, 7(6): 34-41
- [6] Mortari D, Neta B. K-Vector Range Searching Techniques[J]. Adv Astronaut Sci, 2000, 105: 449-464
- [7] Zhang Guangjun, Wei Xinguo, Jiang Jie. Full-Sky Autonomous Star Identification Based on Radial and Cyclic Features of Star

4 结 论

星传感器是以恒星为探测目标的高精度姿态测量设备,具备独立自主工作能力强和可靠性高等有点,已经广泛应用在各类航天器。其中星图识别是影响星传感器获取初始姿态速度的关键步骤。基于传统哈希算法的星图识别算法是目前已有算法中识别速度最快的,为进一步提高星图识别速度,本文提出基于局部敏感哈希的快速星图识别算法,在考虑星图识别误差的影响情况下,建立星角距的局部敏感映射关系,以星图识别角距误差限为基准,量化星图识别模式中角距,利用整数哈希函数建立“一对多”的星图识别模式哈希表,可以通过哈希表直接快速匹配星图识别模式,提高星图识别速度。总体上讲,本文提出的基于局部敏感哈希的快速星图识别算法具有以下特点:

1) 虽然本文提出算法的时间复杂度与传统哈希算法相同,但是在需要匹配同样星角距数量情况下,本文提出算法的识别速度比传统哈希算法快3.4倍左右,哈希表占用内存量约为传统哈希算法1.3倍;

2) 本文提出快速星图识别算法在识别速度方面有很好的鲁棒性,受到星角距误差影响比较小;

3) 考虑工程应用的情况,可以选择星角距误差限为1个像素对应角距,角距数量 $N = 3$,此时星图识别过程中哈希表的冲突率为0.74%,平均搜索次数为1.0074,星图平均识别时间22 μs ,哈希表存储容量904 kB。

- Pattern[J]. Image Vision Computing, 2008, 26 (7): 891-897
- [8] Jiang Jie, Ji Feilong, Yan Jinyun, et al. Redundant-Coded Radial and Neighbor Star Pattern Identification Algorithm[J]. IEEE Trans on Aerospace and Electronic Systems, 2015, 51(4): 2811-2822
- [9] 梁斌, 杨君, 宋靖雁, 等. 一种快速搜索导航星表的方法[R]. CN101995248A, 2011
Liang Bin, Yang Jun, Song Jingyan, et al. A Fast Method for Guide Star Search[R]. CN101995248A, 2011 (in Chinese)
- [10] 张磊, 何昕, 魏仲慧, 等. 三角形星图识别算法的改进[J]. 光学精密工程, 2010, 18(2): 458-463
Zhang Lei, He Xi, Wei Zhonghui, et al. Modification of Triangle Identification Algorithm[J]. Opt Precision Eng, 2010, 18 (2): 458-463 (in Chinese)
- [11] 熊雪, 王庆. 基于多视场星敏感器的三角形星图识别方法[J]. 计算机测量与控制, 2014, 22(1): 225-228
Xiong Xue, Wang Qing. An Improved Triangle Star Identification Method Based on Multi-FOV Star Sensor[J]. Computer Measurement & Control, 2014, 22(1): 225-228 (in Chinese)
- [12] Rao G N, Bhat M S, Alex T K. Fast Access Method for Onboard Star Catalog[J]. Journal of Guidance Control & Dynamics, 2005, 28(5): 1032-1037
- [13] 郭磊. 纳型星敏感器星图识别算法优化研究[D]. 北京: 中国科学院国家空间科学中心, 2017
Guo Lei. Research on Optimization of Star Pattern Recognition Algorithm for Nano Star Tracker[D]. Beijing, National Space Science Center, University of Chinese Academy of Sciences, 2017 (in Chinese)

Fast Access for Star Catalog Based on Locality-Sensitive Hashing

Zhu Hailong, Liang Bin, Zhang Tao

(Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: In order to improve the access speed and robustness of star catalog database during star identification, an algorithm based on locality-sensitive hashing is proposed. First, according to principle of star identification, the angle distances are quantified on the basis of angle distance error limit, and the order star set pattern is transformed into an array of integers, which has locally sensitive hash characteristics. Then key of hash obtain by hashing the array of integers with Stlport, and the value of hashing is a set of central star number in the ordered star point set pattern. Numerical simulation results indicate that the time complexity of proposed algorithm is $O(1)$, which is much better than direct search, binary search and k-vector search technology. In addition, the proposed algorithm is robustness due to the affect is not significant as the performance influenced by angle distance error limit. Considering practical application, the error limit of angle distance could be chose as 1 pixel, the number of quantified angle distances could be chosen as 3. Under this condition, the collision rate in hashing table is 0.74%, the average searching time is 1.007 4 and the average consuming time is 22 μ s during star identification.

Keywords: star identification; ordered star points set; locality-sensitive hashing; quantized star angle distance; error limit of angle distance; design of experiments