

MVDT-SI: 一种多视图双三角形星图识别算法

孙立健 周鋈

国防科技大学信息系统工程重点实验室

摘要: 近年来, 三角形算法因其简单方便而成为应用最为广泛的恒星识别算法, 其中星等信息在星图特征的构建中起着关键的作用。然而, 在实际应用中, 观测星图的星等信息往往很难使用, 因为它们在某些最坏的情况下可能包含误差或丢失。为了解决这一问题, 本文提出了一种多视图双三角形星图识别算法。该算法利用星点的角度和距离信息构造恒星的双三角形特征。此外, 为了减少噪声干扰对模型辨识精度的影响, 我们对观测星图建立了多视图双三角形特征, 提高了算法的鲁棒性。综合实验和实际实验表明, 该算法对“假星”噪声和“缺星”噪声具有 98.4% 以上的识别精度, 且不受星敏传感器焦距和拍摄角度的影响。实验结果还表明, 该算法具有鲁棒性好、辨识时间短、存储成本低等优点, 具有一定的实用价值。

关键字: 多视图; 双三角形; 星图识别

自主天文导航, 又称天文导航, 是一种独立的、无条件的、全球性的自主导航系统, 在世界许多国家得到了广泛的应用[1]。星敏传感器/跟踪器是天体导航系统[2]的主要组成部分, 是利用光电池或照相机测量恒星姿态信息的光学测量装置。星敏传感器的主要特征包括: 星质心估计、星识别和姿态计算。恒星质心的估计方法有很多, 如[3-5], 它们都是用来计算恒星质心的。这些方法对提高星敏传感器姿态测量精度具有重要意义。在这里, 最具挑战性的部分是恒星识别, 它创建了观测到的恒星和编目的恒星之间的对应关系[6]。

星敏传感器的识别通常包括两个步骤: 1) 导星目录的制作; 2) 识别算法的设计与实现, 影响星敏传感器的姿态识别精度和姿态输出速度。因此, 一个好的恒星识别算法是非常重要的。

由于不同应用背景的差异, 各种星敏传感器的具体指标存在较大差异, 目前还没有统一的、公认的星敏传感器识别评价标准。然而, 如前几篇参考文献所述[6-8], 一个好的星图识别算法通常具有识别精度高、识别时间短、存储空间小(星型数据库 SPD)等优点。

传统的星图识别算法通常采用角距离及其导数形式来提取观测星图的特征, 这种方法相对简单, 且通常需要较大的存储空间。而且其识别精度普遍较低, 不利于实际应用。

此外, 传统的恒星识别算法对观测星图中的大部分信息进行处理, 如星心位

置和星等信息，以提高识别精度。然而，当星等信息丢失时，这些算法大多无法识别观测到的星图，或者鲁棒性差[9,10]。因此，有必要探索一种不基于观测星图星等信息的更可靠的恒星识别算法。

为了解决这些问题，我们提出了一种新的星识别算法多视图双三角形星图识别算法(MVDT-SI)，该算法具有较好的鲁棒性、适用性和灵活性。本文的创新点概括如下：

新的星图识别算法不依赖于观测星图的星等信息；

该算法构造了多视图双三角形特征来识别恒星，有效地提高了算法的识别精度和鲁棒性。

新的恒星识别算法不受星敏感器焦距和拍摄角度的影响。

一、相关工作

星敏感器测量飞行器姿态主要有两个步骤：质心估计和星识别。第一步——质心估计——已经得到了很好的研究。例如，Xin[5]提出的星质心位置估计算法将系统误差降低到 10^{-7} 像素的 3.0 倍以下。因此，本文着重研究第二步——恒星识别，这仍然是一个挑战。

星识别算法一般分为两类：基于子图同构的算法和基于模式的算法。

三角形算法是一种重要的基于子图同构的算法。为了解决人工恒星识别效率低、易出错的问题，Junkins[11]于 20 世纪 70 年代提出了经典的三角形特征识别算法。然后，Van[12]根据探测到的恒星的大小给每颗恒星分配一个概率，从而选择最可靠的三角形。随后，Mortari[13]提出了金字塔识别算法，该算法引入了一个星来构造四面体来解决“假星”问题。基于子图同构的算法也发展了一种基于角距离计数的投票策略[14]。2018 年，Wang 等人[15]使用哈希映射加速金字塔识别算法。金字塔识别算法对位置噪声具有很强的鲁棒性，但当虚假星数增加而真星数不足时，该算法可能不会收敛。

网格算法和神经网络算法是两种典型的基于模式的算法。网格算法是 Padgett 在 20 世纪 90 年代[16]首次提出的。该算法不需要通过各种组合构造三角形，而是将星图转换成网格形状，并利用特征矩阵进行匹配。2007 年，张等人[17]针对网格算法的不足，提出了一种基于径向和循环特征的改进算法。为了克服分步匹配策略的速度问题，Zhao 等人提出了用 K-L 变换形成粗匹配，并用星走过程进行精确识别星体[18]。与三角形算法相比，网格算法具有存储量小、识别速度快的优点。

近年来，随着人工智能的发展，神经网络[19]等先进算法（如遗传算法[20]）被应用于恒星识别。本文将神经网络的思想引入到[21]的恒星识别算法中。在该算法中，Hong 等人构造特征向量对网络进行星识别训练。神经网络算法具有辨识精度高、辨识时间短的优点。然而，神经网络的训练时间通常很长，这是目前最大的问题之一。在 2019 年，最近的一项工作[22]提出了一种基于表示学习的星型识别网络 RPNNet，通过简化网络结构来减少模型训练时间。与基于模式的算

法相比，三角形算法以其简单性得到了广泛的应用和实现。

值得注意的是，在构造星图特征时，上述星图识别算法都选择了观测星图的星心位置和星图的星等信息。这些利用星等信息选择参考星或识别参考星的算法更容易失败，因为观测星图和导航星表中的恒星亮度顺序。[23,24]中的研究还表明，恒星的星等信息是一个模糊的特征，应加以避免。另外，当观测星图中的恒星亮度发生畸变或丢失时，将无法准确获得观测星图的星等信息。此时，上述恒星识别算法将难以准确识别观测到的星图。

为了解决这一问题，其他学者根据恒星之间的距离（恒星与观测星图中心的距离）或扫描技术选择参考星。2001 年，莫塔里等. 提出了不使用星等信息的金字塔算法[25]。但是，通过扫描三角形组合来查找唯一的三角形需要额外的内存。为了实现快速识别，专门开发了低内存 SPD[26]、优化 SPD（SPOD）[27]等星识别算法。另外，科洛门金等. 提出了一种基于几何投票方案的几何投票算法（GMV），如果两对恒星之间的角距离相似，则目录中的一对恒星将投票给观测星图中的一对恒星。2018 年，德瓦尔等. 提出了一种星 ID 短名单（SISS）[29]的星识别算法，该算法在识别可靠性和速度上都有明显的优势。然而，当观测到的星图中存在大量的假/缺星时，这些算法的性能将大大降低。

为了解决这一问题，本文提出了一种不考虑观测星图的星等信息的改进算法。该算法通过构造观测星图的多视图双三角形特征，具有良好的识别精度和鲁棒性。

二、基本星表

星表是基于星敏感器的恒星识别中不可缺少的一部分。它可以用来建立恒星三角特征数据库，用于恒星识别。

基本星表包含上万颗恒星，每颗恒星都包括恒星的位置、亮度、光谱等参数。然而，由于星敏传感器只能探测到一定亮度范围内的恒星，因此需要对星敏感器的基本星表进行处理。星敏传感器中使用的恒星参数主要包括恒星的位置（赤经、赤纬坐标）和亮度。我们可以根据基本星表中的极限星等提取所需的参数，为星敏传感器创建一个特定的星表。这个特定的目录通常被称为指南星目录。

为了创建引导星目录，从基本星目录中选择亮度大于（小于）极限星等的星。在基本星表中选择的星称为导星。假设星敏感器的极限星等为 6mv，则选择亮度大于（或等于）6mv 的 4908 颗恒星组成导星目录。在获得导星星表后，通过比较观测星图与导星星表的差异，可以将观测星图与导星星表进行匹配。

三、星识别的多视图双三角形算法（MVDT-SI）

3.1 坐标变换

天球是一个半径无限大的球体，恒星在天球中的位置由导星目录中的赤经和

赤纬坐标决定。因此，识别前需要将导星投射到星敏感器上。当天球坐标系转换为星敏感器坐标系时，它们的原点可以认为是重合的。因此，导星在单位球面切平面上的投影可以看作是星敏感器上的映射。

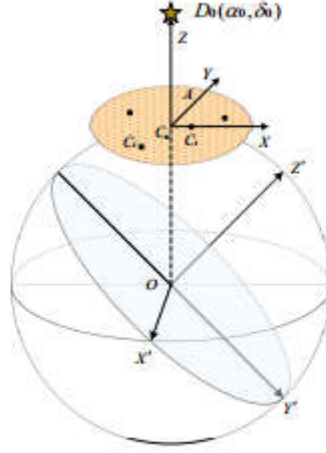


Figure 1. The coordinate transformation from the celestial sphere to the two-dimensional plane

3.2 天球到二维平面的坐标变换

假设 $D_0(\alpha_0, \delta_0)$ 是天球上的一颗引导星， C_0 是单位球上 D_0 的投影点，其中 α_0, δ_0 分别是 D_0 恒星的赤经和赤经坐标。在单位球面坐标系中， C_0 点 (α_0, δ_0) 可以表示为：

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \cos \alpha_0 \cos \delta_0 \\ \sin \alpha_0 \cos \delta_0 \\ \sin \delta_0 \end{pmatrix}$$

在这种情况下，单位球面在 C_0 点的切平面可以表示为：

$$\cos \alpha_0 \cos \delta_0 x + \sin \alpha_0 \cos \delta_0 y + \sin \delta_0 z = 1$$

其中 (x, y, z) 是相切平面上的任意点。对于任何星 $D_i(\alpha_i, \delta_i)$ ，除了指南星表中的 D_0 ，可以从球体 O 的中心到 D_i 绘制光线。参数方程如下：

当光线不平行于切线平面时，光线和切线平面的交点 C_i 可以通过公式 1 和 3 来计算。此时，交点 C_i 是导星 D_0 在以 C_0 为中心的切面上的投影点。

当所有的恒星都投影到切面上时，切面上的恒星数目可以根据星敏感器的视场（FOV）来调整。让 S 作为星敏感器的视场

在相切平面上，以 D_0 为中心，可以生成半径为 $L/\sqrt{2}$ 的圆。此时，圆中所有的投影星点都会生成集合 C 。显然， C 中包含的恒星都是星敏感器中的导星，其中心为参考星 D_0 ，视场为 S 。

为了使参考星点 C_0 和集合 C 在一个唯一的坐标系中，我们在导星目录中用导星定义了平面坐标系。也就是说，我们选择了 C_1 （这是最接近 C 中 C_0 的点）作为辅助星来建立坐标系 Z 轴。

集合 C 中的星点可以在新的坐标系 XYZ 中表示。在这个坐标系中， Z 轴上所有星点的坐标总是 0。换句话说，集合 C 中的所有恒星都投影到 C_0C_1 二维平面上。

最后，以 $\overline{C_0C_1}$ 为单位长度，对平面上所有恒星的坐标进行了标准化。

对于指南星表中的每个参考星 D ，我们可以计算 FOVS 中星点的坐标。因此，我们可以根据不同的参考星，得到所有导航星在星敏感器上的投影图。

3.3 观测星图的旋转不变性

在对星敏感器捕捉到的星图进行去噪（质心估计等）后，由于星敏感器的拍摄角度不同，星图上的星点往往具有不同的坐标。为了保证算法不受星敏感器拍摄角度（旋转不变性）的影响，通过从观测星图中选取恒星来定义平面坐标系。

首先从观测星图中选取一颗恒星作为参考星，然后在参考星附近找到辅助星。接下来，我们可以通过由参考星和辅助星组成的向量构造 X -轴，并通过其逆时针正交向量构造 Y -轴。定义了坐标轴和单位长度后，星图具有旋转不变性。同时，当确定参考星和辅助星时，观测到的星图是唯一的。这个过程可以总结如下：

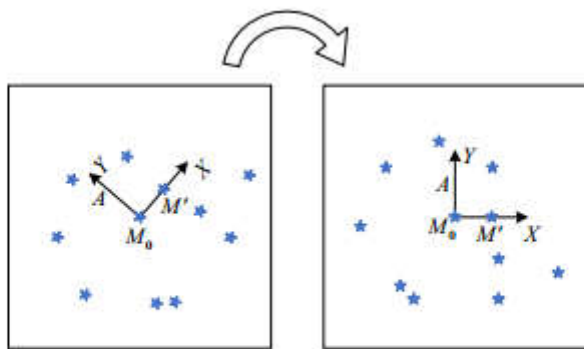


Figure 2. The establishment of the coordinate system of the observed star map.

我们在观察到的星图中随机选择了一颗恒星，命名为参考星 M_0 ，然后根据一定的规则选择了辅助星 M' （如节 4.3 所示）；

我们以 $\overrightarrow{M_0M'}$ 矢量为 X 轴，其逆时针正交矢量 $\overrightarrow{M_0A}$ 为 Y 轴建立新的坐标系；项目观测星图中所有恒星的坐标用新的坐标系表示；我们以矢量 $\overrightarrow{M_0M'}$ 为单位长度，对新坐标系中所有恒星的坐标进行了标准化。

利用上述方法，我们可以得到以参考星 M_0 为中心，以 M' 为辅助星的观测星图的唯一坐标。当选择不同的参考星和辅助星时，可以得到不同视图下观测星图的坐标表示，从而可以从多视图生成观测星图特征。

3.3 观测星图的多视图双三角形特征

在实际的工程应用中，星敏感器捕捉到的观测星图的星等信息往往是丢失或噪声的。因此，我们的 MVDT-SI 算法在构造特征时排除了星等信息，使用了观测星图中恒星质心的坐标信息。

在对导星目录中的导星和观测星图中的观测星进行预处理后（如第 1 节和第 2 节所述），确定参考星和辅助星意味着每个观测视图中的恒星的单位长度和坐标是唯一的。因此，在一个观察视图中，每个三角形的三个角和边可以被构造成一个固定的矢量，因此我们可以利用三角形来构造观察视图的特征。

首先，我们随机选取一颗恒星作为观测星图中的参考星 M_0 ，并计算出参考星 M_0 与观测星图中其他恒星之间的距离。接下来，我们选择最近的四个点，依次是 M_1 、 M_2 、 M_3 、 M_4 ，并利用这五个点构造多视图的双三角形特征。

当选取不同的辅助星时，可以构造多组三角形来描述观测星图的特征。因此，我们在多个视图中构造三角形特征，以减少噪声点对观测星图的影响。

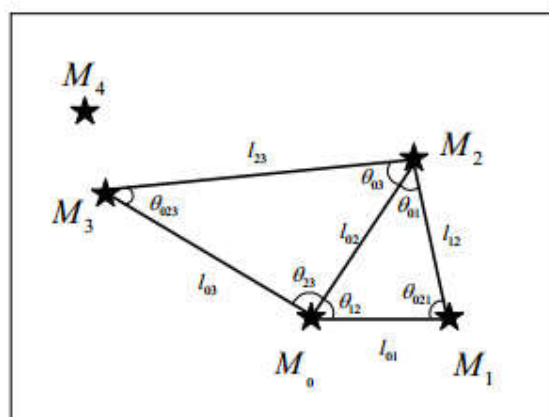


Figure 3. Illustration of the double-triangle feature of the observed star map in the first view. Four stars located at the two-dimensional coordinate system form the double triangle feature. Each vertex of the triangle is a star, and each edge of the triangle is the distance between two stars in the two-dimensional plane.

形特征。由于导星目录中没有噪声点，我们在第一个视图中构造了星特征。通过计算和排序从二维平面上的每个星点到参考星 D_0 的距离，我们可以从参考星 C_0 中找到最近的三个点 C_1, C_2, C_3 （ C_0 是星敏感器上 D_0 的投影点）。

$H = [\theta_{01}, \theta_{021}, \theta_{12}, \theta_{023}, \theta_{03}, \theta_{23}, l_{01}, l_{02}, l_{12}, l_{02}, l_{03}, l_{23}]$ 。因此，当指南星表中有 K 颗恒星时，我们可以得到双三角形特征数据库 $\{H_1, H_2, \dots, H_K\}$ 。

3.4 MVDT-SI 算法流程

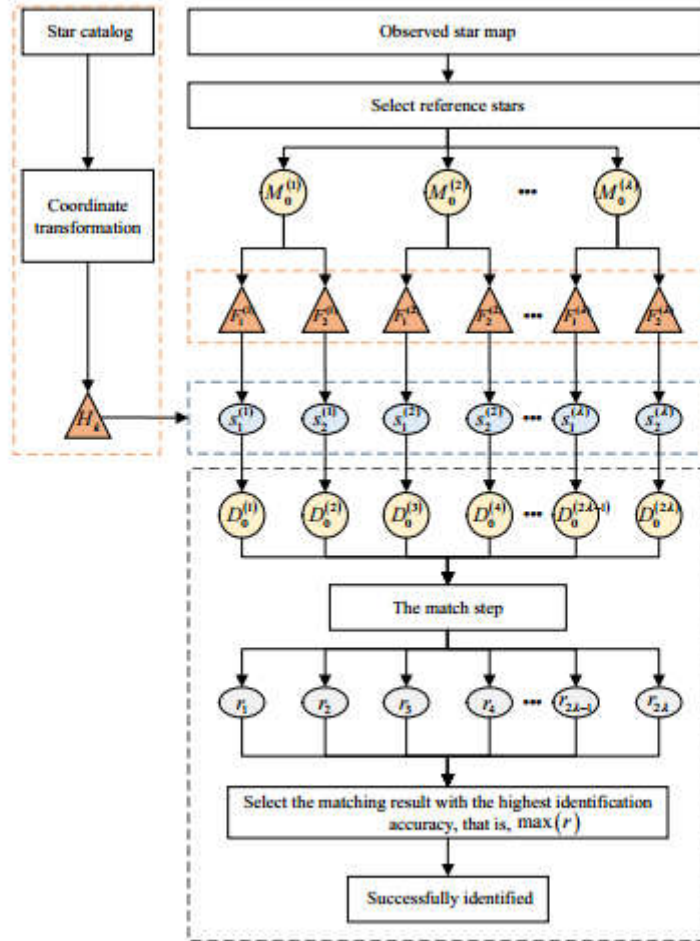


Figure 5. Multi-View Double-Triangle Algorithm for Star Identification Flow.

多视图双三角形恒星识别流算法。

假设观测星图上有 N 个恒星点， λ 是所选参考星的最大数目。如果 $N < 4$ ，则无法识别观测到的恒星；如果 $4 \leq N \leq \lambda$ ，则使用观测星图上的一个一个的星点作为参考星来构造双三角形特征；如果 $N > \lambda$ ，则随机选择 λ 星点作为参考星，

构造双三角形特征，即集合参考星的数目是 $\{M_0^{(1)}, M_0^{(2)}, \dots, M_0^{(\lambda)}\}$ 。

第二步是确定观测图中的参考星。对于第 i -th 个参考星 $M_0^{(i)} (i=1, 2, \dots, \lambda)$ ，它们的双三角形特征 $F_j^{(i)} (j=1, 2)$ 是基于不同的辅助星 $M_1^{(i)}$ 和 $M_2^{(i)}$ 分别构造的。换句话说，每个引用星 $M_0^{(i)}$ 都具有双视图的特性。然后，我们将观察到的星图的不同视图上的特征信息 $F_j^{(i)}$ 与特征数据库 $H_k (k=1, 2, \dots, K)$ 中的特征 $\{H_1, H_2, \dots, H_K\}$ 进行了比较。换言之，我们计算了它们特征之间的差异之和，也就是说，

$$s_j^{(i)} = \min_{k=1, 2, \dots, K} \left(\sum_{c=1}^6 (|\Delta\theta_c| + \Delta l_c) \right), j=1, 2$$

为了降低 $s_j^{(i)}$ 的计算复杂度，我们只在特征数据库 $\{H_1, H_2, \dots, H_K\}$ 中选择 K_s 星进行匹配。将选择满足以下条件的 K_s 星：

$$l_{H_{k_s}} - l_{F_r} < \min(\min_{k=1, 2, \dots, K} (l_{H_k} - l_{F_r}) * 200, 50), \quad k_s = 1, 2, \dots, K_s$$

$$l = \sum_{c=1}^6 l_c$$

其中 l_{F_r} 是观测星图中参考星的边长特征之和， $\{l_{H_k}\}$ 是特征数据库中第 k 颗星的边长特征之和， $l_{H_{k_s}}$ 是特征数据库中第 k_s 颗星的边长特征之和。

$\min_{k=1, 2, \dots, K} (l_{H_k} - l_{F_r})$ 表示观察到的星图中的参考星的边长度与特征数据库 $\{H_1, H_2, \dots, H_K\}$ 中的星的边长度之和之间的最小值。经过选择， K_s 仅为 K （指南星目录号）的 3% 左右。此时，所选的特征数据库为 $\{H_1, H_2, \dots, H_{K_s}\}$ ，并按如下方式计算 $s_j^{(i)}$ ：

$$s_j^{(i)} = \min_{k=1, 2, \dots, K_s} \left(\sum_{c=1}^6 (|\Delta\theta_c| + \Delta l_c) \right), j=1, 2$$

接下来，我们分别选择导星目录中差异最小的 $s_j^{(i)}$ 对应星号作为观测星图中

的 $M_0^{(i)}$ 星号。 $\{D_0^{(1)}, D_0^{(2)}, \dots, D_0^{(2\lambda)}\}$ 。

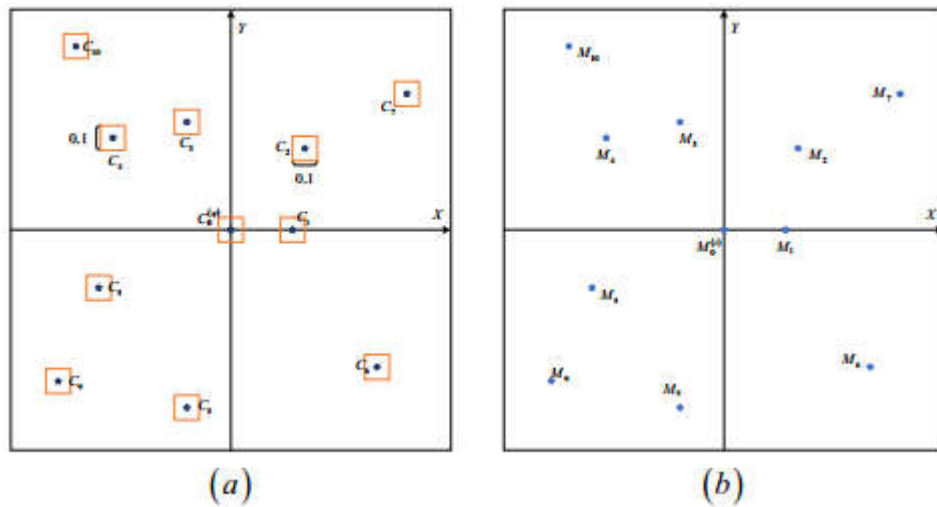


Figure 6. Star point number matching in the observed star map.

第三步是根据参考星识别观测图中的所有星点。基于第二步，我们可以得到观测星图中的参考星 $M_0^{(i)}$ 的个数，它对应于导星目录中的星 $D_0^{(q)}$ （投影点是 $C_0^{(q)}, q=1, 2, \dots, 2\lambda$ ）。因此，通过对 3.1 和 3.2 节两部分的操作，可以得到导星图和观测星图的标准化星图。首先，我们创建了一个盒子（大小为 0.1×0.1 ），其中中心是导航星图中的恒星。接下来，我们确定观测到的星图中的恒星是否在这些盒子里。如果观测星图上的一个星点落入导星图的方框中，则该星点将标有相应的导星号。如果星点同时落入多个方框中，则用方框内最小星等对应的导星号标记该点。否则，它被标记为-1。

最后，计算了不同视图下不同星点的识别结果。恒星识别结果 r 的统计值每成功一次增加 1。换言之，恒星识别结果 r 是观测地图上未标记为-1 的恒星点的数目。根据参考星在不同视图下的识别结果，选择识别最成功的结果作为观测星图中的最终星点数。

四、实验

4.1 实验设置

本文提出了一种基于多视点双三角形的恒星识别算法，并将其命名为 MVDT-SI。最新的恒星识别算法声称提供了高鲁棒性和快速识别的实现。我们比较了一种具有星 ID 短名单 (SISS) [29]、基于优化数据库搜索树的高速算法 (STOD) [27]、一种具有高鲁棒性的几何投票算法 (GMV) [28] 和我们提出的算法 MVDT-SI。此外，我们还将 MVDT-SI 算法与以下算法设置进行了比较：

单视图单三角形恒星识别算法 (SVST-SI)：该算法利用参考星和辅助星的第

一个视图中的单个三角形来构造观测星图的特征。换言之，SVST-SI 算法使用由 $\{M_0, M_1, M_2\}$ 组成的三角形来构造 $\lambda = 1$ 的参考星的特征。

星敏感器的姿态识别算法是在现实世界中用来捕获星敏感器的姿态信息，其识别时间短，识别精度高。此外，由于星敏感器的存储容量有限，星敏感器的识别算法必须考虑空间复杂度。

另外，在实际观测到的星图中总是存在着许多干扰。因此，在一定的干扰条件下，用星识别算法来衡量系统的鲁棒性。一般来说，干扰可以分为干扰噪声和干扰星。

干涉噪声是指恒星的位置噪声和星等噪声。星点位置偏差主要来源于星敏感器的标定误差（如焦距测量误差、透镜畸变、光轴偏差误差等）和星点位置算法的误差。星等噪声反映了星敏感器对恒星亮度的敏感性。

干扰星有两种，一种是“假星”，如行星、星云尘埃、空间碎片等，从观测星图上很难区分其成像目标和普通星点目标。此外，由于星敏感器对星等的分辨率有限，一些亮度较弱的恒星也可能被捕获，但它们无法从导星目录中找到相应的匹配。另一种干扰星是“缺星”，即本应被捕获的恒星由于某种原因没有出现在观测图上。

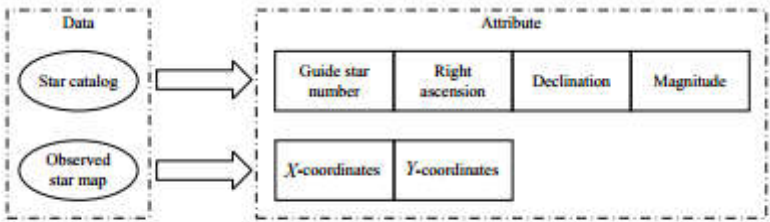


Figure 7. Data attributes of the guide star catalog and the observed star map.

在这个实验中，所有的算法都在一个时钟频率为 2.5ghz 的桌面处理器上测试。为了验证算法的性能，我们选择了公开可用的星标识数据集,用于 2019 年中国研究生数学建模竞赛。数据集由两部分组成。第一部分是预定义的星载星表，其星等阈值为 6mv。这个星表中有 4908 颗恒星，每颗都包含四个属性：“引导星号”、“赤经”、“决定”和“星等”。数据集的第二部分在质心估计之后提供了四个真实的观测星图。观测到的星图的属性是恒星质心的坐标：`\$X\$-坐标"和`\$Y\$-坐标'。图 1 显示了导航星表中使用的 MVDT-SI 算法的数据属性，以及在恒星识别过程中观察到的星图。

4.2 仿真星图实验

根据 3.1 节，我们可以计算出导星目录中所有恒星在星敏感器上的投影。此时，我们可以得到投影星图中每颗恒星的坐标。在上述投影星图的基础上，随机选取 100 个星图，加入噪声，验证了算法的性能。为了考虑模拟图中恒星数目对算法的影响，我们分别在 $12^\circ \times 12^\circ$ 和 $16^\circ \times 16^\circ$ 的视场下进行了验证。

同时，为了验证算法的鲁棒性，证明算法不受星敏感器焦距和星图拍摄角度的影响，我们在上述投影星图的基础上，通过以下步骤生成了 100 幅模拟星图。

1) 我们随机地将模拟星图上的恒星坐标放大或缩小 1-5 倍。当星敏感器的焦距不同时，观测到的星图中的恒星之间的距离也会不同。因此，我们通过模拟这个现象；

2) 我们随机丢弃模拟星图上的 φ 星点以模拟“缺星”的干扰，随机生成模拟星图上的 σ 星点以模拟“假星”的干扰。完成上述步骤后，我们将模拟地图中的“假星”点标记为“1”。

3) 我们随机旋转模拟星图 $0^\circ \sim 360^\circ$ ，以模拟星敏感器的不同拍摄角度。

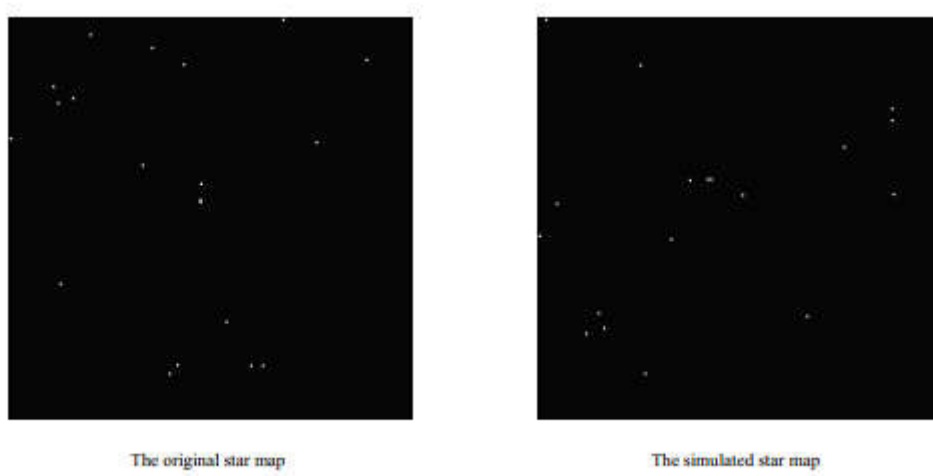


Figure 8. The explanation for the generation of the simulated map.

在理想情况下，观测到的星图不会出现假星/缺星的问题（ $\sigma = 0$ 和 $\varphi = 0$ ）。

表 1 和表 2 总结了在具有不同 FOV 值的理想情况下，星标识算法的性能。

Table 1. Benchmarking of star identification algorithms in the ideal case when the FOV is $12^\circ \times 12^\circ$

Technique	Identification accuracy (%)	Identification-time (s)	Storage size (MB)
GMV [28]	92.97	0.758	1.31
STOD [27]	91.26	0.415	1.40
SISS [29]	99.08	0.420	1.70
SVST-SI	27.72	0.031	0.23
MVDT-SI($\lambda = 10$)	99.51	0.102	0.44
MVDT-SI($\lambda = N$)	99.75	0.285	0.44

从表 1 的结果可以看出，MVDT-SI（ $\lambda = N$ ）算法的识别精度优于比较的星识别算法。此外，SVST-SI 算法速度最快，但其识别精度不可接受。同时，我们注意到，当我们将 MVDT-S 算法的超参数 λ 调整为 10 时，其精度仅比 $\lambda = N$ 降低了 0.24% 和 1.27%，但其识别时间明显缩短。当 $\lambda = 10$ 时，MVDT-SI 算法仅需 0.11s 就能达到 98% 以上的识别精度。如果我们需要更高的识别精度来满足现实世界的需要，可以通过进一步增加 λ 的值来实现 MVDT-SI 算法。与其他三种算

法相比，MVDT-SI ($\lambda = N$) 算法速度最快。虽然在实际应用中，存储算法达到 2 MB 在片上存储器中不是问题，但是存储大小仍然是一个重要的考虑因素。每个算法的导星目录特性的存储大小显示在表 1 和表 2 中。与 GMV、STOD 和 SISS 算法相比，MVDT-SI 算法需要的存储空间最小。

Table 2. Benchmarking of star identification algorithms in the ideal case when the FOV is $16^\circ \times 16^\circ$

Technique	Identification accuracy (%)	Identification-time (s)	Storage size (MB)
GMV [28]	94.73	0.767	1.31
STOD [27]	91.88	0.412	1.40
SISS [29]	99.57	0.501	1.70
SVST-SI	17.36	0.034	0.23
MVDT-SI($\lambda = 10$)	98.56	0.112	0.44
MVDT-SI($\lambda = N$)	99.83	0.483	0.44

在本文中，观测到的星图中总是存在“假星”或“缺星”。“假星”或“缺星”情况下的星识别算法性能如图 1 所示。如果观测到的星图中有假星或缺星，GMV 和 STOD 将彻底失效。在这种情况下，基于邻近恒星与参考星的欧几里德距离投票的 GMV 失败，因为由于遗漏了邻近恒星，参考星的投票计数减少。STOD 也失败了，因为它搜索的树数据库的参数是相邻恒星的数目。在这种情况下，SISS 算法和我们的算法保持了很高的鲁棒性。可以看出，我们的 MVDT-SI ($\lambda = N$) 算法具有更强的算法鲁棒性。

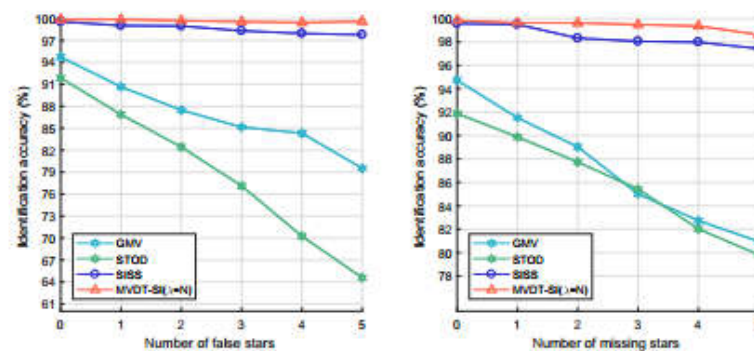


Figure 9. Performance of star identification algorithms in the scenario of “false stars” or “missing stars” when the FOV is $16^\circ \times 16^\circ$.

在某些情况下，观测到的地图中可能同时存在假星和缺星。从表 3 可以看出，当同时存在多个假星和缺星时，MVDT-SI 算法 ($\lambda = N$) 仍能达到 98.47% 以上的识别精度。当只有缺星或假星 ($\sigma = 0$ 或 $\varphi = 0$) 时，在任意视场 ($\{12^\circ\}$ 倍 $\{12^\circ\}$ 或 $\{16^\circ\}$ 倍 $\{16^\circ\}$) 下，MVDT-SI ($\lambda = N$) 算法的识别精度可达 99.56% 以上，表明该算法具有良好的鲁棒性。

Table 3. Identification accuracy of MVDT-SI($\lambda = N$) algorithm in the scenario of “false stars” and “missing stars” with different values of the FOV ($12^\circ \times 12^\circ$ and $16^\circ \times 16^\circ$)

φ	0	1	2	0	1	2
σ	$12^\circ \times 12^\circ$			$16^\circ \times 16^\circ$		
0	99.75%	99.67%	99.64%	99.83%	99.63%	99.62%
1	99.88%	99.72%	99.52%	99.88%	99.69%	99.67%
2	99.56%	98.54%	98.47%	99.70%	99.43%	99.44%

4.3 真实星图实验

为了进一步验证算法的有效性，我们在上述数据集的基础上选取了四个真实的星图进行星识别。这四张星图是从真实的夜空数据中提取的。我们只知道每个观测星图中星点的质心坐标。前两个星图记录的星敏感器视场为 $12^\circ \times 12^\circ$ ，像素数为 512×512 ；后两个星图记录的星敏感器视场为 $20^\circ \times 20^\circ$ ，像素数为 1024×1024 。为了使算法的结果可视化，我们对星点的质心坐标进行了旋转和标准化。星号识别结果如图所示。

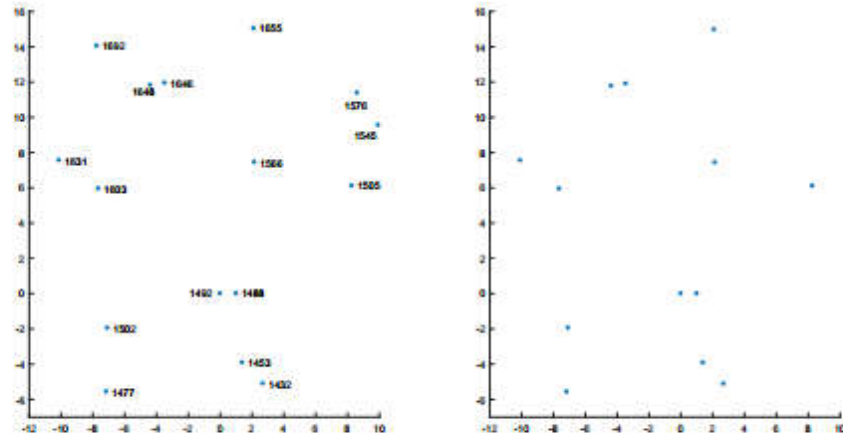


Figure 10. The identification results of our MVDT-SI algorithm on the first real star map. The right image is the real observed map of star sensor after standardization, and the left image is the identification results (star numbers) of our MVDT-SI algorithm in the guide star catalog.

我们的 MVDT-SI 算法在第一个真实星图上的识别结果。右图是星敏感器标准化后的真实观测图，左图是导星目录中 MVDT-SI 算法的识别结果（星号）

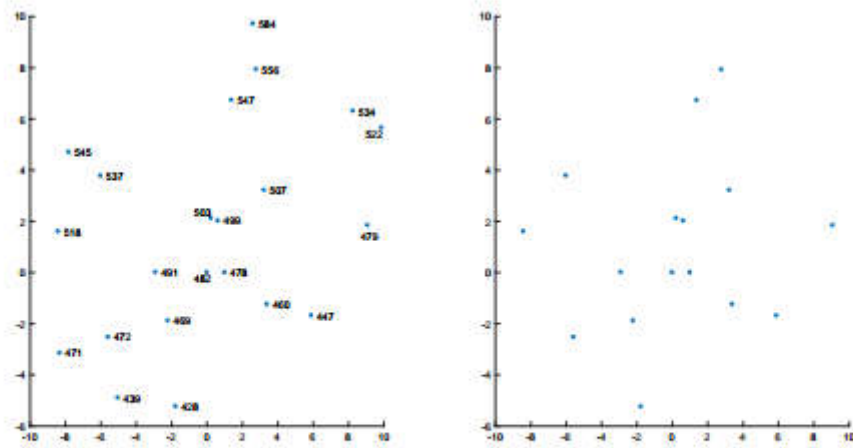


Figure 11. The identification results of our MVDT-SI algorithm on the second real star map. The right image is the real observed map of star sensor after standardization, and the left image is the identification results (star numbers) of our MVDT-SI algorithm in the guide star catalog.

我们的 MVDT-SI 算法在第二个实星图上的识别结果。右图是星敏感器标准化后的真实观测图，左图是导星目录中 MVDT-SI 算法的识别结果（星号）

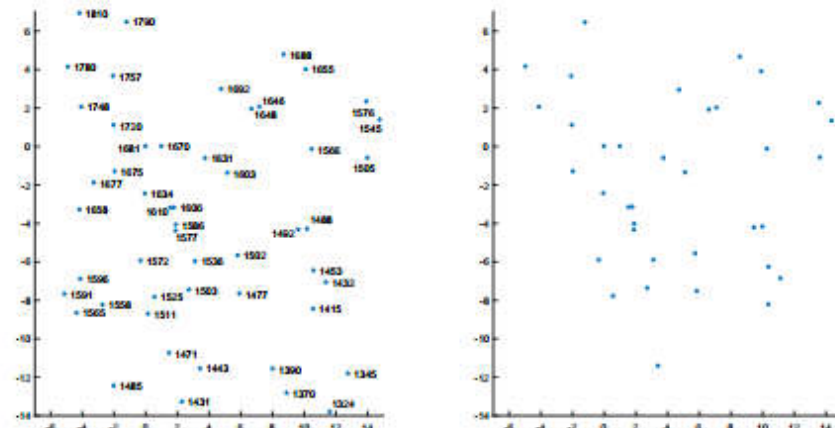


Figure 12. The identification results of our MVDT-SI algorithm on the third real star map. The right image is the real observed map of star sensor after standardization, and the left image is the identification results (star numbers) of our MVDT-SI algorithm in the guide star catalog.

我们的 MVDT-SI 算法在第三个实星图上的识别结果。右图是星敏感器标准化后的真实观测图，左图是导星目录中 MVDT-SI 算法的识别结果（星号）

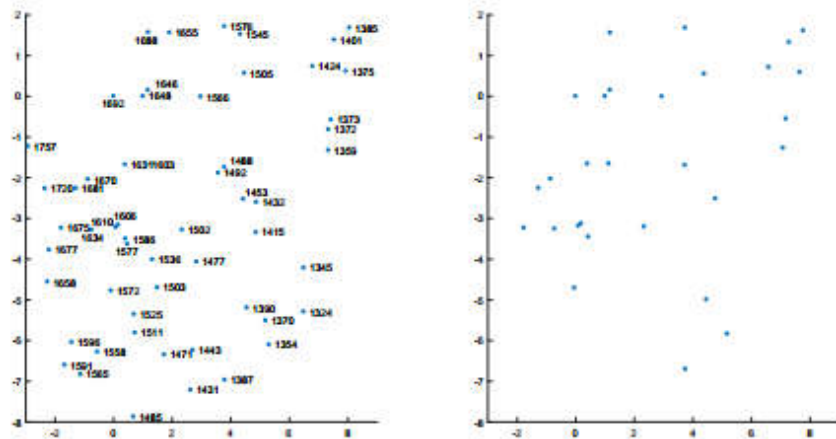


Figure 13. The identification results of our MVDT-SI algorithm on the fourth real star map. The right image is the real observed map of star sensor after standardization, and the left image is the identification results (star numbers) of our MVDT-SI algorithm in the guide star catalog.

我们的 MVDT-SI 算法在第四个实星图上的识别结果。右图是星敏传感器标准化后的真实观测图，左图是导星目录中 MVDT-SI 算法的识别结果（星号）

我们可以看到，即使在四个观测星图中有许多噪声点，我们的 MVDT-SI 算法仍然可以成功地完成识别。

综上所述，实验结果表明，我们的 MVDT-SI 算法在不需要合成星图和真实星图的星等信息的情况下，成功地识别出观测星图，具有良好的鲁棒性。与其他星型识别算法相比，MVDT-SI 算法具有识别精度高、识别时间短、存储空间小等优点。

五、结论

本文提出了一种多视图双三角形恒星识别算法。当星敏传感器在“丢失空间 (LIS)”模式下工作，丢失观测星图的星等信息时，MVDT-SI 算法仍能克服“假星”和“缺星”问题，在短时间内识别出观测星图。此外，值得注意的是，当观测星图中的恒星数目非常小 ($N < 10$) 且观测星图中同时存在大量的假/缺星 ($> 50\%$) 时，我们的 MVDT-SI 算法的性能可能会下降。为了克服这一缺点，应该增加星敏传感器的视场。

MVDT-SI 算法包含四个步骤。首先，对导星目录的坐标系进行变换，得到不同参考星下各导星的二维坐标表示。然后，为了保证观测星图的坐标和特征不受星敏传感器拍摄角度的影响，定义了确定观测星图坐标系的规则。然后，构造了观测星图的多视图双三角形特征和导星目录的导星特征。最后，通过比较它们的特征差异，我们得到了观测星图上每个星点的数目。

实验结果表明，该算法在“假星”噪声和“缺星”噪声干扰下具有较好的鲁棒性和较短的识别时间。此外，实验结果表明，我们的 MVDT-SI 算法可以应用于真实的星图，因此可以在未来的自主导航系统中嵌入。

参考文献：

1. Wang, A.G. Modern celestial navigation and the key techniques. Tien Tzu Hsueh

Pao/Acta Electronica Sinica **2007**, 35, 2347–2353.

2. Liebe, C.C. Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems* **2002**, 38, 587–599. doi:10.1109/TAES.2002.1008988.

3. Xu, W.; Li, Q.; Jun Feng, H.; Hai Xu, Z.; Ting Chen, Y. A novel star image thresholding method for effective segmentation and centroid statistics. *Optik* **2013**, 124, 4673 – 4677. doi:<https://doi.org/10.1016/j.ijleo.2013.01.067>.

4. Ozkan, S.; Tola, E.; Soysal, M. Performance of star centroiding methods under near-real sensor artifact simulation. 2015 7th International Conference on Recent Advances in Space Technologies (RAST), 2015, pp. 615–619. doi:10.1109/RAST.2015.7208417.

5. Wei, X.; Wen, D.; Song, Z.; Xi, J.; Zhang, W.; Liu, G.; Li, Z. A Systematic Error Compensation Method Based on an Optimized Extreme Learning Machine for Star Sensor Image Centroid Estimation. *APPLIED SCIENCES-BASEL* **2019**, 9.

6. IV, B.B.S.; Mortari, D. A Survey on Star Identification Algorithms. *Algorithms* **2009**, 2, 93–107. doi:10.3390/a2010093.

7. Vogiatzis, C.; Camur, M.C. Identification of Essential Proteins Using Induced Stars in Protein-Protein Interaction Networks. *INFORMS Journal on Computing* **2019**, 31, 703–718. doi:10.1287/ijoc.2018.0872.

8. Mehta, D.S.; Chen, S.; Low, K.S. A Hamming Distance and Spearman Correlation Based Star Identification Algorithm. *IEEE Trans. Aerospace and Electronic Systems* **2019**, 55, 17–30. doi:10.1109/TAES.2018.2845198.

9. Accardo, D.; Rufino, G. Brightness-independent start-up routine for star trackers. *IEEE Transactions on Aerospace and Electronic Systems* **2002**, 38, 813–823.

10. Padgett, C.; Kreutz-Delgado, K.; Udomkesmalee, S. Evaluation of Star Identification Techniques. *Journal of Guidance, Control, and Dynamics* **1997**, 20, 259–267, [<https://doi.org/10.2514/2.4061>]. doi:10.2514/2.4061.

11. Junkins, J.; III, C.; Turner, J. Star pattern recognition for real time attitude determination. *Journal of The Astronautical Sciences - J ASTRONAUT SCI* **1977**, 25, 251–270.

12. der Heide, E.V.; Kruijff, M.; Avanzini, A.; Liedtke, V.; Karlovsky, A. Thermal Protection Testing of the Inflatable Capsule for YES2. 54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law, Bremen, Germany, 2003.

13. Mortari, D.; Samaan, M.; Bruccoleri, C.; Junkins, J. The Pyramid Star Identification Technique. *NAVIGATION* **2004**, 51. doi:10.1002/j.2161-4296.2004.tb00349.x.

14. Li, J.; Wei, X.; Zhang, G. Iterative algorithm for autonomous star identification. *IEEE Trans. Aerospace and Electronic Systems* **2015**, 51, 536–547. doi:10.1109/TAES.2014.130729.

15. Wang, G.; Li, J.; Wei, X. Star Identification Based on Hash Map. *IEEE Sensors Journal* **2018**.

16. Padgett, C.; Kreutz-Delgado, K. A grid algorithm for autonomous star identification. *IEEE Transactions on Aerospace and Electronic Systems* **1997**, 33, 202–213. doi:10.1109/7.570743.

17. Zhang, G.; Wei, X.; Jiang, J. Full-sky autonomous star identification based on radial and cyclic

features of star pattern. *Image Vision Comput.* **2008**, 26, 891–897. doi:10.1016/j.imavis.2007.10.006.

18. Zhao, Y.; Wei, X.; Li, J.; Wang, G. Star Identification Algorithm Based on KL Transformation and Star Walk Formation. *IEEE Sensors Journal* **2016**, 16, 5202–5210. doi:10.1109/JSEN.2016.2553245.

19. Zhang, G., Star Identification Utilizing Neural Networks; 2017. doi:10.1007/978-3-662-53783-1_5.

20. Paladugu, L.; Schoen, M.; Seisie-Amoasi, E.; Williams, B. Intelligent Star Pattern Recognition for Attitude Determination: the "Lost in Space" Problem. *Journal of Aerospace Computing Information and Communication - J AEROSP COMPUT INF COMMUN* **2006**, 3, 538–549. doi:10.2514/1.19398.

21. Hong, J.; Dickerson, J. Neural-Network-Based Autonomous Star Identification Algorithm. *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM* **2000**, 23, 728–735. doi:10.2514/2.4589.

22. Xu, L.; Jiang, J.; Liu, L. RpNet: A Representation Learning-Based Star Identification Algorithm. *IEEE Access* **2019**, 7, 92193–92202. doi:10.1109/ACCESS.2019.2927684.

23. Accardo, D.; Rufino, G. Brightness-independent start-up routine for star trackers. *IEEE Transactions on Aerospace and Electronic Systems* **2002**, 38, 813–823.

24. Sohrabi, S.; Shirazi, A.B. A novel, smart and fast searching method for star pattern recognition using star magnitudes. 2010 25th International Conference of Image and Vision Computing New Zealand. IEEE, 2010, pp. 1–6.

25. Mortari, D.; Junkins, J.L.; Samaan, M. Lost-in-space pyramid algorithm for robust star pattern recognition. *Guidance and control* 2001, 2001, pp. 49–68.

26. Needelman, D.D.; Alstad, J.P.; Lai, P.C.; Elmasri, H.M. Fast Access and Low Memory Star Pair Catalog for Star Pattern Identification. *Journal of Guidance, Control, and Dynamics* **2010**, 33, 1396–1403, [<https://doi.org/10.2514/1.48568>]. doi:10.2514/1.48568.

27. Pham, M.D.; Low, K.; Chen, S. An Autonomous Star Recognition Algorithm with Optimized Database. *IEEE Transactions on Aerospace and Electronic Systems* **2013**, 49, 1467–1475.

28. Kolomenkin, M.; Pollak, S.; Shimshoni, I.; Lindenbaum, M. Geometric voting algorithm for star trackers. *IEEE Transactions on Aerospace and Electronic Systems* **2008**, 44, 441–456.

29. Mehta, D.S.; Chen, S.; Low, K.S. A robust star identification algorithm with star shortlisting. *Advances in Space Research* **2018**, 61, 2647–2660