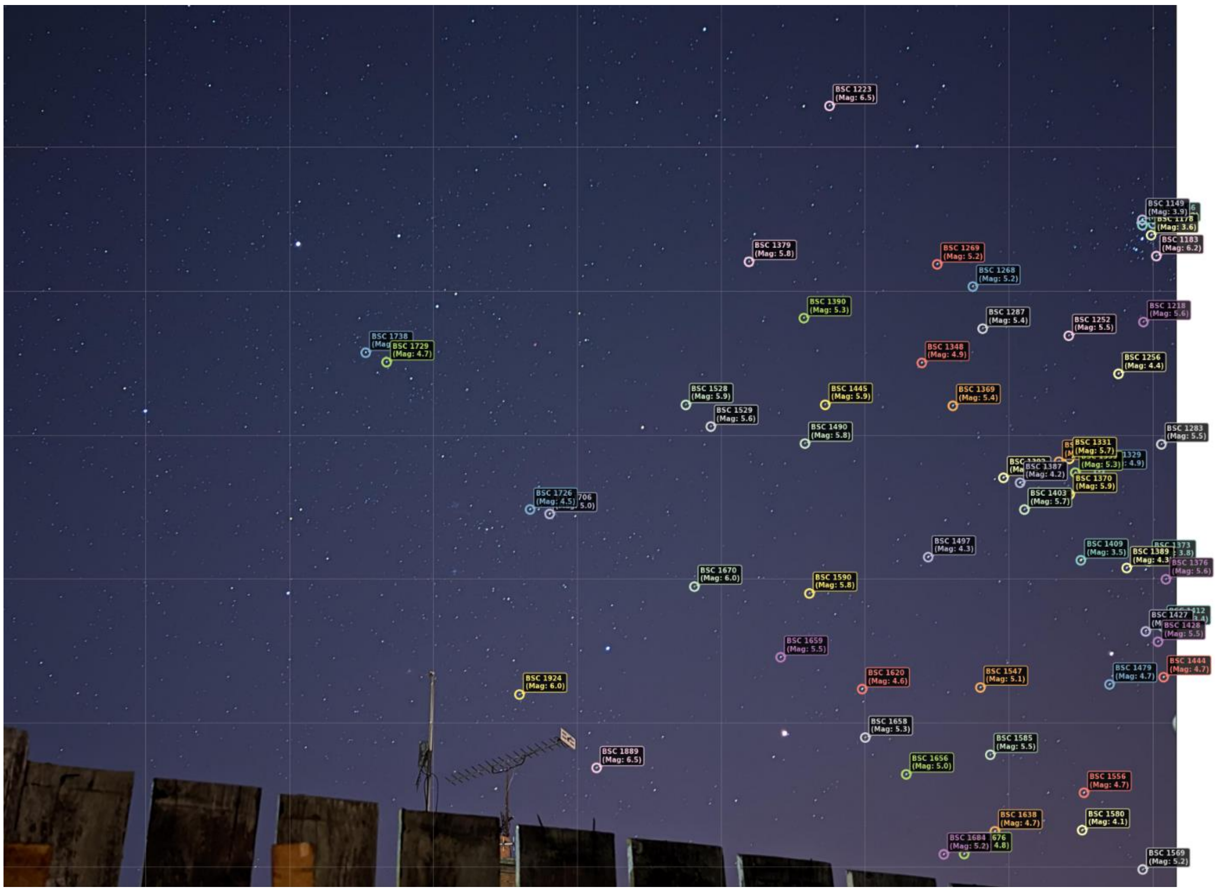
Introduction to Space Engineering – Project Document

By: Yair Jacob



Project Overview

A Python application to identify stars of magnitude <=7 in pictures of the night sky. The project builds upon a modified version of the tetra3 library, which is a fast lost-in-space plate solver - meaning it provides star identification capabilities without requiring any prior pointing information. The application serves as a practical implementation of cutting-edge star pattern recognition algorithms for educational and research purposes.

Lost-in-Space

The "lost-in-space" problem refers to the challenge of determining spacecraft or camera position with no prior knowledge - not even lens parameters such as field-of-view or distortions. This broad approach is essential for autonomous space navigation and ground-based astronomical applications where initial pointing information is unavailable or unreliable.

Program Features

Multi-Mode Operation

The implementation provides three distinct operational modes:

1. Single Image Analysis: Detailed star identification with comprehensive solution metrics
2. Batch Processing: Automated analysis of image collections with summary statistics
3. Cross-Image Matching: Comparative analysis identifying common stars between multiple images

Displaying The Results

The application generates comprehensible visualizations and statistics:

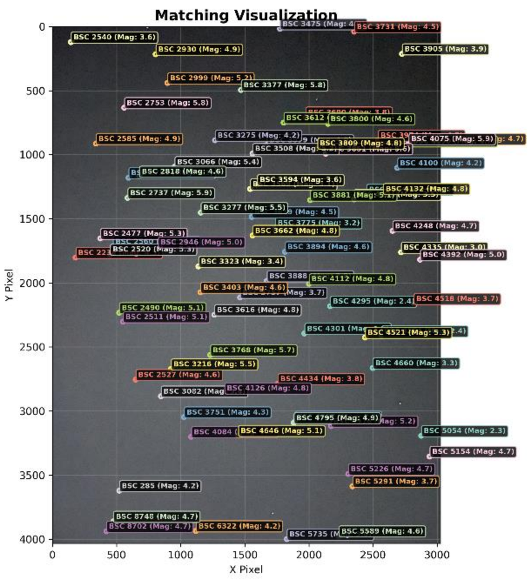
- Annotated Images: Labeled identification of matched stars with catalog information
- Solution Metrics: Comprehensive attitude, field-of-view, and accuracy data
- Composite Images: Aligned combinations of multiple images based on star pattern matching

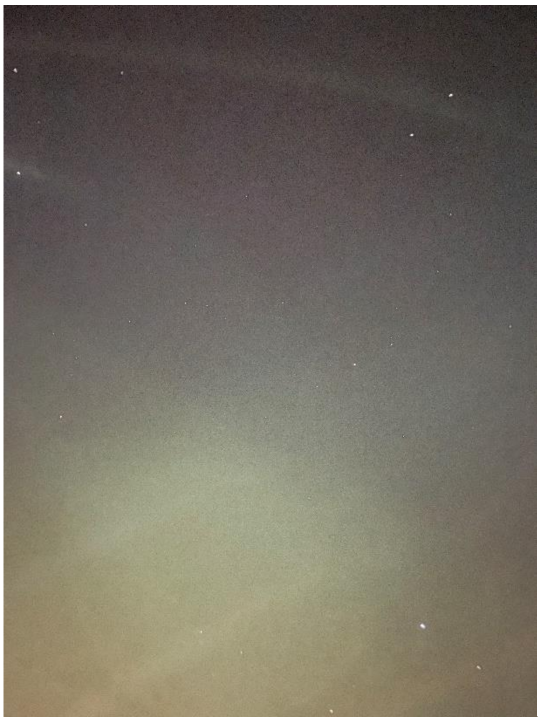
Performance

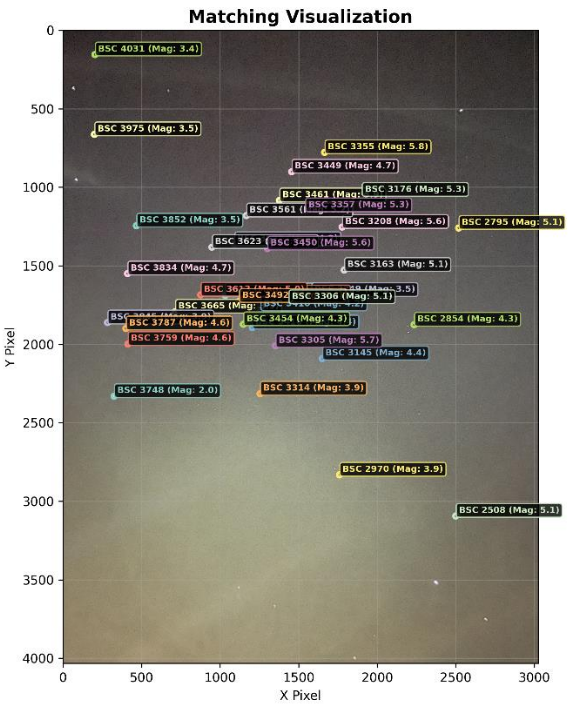
The program was tested on a diverse collection of 50 pictures of the night sky that were taken at different locations, with varying FOVs, quality, dimensions, noise patterns, and star counts. The performance of the program can be summarized as such: provided a night-sky picture taken at FOV between 1°-80° with multiple visible (bright) stars in it, the program has about 80% chance to (correctly) identify stars in the picture in about 0.5 seconds. astrometry.net was used to determine whether the program was correct in its identification of the stars in pictures.

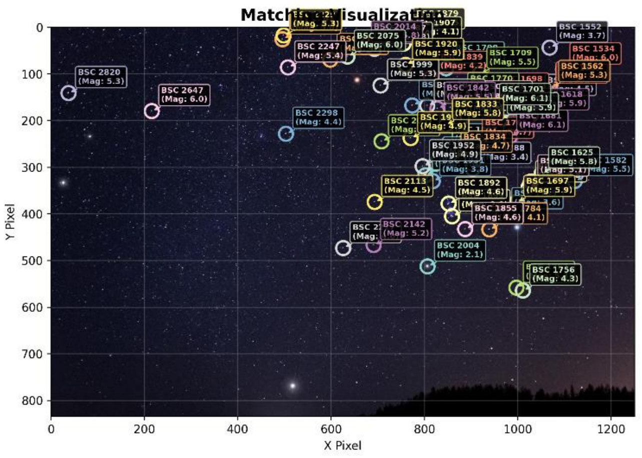
Visualization of a Few Results

Mode 1 (single picture):

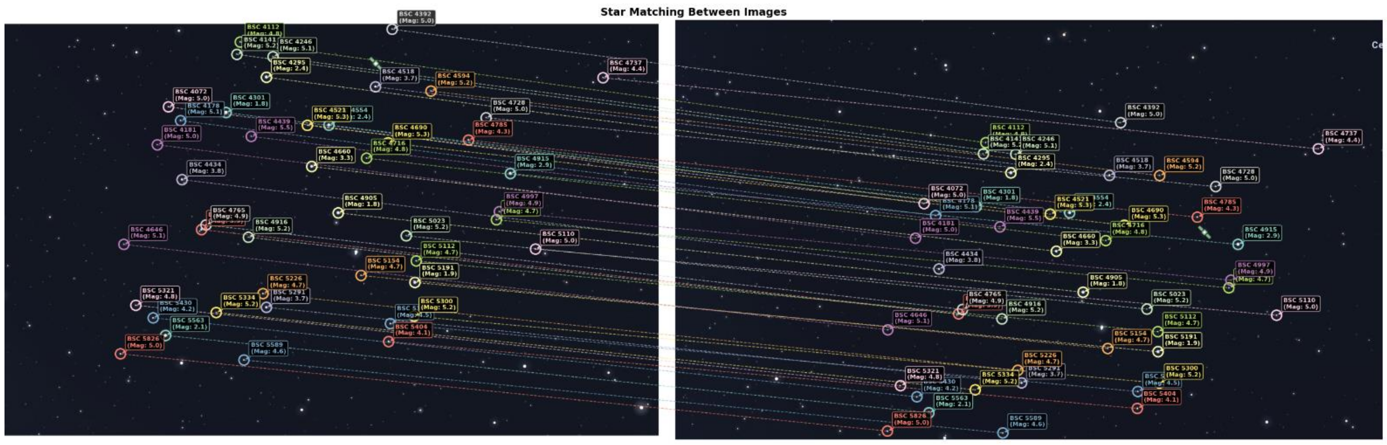


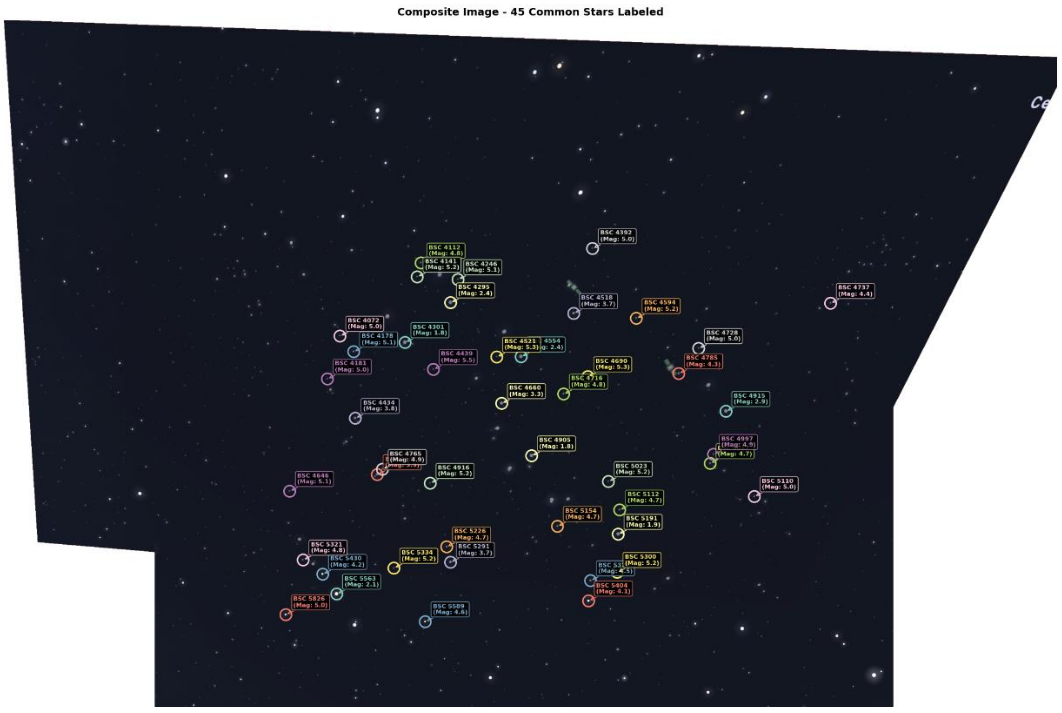




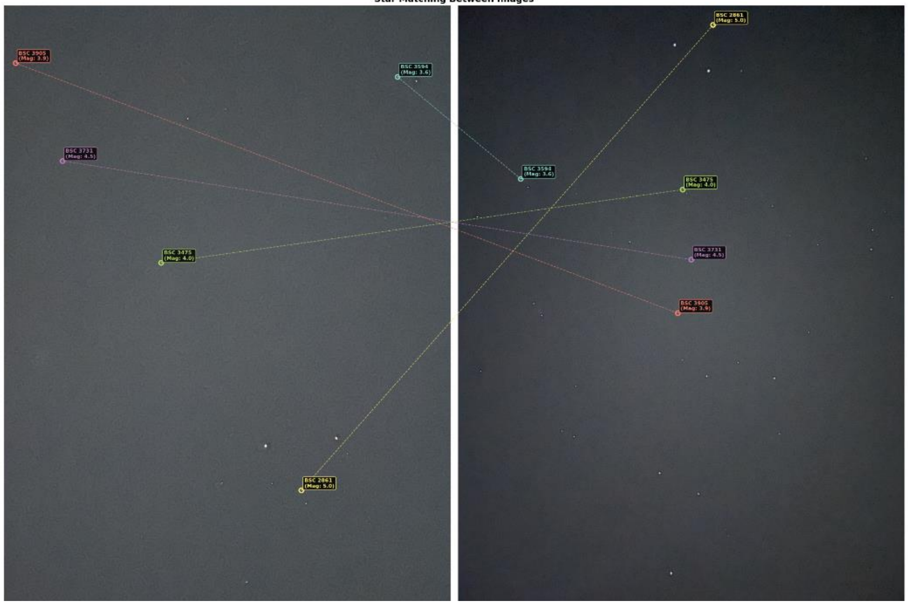


Mode 3 (matching between pictures):

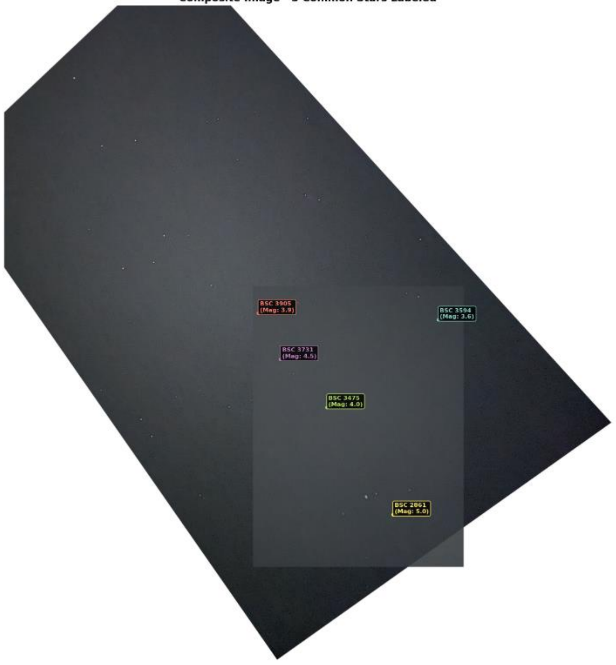




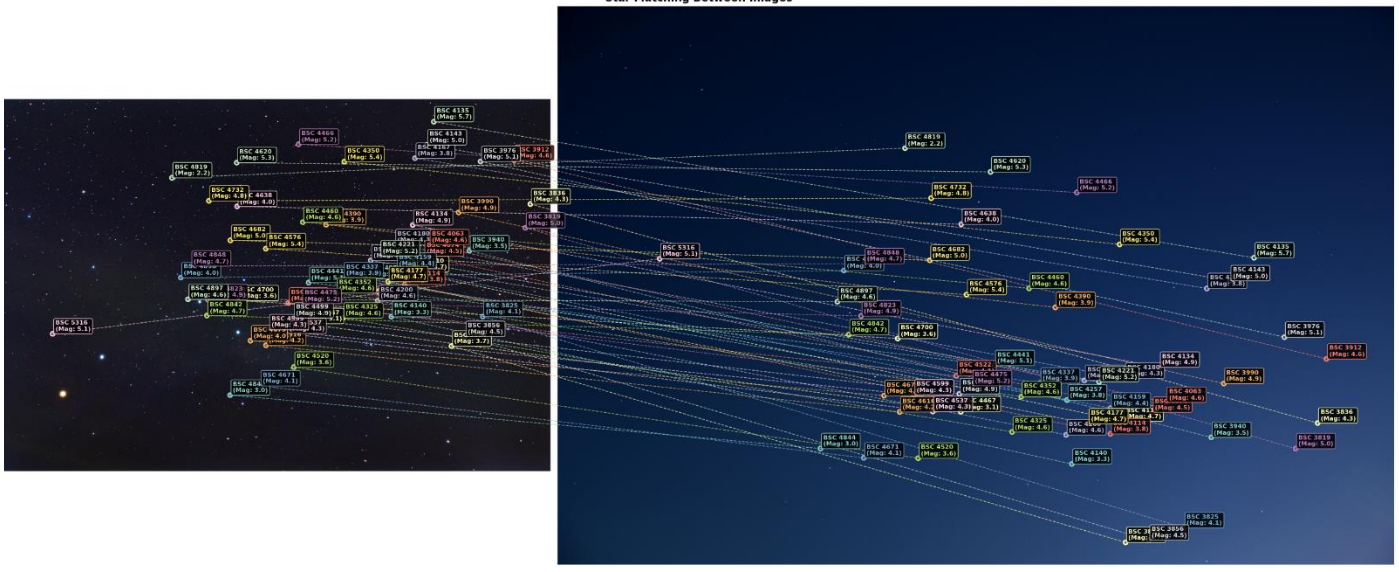
Star Matching Between Images



Composite Image - 5 Common Stars Labeled



Star Matching Between Images



The TETRA Algorithm

The underlying algorithm is based on TETRA (originally developed by Brown, Stubis, and Cahoy in 2017), which uses geometric hashing with hash tables to achieve fast computation times and access to operations with a database.

Geometric Pattern Recognition

The core methodology involves analyzing patterns of four stars (quadrilaterals), creating "fingerprints" by calculating distances between every pair of stars in the pattern and normalizing by the longest distance to create a set of five numbers between zero and one. This approach provides several key advantages:

1. **Scale Invariance**: Normalizing distances by the longest edge makes patterns independent of image scale and field of view
2. **Rotation Invariance**: The geometric relationships remain constant regardless of camera orientation
3. **Robustness**: Four-star patterns provide sufficient geometric constraints while maintaining decent computational efficiency

Hash Table Implementation

The algorithm uses a directly-addressed hash table structure enabling star pattern identification with a single database access. The hash table approach works as follows:

1. **Pattern Key Generation**: Converting the five edge ratios into discrete bins to create a unique hash code for each star pattern

2. **Database Lookup**: Using the hash code to directly access potential pattern matches in the star catalog
3. **Collision Handling**: Managing cases where multiple patterns produce the same hash code through probing techniques

Star Detection and Centroiding

The application incorporates sophisticated image processing techniques for star detection:

- **Adaptive Thresholding**: Multiple background subtraction modes (local/global, median/mean) to handle varying imaging conditions
- **Noise Analysis**: Statistical methods for determining optimal detection thresholds based on image noise characteristics
- **Centroiding Algorithms**: Sub-pixel accuracy star position determination using moment-based calculations

Database Generation

The database generation involves creating optimized pattern catalogs for specific field-of-view (FOV) ranges. For example, if a FOV range of 10-30 was selected, 4 distinct datasets will be generated with FOVs 10, ~16, ~23, 30. Tetra3 inherently supports 3 star catalogs:

- **Yale Bright Star Catalog (BSC5)**: 9,110 stars complete to magnitude 7, suitable for wide field applications
- Hipparcos Catalogue: 118,218 stars providing higher star density for moderate field views
- Tycho Catalogue: Over 1 million stars for high-precision narrow field applications

That said, in this project we only use **BSC5** due to the project's requirements and scale.

Computational Efficiency

TETRA outperforms earlier algorithms in runtime, centroiding error sensitivity, and field-of-view error sensitivity through its optimized hash table approach. The algorithm's time complexity benefits significantly from the direct hash addressing scheme.

Robustness Factors

- **Error Tolerance**: Built-in tolerance for centroiding errors and field-of-view uncertainties
- **False Positive Control**: Statistical validation prevents incorrect pattern matches
- **Magnitude Range**: Effective operation with stars ranging from very bright to magnitude 7-8

Conclusion

The Bright Star Identifier project successfully demonstrates the practical application of advanced geometric hashing algorithms for autonomous star identification. By implementing the TETRA algorithm's hash table approach, the system achieves fast computation time and database access requirements for solving the lost-in-space problem.

The project serves as both an educational tool for understanding modern plate solving techniques and a practical implementation suitable of it.