Efficient Star Identification Algorithm for Nanosatellites in Harsh Environment

IAA-AAS-CU-17-05-02

Vincenzo Schiattarella[1], Dario Spiller[2], Fabio Curti[3]

[1] Post-graduate Student, School of Aerospace Engineering, Sapienza University of Rome, Rome 00138, Italy. Email: vincenzo.schiattarella@uniroma1.it
[2] PhD Student, School of Aerospace Engineering, Sapienza University of Rome, Rome 00138, Italy. Email: dario.spiller@uniroma1.it
[3] Associate Professor, School of Aerospace Engineering, Sapienza University of Rome, Rome 00138, Italy. Email: fabio.curti@uniroma1.it

SAPIENZA UNIVERSITÀ DI ROMA

Automation Robotics and Control for Aerospace Laboratory

Introduction

Next CubeSat generations will require more accurate attitude estimations provided by stars sensors.

Introduction

Next CubeSat generations will require more accurate attitude estimations provided by stars sensors.

False objects represent the major issue in star identification since star trackers for CubeSat generally have low radiation shielding.

Introduction

Next CubeSat generations will require more accurate attitude estimations provided by stars sensors.

False objects represent the major issue in star identification since star trackers for CubeSat generally have low radiation shielding.

This work proposes a feasibility study for a **third improvement** of the ***Multi-Poles Algorithm (MPA)***.

Introduction

Next CubeSat generations will require more accurate attitude estimations provided by stars sensors.

False objects represent the major issue in star identification since star trackers for CubeSat generally have low radiation shielding.

This work proposes a feasibility study for a **third improvement** of the ***Multi-Poles Algorithm (MPA)***.

The MPA is a ***star identification*** algorithm for ***Lost in Space***.
It is especially designed to be ***robust*** in presence of a large number of false objects in the image.

Multi-Poles Algorithm

Original Idea

The ***original idea*** has been published in *Advances in Space Research, Vol. 59, Issue 8, Pages 2133-2147*, “A novel star identification technique robust to high presence of false objects: The Multi-Poles Algorithm”

Multi-Poles Algorithm

Original Idea

The ***original idea*** has been published in *Advances in Space Research, Vol. 59, Issue 8, Pages 2133-2147*, “A novel star identification technique robust to high presence of false objects: The Multi-Poles Algorithm”

MPA main characteristics:

- Recognize stars using **THREE** phases
- Magnitude data are **NOT** required
- Stop when the desired number of stars are recognized

Multi-Poles Algorithm

Original Idea

The ***original idea*** has been published in *Advances in Space Research, Vol. 59, Issue 8, Pages 2133-2147*, “A novel star identification technique robust to high presence of false objects: The Multi-Poles Algorithm”

MPA main characteristics:

- Recognize stars using **THREE** phases
- Magnitude data are **NOT** required
- Stop when the desired number of stars are recognized

MPA is able to return correct identifications for a *number of false objects **up to 6 times** the number of the cataloged stars* in the image.

Multi-Poles Algorithm

Participation to ESA contest - Star Trackers: First Contact

Goal: Propose new and fast star identification algorithms for star trackers that are robust to measurement uncertainties and artifacts.

Multi-Poles Algorithm

Participation to ESA contest - Star Trackers: First Contact

Goal: Propose new and fast star identification algorithms for star trackers that are robust to measurement uncertainties and artifacts.

Difference w.r.t. MPA (original idea) :

- Recognize stars using **TWO** phases
- Magnitude data are **REQUIRED**
- Stop when the maximum number of stars are recognized

Multi-Poles Algorithm

Participation to ESA contest - Star Trackers: First Contact

Goal: Propose new and fast star identification algorithms for star trackers that are robust to measurement uncertainties and artifacts.

Difference w.r.t. MPA (original idea) :

- Recognize stars using **TWO** phases
- Magnitude data are **REQUIRED**
- Stop when the maximum number of stars are recognized

MPA has reported the **second** highest accuracy score and speed:

- **9703.4313** (w.r.t. a perfect score of 9921.0)
- 0.58 s (Total time required for 10000 scenes) **58 µs** per scene

Star Trackers: First Contact

Lost in Space

Ended Sept. 1, 2017

Multi-Poles Algorithm

New version

The new MPA is a ***mixed approach*** between the two former versions of the algorithm:

- Recognize stars using **TWO** phases:
  the ***acceptance phase*** and the ***check phase***.
- Magnitude data are **REQUIRED**
- Stop when the desired number of stars are recognized

Multi-Poles Algorithm

New version

The new MPA is a ***mixed approach*** between the two former versions of the algorithm:

- Recognize stars using **TWO** phases:
  the ***acceptance phase*** and the ***check phase***.
- Magnitude data are **REQUIRED**
- Stop when the desired number of stars are recognized

The ***on-board catalog*** is based on the Hipparcos identifier h and the magnitude m of 2 stars.

h_{01}	h_{02}	m_{01}	m_{02}
h_{11}	h_{12}	m_{11}	m_{12}
$\vdots$	$\vdots$	$\vdots$	$\vdots$

on-board catalog

Multi-Poles Algorithm

New version

The new MPA is a ***mixed approach*** between the two former versions of the algorithm:

- Recognize stars using **TWO** phases: the ***acceptance phase*** and the ***check phase***.
- Magnitude data are **REQUIRED**
- Stop when the desired number of stars are recognized

h_{01}	h_{02}	m_{01}	m_{02}
h_{11}	h_{12}	m_{11}	m_{12}
$\vdots$	$\vdots$	$\vdots$	$\vdots$

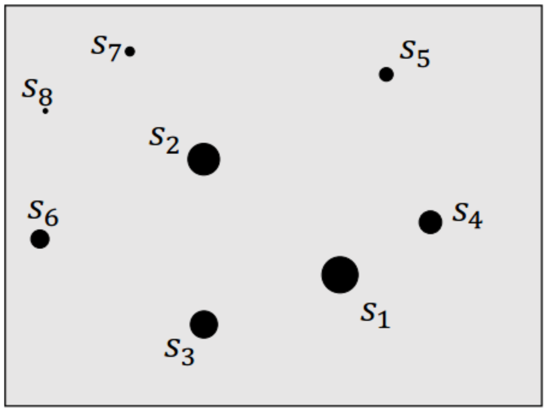
on-board catalog

The ***on-board catalog*** is based on the Hipparcos identifier h and the magnitude m of 2 stars.

First of all a function to project ***spikes*** image coordinates onto sensor reference frame is required. Each spike will be associated to a 3D unit vector $\boldsymbol{v_i}$ and an estimated magnitude value m_i.

$$\boldsymbol{s_i} = [\boldsymbol{v_i}, m_i]$$

Sorted by ascending magnitude

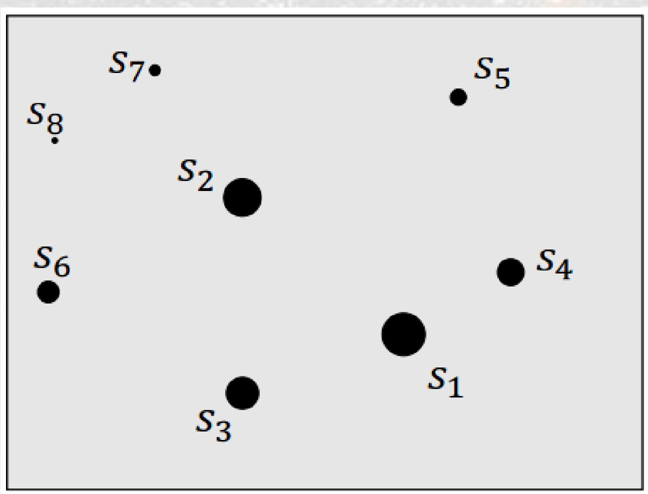


Multi-Poles Algorithm

Acceptance phase

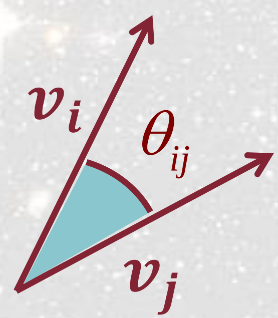
The acceptance phase is based on a polar approach and returns a set of ***Accepted stars***. This phase consists of the following steps:

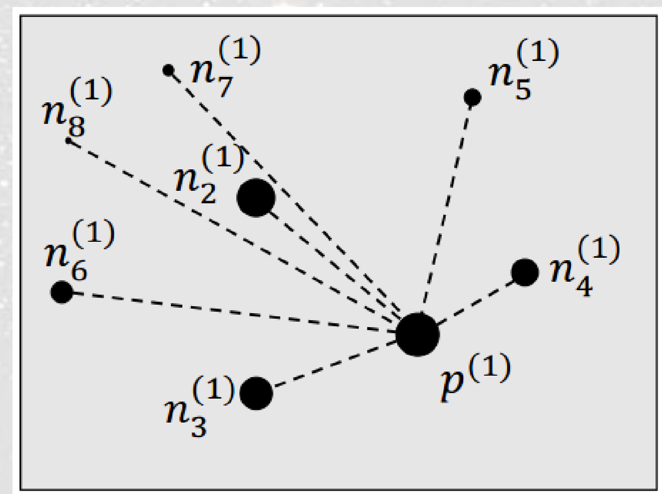
1. Select a spike $\boldsymbol{s_i}$ named ***pole*** $p^{(i)}$, and computes the **angular distances** θ_{ij} between the pole and the other spikes $\boldsymbol{s_j}$ in the image, named ***neighbors*** $n_j^{(i)}$.



Image

$$\boldsymbol{s_i} = [\boldsymbol{v_i}, m_i]$$





First pole selection

Multi-Poles Algorithm

Acceptance phase

2. Search for each **angular distance** θ_{ij} within a given tolerance the stars in the on-board catalog.

h_{01}	h_{02}	m_{01}	m_{02}
h_{11}	h_{12}	m_{11}	m_{12}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
h_{l1}	h_{l2}	m_{l1}	m_{l2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\theta_{ij} \rightarrow$

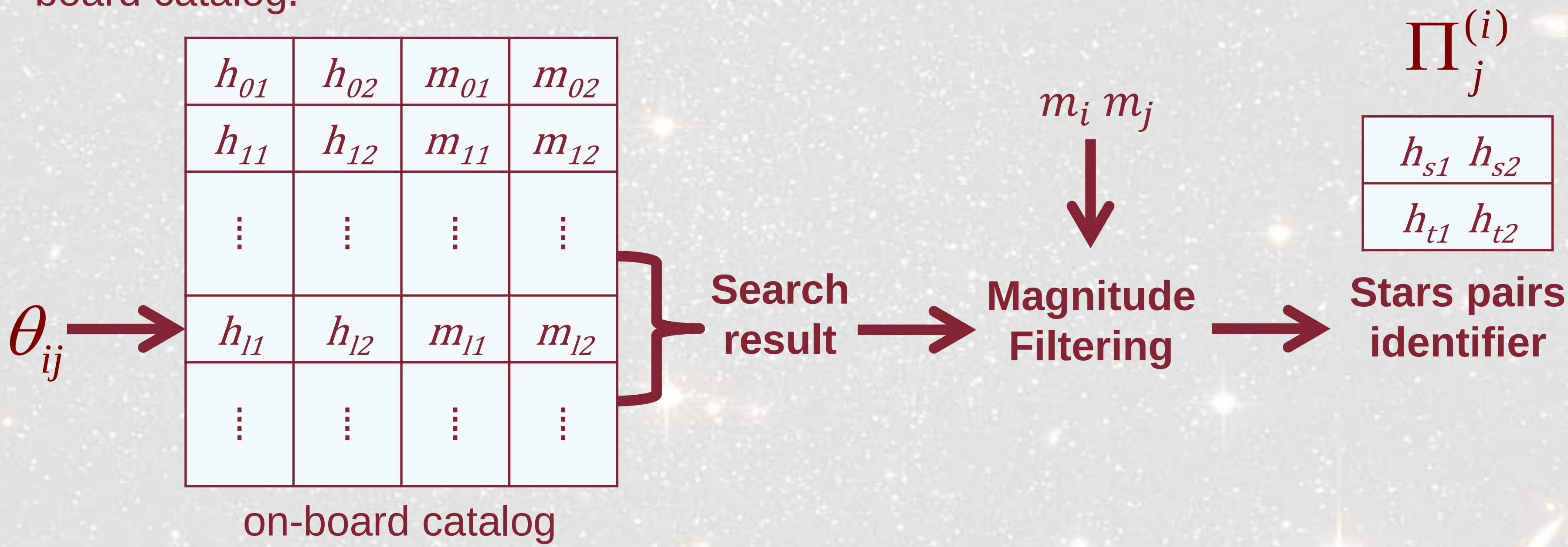
Search result

on-board catalog

Multi-Poles Algorithm

Acceptance phase

2. Search for each **angular distance** θ_{ij} within a given tolerance the stars in the on-board catalog.



3. Restrict the number of resulting stars through **magnitude** information making lists of stars pairs Hipparcos identifiers $\Pi_j^{(i)}$.
   The **pairs** are the admissible pole's and neighbors' identifiers.

Multi-Poles Algorithm

Acceptance phase

3. Accept as ***Pole Star*** the Hipparcos identifier which has the maximum number of appearances through the lists of pairs.

$\Pi_2^{(1)}$	$\Pi_3^{(1)}$	$\Pi_4^{(1)}$	$\Pi_5^{(1)}$	$\Pi_6^{(1)}$	$\Pi_7^{(1)}$	$\Pi_8^{(1)}$
31,40	*100,102*	*1346,1352*	*5,**<u>10</u>***	*81,97*	*993,999*	*534,546*
500,512	***<u>10</u>**,16*	*416,425*	*44,65*		*2023,2031*	*876,893*
	231,242	*8,**<u>10</u>***				*3648,3742*
	1028,1032					***<u>10</u>**,21*

Multi-Poles Algorithm

Acceptance phase

3. Accept as ***Pole Star*** the Hipparcos identifier which has the maximum number of appearances through the lists of pairs.

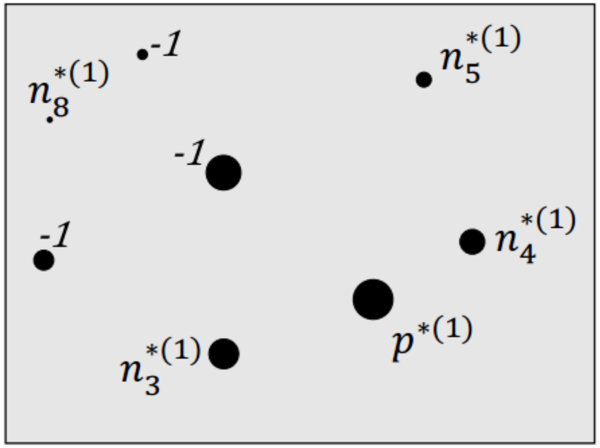
$\Pi_2^{(1)}$	$\Pi_3^{(1)}$	$\Pi_4^{(1)}$	$\Pi_5^{(1)}$	$\Pi_6^{(1)}$	$\Pi_7^{(1)}$	$\Pi_8^{(1)}$
31,40	*100,102*	*1346,1352*	*5,10*	*81,97*	*993,999*	*534,546*
500,512	*10,16*	*416,425*	*44,65*		*2023,2031*	*876,893*
	231,242	*8,10*				*3648,3742*
	1028,1032					*10,21*

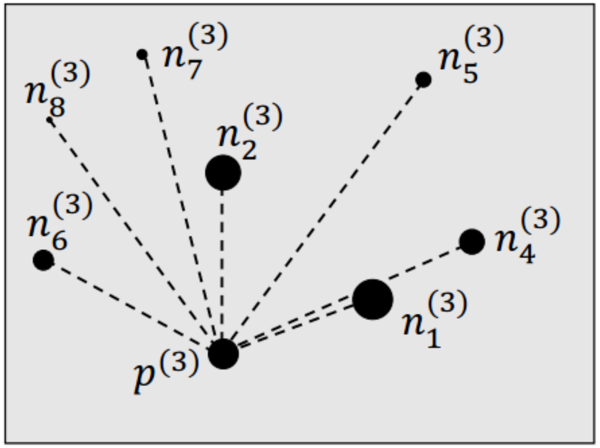
4. Search in the lists of pairs the Hipparcos identifier coupled with ***Pole Star*** and accept it as ***Neighbor Star***. The ***Accepted stars set*** will be:

$p^{*(1)}$	***F***	$n_3^{*(1)}$	$n_4^{*(1)}$	$n_5^{*(1)}$	***F***	***F***	$n_8^{*(1)}$
10	*-1*	*16*	*8*	*5*	*-1*	*-1*	*21*

Multi-Poles Algorithm

Next poles selection



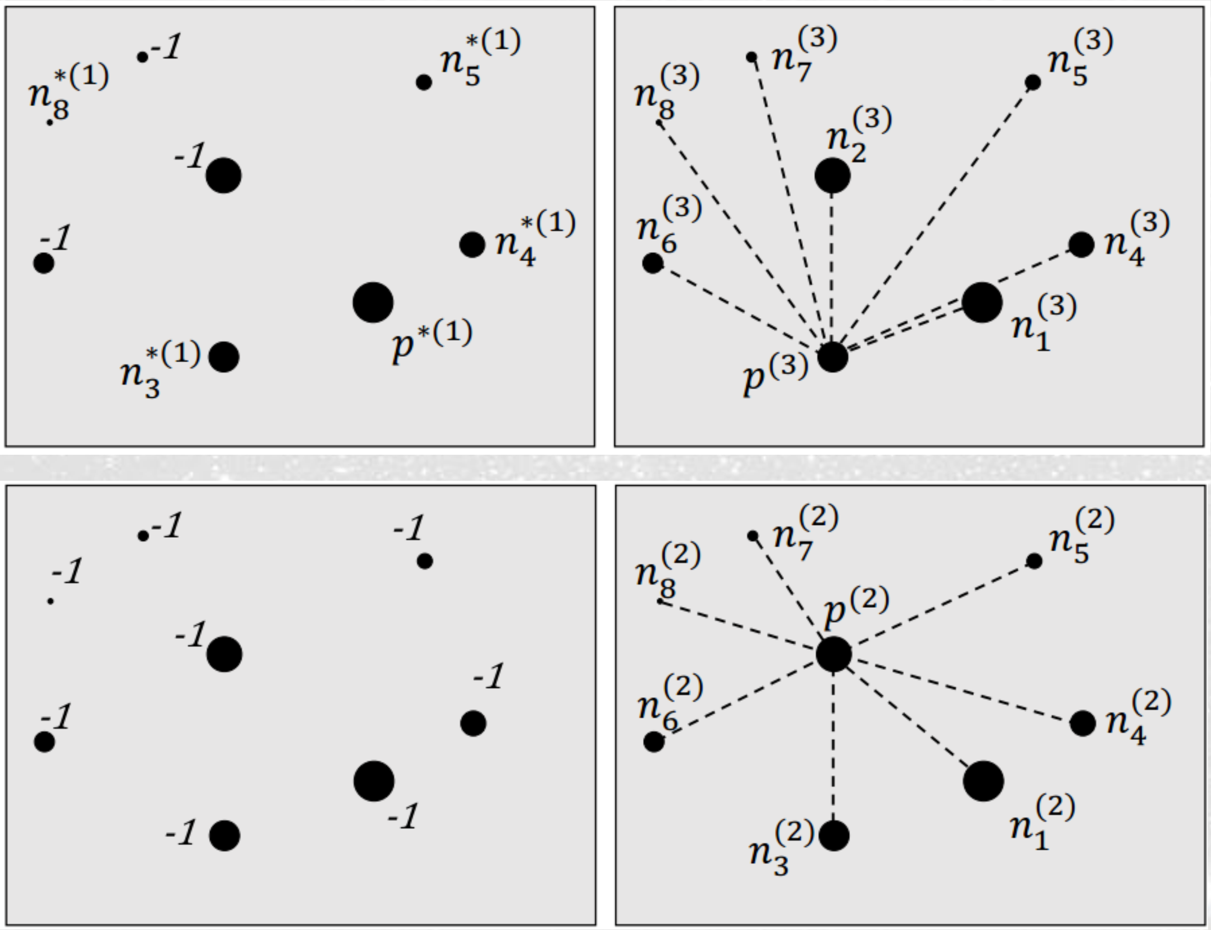


<u>Accepted stars **returned**</u>

New poles are selected within the spikes previously accepted as neighbor stars.

Multi-Poles Algorithm

Next poles selection



Accepted stars **returned**

New poles are selected within the spikes previously accepted as neighbor stars.

Accepted stars **NOT** returned

Selection criterion remains spike magnitude: next spike is selected as pole.

Multi-Poles Algorithm

Check phase

The check phase performs a cross-check between **two sets** of accepted stars. Check if the number of stars belonging to the ***intersection*** of two accepted stars sets is greater than a threshold value *t**, which is an user defined minimum number of required stars.

$p^{*(1)}$	F	$n_3^{*(1)}$	$n_4^{*(1)}$	$n_5^{*(1)}$	F	F	$n_8^{*(1)}$
10	*-1*	*16*	*8*	*5*	*-1*	*-1*	*21*

First accepted stars set

$n_1^{*(3)}$	F	$p^{*(3)}$	$n_4^{*(3)}$	F	$n_6^{*(3)}$	F	$n_8^{*(3)}$
10	*-1*	*16*	*8*	*-1*	*11*	*-1*	*21*

Second accepted stars set

Multi-Poles Algorithm

Check phase

The check phase performs a cross-check between **two sets** of accepted stars. Check if the number of stars belonging to the ***intersection*** of two accepted stars sets is greater than a threshold value *t**, which is an user defined minimum number of required stars.

$p^{*(1)}$	*F*	$n_3^{*(1)}$	$n_4^{*(1)}$	$n_5^{*(1)}$	*F*	*F*	$n_8^{*(1)}$
10	-1	16	8	5	-1	-1	21

First accepted stars set

$n_1^{*(3)}$	*F*	$p^{*(3)}$	$n_4^{*(3)}$	*F*	$n_6^{*(3)}$	*F*	$n_8^{*(3)}$
10	-1	16	8	-1	11	-1	21

Second accepted stars set

↓

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
10	-1	16	8	-1	-1	-1	21

Recognized stars set

Tests Definition

The adopted **simulator** has been provided by ESA in the frame of Kelvins ESA contest “Star Trackers: First Contact”.

Results

Tests Definition

The adopted **simulator** has been provided by ESA in the frame of Kelvins ESA contest “Star Trackers: First Contact”.

Success for each simulation is evaluated as the percentage of image with no false star wrongly identified and at least 3 correctly identified stars

Results

Tests Definition

The adopted **simulator** has been provided by ESA in the frame of Kelvins ESA contest “Star Trackers: First Contact”.

Success for each simulation is evaluated as the percentage of image with no false star wrongly identified and at least 3 correctly identified stars

Accuracy is evaluated trough a scoring function depending on number of correctly and wrongly identified stars. ***Relative accuracy*** is evaluated as the percentage of accuracy where at least 3 actual stars are in the image.

Results

Tests Definition

The adopted **simulator** has been provided by ESA in the frame of Kelvins ESA contest “Star Trackers: First Contact”.

Success for each simulation is evaluated as the percentage of image with no false star wrongly identified and at least 3 correctly identified stars

Accuracy is evaluated trough a scoring function depending on number of correctly and wrongly identified stars. ***Relative accuracy*** is evaluated as the percentage of accuracy where at least 3 actual stars are in the image.

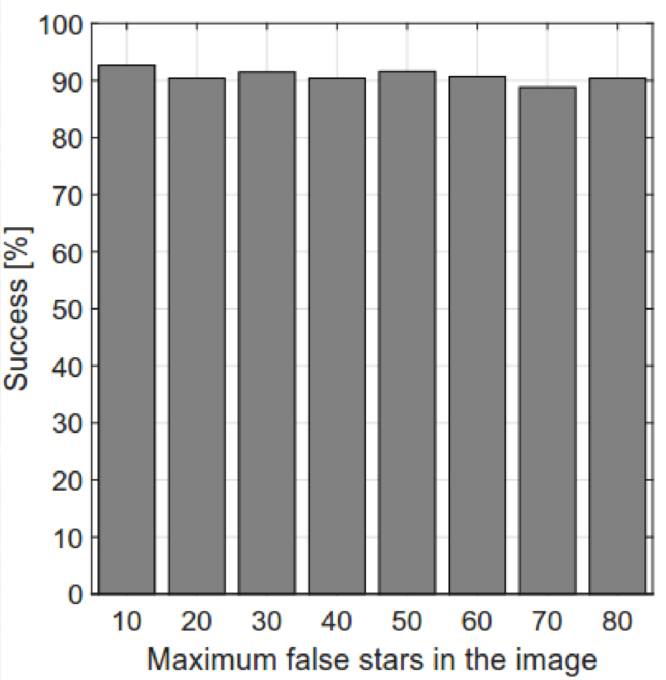
We perform the following **simulation campaigns**:

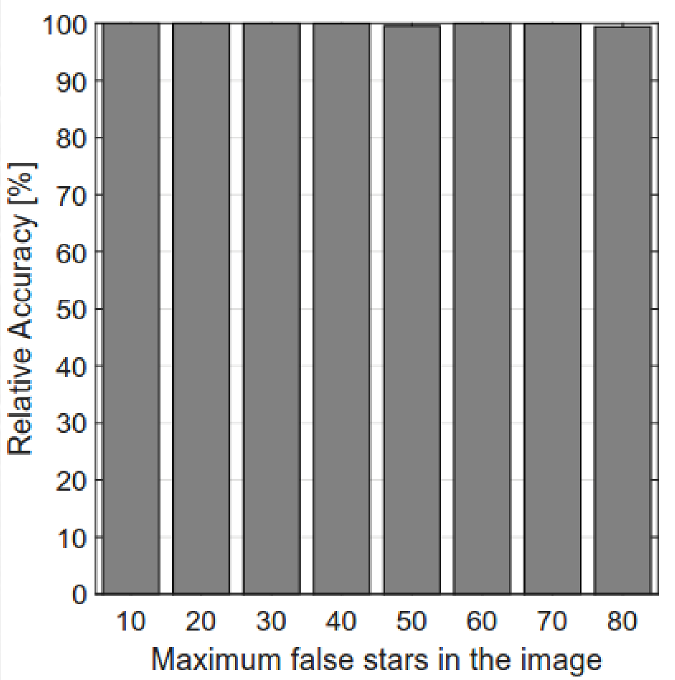
- Tests for increasing number of false objects
- Tests with ***at least η*** real stars in the image
- Tests with ***only η*** real stars in the image

Results

Results for increasing number of false objects in the image

The simulation campaign deals with tests performed for ascending maximum number of false objects in the image. For each simulation, 1000 images with random pointings have been checked. The number of actual stars per image is at least 0 (i.e. the most generic conditions).

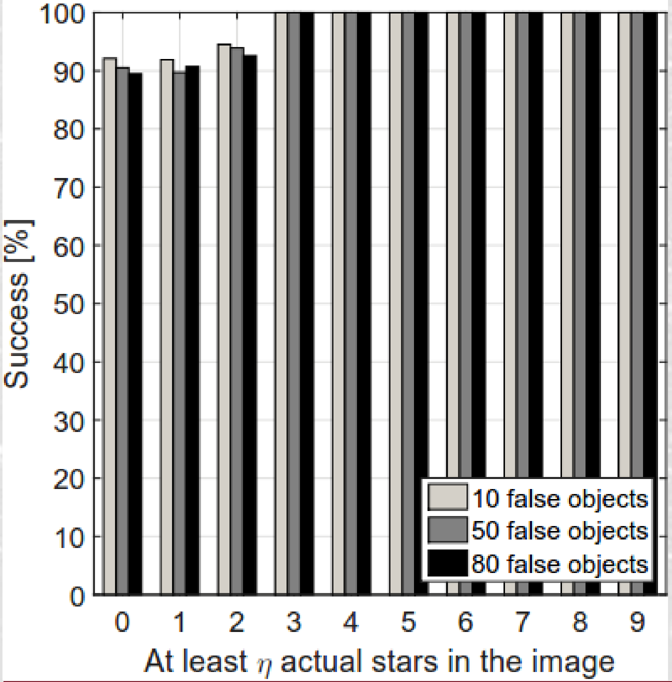


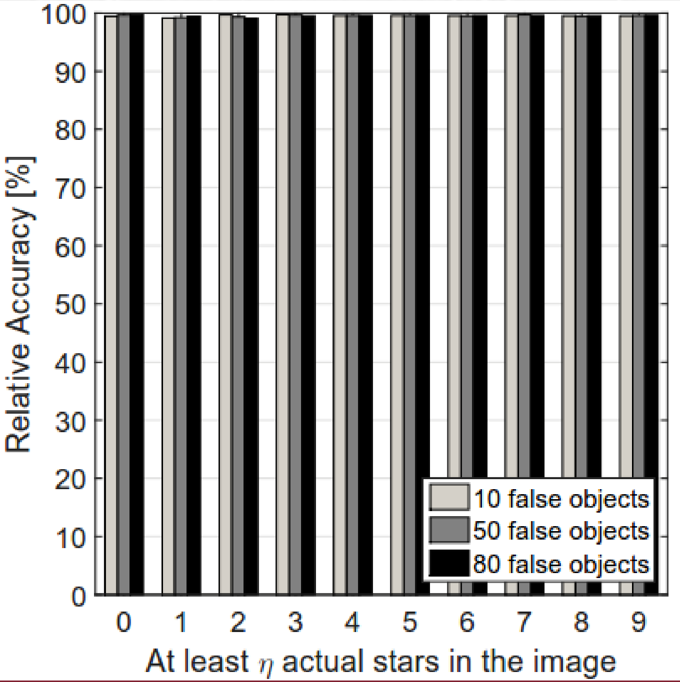


Results

Results with <u>at least</u> η real stars in the image

Defining η as the minimum value of actual stars in the image, the simulation campaign deals with tests performed for ascending η ranging from 0 to 9 stars.
For each simulation, 1000 images with random attitude have been checked.

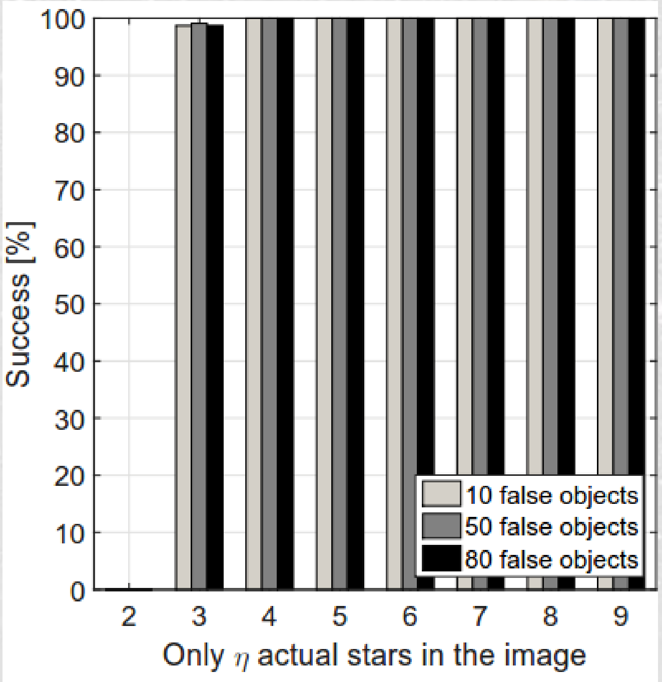


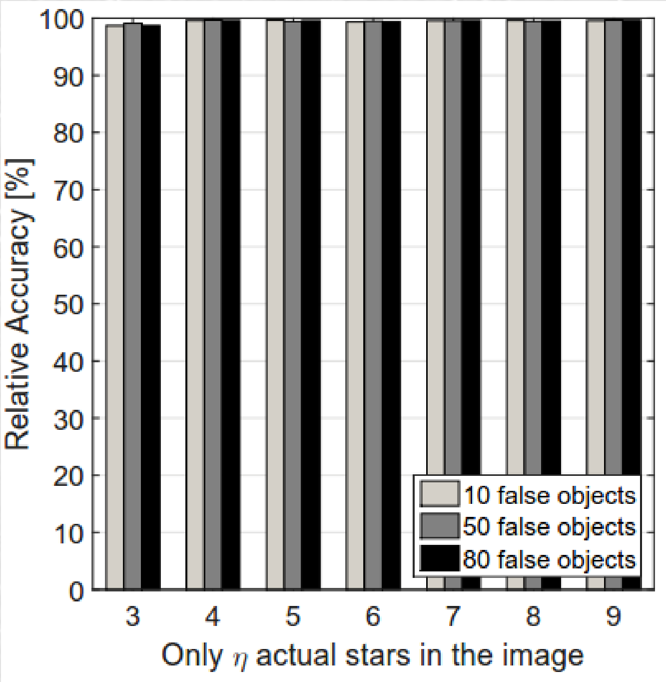


Results

Results with only η real stars in the image

Defining η as the value of actual stars in the image, the simulation campaign deals with tests performed for ascending η ranging from 2 to 9 stars.
For each simulation, 1000 images with random attitude have been checked.



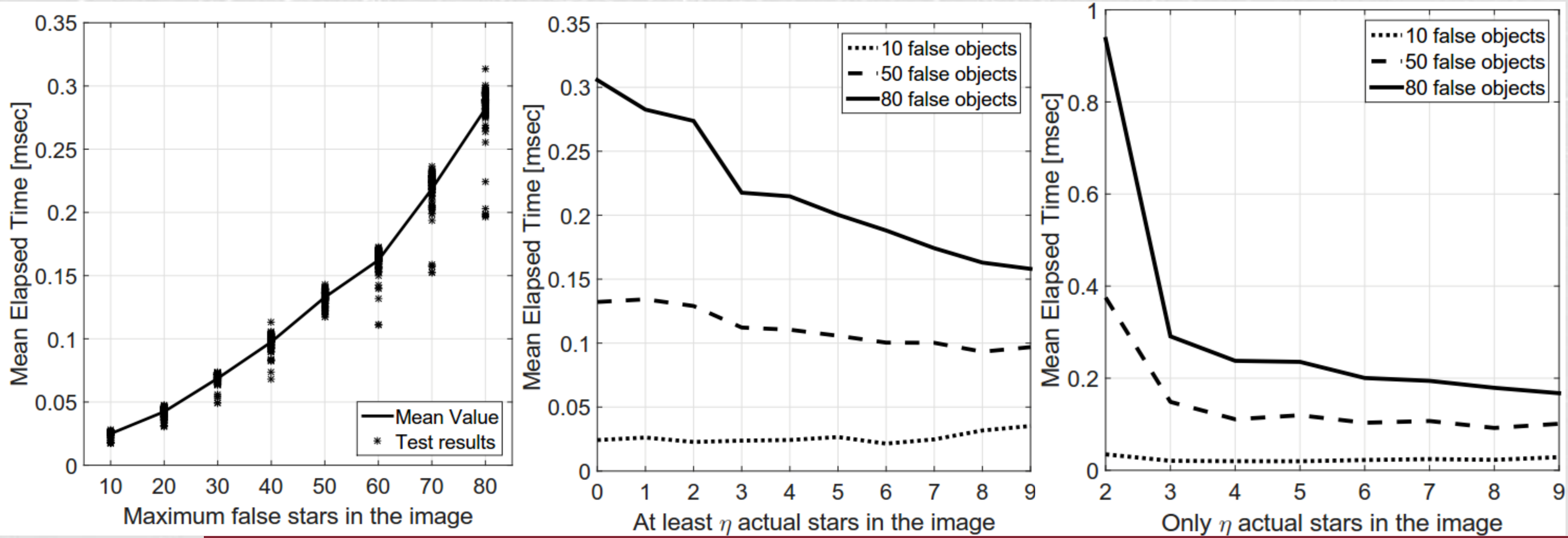


Results

Time Performances

The MPA C source code has been profiled on an Intel® Core 2 Quad Processor Q6600 @ 2.40 GHz machine CPU architecture.

The elapsed time for simulation depends on number of stars in the image.



Conclusion

The results show that MPA **success rate** ranges from 99% to 100% for simulation with at least 3 actual stars per image. In case with less than 3 actual stars in the image **no wrong** identification has occurred.

Conclusion

The results show that MPA **success rate** ranges from 99% to 100% for simulation with at least 3 actual stars per image. In case with less than 3 actual stars in the image **no wrong** identification has occurred.

With a low computational performances CPU architecture, the mean **elapsed time** for random simulations ranges from a minimum of $3 \cdot 10^{-5}$ seconds with 10 false stars to a maximum of $2.7 \cdot 10^{-4}$ seconds with 80 false stars.

Conclusion

The results show that MPA **success rate** ranges from 99% to 100% for simulation with at least 3 actual stars per image. In case with less than 3 actual stars in the image **no wrong** identification has occurred.

With a low computational performances CPU architecture, the mean **elapsed time** for random simulations ranges from a minimum of $3{\cdot}10^{-5}$ seconds with 10 false stars to a maximum of $2.7{\cdot}10^{-4}$ seconds with 80 false stars.

The algorithm **robustness capability** and its **modest computational** requirement enhance the opportunity to adopt MPA in star trackers for CubeSat, for mission where an accurate attitude knowledge is required.

Conclusion

The results show that MPA **success rate** ranges from 99% to 100% for simulation with at least 3 actual stars per image. In case with less than 3 actual stars in the image **no wrong** identification has occurred.

With a low computational performances CPU architecture, the mean **elapsed time** for random simulations ranges from a minimum of $3 \cdot 10^{-5}$ seconds with 10 false stars to a maximum of $2.7 \cdot 10^{-4}$ seconds with 80 false stars.

The algorithm **robustness capability** and its **modest computational** requirement enhance the opportunity to adopt MPA in star trackers for CubeSat, for mission where an accurate attitude knowledge is required.

Future improvements will require increasing centroiding noise and finding best parameters configuration.

Thanks for your attention

School of Aerospace Engineering

SAPIENZA
UNIVERSITÀ DI ROMA

Automation Robotics and Control for Aerospace Laboratory