A novel thresholding method for automatically detecting stars in astronomical images

Alejandro Cristo, Antonio Plaza, David Valencia
Department of Technology of Computers and Communications, University of Extremadura
Escuela Politecnica de Caceres, Avda. de la Universidad s/n, E-10071 Caceres, SPAIN
Phone: +34 927 257195 – Fax: +34 927 257203 – E-mail: {acristo, aplaza, davaleco}@unex.es

***Abstract*—Tracking the position of stars or bright bodies in images from space represents a valuable source of information in different application domains. One of the simplest approaches used for this purpose in the literature is image thresholding, where all pixels above a certain intensity level are considered stars, and all other pixels are considered background. Two main problems have been identified in the literature for image thresholding-based star identification methods. Most notably, the intensity of the background is not always constant; i.e., a sloping background could give proper detection of stars in one part of the image, while in another part every pixel can have an intensity over the threshold value and will thus be detected as a star. Also, there is always some degree of noise present in astronomical images, and this noise can create spurious peaks in the intensity that can be detected as stars, even though they are not. In this work, we develop a novel image thresholding-based methodology which addresses the issues above. Specifically, the method proposed in this work relies on an enhanced histogram-based thresholding method complemented by a collection of auxiliary techniques aimed at searching inside diffuse objects such as galaxies, nebulas and comets, and thus enhance their detection by eliminating noise artifacts. Its black-box design and our experimental results indicate that this novel method offers potential for being included as a star identification module in already existent techniques and systems that require accurate tracking and recognition of stars in astronomical images.**

***Index Terms*—astronomical image thresholding, identification of stars, bright body detection, histogram analysis.**

I. Introduction

Current star sensors are able to image an arbitrary section of the sky at very high resolution, thus allowing for the possibility of determining the correspondence between the imaged stars and a catalog of reference stars [1]. A star sensor can also be used to track known stars [2], thus providing continuous and extremely accurate estimates regarding angular displacements. Such star-tracking methodologies currently allows intelligent orientation of space satellites and determination of their attitude along their orbit around the Earth. However, due to the increased resolution of the images collected by such sensors, advanced and noise-robust techniques are required for automatic identification of stars in the image data provided by these instruments.

A number of algorithms have been developed in the past few years for star identification [3], [4], [5], [6]. Some of these algorithms are mainly intended to automatically determine the correspondence between the viewed star field and a set of catalog stars in a known reference frame [7]. Among such algorithms, general-purpose image thresholding techniques represent a simple and computationally efficient approach to automatic (yet parameterized) identification of stars in astronomical images. In such techniques, all pixels above a certain intensity level are considered stars while all other pixels are considered background. Existing techniques include the well-known *Otsu* method [8], which thresholds an image by considering its histogram as bimodal and finding the optimum value that separates the classes using several iterations with different threshold values. Another popular technique is the *Tsai* method [9], a moment-preserving technique.

In addition to general-purpose thresholding techniques, other more specialized algorithms have been specifically developed for astronomical applications. For instance, the *Misao* project [10] is an ongoing effort in which time sequences of sky photographies, taken by amateur astrophotographers, are collected and analyzed by means of a software suite which includes a module aimed at detecting the stars that they have might modified their brightness. Such module uses a quick thresholding-based algorithm, while computes the background level and each pixel's standard deviation with regards to such level. Another ongoing project is *GCX* [11], an application that allows to obtain astronomical images from cameras or telescopes, and to process them in fully automatic fashion. Among several advanced options implemented in this system, there is an automatic thresholding-based star detection module, which is performs background level calculation by considering a fixed standard deviation, along with several additional steps such as elimination of pixels in the spatial neighborhood of the brightest pixels, or elimination of large bodies (an object is considered a large body if one of the sides of the convex hull region that contains such object exceeds a certain threshold).

Despite the suitability of thresholding algorithms (such as those mentioned above) for approaching the problem of automatically detecting stars in astronomical images, several problems have been identified in the literature regarding the use of image thresholding-based identification methods in astronomical applications [6]:

1) The stars tend not to be point-like objects in the image, so a bright star can look bigger than a faint star, resulting in more than one pixel responding to the thresholding of a single star.
2) The intensity of the background is not always constant. A sloping background could give proper detection of stars in one part of the image, while in another part every pixel can have an intensity over the threshold value and will thus be detected as a star.
3) There is always some degree of noise present in the image which may create spurious peaks in the intensity, which can be detected as stars even though they are not.

978-1-4244-3555-5/08/$25.00 ©2008 IEEE

In this work, we develop a novel image thresholding-based methodology which addresses the three main issues listed above. Specifically, the method proposed in this work is based on the development of an enhanced histogram-based thresholding method which is complemented by a collection of auxiliary modules aimed at searching inside diffuse objects such as galaxies, nebulas and comets, thus enhancing detection accuracy by eliminating noise artifacts. The remainder of the paper is organized as follows. Section II describes the proposed methodology. Section III provides a quantitative and comparative experimental validation of the proposed approach using both simulated and real astronomical images with ground-truth. Finally, section IV concludes with some remarks and hints at plausible future research lines.

II. Proposed Methodology

The proposed star detection system has been designed in the form of a black box, in which an astronomical image serves as an input, and the system provides information about every detected body. The overall system is illustrated in Fig. 1. This black box design may allow the proposed technique to be incorporated into already available systems. Specifically, the outputs of the proposed system are the following ones:

- *Objectmat* – A matrix analogous to the astronomical image, where each cell stores a '1' value if the corresponding pixel is labeled as a star, or a '0' value otherwise.
- *Objects* – A matrix containing information about detected star, where each column represents the body identifier, and each line stores the number of pixels belonging to the star and the spatial coordinates of the region that the star covers.
- *Objectnum* – The total number of bodies (stars) detected in the astronomical image.
- *Filteredim* – The original astronomical image with annotations (each detected star is marked in a different colour).

The proposed black box system is composed by a sequence of modules, each one with a well established function that will perform a sequence of transformations to the original astronomical image. These modules are graphically represented in Fig. 2 and individually described below.

A. Unispectral transformation module "ImageMatrix"

This module transforms an original (RGB) astronomical image into a single channel by summing up all the channels and storing these values in an auxiliary matrix analogous to the original image. This allows application of the proposed image processing techiques irrespectively of the number of channels in the input data.

B. "Antineb" filter

A high pass filter in which a 3×3-pixel kernel is used to assign the maximum value in the neighborhood within the kernel to the central pixel. This makes bright bodies widen and diffuse objects more compact while reducing the noise.

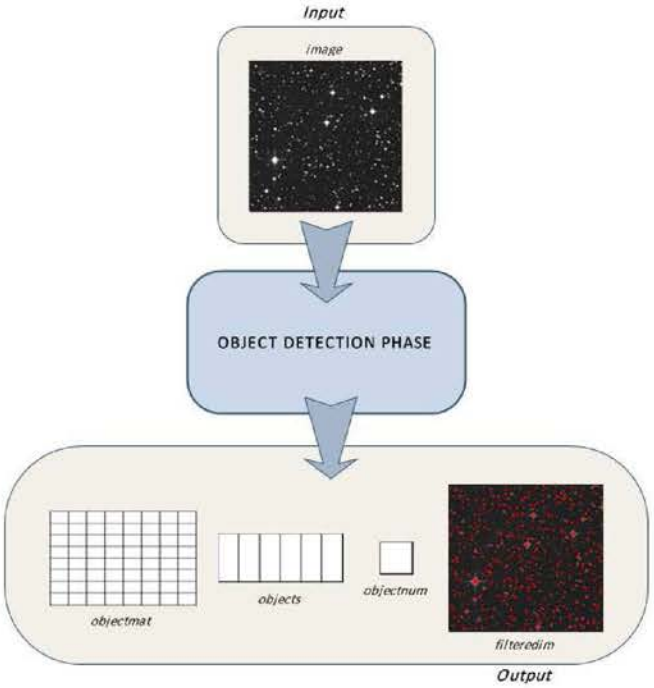

Fig. 1. Flowchart depicting the proposed detection system as a black box.

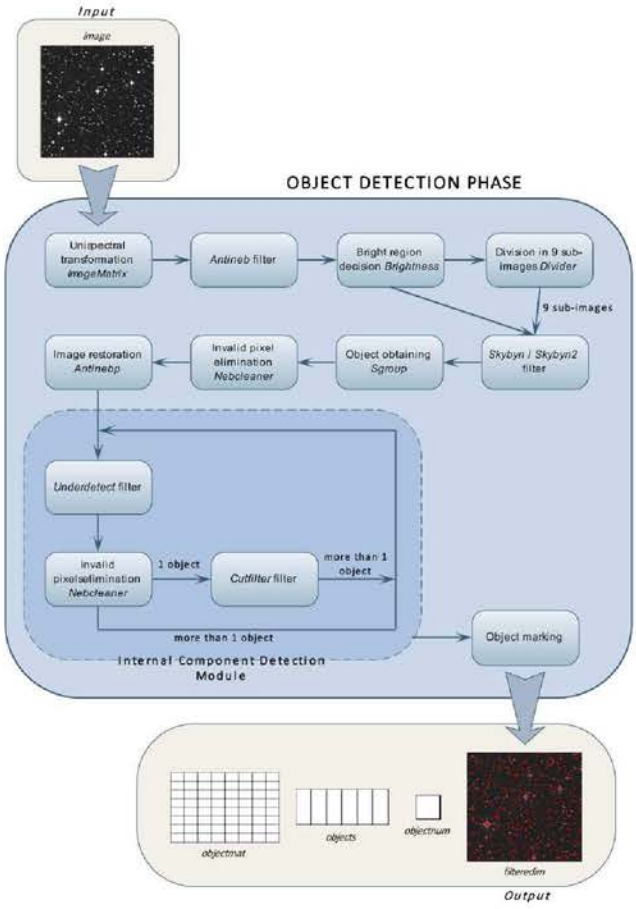

Fig. 2. Detail of the proposed detection system, explaining the sequence of modules which are applied inside the black box depicted in Fig. 1.

C. "Skybyn" filter

This filter estimates the optimum histogram value that will allow us to consider a pixel as belonging to a star or to the background (sky). This value is obtained by following the steps listed below:

1) Histogram morphological transformation using the following expression:

$$f'(x) = f(x) \times (255 \times nchannels - f(x)). \quad (1)$$

The expression above will be applied to every possible value of the histogram. The aim of this step is to obtain a typical atronomical histogram aspect (a lot of high values near to zero as background, and some picks in the opposite site as stars), independently of its morphology, by making darker values brighter and brighter values darker. In (1), *nchannels* is the number of channels or bands of the original astronomical image.
2) Calculation of the maximum value of the histogram and, using this value, calculation of the last minimum value of the histogram.
3) Joining of the maximum values inside the previous range to smooth the curve, so zero values caused by the absence of pixels for certain colours are deleted.
4) New calculation of the maximum value and the minimum value nearest to it.
5) Translation of the minimum-maximum range in such a way that the minimum point is placed in spatial location $(0, 0)$, and rotation in such a way that the maximum point is placed in *y=0*.
6) Calculation of the maximum value of the transformed range.
7) The threshold point will be the previous calculated maximum in the original coordinate system.

D. "Skybyn2" filter

This module is not simply a variation of the previous filter but, instead, a new method for estimating the optimum histogram value that will allow us to consider a pixel as belonging to an object or belonging to the sky. This value is obtained by following these steps:

1) Execution of steps (1)–(3) of the *Skybyn* filter.
2) Calculation of the weighted mean of all the histogram values, so that darker values will have more weight and brighter values will have less weight, in such a way that if a value is positioned in position (x) of the histogram, its weight will be calculated as $256 \times nchannels - x$. With this, a straight line higher than the normal mean is obtained.
3) The cut-point of the straight line previously calculated with the histogram values will be considered as the new threshold value.

E. Object extraction module "Sgroup"

Once objects have been detected with the previous filter it is necessary to extract each one and to extract as much information about them as possible, including the number of pixels, the image coordinates where they are located, etc. To achieve this, a new algorithm has been designed and implemented which allows to determine the number of pixels that each detected body comprises and the positions (x, y) of the upper leftmost and bottom rightmost corner of the convex hull region that contains each body, using only two passes of a simple window moving-based strategy.

F. Invalid pixel elimination module "Nebcleaner"

This simple filter, statistically calculated, eliminates the pixels from diffuse bodies which are detected as stars. To achieve this, we simply eliminate the objects whose maximum value is below the following threshold:

$$threshold' = \frac{125}{15 \times (maximum - threshold)}, \quad (2)$$

where *threshold* is obtained via either *Skybyn* or *Skybyn2* filters.

G. Image restoration module "Antinebp"

For enhancing processing by all previously described modules, the proposed system makes use an *Antineb*-filtered image (section II-B), in which diffuse objects appears more compact in order to detect them more accurately. This module filters the original image in the way the *Antineb*-filtered one is in a pixel-per-pixel fashion.

H. Internal Component Detection Module "ICDM"

This module analyzes each detected object with the aim of finding small bodies inside large objects (generally in large and diffuse objects such as galaxies and nebulas). Each detected body inside an object is analyzed too. To accomplish this, we use two filtering algorithms, where each one calculates a different threshold. If new bodies are not found inside an object after applying the first threshold, then the second threshold is resorted to. The filtering algorithms used are the following ones:

- *Underdetect* – the threshold calculated by this method is the mean of the values of the current object. The pixels whose value does not exceed this threshold will be deleted.
- *Cutfilter* – The threshold calculated by this method is the average value of the maximum and minimum values of the region that covers the current object.

I. Bright image treatment module

Certain images have very bright and large isolated regions that can negatively influence the performance of the *Skybyn* filter threshold. As a result, darker objects cannot be detected as accurately. To solve this problem, large regions may be detected and isolated before conducting the object search. The

two modules described below have been designed to perform this task:

- Bright region decision (*Brightness*) – This module evaluates if the image is bright enough to present the problem described above. The module considers that this is the case if the position of the maximum value of the histogram exceeds one-fourth of the total ($256 \times nchannels$). In that case, the *Divider* module described below is applied.
- Division of the image (*Divider*) – This module applies a simple *Skybyn* filter to the original image and, using *Sgroup*, the largest object is located. Then, the image is divided in several sub-regions, with the large bright region located in the centre. Once the image has been divided, the *Skybyn* filter is applied to each sub-image independently, and after this, the individual parts are recombined to continue the object detection process.

III. Experimental Results

In order to test the performance of the proposed thresholding-based method, we have compared its performance to that of other thresholding algorithms in widely available methodologies, such as the *Misao* and *GCX* projects. For reference, we have also included the standard *Otsu* and *Tsai* methods in our comparison. In order to allow a fair comparison, some of the features in our proposed methodology were applied prior to the processing using other methods, including the unispectral transformation *ImageMatrix*, internal component searching using the *ICDM* module, or quick object information obtaining using *Sgroup*. In order to carry out the quantitative and comparative experiments, we have used four different data sets:

1) *Ideal image* – this image has been simply created using Microsoft Paint application. Over a black background, a group of spots in different colours and sizes have been painted.
2) *Semi-ideal image* – this image was directly obtained from Stellarium application. It is an image similar to the ideal, but the objects have different gradients of colour and a soft contour blurred with the dark background.
3) *B&W (greyscale) real image* – this is a greyscale real image with only one channel taken from a small region of Virgo.
4) *RGB (color) real image*– this is a color real image with three channels (red, green and blue), taken from a region of Botes.

For illustrative purposes, Fig. 3 shows the four scenes considered in experiments with the ground-truth superimposed in red. The first two images [displayed in Fig. 3(a) and Fig. 3(b), respectively] are synthetic in nature and, therefore, ground-truth information regarding the pixels which actually belong to stars is available *a priori*. These images were simulated using different noise levels and considering four different types of noise, all of them representative of available astronomical instruments:

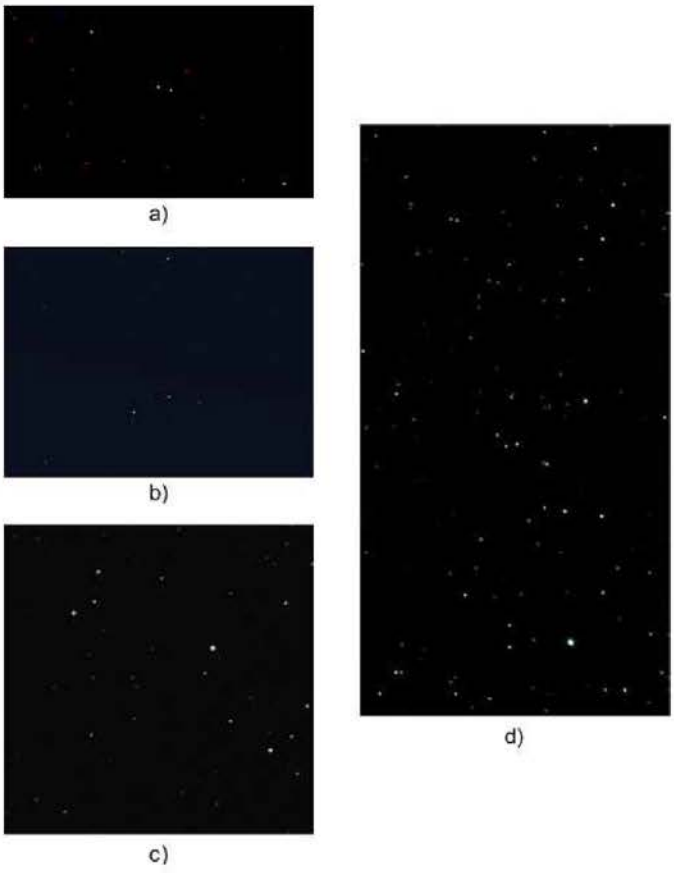

Fig. 3. Image data sets considered in experiments with ground-truth annotations superimposed: (a) Ideal image; (b) Semi-ideal image; (c) Greyscale real image; (d) Color real image.

- *Gaussian noise* with variance V, it is the most common noise that we can find in any kind of image, caused by the thermal noise of the optical device.
- *Poisson noise*, in which the Poisson distribution was calculated from the information present in the image, instead of generating it artificially.
- *Salt & pepper noise*, which makes possible that some pixels get extreme values, i.e., either very dark or very bright, with an adjustable density D.
- *Speckle noise*, which is dependent on the actual image noise through the expression $J = I + n \times I$, where I is the input image, and n is an uniformly-distributed random noise with mean 0 and variance V.

It should be noted that, in real scenes, noise is coming from different external sources, such us atmosphere, camera, etc. In order to generate ground-truth information in the real scenes, an expert astronomer manually annotated the stars based on previous knowledge about the regions imaged [see Fig. 3(c) and Fig. 3(d), respectively]. The four considered images (ideal, semi-ideal, greyscale, and color) were used to substantiate the considered thresholding algorithms in terms of the following metrics of detection accuracy:

	Histogram1	Histogram2	Misao	GCX	Otsu	Tsai
Number of objects						
Ideal (47)	32	47	47	23	47	32
Semi-ideal (98)	98	98	99	73	35	43
B&W (59)	56	56	1521	106	39	39
RGB (151)	147	142	674	339	99	96
% True positive rate						
Ideal	83.04	100	100	96.19	100	83.04
Semi-ideal	79.18	79.18	100	88.47	59.47	66.54
B&W	72.3	72.3	100	100	77.13	76.48
RGB	73.87	71.99	99.24	98.68	54.13	50.56
% False positive rate						
Ideal	0	0	0	0	0	0
Semi-ideal	0.01	0.01	0.01	0.06	0	0.01
B&W	0.04	0.004	2.22	0.38	0.01	0.01
RGB	0.09	0.07	2.17	1.56	0.02	0.02
Time (sec)						
Ideal	2.2432	2.2232	0.96138	1.2418	1.0816	1.1016
Semi-ideal	2.7039	2.7039	1.1416	1.5022	1.2218	1.2919
B&W	1.1216	1.1316	0.73105	0.75108	0.58084	0.60086
RGB	0.78112	0.74107	0.45065	0.41059	0.33048	0.40058

Fig. 4. Detection and computing time results obtained by the two implementations of the proposed thresholding method (*Histogram1* and *Histogram2*) and by other standard methodologies such as Otsu's and Tsai's method, and the thresholding methods available in the *Misao* and *GCX* projects, using two simulated scenes and two real scenes.

- Number of detected stars by a certain algorithm (we assume that a star is detected if all the pixels belonging to the star are correctly detected by the algorithm).
- True positive rate achieved by a certain algorithm (percentage of pixels belonging to stars which are correctly labeled as stars by the algorithm).
- False positive rate achieved by a certain algorithm (percentage of pixels belonging to background which are incorrectly labeled as stars by the algorithm).
- Computing time in seconds spent by each algorithm in detecting the stars. The computing system used in experiments consisted of a 2.08 GHz CPU with 512 MB of RAM memory and the Windows XP operating system (all the algorithms were implemented and executed on the Matlab 7.0 environment).

Fig. 4 displays the results obtained by the two implementations of the proposed thresholding method (*Histogram1* and *Histogram2*, resulting from the application of *Skybyn1* and *Skybyn2* filters, respectively) and by other standard methodologies such as Otsu's and Tsai's method, and the thresholding methods available in the *Misao* and *GCX* projects, using the four considered scenes. In order to analyze the impact of simulated noise on the obtained results, Figs. 5, 6, and 7 report (in graphical form) the results obtained by the four considered metrics after applying different types of noise (Gaussian, salt & pepper and speckle) to the semi-ideal image. Finally, Fig. 8 shows the detection results obtained after applying Poisson noise to both the ideal and semi-ideal scenes.

A. Analysis of the results

With the above observations in mind, our experimental results reveal that the best results were generally reported

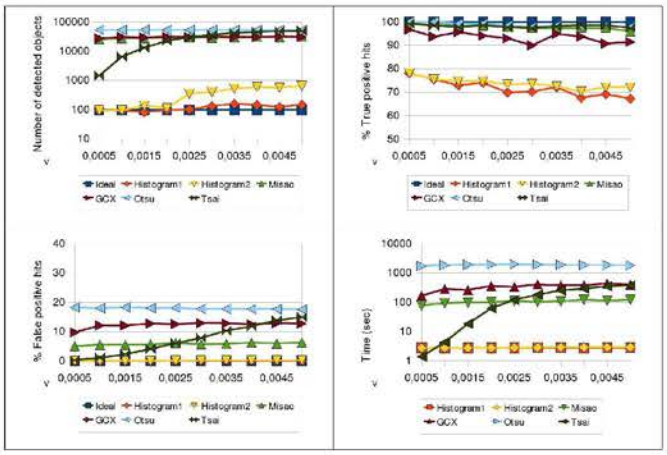

Fig. 5. Detection results obtained by all the tested methodologies after applying Gaussian noise to the semi-ideal image.

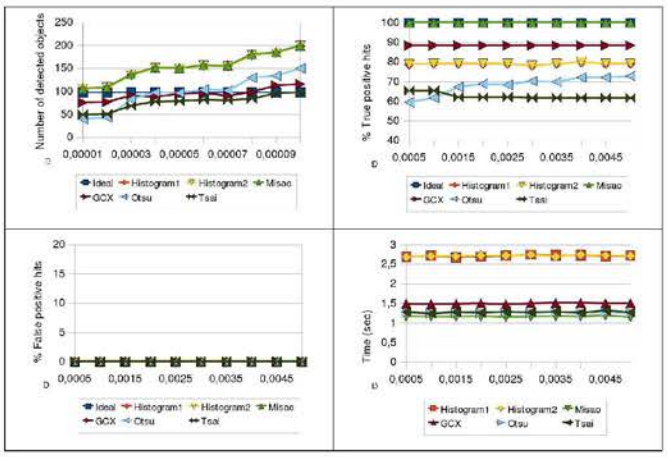

Fig. 6. Detection results obtained by all the tested methodologies after applying salt & pepper noise to the semi-ideal image.

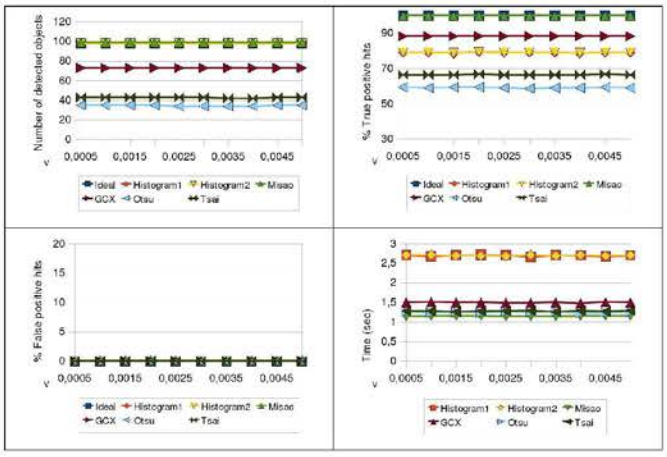

Fig. 7. Detection results obtained by all the tested methodologies after applying speckle noise to the semi-ideal image.

	Histogram1	Histogram2	Misao	GCX	Otsu	Tsai
Number of objects						
Semi-ideal (98)	93	93	25473	22466	35	402
% True positive rate						
Semi-ideal	77.69	77.69	100	94.42	60.96	99.62
% False positive rate						
Semi-ideal	0.01	0.01	5.15	6.84	0	0.09
Time (sec)						
Semi-ideal	2.7239	2.7339	85.3027	98.3314	1.2518	1.3019

Fig. 8. Detection results obtained by all the tested methodologies after applying Poisson noise to the ideal and semi-ideal image.

by the *Histogram1* and *Histogram2* methods, although their execution time is slightly higher than that reported for the other algorithms. The rest of the algorithms, in spite of having lower execution times, result in acceptable scores for both ideal and semi-ideal (synthetic) images, but not as good as expected for real images, in which natural noise is indeed present. Regarding the Otsu and Tsai thresholding methods, these algorithms are very fast but, as general-purpose methods, their detection scores are not as high as expected, and their results can vary in the experiments with very noisy images. Finally, the *Misao* and *GCX* methods are computationally efficient because of their simplicity, and are also able to provide good results when the astronomical images are clean and do not present any kind of noise. However, when noise exists, these methods are outperformed by the two proposed approaches. Some additional observations regarding the impact of noise in the considered thresholding methods are noteworthy:

- The behaviour of all considered algorithms in the presence of high Gaussian noise is not very good (see Fig. 5). This is because this kind of noise creates an irregular background which tends to be detected as a single bright, homogeneous object. Only the two proposed methods, *Histogram1* and *Histogram2*, seem fairly robust in the presence of this effect.
- All considered algorithms (except perhaps *Misao* and *GCX*) seem to perform accurately in the presence of Poisson noise, with the proposed *Histogram1* and *Histogram2* methods evidencing the best results (see Fig. 8).
- The salt & pepper noise tends to add very bright and very dark pixels to the image, and thus the stars are very difficult to be detected after addition of this type of noise. As a result, this noise affects quite negatively all developed algorithms (see Fig. 6). It is worth noting that, still, the *Histogram1*, *Histogram2*, and *Misao* algorithms are able to detect all labeled stars under Poisson-generated noise.
- Finally, the speckle noise does not alter the original image significantly and, as a result, the performance of the considered algorithms is quite similar to the ideal case, even when the amount of speckle noise increases.

To conclude our discussion of results, it is important to emphasize that the *Misao*, *GCX*, *Otsu* and *Tsai* methods would have resulted in less accurate detection results if some of the modules designed for *Histogram1* and *Histogram2* methods (e.g., the *ICDM*) had not been included as part of the processing of such methods. With the above observation in mind, we strongly believe that the experimental comparison conducted in this section provides a fair overview of the performance of the proposed methods as compared to available approaches, revealing the capacity of our method to achieve consistently high detection accuracies in the presence of noise.

IV. Conclusion

In this work, we have developed a novel thresholding algorithm for automatic detection of stars which is based on the search for a histogram threshold value that allows a simple decision when evaluating whether a pixel belongs to a star or the sky (background). The proposed method is not as simple as standard thresholding algorithms commonly available in existing change detection systems. Instead, our approach has been complemented with a set of specific auxiliary modules for image preprocessing and postprocessing with the ultimate goal of addressing some problems identified for already available algorithms in the literature. Our experimental results using both simulated and real astronomical images with ground-truth indicate that the proposed algorithm is highly competitive in terms of both detection accuracy and robustness in the presence of noise, outperforming several standard algorithms in the considered case studies. Another advantage of the proposed methodology is its design in the form of a black box, which allows potential incorporation of the algorithm to already existing systems. Although the results reported in this work are quite encouraging, further experiments using additional scenes and algorithms are required to fully substantiate the contributions of our proposed thresholding-based methodologies.

References

[1] D. S. Clouse and C. W. Padgett, "Small Field-of-View Star Identification Using Bayesian Decision Theory," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 773–783, 2000.

[2] C. C. Liebe, "Accuracy Perfomance of Star Trackers: A Tutorial," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, 2002.

[3] M. A. Samaan, D. Mortari, J. L. Junkins, "Recursive Mode Star Identification Algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1246–1254, 2005.

[4] B. Balaji and A. Damini, "New Robust Star Identification Algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 3, pp. 1126–1131, 2006.

[5] C. Padgett and K. Kreutz-Delgado, "A Grid Algorithm for Autonomous Star Identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, 1997.

[6] E. Silani and M. Lovera, "Star Identification Algorithms: Novel Approach & Comparison Study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, pp. 1275–1288, 2006.

[7] D. Lindler, S. Heap, J. Holbrook, E. Malumuth, D. Norman and P. C. Vener-Saavedra, "Star Detection, Astrometry and Photometry in Restored PC Images," Available online: http://www.stsci.edu/stsci/meetings/irw/proceedings/lindlerd.dir/lindlerd.html

[8] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, 1979.

[9] S.-C. Cheng and W.-H. Tsai, "A Neural Network Implementation of the Moment-Preserving Technique and its Application to Thresholding," *IEEE Transactions on Computers*, vol. 42, no. 4, pp. 501–507, 1993.

[10] *The MISAO Project*, Available online: http://www.aerith.net/misao

[11] R. Corlan, "GCX User's Manual v0.7.6," Available online: http://astro.corlan.net/gcx/html/gcx.html