

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/265934269>

Astronomical Image Data Reduction for Moving Object Detection

Conference Paper · November 2013

DOI: 10.1007/978-3-642-41827-3_15

CITATIONS

2

READS

51

4 authors:



Kevin Allekotte

University of Buenos Aires

1 PUBLICATION 2 CITATIONS

[SEE PROFILE](#)



Pablo De Cristóforis

University of Buenos Aires

23 PUBLICATIONS 146 CITATIONS

[SEE PROFILE](#)



Mario Melita

National Scientific and Technical Research Council

51 PUBLICATIONS 385 CITATIONS

[SEE PROFILE](#)



Marta Mejail

University of Buenos Aires

88 PUBLICATIONS 558 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



S-PTAM [View project](#)



Trans Neptunian Objects [View project](#)

Astronomical Image Data Reduction for Moving Object Detection

Kevin Allekotte¹, Pablo De Cristóforis¹, Mario Melita², and Marta Mejail¹

¹ Departamento de Computación, Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires, Argentina

² Instituto de Astronomía y Física del Espacio, Consejo Nacional de Investigaciones
Científicas y Técnicas, Argentina
{kallekotte,pdecris,marta}@dc.uba.ar, melita@iafe.uba.ar

Abstract. In this work we present a system for autonomous discovery of asteroids, space trash and other moving objects. This system performs astronomical image data reduction based on an image processing pipeline. The processing steps of the pipeline include astrometric and photometric reduction, sequence alignment, moving object detection and astronomical analysis, making the system capable of discovering and monitoring previously unknown moving objects in the night sky.

Keywords: astronomical images, data reduction, moving object detection.

1 Introduction

One of the goals of modern astronomy is the exhaustive study and mapping of celestial bodies, in particular the moving bodies of the solar system. In the past, observations were made and analyzed manually, nowadays we have robotic telescopes with CCD cameras. This allows the development of systems that control the telescope and automate the data extraction and reduction from astronomical images. The replacement of rutinary tasks performed by astronomers using autonomous systems has multiple advantages and enables a continuous and long exploration (even in hostile environments) resulting in higher probabilities of achieving astronomical discoveries.

In this work we present a method for autonomous discovery of asteroids, space trash and other moving objects. This method performs astronomical image data reduction based on an image processing pipeline to find moving objects in the night sky and control the telescope for an automated tracking. The processing steps of the pipeline include astrometric and photometric reduction, sequence alignment, moving object detection and astronomical analysis, making the system capable of discovering and monitoring previously unknown moving objects. In Figure 1, the pipeline data flow and a sample of an astronomical image are shown.

Some of the previous related works include IRAF [1] (Image Reduction and Analysis Facility), a well known system that implements most methods for astronomical data reduction. The main drawback of this system is that it is designed

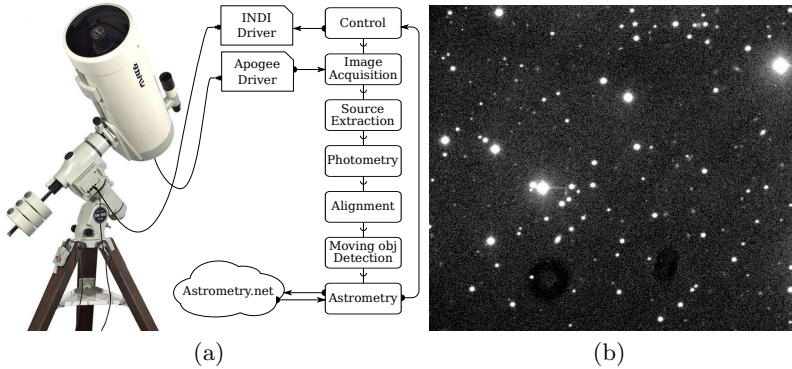


Fig. 1. (a) System's pipeline data flow and (b) sample of astronomical image captured with Takahashi Mewlon-210 telescope and Apogee Alta F16 CCD camera.

for manual analysis of the data, so it is rather complicated to use as a library with a programmable interface and, more importantly, has no moving object detection routines. The purpose of this paper is to address these issues.

The remainder of this paper is organized as follows: Section 2 details the proposed method, section 3 presents the results and section 4 summarizes our work.

2 Proposed Method

The images are captured with a CCD camera and stored in the FITS (Flexible Image Transport System) file format. The images usually contain a few hundred light sources, depending on the region of the sky and the angular field of the telescope. The profile of each light source can be approximated with a 2D Gaussian curve (if the camera is in focus). The one-dimensional profile of this Gaussian curve is characterized in astronomy by the FWHM (Full Width at Half Maximum) which depends on the dispersing process in the atmosphere and therefore is approximately constant for all point light sources of the image [2]. These images also contain background noise that is characterized by an additive and a multiplicative component. Furthermore, there are spurious detections caused by defective pixels or cosmic rays that randomly reach the sensor and cause peaks in the image. Figure 2 shows examples of light sources in astronomical images.

The moving objects we are interested in finding appear in the image with a very similar profile as the stars and satisfy the following conditions: a) $v \ll \frac{FWHM}{\Delta t_{exp}}$, b) $\Delta t \gg \frac{FWHM}{v}$ and c) $\frac{\#D}{N\Delta t} > v$, where v is the velocity of the object, Δt_{exp} is the exposure time, Δt is the time interval of the sequence, $\#D$ is the size of the captured image D and N is the number of images (frames) in the sequence (typically $N \geq 4$). The time interval between the images of the sequence, Δt , is chosen according to v .

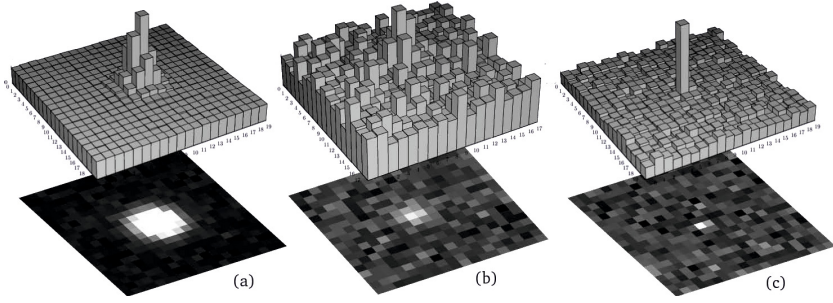


Fig. 2. Examples of light sources in a image: (a) bright star, (b) faint star and (c) a noise artifact (cosmic ray)

Source Extraction. The first step in the image processing pipeline is extracting the location of light sources, including stars and possible moving objects. In this work we explore two options: SExtractor (Source Extractor library) [3], which uses neural network algorithm to find point light sources and also other astronomic features; and DAOPHOT [4] a source extraction method which finds the location of sources by filtering the image with a gaussian convolution and then extracting the local maximums which values are above a threshold. A performance comparison between both methods is presented in section 3.

Some of the detected features in the images do not correspond to point light sources, but are rather artifacts of noise like cosmic rays or defective pixels in the CCD sensor. To filter out these spurious detections, we analyze each feature's roundness and sharpness parameters.

Let $\mathcal{N}(i_0, j_0)$ be a neighborhood around (i_0, j_0) . The sharpness of a feature centered in (i_0, j_0) is defined as $\text{sharp}_{i_0, j_0} = \frac{D_{i_0, j_0} - \langle D_{i, j} \rangle}{H_{i_0, j_0}}$, where $\langle D_{i, j} \rangle$ is the mean of the $D_{i, j}$, $(i, j) \in \mathcal{N}(i_0, j_0) - \{(i_0, j_0)\}$ and H_{i_0, j_0} is a filtered version of D_{i_0, j_0} . The roundness parameter is defined as $\text{round} = 2 \frac{h_y - h_x}{h_y + h_x}$ where h_y and h_x are one-dimensional convolutions of D corresponding to the gaussians: $g_x(\Delta i; \sigma) = e^{-\frac{\Delta i^2}{\sigma^2}}$ and $g_y(\Delta j; \sigma) = e^{-\frac{\Delta j^2}{\sigma^2}}$, respectively.

In Figure 3(a) the distribution of sharpness/roundness of detected features in the images can be seen. Figure 3(b) shows the detected light sources (green) and the spurious detections (red) filtered by sharpness/roundness parameters.

Photometry. The goal of the photometry reduction is to obtain a light flux estimation for each detected source, which represents the number of photons (amount of light) received in an area near the position of the source subtracting the background. This flux is stable over the frames for the same celestial body so it can be used as a descriptor of the source that distinguishes each other. Moreover, to eliminate the effect of the atmosphere the air mass has to be subtracted.

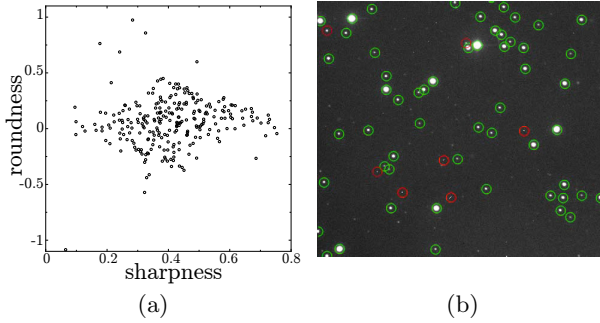


Fig. 3. (a) Sharpness/roundness distribution of detected features and (b) detected light sources (green) and the spurious detections (red) filtered by sharpness/roundness

To estimate the background of the whole image we define BG as an erosion filter of D , $BG_{i_0, j_0} = \min_{(i, j) \in \mathcal{N}(i_0, j_0)} D_{i, j}$. Then, the light flux of one source placed in (i_0, j_0) can be calculated as follows: $FLUX_{i_0, j_0} = \sum_{(i, j) \in \mathcal{N}(i_0, j_0)} (D_{i, j} - BG_{i, j})$.

In sparsely populated areas, we can calculate the flux from a source without estimating the background, taking as reference baseline a ring around the source. In this case the $FLUX_{i_0, j_0}$ can be defined as:

$FLUX_{i_0, j_0} = \sum_{(i, j) \in \text{circ}(i_0, j_0)} D_{i, j} - \sum_{(i, j) \in \text{ring}(i_0, j_0)} D_{i, j} \times \frac{\#\text{circ}(i_0, j_0)}{\#\text{ring}(i_0, j_0)}$, where $\text{circ}(i_0, j_0)$ and $\text{ring}(i_0, j_0)$ are circular and an annular regions around (i_0, j_0) , respectively.

Alignment. The sequence of acquired images is obtained under the same sky coordinates. The robotic telescope corrects for the rotation of the earth as time passes, but still some mechanical errors accumulate. The result is that the light sources appear displaced a few pixels in both X and Y coordinates or even slightly rotated relative one image to another. Then, the sequence of images need to be aligned. The scale of the images does not change because the focal length of the telescope is fixed.

Since the misalignment of the images is given in three degrees of freedom (movement in X , Y and rotation). The problem is reduced to finding a rigid transformation to bring all images to the same reference system. The method ICP (Iterative Closest Point) [5] assumes that the features are not very far from their original position, which can be assumed in our case because we want to correct alignment errors due to perturbations in the motion of the telescope. The idea of this method is to establish a correspondence between each feature of an image with the feature closest to it in the second image. Then the transformation that fits all correspondances best is computed using RANSAC (Random Sample and Consensus).

Moving Object Detection. Since the objects of interest move at an approximately constant speed and the field of vision of the telescope is small (long focal

length), we can consider that the trajectory of the object describes a straight line. Then, the problem of finding moving objects is then reduced to detect a sequence of features that are collinear and are spacially distributed proportionally to the time intervals of the frames¹. If we think of the features as points in a three dimensional space with coordinates (x, y, t) , we are looking for a sequence of collinear points in this space (see Figure 4).

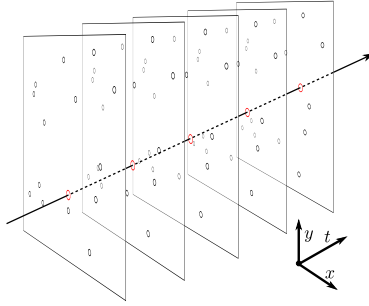


Fig. 4. Set of features aligned in space-time coordinates, corresponding to a moving object along the images sequence of frames.

The first step is discarding the features that appear repeatedly in the same position (x, y) along the sequence of frames, i.e., corresponding to sources that do not move (mostly stars). Some of the features that remain correspond to noise and artifacts that randomly appear in a frame, and some of them might correspond to a moving object that appears at different positions along the sequence of frames. The challenge is to find which features are aligned in a trajectory, in an efficient way.

We also have to keep in mind that we might possibly detect the object only in a subset of the frames, so we have to search for collinearities in different subsets of frames. To achieve this we choose three frames (f_1, f_2, f_3) randomly and seek for triplets of features which are collinear in (x, y, t) .

For each pair of features (p_1, p_2) , $p_1 \in f_1$ and $p_2 \in f_2$, we calculate the spatial displacement vector $p_1 \rightarrow p_2$ and estimate where we should find a feature $\hat{p}_3 \in f_3$ to complete the collinear triplet. Then we compare each estimated \hat{p}_3 with each real $p_3, \forall \hat{p}_3, p_3 \in f_3$, using nearest neighbor matching in high-dimensional spaces (implemented in the FLANN library [6]), to find which pairs (p_1, p_2) have a p_3 in f_3 that completes a space-time collinear triplet.

Once we have these possible traces of three points we calculate the straight line that joins them. Then, we evaluate the remaining frames to find out whether they have features on this line. If the moving object is detected in all frames, a feature in each frame will be found that satisfies the equation of the line. In practice, it is very difficult to find the same feature in all frames, however, if a feature is found in a few frames we can be pretty sure that this feature

¹ When moving at constant speed, the distance traveled between two frames is proportional to the time difference of these frames.

corresponds to a moving object. Typically, four frames are enough to have very good certainty of a positive discovery. We repeat this process a many times (approximately the total number of frames) with random triplets of frames, and merge the found traces if they belong to the same object (see Algorithm 1).

Algorithm 1. Pseudocode to find collinear traces of features.

```

traces  $\leftarrow$   $\emptyset$ 
for  $i=0$  to #frames do
    frame1, frame2, frame3  $\leftarrow$  random_sample(frames, 3)
    factor  $\leftarrow$   $\frac{t_{f3} - t_{f2}}{t_{f2} - t_{f1}}$ 
    triplets  $\leftarrow$  get_linear_triplets(features(frame1), features(frame2),
    features(frame3), factor)
    foreach triplet  $\in$  triplets do
        line  $\leftarrow$  fit_line(triplet)
        trace  $\leftarrow$  {triplet[1], triplet[2], triplet[3]}
        foreach frame  $\in$  frames  $\setminus$  {frame1, frame2, frame3} do
            foreach feature  $\in$  frame do
                if feature  $\propto$  line then
                    trace  $\leftarrow$  trace  $\cup$  {feature}
            if  $\exists$  trace'  $\in$  trace  $\cong$  trace' then
                merge(trace, trace')
            else
                traces  $\leftarrow$  traces  $\cup$  {trace}
return traces

```

As a result we obtain sets of traces, i.e. features that are aligned in space and time corresponding to objects moving in a straight line at constant speed. The equations for finding the line that best fits each trace of feaures form an overdetermined system, so we use the least squares technique. This line gives an estimation of the position and the velocity of the moving object.

3 Results

In this section we present some results of this work. First a comparison between DAOPHOT and SExtractor methods is considered. Figure 5 shows the performance of each method and the results of the photometry reduction (flux estimation). Figure 5(a) shows the execution time of each method as a function of the image size with a constant number of stars (300 aprox.). Figure 5(b) is the same with constant star density (20 stars for each 100 \times 100 pixels). As can be seen, if the density of star is constant, the behaviour of the DAOPHOT method is better than that of SExtractor. Figures 5(c) and 5(d) show the photometry computation (flux estimation) using DAOPHOT and SExtractor, respectively. In both cases, flux estimation does not have large variations, allowing the detection of the point light sources in different frames.

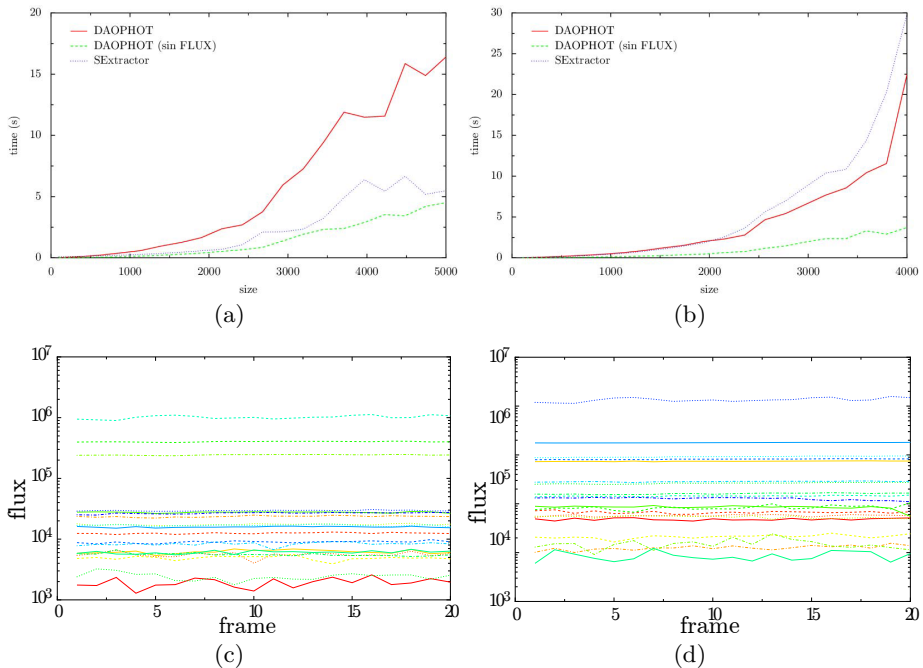


Fig. 5. Execution time comparison between DAOPHOT and SExtractor: (a) for different image size and constant amount of stars, and (b) for different image sizes and constant star density. Flux estimation using (c) DAOPHOT and (d) SExtractor.

For the case of real images acquired with a telescope, very satisfactory results are obtained. As test data set, we use image sequences containing moving objects from the CASLEO Observatory located in the Leoncito, San Juan, Argentina. Figure 6(b) shows the result of applying our proposed method to an image sequence within the data set corresponding to an observation of the asteroid 41427. This asteroid was correctly detected and another object (on the left side) was also found that is virtually imperceptible to the human eye.

To calculate the transformation between image coordinates and sky coordinates we use Astrometry.net [7]. Once we get the sky coordinates of the detected celestial bodies, we can compare the results with USNO-B1 catalog. In Figure 6(a) the positions of the detected and the catalogued celestial bodies, marked with circles and crosses, respectively, are plotted. In most cases they coincide, indicating a good alignment of the image. Then, we can conclude that mobile objects' sky coordinates were calculated accurately.

4 Conclusions

In this work we present a method that automates the detection of moving objects in the night sky. To extract the locations of the point light sources we analyze

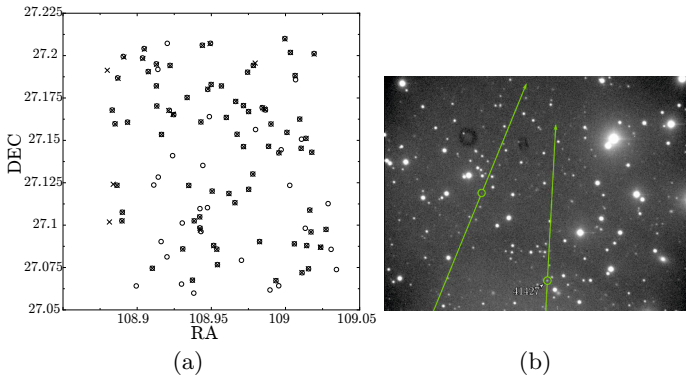


Fig. 6. (a) Positions (in sky coordinates) of the detected and the catalogued celestial bodies, marked with circles and crosses, respectively. (b) Discovered moving objects in the image sequences from CASLEO data set, one of them is the asteroid 41427.

both DAOPHOT and SExtractor methods and compare them. We propose an algorithm for moving object detection in image sequences, which operates looking for collinearity in the sets of the detected point sources. Experimentation with synthetic and real images shows that this method is very effective for the detection of asteroids and even other faint objects. Moreover, it can be used with a robotic telescope to achieve an autonomous system for astronomical discoveries.

References

1. Tody, D.: The iraf data reduction and analysis system. In: 1986 Astronomy Conferences, International Society for Optics and Photonics, pp. 733–748 (1986)
2. Starck, J., Murtagh, F.: Astronomical image and data analysis. Astronomy and astrophysics library. Springer (2002)
3. Bertin, E., Arnouts, S.: SExtractor: Software for source extraction. Astronomy & Astrophysics, Supplement, 393–404 (June 1996)
4. Stetson, P.B.: Daophot: A computer program for crowded-field stellar photometry. Publications of the Astronomical Society of the Pacific, 191–222 (1987)
5. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision (2), 119–152 (1994)
6. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: International Conference on Computer Vision Theory and Application, VISSAPP 2009, pp. 331–340. INSTICC Press (2009)
7. Lang, D., Hogg, D.W., Mierle, K., Blanton, M., Roweis, S.: Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. The Astronomical Journal (5), 1782 (2010)