A-Track: A New Approach for Detection of Moving Objects in FITS Images

T. Atay[a], M. Kaplan[a], Y. Kilic[a], N. Karapinar[a]

[a]*Akdeniz University, Department of Space Sciences and Technologies, Antalya, Turkey*

Abstract

We have developed a fast, open-source, cross-platform pipeline, called A-Track, for detecting the moving objects (asteroids and comets) in sequential telescope images in FITS format. The pipeline is coded in Python 3. The moving objects are detected using an improved line detection algorithm for asteroids, called ILDA. We tested the pipeline on astronomical data acquired by an SI-1100 CCD with a 1-meter telescope at TUBITAK National Observatory, Antalya, Turkey. We found that A-Track performs quite well in terms of detection efficiency, stability, and processing time. The code is hosted on GitHub under the GNU GPL v3 license.

Program summary
Program title: A-TRACK
Catalogue identifier: a-track v1.0
Program summary URL: https://github.com/akdeniz-uzay/A-Track
Program obtainable from: https://github.com/akdeniz-uzay/A-Track
Licensing provisions: Standard GNU General Public Licence, http://www.gnu.org/licenses/gpl-3.0.en.html
No. of lines in distributed program, including test data, etc.: 1027
No. of bytes in distributed program, including test data, etc.: 134262784
Distribution format: .zip
Programming language: Python3
Computer: Personal Computer
Operating system: Any OS where Python 3 are installed.

Subprograms used: Numpy, Pandas, SExtractor, PyFITS, Alipy, f2n, docopt

Keywords: methods: data analysis, minor planets, asteroids, techniques:

Email address: `atay@akdeniz.edu.tr` (T. Atay)

image processing

2015 MSC: 10-13, 99-00

1. INTRODUCTION

Many research groups from all over the world contribute their best efforts to detect and track asteroids and comets. There are three main reasons behind this interest in detection of these moving objects: 1) They are the debris left over from the process that formed our solar system. Studying them gives us information about the formation of the solar system. 2) Our planet is hit by tons of particles everyday. Even though most of them are so small that they are destroyed in the atmosphere before they can reach the ground, relatively large objects do strike the Earth and cause significant damage ocassionally [Tunguska meteor (60m) 1908, Chelyabinsk meteor (20m) 2013]. Such potentially dangerous objects should be carefully studied to understand their sizes and trajectories. 3) Asteroids and comets offer a rich supply of minerals and raw materials. We can benefit from studying their compositions and structures.

Most of these research groups use robotized telescopes with high-resolution cameras and optical systems to scan the sky all night long. However, to our best knowledge, none of these groups use an open-source/free software that detects asteroids and comets automatically. Some independent researchers prefer commercial software such as Astrometrica (Raab, 2012) or CoLiTec (Savanevich et al., 2012). These are relatively easy to use, yet inadequate for large archives as they need human interaction.

In this work, we introduce a new Improved Line Detection Algorithm for the detection of celestial moving objects, called ILDA. The idea comes from Chen's Randomized Line Detection (RLD) algorithm (Chen and Chung, 2001). We modified and improved RLD to make it applicable to sequential telescope images in FITS format. Our algorithm needs at least 3 sequential images of the same sky region taken one after the other. Even though the images don't have to be taken on the same night, the duration of the observation should not be too long as an object to be detected can leave the region.

We used ILDA to develop an open-source (licensed under GPL v3), cross-platform pipeline called A-Track. It is easy to use, it is fast, and it doesn't need human interaction while running. The coding was done in Python 3 as it is easy to understand and has many astronomical modules to work with. It will run on any operating system where Python 3 and the rest of the dependencies are installed. We tested A-Track on GNU/Linux (Fedora 20, Ubuntu 14.04) and Mac OS X (10.10.4, Yosemite) without any problems.

The workflow of the pipeline, the packages used, and details about the developing stage are given in sections 2,3,4. In section 5, we present some test results, moving objects detected by A-Track on two data sets.

2. A-TRACK: OVERVIEW

In principle, the moving objects in sequential CCD images can be identified by tracking their motion with respect to the stationary objects (stars). One requirement is that the moving object is detected by the telescope/CCD system in most (if not all) of the images. Another requirement is that the moving object is fast enough that its motion from image to image is detectable. Our algorithm is based on the idea that, over short enough intervals of observation time (which can be days or even weeks depending on the position and speed of the object) the trajectory of a moving celestial object can be approximated by a line segment.

The workflow of A-Track is shown in Fig. 1. The program consists of five main files: atrack.py, sources.py, asteroids.py, visuals.py, and atrack.config. atrack.py is the main file to be executed with Python 3 via the command-line. On any OS where all dependencies are installed, it can be invoked with

```
python3 atrack.py <fits_dir> [--ref=<ref_image>]
          [--skip-align] [--skip-cats] [--skip-pngs] [--skip-gif]
```

`<fits_dir>` is the path of the directory where the telescope images to be analyzed are located. Unless the `--ref=<ref\_image>` option is used, the first image file in lexicographical order is used as the reference image for alignment. The rest of the options will be discussed in the following sections.

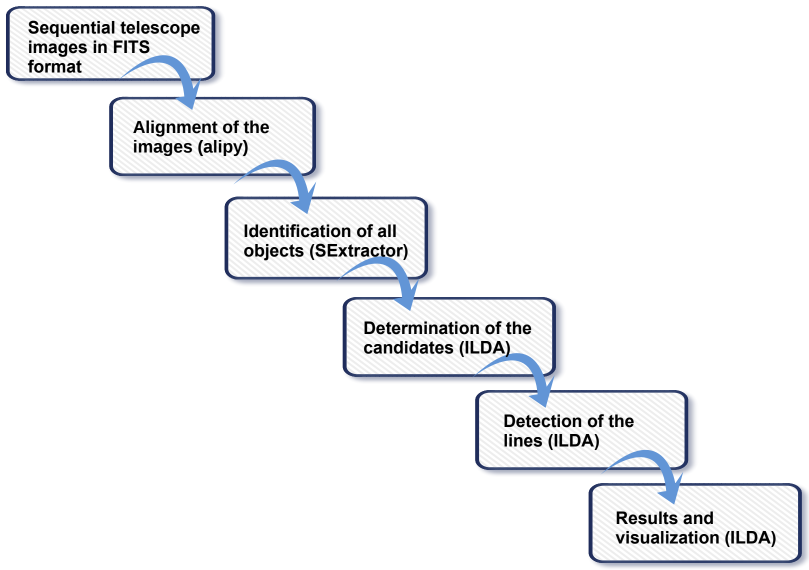

Sequential telescope images in FITS format
Alignment of the images (alipy)
Identification of all objects (SExtractor)
Determination of the candidates (ILDA)
Detection of the lines (ILDA)
Results and visualization (ILDA)

Figure 1

sources.py, asteroids.py, and visuals.py are the three modules imported by atrack.py to handle the alignment of the images and the identification of the objects, the detection of the moving objects, and the visual outputs, respectively. Usually, the detection of the moving objects is the most CPU-demanding step. Therefore, to reduce the processing time, ILDA was coded using the parallel processing architecture in Python 3.

All of the parameters explicitly used by A-Track can be found, together with their descriptions, in the configuration file atrack.config. Some of the default parameters in this file are specific to the telescope/CCD system used, while the rest of them are purely empirical. The user is expected to use the parameter values appropriate for his/her own images.

All of the FITS images in `<fits_dir>` are used for detection. Also, unless it already exists, a directory named atrack is created in `<fits\_dir>` where all outputs from A-Track for that particular image set are saved.

3. IMAGES, ALIGNMENT, and OBJECT IDENTIFICATION

There are a few requirements for the images to be used with A-Track: They need to be in FITS format. The FITS-headers need to comply with the FITS standarts (**?**) as the key parameters DATE-OBS, EXPTIME, XBIN, YBIN, NAXIS1, and NAXIS2 are explicitly used by ILDA. A-Track needs at least 3 sequential FITS images to run. Usually, 5 to 10 images are enough to detect all of the moving objects detectable (limited by the image quality) in the images. It can work with raw images (no pre-reduction is needed). It assumes the images are lexicographically ordered in time. The x and y binnings of the images need to be the same. The parameters in the [sources] section of the configuration file are used for alignment and identification.

To align the images, A-Track uses the Python module alipy (Tewes, 2015), selecting one of the images as the reference. alipy first determines the affine transformation matrices between the images and the reference, following Lang's method (Lang et al., 2010). It then uses the affine_transform() function of the scipy module to align all of the images with respect to the reference. The aligned images are saved in the atrack/ directory in FITS format with the suffix '_affineremap'. If the images to work with are already aligned, this process can be skipped using the `--skip-align` option in the command-line interface.

A-Track uses the latest stable version 2.19.5 of SExtractor to identify the (light) sources in the CCD images. It is invoked via the pysex submodule (wrapper) in alipy. The default values for the user-defined parameters used for the identification process can be found in the [sources] section of the configuration file. For each image, a catalog file with the extension .pysexcat is created in the atrack/ directory. This catalog file contains the internal flags (FLAGS), the x and y coordinates (X_IMAGE, Y_IMAGE), the flux and its RMS error (FLUX_AUTO, FLUXERR_AUTO), the local sky background

(BACKGROUND), the FWHM (FWHM_IMAGE), and the elongation (ELONGATION) for each object identified by SExtractor in that particular image (Bertin and Arnouts, 1996). If the objects in the images are already identified, this process can be skipped using the `--skip-cats` option in the command-line interface.

Once all objects in all images are identified, that is, a catalog file is created for each image, A-Track treats the objects as points for the purpose of moving object detection, with coordinates given in the respective catalog file. We will use the words 'object' and 'point' interchangeably in the rest of the paper.

4. MOVING OBJECT DETECTION

4.1. Elimination of Stationary Objects

The first step in moving object detection is to eliminate the objects that are clearly stationary from the catalogs so that one is left with the moving object candidates for further analysis. The elimination process is based on the user-defined parameters found in the [asteroids] section of the configuration file. In addition to truly moving objects, these candidates may contain cosmic rays as well as objects that appear to move due to atmospheric seeing. Once the candidates are determined for each image, the internal flags (FLAGS), the x and y coordinates (X_IMAGE, Y_IMAGE), the flux (FLUX_AUTO), and the local sky background (BACKGROUND) values for these candidates are saved in the atrack/ directory as a file with the extension .cnd.

To decide whether or not an object is a candidate, A-Track performs several checks. The FLAGS value of a candidate cannot be greater than FLAG_MAX in the configuration file. The default value 31 of FLAG_MAX ensures that the objects with bright and close neighbors, the objects blended with another object, the objects with saturated pixels, and the objects close to or on the image boundary are not discarded during the identification. The FWHM of a candidate cannot be smaller than FWHM_MIN set in the configuration file. The user can set the maximum FWHM via the FWHM_COEFFICIENT parameter in the configuration file (the default value is 2.5). A candidate object's FWHM cannot be greater than FWHM_COEFFICIENT times the average FWHM of all identified objects. The FLUX_AUTO value of a candidate has to be between BACKGROUND and FLUX_MAX, both of which are set in the configuration file. The FLUX_AUTO/FLUXERR_AUTO ratio of a candidate cannot be smaller than SNR_MIN set in the configuration file. The ELONGATION value of a candidate cannot be greater than ELONGATION_MAX set in the configuration file.

To perform the final check, first, the catalog files generated by SExtractor are merged into one large catalog file called master.pysexcat, which is also saved in the atrack/ directory. Then, A-Track checks if the object has the same x and y coordinates (within a tolerance of TRAVEL_MIN set in the configuration file) in more than one image. To do this, it calculates the distances, one by one, between the object and all of the objects in the master catalog file. It uses the Logic Equation EQhnak to determine the number of occurences where the calculated distance is less than TRAVEL_MIN. If this number is 1 for the object, then the object is tagged as a candidate.

$$numberofoccurencesforcandidate = \sum_{objectsinmaster.pysexcat} (\sqrt{((C_x - O_x)^2 + (C_y - O_y)^2)} <= TRAVEL\_MI \tag{1}$$

where C is the candidate being investigated, O is the object in the master.pysexcat file that is being compared, and the subscripts x and y denote the x and y coordinates, respectively. The result of the summand is 1 if the inequality holds, 0 otherwise.

4.2. ILDA: An Improved Line Detection Algorithm

Once the candidates are found for each image (catalog file), ILDA looks for points from different images that would form a line when plotted on the same graph. This is explained in Fig. 2. The lines ILDA seeks will not be on any particular image, but rather will form when all points in all images are plotted on the same graph. Each object on any of these lines will belong to a different image.

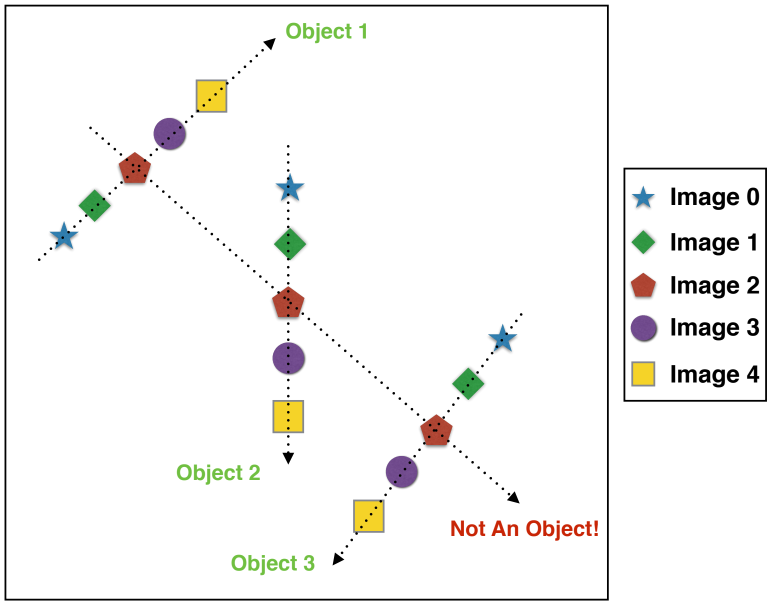


Figure 2

In addition to this forming-a-line requirement, ILDA also applies some time constraints that will be explained below. That is, forming a line is not enough for those objects to be considered as moving objects. They need to be at the right positions on that line. The time information (observation time and exposure time) of the images are read from the FITS-headers. They need to be in `%Y-%m-%dT%H:%M:%S.%f` or `%Y-%m-%dT%H:%M:%S` format. The parameters in the [asteroids] section of the configuration file are used for line detection.

The first step of ILDA is to find all of the 3-point segments that would belong to a kind of line described above. To do this, ILDA investigates all of the possible 3-point combinations obtained from all of the candidate files (.cnd), each point coming from a different image. For each 3-point combination, it performs three checks.

First, it looks at the first two points to perform a distance check. It demands that the distance between these two points is smaller than d_max given by

$$d_{max} = (OT_2 - OT_1 + (ET_2 - ET_1)/2) * v_{max}/(S * xbin) \qquad (2)$$

where OT is the observation time, ET is the exposure time, v_max is the maximum angular velocity of the sought object (V_MAX in the configuration file), S is the pixel scale subtended by the telescope/CCD system (SCALE in the configuration file), xbin is the x-binning of the image the first point belongs to (read from the FITS-header), and the subscripts 1 and 2 denote the first and the second points, respectively. If the distance between the first two points of a 3-point combination is not smaller than d_max, ILDA eliminates that 3-point combination.

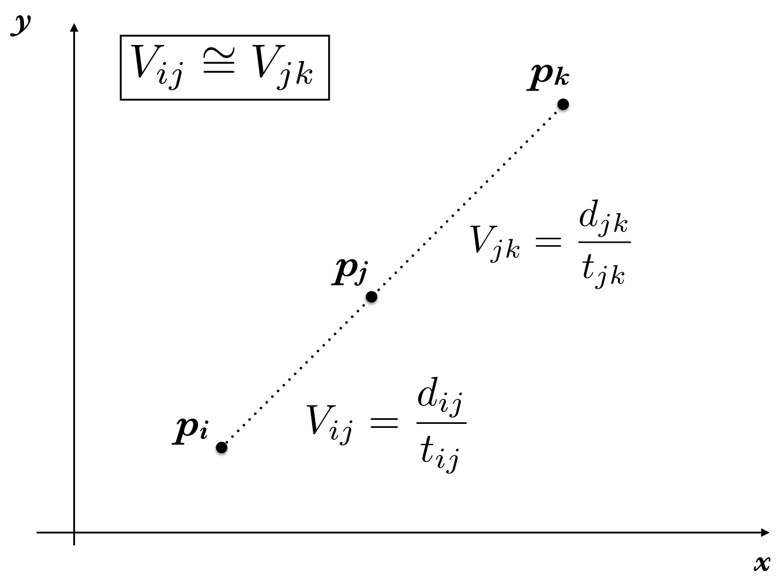
If a 3-point combination passes the distance check, then ILDA performs a speed check. As shown in Fig 3. The speed of the object between points 1 and 2 need to be approximately equal to the speed between points 2 and 3. Thus, ILDA demands that the following inequality is satisfied

$$abs(d_{23} - d_{12}/t_{12} * t_{23}) <= TOL \qquad (3)$$

where d_ij and t_ij are the distance and the time, respectively, between points i and j and TOL is the tolerance in pixels (TOLERANCE in the configuration file).

If a 3-point combination passes both checks (distance and speed), then ILDA performs a final check, collinearity. Considering the triangle formed by these 3 points, ILDA demands that the length of the longest side of this triangle is greater than 2*TRAVEL_MIN and the height associated with this side is smaller than HEIGHT_MAX, as shown in Fig. 4. Both of these parameters are set in the configuration file.

Once ILDA determines all of the 3-point segments that pass the distance, the speed, and the collinearity criteria, it merges the segments that belong to the same line. In the end, there will be as many lines as the number of moving



y
$\boxed{V_{ij} \cong V_{jk}}$
p_k
$V_{jk} = \frac{d_{jk}}{t_{jk}}$
p_j
p_i
$V_{ij} = \frac{d_{ij}}{t_{ij}}$
x

Figure 3

objects in the images. Each line, hence each moving object, is assigned an ID number (starting from 1). The moving objects are divided into two groups: FAST and SLOW. If the speed of a moving object is smaller than SPEED_MIN set in the configuration file, then it is classified as SLOW. Otherwise, it is classified as FAST. The default value for SPEED_MIN is 0.1 (pixels/min). The detected objects with speeds smaller than this value are usually not moving objects but artifacts. ILDA leaves the decision to the user.

The detected moving objects are printed on the screen and saved in the results.txt file in atrack/ directory. The first few lines of a sample results.txt file are shown in Fig 5. Each point on the detected lines is included in the result. ObjectID is the ID number of the object (line) the point belongs to. Hence, the highest ObjectID is the number of different moving objects detected in the images. FileID is the number of the image file the point belongs to (the image files are numbered in lexicographical order, starting from 0). Flags is the

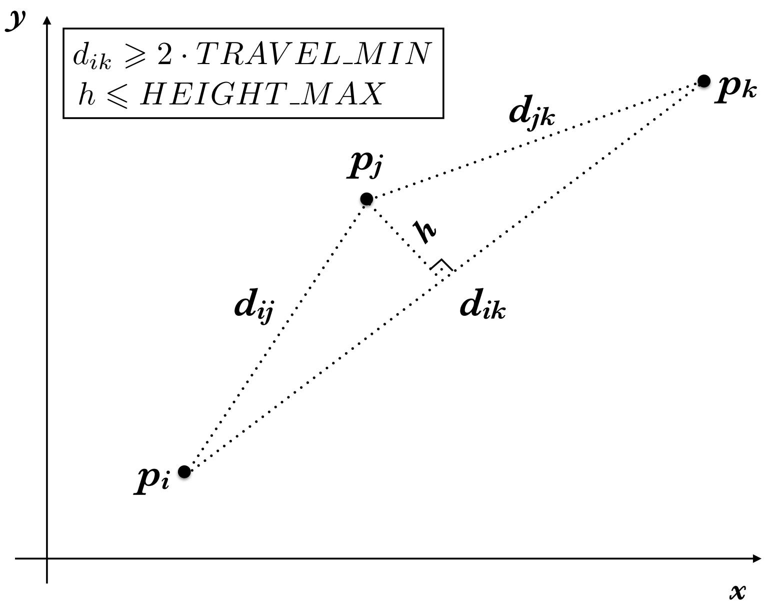
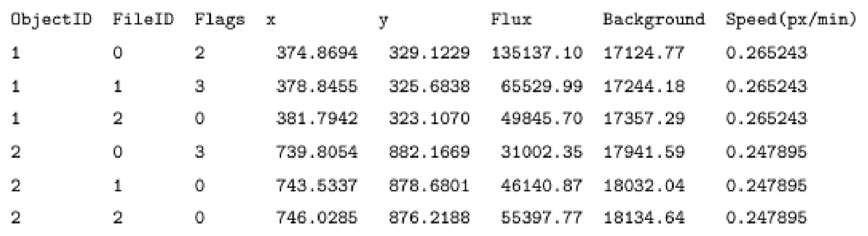

$d_{ik} \geqslant 2 \cdot TRAVEL\_MIN$
$h \leqslant HEIGHT\_MAX$
y
p_k
d_{jk}
p_j
h
d_{ij}
d_{ik}
p_i
x

Figure 4

internal flags code generated by SExtractor for that particular point. x and y values are the x and y coordinates of the point in the (aligned) image it belongs to. Flux, Background, and Speed values for the point are also included in the result.



```
ObjectID  FileID  Flags  x          y          Flux       Background  Speed(px/min)
1         0       2       374.8694   329.1229  135137.10  17124.77    0.265243
1         1       3       378.8455   325.6838   65529.99  17244.18    0.265243
1         2       0       381.7942   323.1070   49845.70  17357.29    0.265243
2         0       3       739.8054   882.1669   31002.35  17941.59    0.247895
2         1       0       743.5337   878.6801   46140.87  18032.04    0.247895
2         2       0       746.0285   876.2188   55397.77  18134.64    0.247895
```

Figure 5

4.3. Visualization

Once the detection of moving objects is complete, the user can check the results in visual form as well. The only parameter used for visualization is SPEED_MIN in the [visuals] section of the configuration file. Using the f2n module, A-Track converts each aligned image into PNG format, encircling each of the detected moving objects on the image. For each aligned image, a file is created in the atrack/ directory with the extension .png. As seen in Fig 6, green and red circles are used to mark FAST and SLOW moving objects, respectively, and the ObjectID associated with the object is printed next to the circle. The PNG conversion can be skipped using the `--skip-pngs` option in the command-line interface.

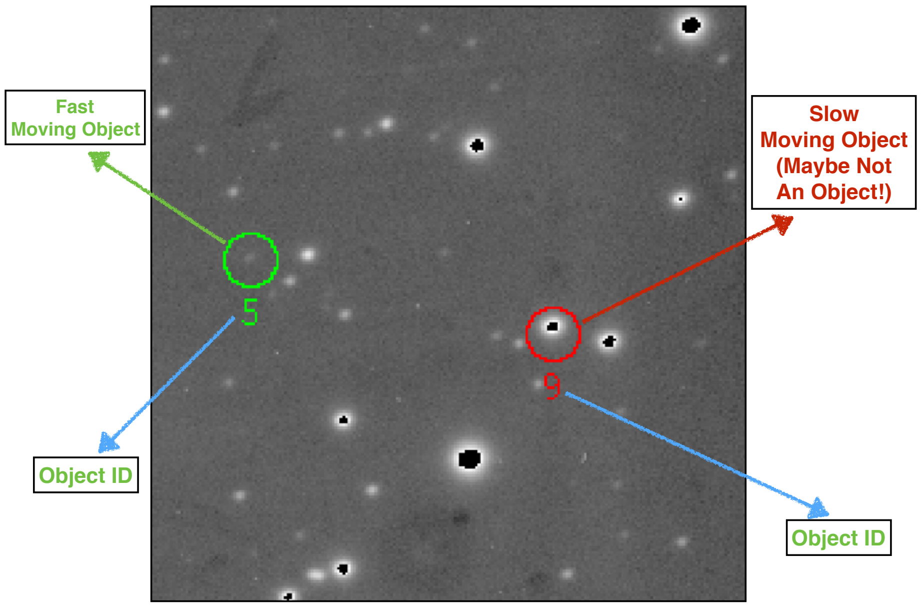


Figure 6

Finally, unless the `--skip-gif` option is used in the command-line interface, an animated GIF file, named animation.gif, is created in the atrack/directory. This file shows the detected moving objects in motion.

5. RESULTS AND DISCUSSION

The parameters used by ILDA that affect the results most significantly are TRAVEL_MIN, HEIGHT_MAX, TOLERANCE, and SNR_MAX. To test the performance effects of these parameters, we ran A-Track on two sets of sequential telescope images: 2000CA30 and Astronomia, named after the asteroids being observed. The 2000CA30 set corresponds to a dense region of the sky, with more than 5000 identified sources; while the Astronomia set (Nurdan'dan gelecek sonulara bal olarak, was intentionally chosen to be of poor quality) corresponds to a sparse region, with less than 400 identified sources. We decided that these two sets are appropriate for testing A-Track as they contain fast, slow, bright, and faint asteroids, all in the same region.

The images were acquired by an SI-1100 CCD with a 1-meter telescope at TUBITAK National Observatory, Antalya, Turkey (Observatory Code: A84). We used raw images for testing (no reduction). There were 7 sequential images in each set. All images had the same XBIN and YBIN value of 2 and the same NAXIS1 and NAXIS2 value of 2048. The observation times are from 20140205T18:31:33.85 to 20140205T20:03:14.46 for the 2000CA30 set and from 20150308T23:09:11.89 to 20150308T23:23:29.55 for Astronomia. The exposure times for the images are 700-800 seconds in 2000CA30 and 80-100 seconds in Astronomia. The tests were done on a 2.6 GHz IntelCore i5 3230M processor (4 cores) with 2 GB 1600 MHz GDDR3 RAM running GNU/Linux Xubuntu 14.04 LTS.

The default parameters in the [Sources] section of the configuration file were used for these tests. These are the parameters recommended by Bertin to catch the faintest objects in the images. As for the four parameters being tested, we kept three of the parameters at their default values while varying the fourth one. The FAST moving objects are tagged as True or False detections after checking the images visually. The results are given in Tables 1-4.

2000CA30					
TRAVEL_MIN (px)	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (s)
0.1	4	7	430	449	1234
0.2	4	7	2	45	175
0.3	4	7	0	8	87
0.4	4	7	0	4	64
0.5	4	7	0	2	59
0.6	4	7	0	2	57
0.7	4	7	0	0	51
0.8	4	7	0	0	51
0.9	4	7	0	0	53
1.00	4	7	0	0	52
1.10	4	7	0	0	51
1.20	4	7	0	0	53
1.30	4	7	0	0	53
1.40	4	7	0	0	51
1.50	4	7	0	0	50
1.60	4	7	0	0	53
1.70	4	7	0	0	51
1.80	4	7	0	0	53
1.90	4	7	0	0	53
2.00	4	5	0	0	53
2.10	4	5	0	0	53
2.20	4	5	0	0	57
2.30	4	5	0	0	51
2.40	4	4	0	0	52
2.50	4	4	0	0	51
2.60	4	3	0	0	52
2.70	4	3	0	0	52
2.80	4	2	0	0	50
2.90	4	2	0	0	51
3.00	4	2	0	0	51

(a) 2000CA30

Astronomia					
TRAVEL_MIN (px)	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (s)
0.1	3	3	3	38	2
0.2	3	3	2	4	24
0.3	3	2	0	0	23
0.4	3	2	0	0	23
0.5	3	2	0	0	23
0.6	3	2	0	0	23
0.7	3	2	0	0	23
0.8	3	2	0	0	23
0.9	3	2	0	0	23
1.00	3	2	0	0	23
1.10	3	2	0	0	23
1.20	3	2	0	0	23
1.30	3	1	0	0	23
1.40	3	1	0	0	23
1.50	3	1	0	0	24
1.60	3	1	0	0	24
1.70	3	0	0	0	11
1.80	3	0	0	0	11
1.90	3	0	0	0	11
2.00	3	0	0	0	11
2.10	3	0	0	0	11
2.20	3	0	0	0	11
2.30	3	0	0	0	11
2.40	3	0	0	0	11
2.50	3	0	0	0	11
2.60	3	0	0	0	11
2.70	3	0	0	0	11
2.80	3	0	0	0	11
2.90	3	0	0	0	11
3.00	3	0	0	0	11

(b) Astronomia

Table 1

The results for TRAVEL_MIN are given in Table1. TRAVEL_MIN defines the minimum distance a truly moving object has to travel between consecutive images. It can be considered as a speed filter. Its default value is 0.5 (pixels). Decreasing this value will help ILDA catch slower moving objects at the expense of an increased number of false detections, mostly because of the variations in the central coordinates of the stars due to different atmospheric conditions between images. Increasing this value too much will result in fewer number of detections since only fast enough objects will be detected by ILDA.

2000CA30					
HEIGHT_MAX (px)	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (s)
0.05	4	6	0	4	71
0.10	4	7	0	6	69
0.15	4	7	1	11	68
0.20	4	7	1	12	72
0.25	4	7	2	16	72
0.30	4	7	2	17	83
0.35	4	7	2	19	80
0.40	4	7	3	21	84
0.45	4	7	4	27	83
0.50	4	7	4	28	83
0.55	4	7	4	28	68
0.60	4	7	4	28	70
0.65	4	7	4	29	67
0.70	4	7	6	27	69
0.75	4	7	6	28	69
0.80	4	7	6	28	73
0.85	4	7	7	29	73
0.90	4	7	8	28	70
0.95	4	7	9	29	68
1.00	4	7	9	29	68

(a) 2000CA30

Astronomia					
HEIGHT_MAX (px)	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (s)
0.05	3	2	0	0	23
0.10	3	2	0	0	23
0.15	3	2	0	0	23
0.20	3	2	0	0	23
0.25	3	2	0	0	23
0.30	3	2	0	2	23
0.35	3	2	0	2	23
0.40	3	2	0	2	23
0.45	3	2	0	3	23
0.50	3	2	0	3	23
0.55	3	2	0	3	23
0.60	3	3	0	3	23
0.65	3	3	0	3	23
0.70	3	3	0	3	24
0.75	3	3	0	3	23
0.80	3	3	0	3	23
0.85	3	3	5	18	47
0.90	3	3	0	3	22
0.95	3	3	0	3	24
1.00	3	3	0	3	23

(b) Astronomia

Table 2

The results for HEIGHT_MAX are given in Table2. TRAVEL_MIN defines the maximum height a 3-point combination can have in order to be considered as collinear (Fig. 4). Its default value is 0.1 (pixels). Increasing this value will help ILDA catch more moving objects at the expense of an increased number of false/slow detections. Again, this is due to the small variations in the central coordinates of the objects identified by SExtractor.

2000CA30					
TOLERANCE (px)	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (s)
0.25	4	7	0	2	85
0.50	4	7	0	4	82
0.75	4	7	0	6	83
1.00	4	7	0	6	83
1.25	4	7	1	8	83
1.50	4	7	1	9	81
1.75	4	7	1	11	81
2.00	4	7	2	11	80
2.25	4	7	2	13	78
2.50	4	7	2	13	80
2.75	4	7	2	15	70
3.00	4	7	2	14	73
3.25	4	7	3	14	73
3.50	4	7	4	12	70
3.75	4	7	4	13	81
4.00	4	7	4	14	81

(a) 2000CA30

Astronomia					
TOLERANCE (px)	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (s)
0.25	4	7	0	2	85
0.50	4	7	0	4	82
0.75	4	7	0	6	83
1.00	4	7	0	6	83
1.25	4	7	1	8	83
1.50	4	7	1	9	81
1.75	4	7	1	11	81
2.00	4	7	2	11	80
2.25	4	7	2	13	78
2.50	4	7	2	13	80
2.75	4	7	2	15	70
3.00	4	7	2	14	73
3.25	4	7	3	14	73
3.50	4	7	4	12	70
3.75	4	7	4	13	81
4.00	4	7	4	14	81

(b) Astronomia

Table 3

The results for TOLERANCE are given in Table 3. TOLERANCE defines the tolerable spatial shift of the third point during the speed check. Its default value is 1.0 (pixels). Assuming the time information in the FITS-headers are correct, there is no need to increase this value. Increasing this value will result in an increased number of false and slow detections.

2000CA30					
SNR_MAX	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (")
2.5	4	7	0	6	83
5	4	7	0	6	69
7.5	4	7	0	6	83
10	4	7	0	6	68
12.5	4	6	0	6	82
15	4	5	0	6	64
17.5	4	5	0	6	69
20	4	4	0	6	56
22.5	4	3	0	6	49
25	4	3	0	5	49
27.5	4	3	0	5	46
30	4	2	0	6	46
32.5	4	2	0	6	42
35	4	2	0	6	43
37.5	4	2	0	5	40
40	4	2	0	5	42

(a) 2000CA30

Astronomia					
SNR_MAX	MPC	A-Track TRUE	A-Track FALSE	SLOW	PROCESS TIME (")
2.5	3	2	0	0	23
5	3	2	0	0	23
7.5	3	2	0	0	23
10	3	2	0	0	23
12.5	3	2	0	0	23
15	3	2	0	0	23
17.5	3	2	0	0	22
20	3	2	0	0	22
22.5	3	2	0	0	23
25	3	2	0	0	23
27.5	3	2	0	0	23
30	3	2	0	0	29
32.5	3	2	0	0	26
35	3	2	0	0	27
37.5	3	2	0	0	26
40	3	2	0	0	26

(b) Astronomia

Table 4

The results for SNR_MIN are given in Table 4. SNR_MIN defines the minimum signal-to-noise ratio for a moving object. Its default value is 10. Increasing this value will reduce the processing time at the expense of fainter moving objects going undetected.

Information about the positions of the moving objects within the observed regions can be found at the ASTPLOT website (Ref (**?**)). According to ASTPLOT, there were 8 asteroids in our 2000CA30 set at the time of observation. A-Track was able to detect 7 of them, whose apparent magnitudes were lower than 21. As for the Astronomia set, 6 asteroids can be seen at the time of observation. A-Track can detect 3 of them, whose apparent magnitudes are lower than 20.5.

Note that the maximum detectable magnitude depends on the overall image quality and may vary between observations. The performance of A-Track is limited by that of SExtractor, which, in turn, is affected by the atmospheric conditions at the time of observation and the condition of the telescope/CCD system used.

6. CONCLUSION

A-Track is a fast and efficient pipeline that uses an improved line detection algorithm (ILDA) to detect the moving objects (asteroids and comets) in sequential telescope images. It can be used to track or confirm known asteroids and comets or to discover new ones. Occasionally, moving objects that are supposed to be within the observed region will not appear in the acquired images due to very low brightness, poor telescope/CCD conditions, or atmospheric effects. A-Track will detect any moving object provided that the object is caught by the telescope/CCD system. Since A-Track doesn't need human interaction while running, we believe it'll become a useful tool for both professional and amateur observers.

A-Track is still under development. Currently, we are working on automating the World Coordinate System (WCS) transformations of the detected moving objects. We plan on adding an automatic reporting (to Minor Planet Center) functionality as well as a user-friendly GUI to our pipeline in the future.

ACKNOWLEDGMENT

This work was supported by The Scientific and Technological Research Council of Turkey (TUBITAK), Project Number 114F477. We would also like to thank TUBITAK for their partial support in using the T100 telescope under project number 14BT100-648.

References

Bertin E, Arnouts S. Sextractor: Software for source extraction. Astronomy and Astrophysics Supplement Series 1996;117:393–404. URL: `http://adsabs.harvard.edu/abs/1996A%26AS..117..393B`.

Chen TC, Chung KL. A new randomized algorithm for detecting lines. Real-Time Imaging 2001;7(6):473–81. doi:`10.1006/rtim.2001.0233`.

Lang D, Hogg DW, Mierle K, Blanton M, Roweis S. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. The Astronomical Journal 2010;139(5):1782–800. doi:10.1088/0004-6256/139/5/1782.

Raab H. Astrometrica: Astrometric data reduction of CCD images. Astrophysics Source Code Library, 2012. URL: http://adsabs.harvard.edu/abs/2012ascl.soft03012R.

Savanevich VE, Kozhukhov AM, Bryukhovetskiy AB, Vlasenko VP, Dikov EN, Ivashchenko YN, Elenin LV. Program of automated asteroids detection colitec — new features and results of implementation. 2012. URL: http://adsabs.harvard.edu/abs/2012LPI....43.1049S.

Tewes M. a python package to quickly, automatically, and robustly identify geometrical transforms between optical astronomical images. 2015. URL: http://obswww.unige.ch/~tewes/alipy/.