Advanced Image Processing Techniques for Automatic Reduction of GEO Survey Data

Vladimir Kouprianov

Central (Pulkovo) Astronomical Observatory
of Russian Academy of Sciences

196140, 65/1 Pulkovskoye chaussee, St.Petersburg, Russia
E-mail: v.k@bk.ru

Currently one of the most efficient approaches to space surveillance problem, as applied in particular to the GEO ring, consists in using automatic wide-field cameras that perform nightly surveys of the whole GEO region visible from the given site. Whatever are the actual survey strategy and sensor details, such cameras produce considerable amount of data per night. An additional important requirement of fast detection and immediate tracking of newly discovered GEO objects implies that these data are being processed in real time, which is a demanding task for data reduction software.

Apex II is an open general-purpose software platform for astronomical image processing, used as a standard tool for initial data reduction by members of ISON collaboration. Its major focus is on consistent use of advanced automatic algorithms for image pre-processing, object detection and classification, accurate positional and photometric measurements, initial orbit determination, and catalog matching. Here we describe a number of these techniques currently used in Apex II to support scanning observations of the GEO region.

Introduction

First of all, the term “GEO” in this paper refers indeed not only to the geostationary orbit itself but rather to many types of medium and high Earth orbits, including GTO, HEO and others. Techniques described below, although they were initially developed mainly to handle GEO objects, are general enough to deal with optical observations of any types of space objects provided their apparent motion differs from that of field stars. If this requirement is met it is possible to adopt an imaging strategy that allows one to clearly distinguish between Earth-orbiting objects and field stars by their morphological properties in a single CCD image, which greatly increases computational efficiency of data reduction software and reliability of automatic object detection.

The most obvious and important of these properties, described qualitatively, is whether a space object or field star is point-like or extended into “streak” by its apparent motion in the field of view of optical sensor during integration. According to this property, we may divide all images into four classes:

1) point-like field stars and space objects;
2) point-like field stars, trailed space objects;

3) trailed field stars, point-like space objects;
4) trailed field stars and space objects.

The first type of images has the obvious advantage that it does not require any specialized image processing techniques. However, its practical use in space surveillance is limited to space objects with apparent motion similar to stars. Applying the same imaging strategy to all other types of space objects by using very short exposure times reduces sensitivity of optical sensor. Furthermore, as mentioned above, in this case there is no way to distinguish between space objects and field stars in a single CCD frame. This results in a need to process all detections in each frame, which has a great impact on the overall computational efficiency of imaging pipeline.

The second type of imaging strategy which implies observations with sidereal tracking seems to have no sense at all as it also has extremely poor sensitivity of imaging system with respect to space objects.

According to this approach, the only reasonable imaging strategy in space surveillance implies tracking the target space object if its apparent velocity is known (like in follow-up observations of a particular object) or assuming some expected velocity if it is not (like in survey-type and discovery observations). Exposure time is then chosen to be long enough to allow one to distinguish between space objects and field stars and to achieve reasonable sensitivity, but not very long to avoid producing extremely long trails of both field stars and space objects, which impacts accuracy and reliability of data reduction.

As it was mentioned above, techniques described here are applicable to a large class of space surveillance tasks and orbit types when the appropriate imaging strategy is used during observations. However, this paper focuses mainly on reduction of GEO survey data obtained with wide-field optical sensors as one of the most computationally challenging problems.

Figure 1 shows an example of a typical raw CCD image, one of about a thousand produced by a 22-cm aperture 5.5°×5.5° field of view optical sensor of ISON network (Molotov et al., 2010) during a routine nightly GEO survey. One of the key requirements in space surveillance is to minimize a delay between exposure and its final result in the form of space object coordinates. Although the overall data rate (≈8 Gigabytes of pixel data and 500–1000 tracks per night) can be considered very moderate for a modern automated imaging system, ISON sensors are installed at locations with no access to supercomputing resources and even often with very limited Internet connection. Moreover, different sensors have slightly varying characteristics and details of implementation of their parts, which affects various properties of images. All these considerations lead to very rigid requirements for data reduction software to be, on the one hand, quite efficient to be able to process data in real time and, on the other hand, sufficiently flexible and versatile to accommodate to a wide range of input data. Unfortunately, it is hard to satisfy these two requirements simultaneously: excessive optimization often makes the system less adaptable to changing environment, while maximum flexibility entails an extra overhead of handling endless possibilities. Thus data reduction software should elaborate a sort of compromise to be able to quickly produce a large amount of reliable data.

Apex II Pipeline for Automatic GEO object detection

Apex II (Devyatkin et al., 2010) is a general-purpose platform for astronomical image processing, modeled after such well-known scientific data analysis packages as IRAF, MIDAS, IDL, and MATLAB. It is implemented mostly in Python, a high-level versatile object-oriented scripting programming language widely adopted by the scientific community. Apex II has easily extendable modular structure consisting of (i) library of astronomical data reduction algorithms and (ii) scripts (high-level interpreted programs) for specific data reduction tasks. Design of Apex II and its applications in low-level analysis of data produced in observations of Earth-orbiting objects are outlined in Kouprianov (2008). Most algorithmic details are also covered in that paper. Here we concentrate only on the most important features and latest algorithmic developments.

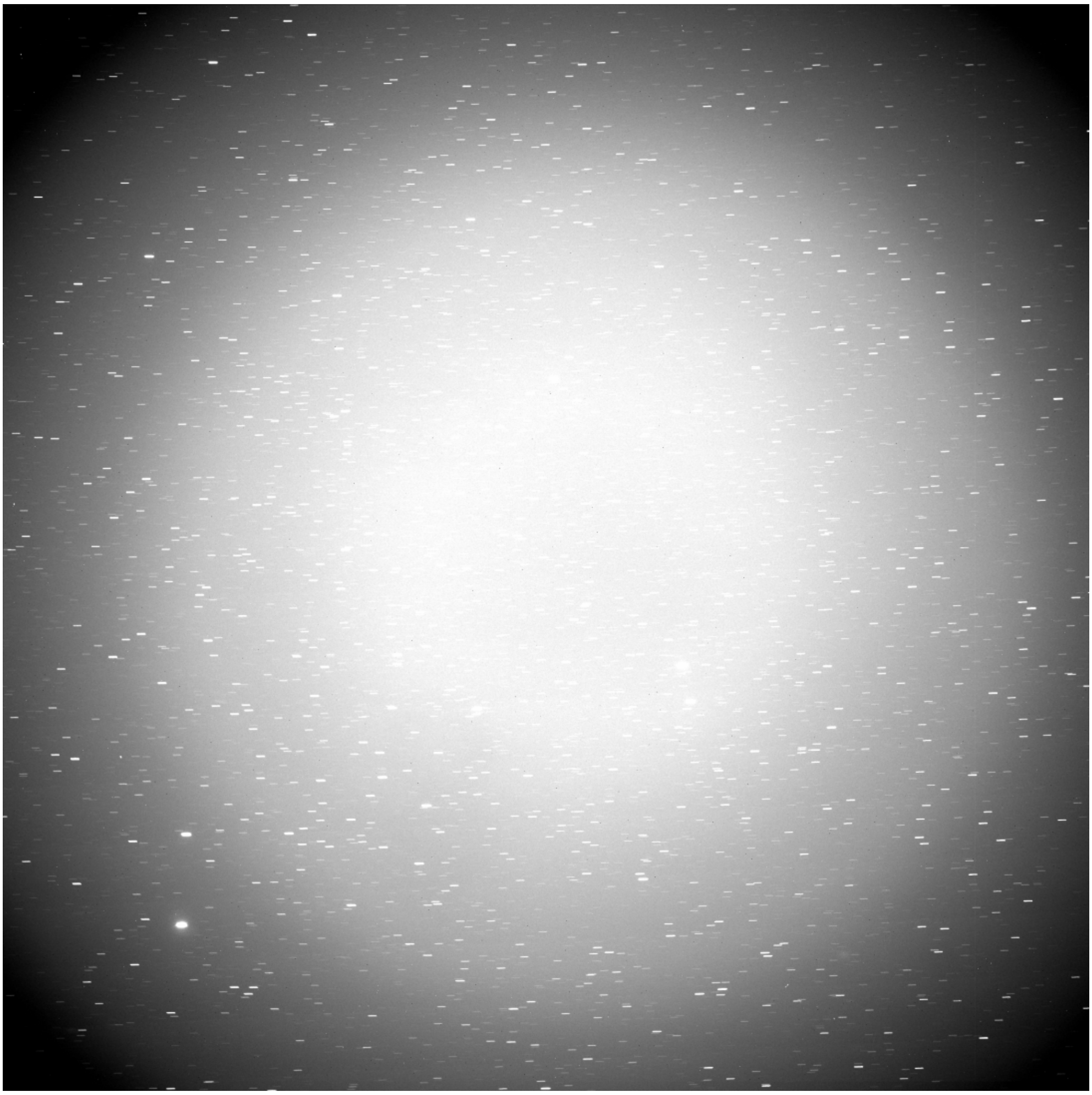

Figure 1. Sample raw CCD image from GEO survey.

Raw CCD image shown in Figure 1 displays several instrumental artifacts that need to be removed. The most obvious (and most annoying) of them is vignetting characteristic to many wide-field systems. As long as accurate CCD photometry is not required, it is sufficient to only flatten background to allow segmentation (separation of objects from background) by simple global threshold; it does not matter whether background comes from sky or vignetting or non-uniformity of CCD chip. Figure 2 shows the result of calibration of the same image; a fast automatic sky background estimation algorithm involved is described in (Kouprianov, 2008). The ultimate goal of initial image processing is to detect space objects (shown enlarged in Figure 2) and accurately determine their position in terms of α and δ.

To achieve this, one needs first to detect reference stars and perform astrometric reduction. Given (i) LSPC (least-squares plate constants) solution and (ii) XY positions of detected GEO objects, their $\alpha\delta$ positions are obtained straightforwardly. $\alpha\delta$ positions of space objects are then (iii) correlated across several adjacent images of the same sky region to obtain tracks of GEO objects and eliminate false detections. These three major stages comprise Apex II pipeline for automatic reduction of space object observations. In the following section we highlight several difficulties that arise in astrometric reduction of CCD images from GEO surveys.

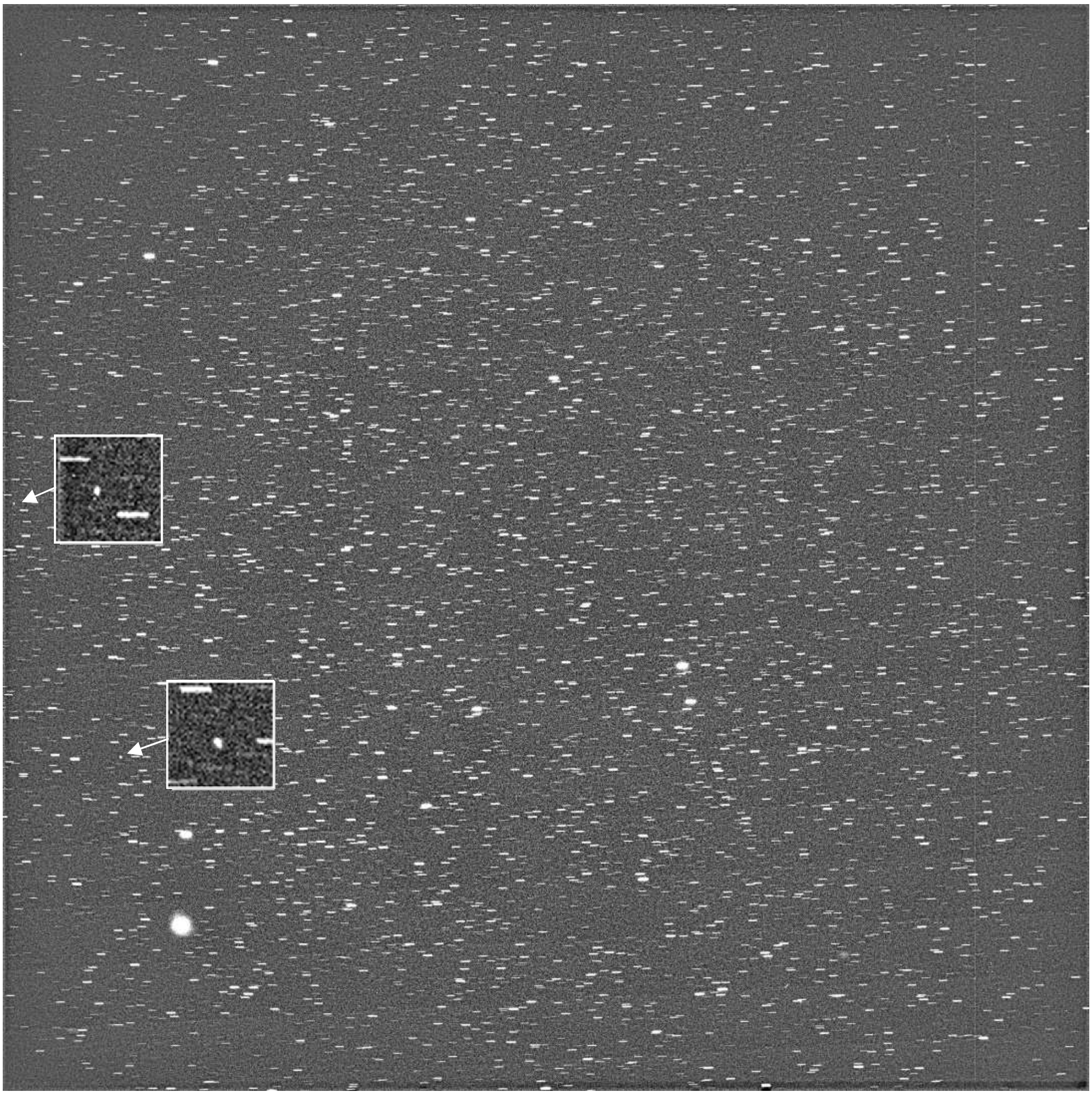

Figure 2. Calibrated CCD image. Regions around two GEO objects are shown enlarged.

Reference Star Detection and Astrometry

As it was mentioned above, the characteristic feature of CCD images from GEO surveys are field stars appearing as streaks. Although this is good for detecting space objects, this apparently complicates dealing with reference stars themselves. Things like atmospheric turbulence, extinction fluctuations, noise, and optical aberrations distort star trails. As a result, a naïve global threshold approach often fails to detect the whole trail, especially for stars close to detection threshold, mostly due to their fragmentation. This is illustrated in Figure 3.

To reduce fragmentation, we use a special kind of binary morphological filter that utilizes properties of star trail shapes known before processing: their length and orientation are easily calculated from pixel scale, exposure duration, and tracking rate, while trail width can be estimated from pixel scale and seeing. The following equation defines the result of filtering:

$$M^{'}(x,y) = \begin{cases} 1 & \text{if } (M \ast K)(x,y) > d \sum K, \\ 0 & \text{otherwise,} \end{cases} \qquad (1)$$

where $M(x,y)$ is the original unfiltered binary image, $M'(x,y)$ is filtered image, "*" denotes convolution, d is filter strength parameter, and filter kernel K is defined as follows:

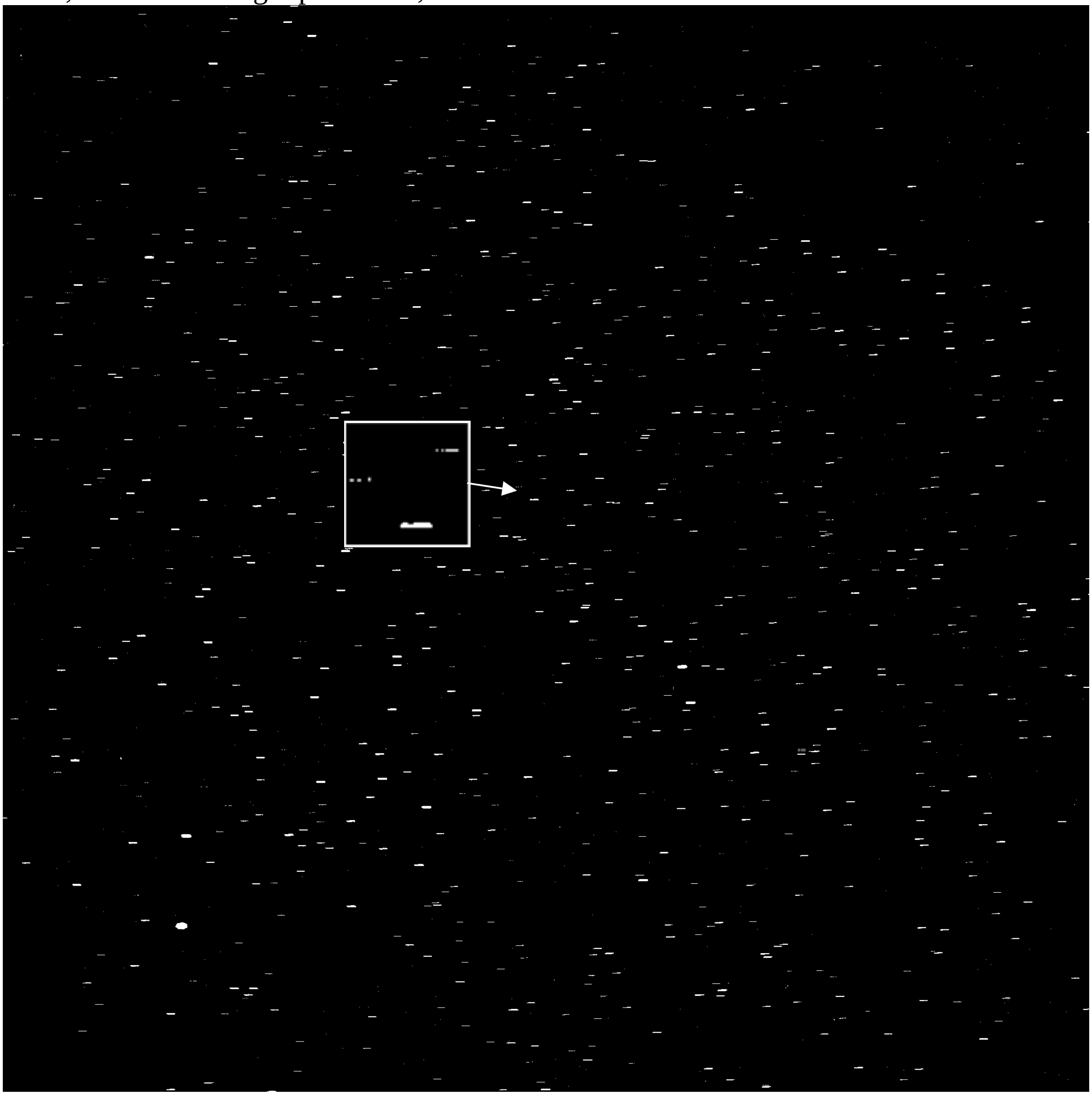

Figure 3. Binary CCD image after segmentation by global threshold: background → black, objects → white. Enlarged region containing three star trails illustrates the effect of fragmentation.

$$K = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{2}$$

In other words, 1's in the filter kernel reproduce the estimated shape of a star trail, with its length and width; if star trail orientation differs from 0 or 180°, kernel is rotated accordingly.

Effect of the above filter is shown in Figure 4. By tuning d, one can achieve the desired balance between the number of fully detected real star streaks and the number of artifacts detected as stars.

We should also mention that the global threshold for reference star detection is chosen automatically based on image histogram. The idea behind this is quite simple: we just select such grayscale level that the number of pixels brighter than this level constitutes the fixed fraction of the total image area. Thus we get always the same fraction of image covered by reference stars, which helps to adapt to varying atmospheric conditions and stellar field densities.

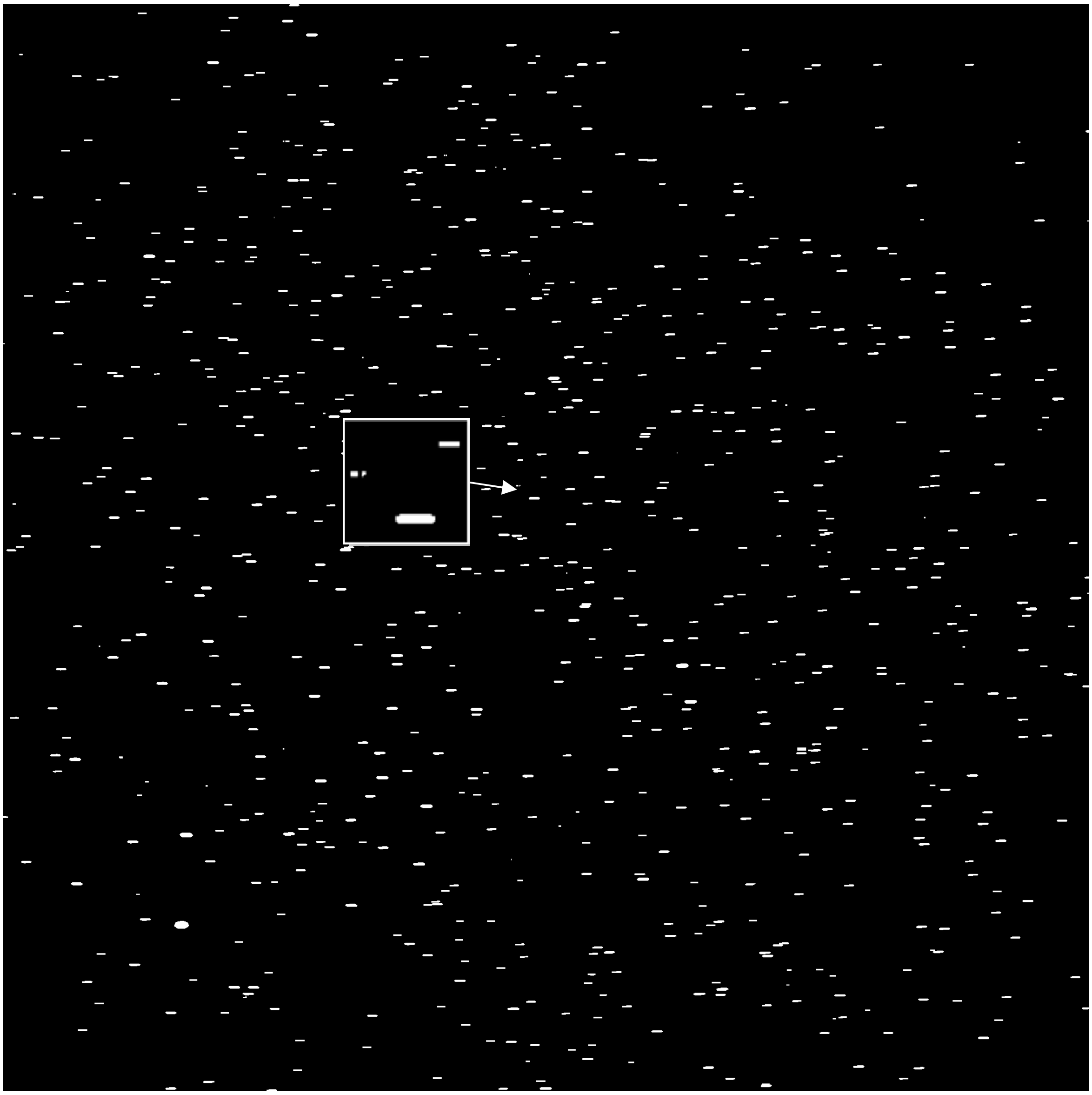

Figure 4. Binary CCD image after morphological filtering for star trail enhancement: background → black, objects → white. Enlarged region is the same as in Figure 3.

In case of sources with noticeable apparent motion in the image, it is important to clearly define which point within the streak left by the source corresponds to which exact moment of time during integration. We choose the easiest approach and assume that the visible center of streak corresponds to mid-exposure time. This has a number of implications, including those related to the times of opening and closing of mechanical shutter and to the possible optical variability of sources (inherent or induced by atmosphere). However, this still remains the most accurate and easy to implement method for most of real-life situations. Thus we need to obtain *XY* positions of centers of all field star trails. This is done by PSF fitting with a special type of point spread function suitable for trailed sources, as described in Kouprianov (2008). Other properties of stellar images, including their lengths, widths, and fluxes, are determined as well. Figure 5 shows a simulated image with real field stars approximated by their ideal models obtained by PSF fitting. One may observe a number of artifacts there – mostly caused by overlapping of multiple star streaks. The latter is very difficult to handle, which imposes two important restrictions on observations: (i) exposure time should not be too long to reduce length of streaks and (ii) one should avoid rich stellar fields in the Milky Way. This also reduces the probability of overlapping with space objects, which is critical for reliability of their detection.

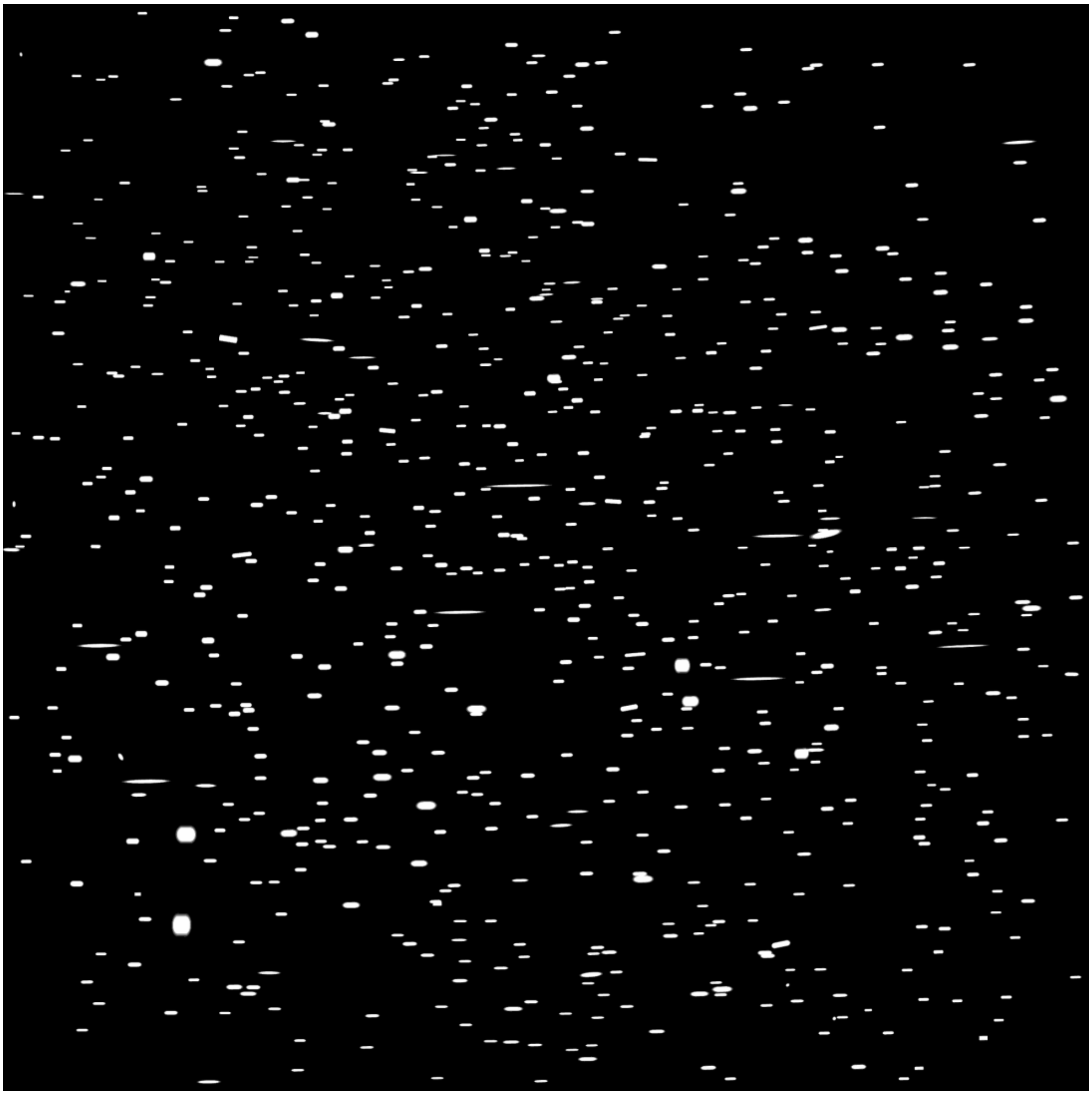

Figure 5. Simulation image of reference stars.

When PSF fitting is complete, several criteria are applied to eliminate false and unreliable detections. Most important of them are constraints on full with at half maximum (FWHM) across the trail and on deviation of measured length and orientation of trail from expected. Stars that pass these criteria undergo the classical differential astrometric reduction sequence that includes matching against reference catalog and obtaining LSPC solution by fitting a parameterized plate model that maps *XY* catalog positions to measured *XY* positions of reference stars (see e.g. Green, 1985). For large fields of view common to survey cameras Tycho–2 (Høg et al., 2000) is the catalog of choice; however, for instruments with fields of view less than about 1° UCAC3 (Zacharias et al., 2009) appears to be also suitable.

Apex II has a number of predefined plate models, both linear and non-linear in parameters. An important issue with wide-field optical systems is the presence of residual optical aberrations, especially at image corners, that need to be eliminated to achieve accurate astrometry across the whole field of view. From the point of view of differential astrometry, any optical aberration can be treated as generalized distortion, i.e. a systematic displacement of centroids of stars; then, if we have enough reference stars, distortion parameters are obtained just as any other plate constants. An example of such model is a simple cubic model with all terms:

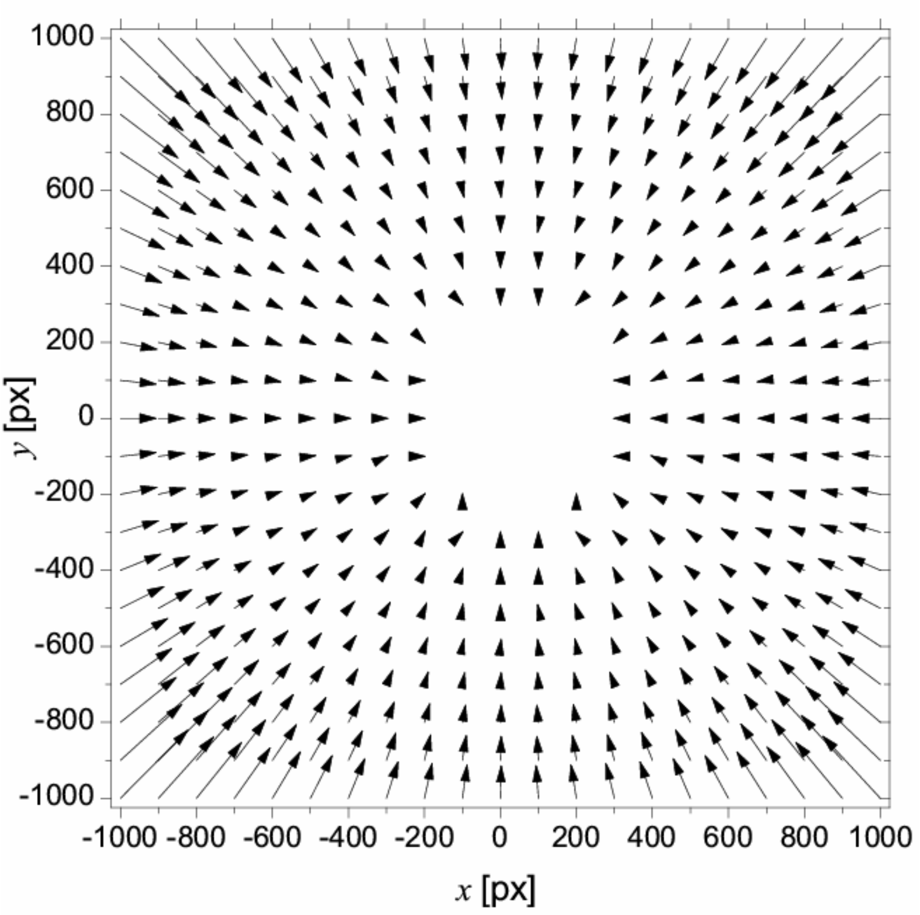

Figure 6. Example of optical distortions of a typical ISON survey camera. Displacement vectors are enlarged by factor 20.

(3)

where *x* and *y* are catalog positions, while *x*′ and *y*′ are measured positions. Another one (Brown, 1966) is suitable for handling pure radial and tangential distortions:

$$x' = A + Bx + Cy + K_1r^2x + K_2r^4x + P_1(r^2 + 2x^2) + P_2xy,$$
$$y' = D + Ex + Fy + K_1r^2y + K_2r^4y + P_2(r^2 + 2y^2) + P_1xy. \qquad (4)$$

where $r^2 = x^2 + y^2$.

However, when optical distortions become extremely large, even choosing the appropriate plate model might appear insufficient. Due to very large deviations of actual reference star positions from their expected positions, especially at the peripheral parts of image, catalog matching algorithm may fail with such stars, so they won’t be included in the final LSPC solution. This will result in systematic errors at image edges. An example of image with strong optical distortions is shown in Figure 6. The solution is to perform all astrometric reduction steps several times, with more and more peripheral stars being included at each iteration as LSPC solution becomes more and more reliable over the whole field of view. This technique leads to excellent accuracy even in the presence of strong distortions provided plate model is adequately chosen. As a real-life example we would mention the pure positional accuracy of about 0.1″ along each axis for pixel scale of 10″/ pixel that is achieved in good atmospheric conditions.

GEO Object Detection

As one can see from Figure 4, space objects may become apparent even in the course of reference star detection, so, at first glance, there is no need in a separate space object detection

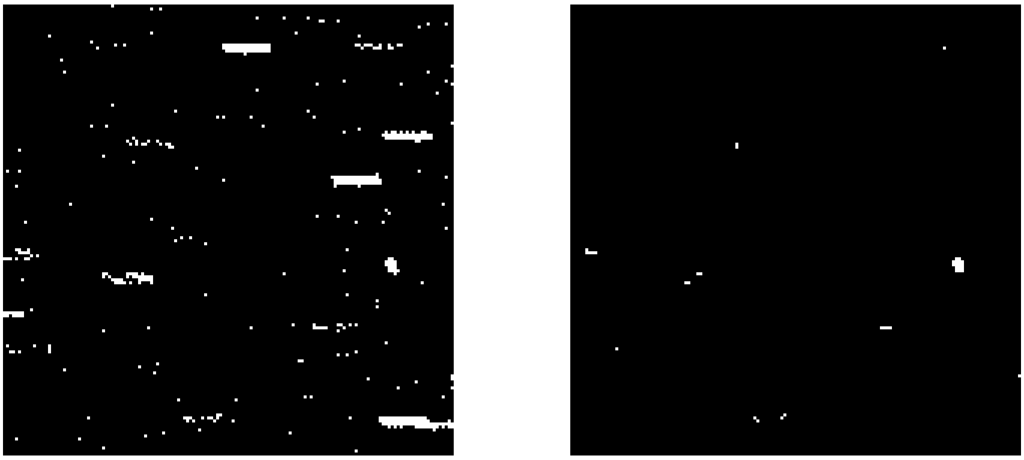

Figure 7. Effect of morphological filter (5), (6) on a fragment of binary image containing star trails and a single GEO object. Left: before filtering; right: after filtering.

step. Unfortunately, this is the case only for bright point-like objects. Fainter objects, as well as those having considerable apparent motion directed across diurnal motion of stars, are wiped out by star trail enhancement filter. Moreover, for better computational performance, the global threshold for reference star detection is chosen comparatively high to take only as many stars as needed for accurate astrometry, so we merely loose faint space objects. Therefore it is necessary to establish a separate space object detection stage, with as low detection threshold as possible (in practice, we use thresholds of down to 2.5σ, where σ is noise level).

Of course, such low threshold values would result in large amounts of false detections (here a field star is also considered a “false detection”). Fully processing all these detections to determine whether they are false or not would be impractical from the computational point of view. Hence we need to work out a process that quickly eliminates most of false detections, including stars, as early as possible, leaving space objects intact. Here the morphological difference between space objects and stars comes into play.

One of the possible ways to remove groups of pixels left by star trails is to use technique similar to that described in the previous section, but acting in the opposite direction. Filtering operator is now defined as

$$M^{'}(x,y)=\begin{cases}1 & \text{if } (M*K)(x,y)<d\sum K \text{ and } M(x,y)\neq 0,\\ 0 & \text{otherwise.}\end{cases} \tag{5}$$

Note that “>” in (1) is replaced by “<” here, which means that we are eliminating structures of the given shape instead of highlighting them. Since we need to remove all streak-like structures of known length and directed along the line of diurnal motion of stars, the corresponding filter kernel *K* is defined as

$$K=\begin{pmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 \cdots 1 & 1 & 1 & 1 & 1 & 1 & 1\end{pmatrix}. \tag{6}$$

And, again, the kernel is rotated appropriately according to the orientation of star trails if it differs from 0 or 180°. Just as before, *d* controls strength of filter and is chosen empirically for each instrument and observation site to find optimum between the efficiency of elimination of star trails and undesirable effect of filter on (especially very bright) space objects.

Figure 7 illustrates how this filter helps to eliminate most of star trails and noise detections. However, considerable amount of noise detections still remains after this process, as one can see from the right part of Figure 7. These are removed by one more filter of the same class as (1) and based on the same idea, but now with a different kernel

$$K = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \quad (7)$$

Here 1's fill the footprint of the conventional image PSF (which is 5×5 pixels in this example), so the filter (1), (7) in fact highlights real structures that are blurred by atmosphere and optics, at the same time eliminating sparsely distributed pixels produced by noise and remains of faint star trails. We should mention that our approach resembles PSF convolution technique widely used to improve detectability of faint objects. However, PSF convolution is very prone to false detections triggered by bright spots from either noise fluctuations or cosmetic defects of CCD chip. On the contrary, morphological filter described here acts on a binary image and is free from this drawback. This can be easily seen in Figure 8.

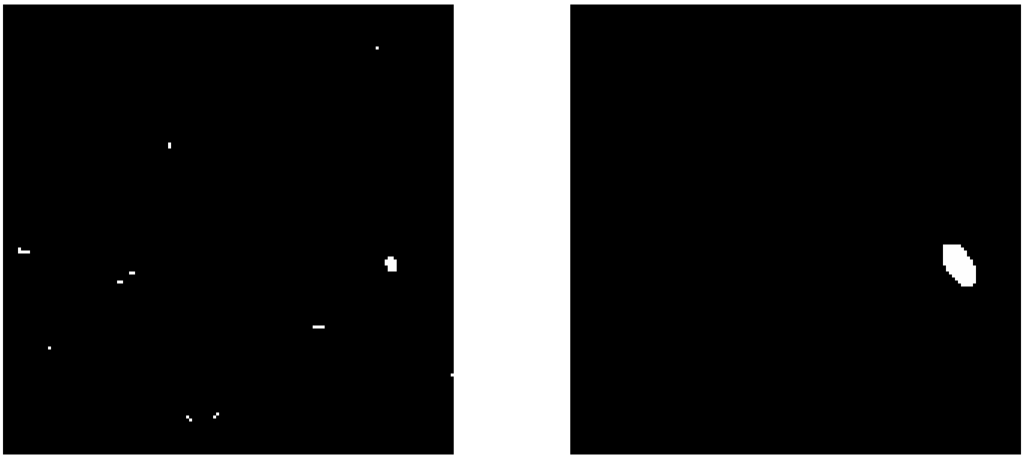

Figure 8. Effect of morphological filter (1), (7) on a fragment of binary image with star trails eliminated by filter (5), (6). Left: before filtering (same as Figure 7 right); right: after filtering.

Then potential space objects detected with the help of these two morphological filters undergo PSF fitting procedure to obtain their accurate positions, fluxes, and shapes. Based on their individual PSF properties, some possible spurious detections are removed. For the rest of them, their $\alpha\delta$ positions are calculated using LSPC solution obtained as described above. This completes the second of three main data reduction stages.

Cross-correlation of Detections

Despite the thorough elimination of possible spurious detections described above, images may still contain some false objects – especially if they were obtained in bad atmospheric conditions. The only way to remove them is to compare a set of adjacent images of the same sky area to find objects that are common to all (or at least most of) these images. Success of this greatly depends on the survey planning strategy: duration of a series should not be too long, or else fast-moving space objects will leave the field of view, and combination of exposure duration and density of stellar field should prevent frequent collisions of space objects with star trails.

Cross-frame correlation is implemented as a kind of brute-force approach: we construct all possible linear or slightly curved paths through all combinations of detections in (α,δ,t) space. Each path is an independent 1st or 2nd degree polynomial fit to $\alpha^{(k)}_{i_k}$ and $\delta^{(k)}_{i_k}$, where k enumerates sequential images in the series and i_k enumerates individual detections in k-th image. Such path is considered valid, i.e. corresponding to a track of a real space object, if it satisfies a certain number of constraints on (i) the absolute value of velocity, (ii) proximity of velocity to apparent velocity of stars due to their diurnal motion, (iii) path curvature, and (iv) deviations of individual detections from their path. Of course, the number of all possible combinations of detections is enormous even for comparatively small number of images in the series (usually 5 to 10) and individual detections in each image (usually one or two dozens), so we exclude deliberately false paths as soon as possible to reduce the number of variants to try. Figure 9 gives an idea of this procedure (the upper object in Figure 2 is close to $x = 0$, while the lower one – to $x = 200$).

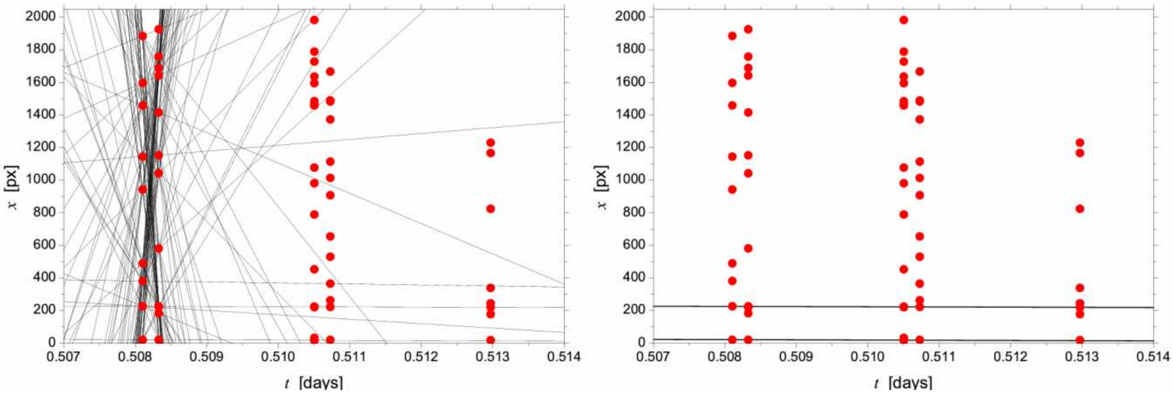

Figure 9. Cross-frame correlation of detections. Left: valid tracks from the first two frames; right: only two valid tracks left after scanning all 5 frames in the series. Positions of individual detections in each of the 5 images are indicated by circles.

In addition to four constraints mentioned above, cross-frame correlation pipeline discards tracks with less than [*N*/2] + 1 or 4 detections, whichever is larger, where *N* is the number of images in the series. Collisions that occur when a single detection belongs simultaneously to multiple tracks are handled by leaving detection in that track where its deviation is smaller. For all tracks that pass the procedure described we perform initial orbit determination, check that the orbit is valid, and exclude outliers.

All tracks produced by the process described above are the final result of initial data reduction for GEO surveys. Apex II pipeline that implements this algorithm can be fine-tuned to achieve the desired balance between reliability of its result (i.e. number of false detections) and the overall sensitivity with respect to faint and/or fast-moving space objects. It is also possible to visually check space objects detected and make any necessary changes to results of automatic data reduction.

Computational Efficiency and Real Time

As we already said at the beginning of this paper, an important requirement for automatic GEO survey data reduction software is its ability to process all data in real time on very moderate computer hardware. Moreover, this should be achieved without excessive optimization to allow working with varying implementations and bundling of optical sensors.

In this respect, solutions based on Python programming language appear to help in achieving the desired balance between these two conflicting prerequisites. Although scripting languages are generally thought to be relatively slow compared to compiled languages, in reality the situation is often just the opposite. The reason is that high-level algorithmic improvements can result in far more dramatic increase in computational efficiency than low-level optimization available to compiled languages. Indeed, Python allows one to combine both types of optimization. Being a high-level language, it allows to easily operate with large blocks of code almost at the level of their algorithmic description, which greatly simplifies complex algorithmic optimizations. At the same time, low-level computations, especially most time-consuming of them, that form Python library for scientific computing are already implemented in C or Fortran and are well tested and tuned over the years of their use. Finally, dynamic nature of the language and its run-time flexibility contribute greatly to adaptability of Apex II to different types of images, instrumentation, and atmospheric conditions.

Python has also built-in support for simple symmetric process-based parallelism which becomes essential when running on multi-processor and multi-core computers. Although the standard Python scientific computing library mostly lacks automatic support for multiple processors, Apex II library eliminates this defect on a higher level. Most of algorithms that are inherently parallel are written in the way that automatically takes advantage of multiple processors; furthermore, multiple input images are also processed in parallel. A combination of these two approaches leads to effective performance increase almost equal to the number of processors available.

To give a rough idea of computational performance of Apex II when processing GEO survey data, we use as an example one of ISON optical sensors with 5.5°×5.5° field of view, 3K×3K CCD chip, and 4-core 2.8GHz computer. The full data reduction pipeline partly described above takes about 45 seconds in average to process 4 images (one per core) in parallel. On the other hand, depending on survey strategy, optical sensor produces 20 to 30 frames in 10 minutes, i.e. less than 4 frames per minute – so data reduction performs at least 1.3 times faster.

Unfortunately, most of CPU time is taken by pixel operations, including mainly background estimation, noise statistics, filtering and PSF fitting. Most of such operations scales roughly as the number of CCD chip pixels, so processing a 4K×4K image would take ≈1.7 more CPU time, which already breaks real-time requirements. However, by slightly adjusting survey strategy and/or making background estimation a bit coarser, we can meet this requirement even in that case.

Conclusions

Here we described several of many challenges arising in the task of real-time automatic reduction of GEO survey data in particular and of wide-field imaging data for various types of high Earth-orbiting objects in general. We described the possible methods to solve some of these problems and how they are implemented in Apex II, a Python-based software platform for astronomical image analysis used by ISON members and other teams to obtain positional and photometric measurements of Earth satellites, space debris, asteroids and other near and deep space objects.

The first group of problems is related to differential astrometric and photometric reduction of CCD images with trailed reference stars – a field that is practically not covered by widely used image analysis software. Algorithms involved include a special class of morphological filters that helps detecting star trails and a generalized PSF fitting technique suitable for trailed sources. A special treatment is also required to handle strong optical distortions of wide-field imagers: in such case iterative LSPC reduction with plate model containing distortions can help.

The second group is connected with detection of space objects in a single CCD image that look differently than star trails. Solution proposed here involves the same class of morphological filters that is used to free binary images from anything except such objects.

A separate problem is correlation of individual detections from several CCD images into a single track of space object. This problem is solved by constructing all possible linear or slightly curved paths in (α,δ) space that approximate measured positions of detections as they move from image to image by a smooth curve, with a set of constraints that help to eliminate false detections remaining.

The last group of problems is the overall computational efficiency of data reduction and a demand for real-time processing on ordinary computer hardware. Apex II faces these problems by choosing C and Fortran implementations of most time-consuming parts of algorithms that are well-tuned and tested. At the same time, most powerful optimizations, including support for multi-processor computers, are done on Python level that almost coincides with the level of algorithmic description.

Therefore, currently Apex II answers all demands for automatic real-time processing of GEO survey data and observations of other similar classes of objects that arise in routine operation of ISON network.

Acknowledgements

The author is thankful to the organizing committee of 8th US–Russian Space Surveillance Workshop for its support in his attending this meeting.

References

Brown, D.C. Decentric Distortion of Lenses. // J. Photogramm. Eng. Remote Seising, 1966, **32**(3), 444–462.

Devyatkin, A.V., Gorshanov, D.L., Kouprianov, V.V., and Verestchagina, I.A. Apex I and Apex II Software Packages for the Reduction of Astronomical CCD Observations. // Solar System Research, 2010, **44**(1), 68–80.

Green, R.M. Spherical Astronomy. 1985, Cambridge Univ. Press, 536 pp.

Høg, E., Fabricius, C., Makarov, V.V., Urban, S., Corbin, T., Wycoff, G., Bastian, U., Schwekendiek, P., Wicenec, A. The Tycho–2 Catalogue of the 2.5 Million Brightest Stars. // A&A, 2000, 355, L27–L30.

Kouprianov, V. Distinguishing Features of CCD Astrometry of Faint GEO Objects. // Adv. Space Res., 2008, **41**, 1029–1038.

Molotov, I., Agapov, V., Akim, E., Sukhanov, S. ISON: Existing Structure, Tasks, Characteristics and Further Development. // Proceeding of 8th US–Russian Space Surveillance Workshop, 2010, in press.

Zacharias, N., Finch, C., Girard, T., Hambly, N., Wycoff, G., Zacharias, M.I. , Castillo, D., Corbin, T., DiVittorio, M., Dutta, S., Gaume, R., Gauss, S., Germain, M., Hall, D., Hartkopf, W., Hsu, D., Holdenried, E., Makarov, V., Martines, M., Mason, B., Monet, D., Rafferty, T., Rhodes, A., Siemers, T., Smith, D., Tilleman, T., Urban, S., Wieder, G., Winter, L., Young, A. Third U.S. Naval Observatory CCD Astrograph Catalog (UCAC3) // Astron. J., 2009, in press.