

RANSACing Optical Image Sequences for GEO and near-GEO Objects

Radim Šára, Martin Matoušek, Vojtěch Franc

Center for Machine Perception

Department of Cybernetics

Czech Technical University in Prague

Czech Republic

{sara, xmatousm, xfrancv}@cmp.felk.cvut.cz

ABSTRACT

This paper describes statistical models and an efficient Monte-Carlo algorithm for detecting tracks of slowly moving objects in optical telescope imagery sequences. The algorithm is based on accurate robust image pre-registration with respect to the star background, hot/warm pixel suppression, extracting dense normalized local image features, pixelwise statistical event detection, segmentation of event maps to putative image primitives, and finding consistent track sequences composed of the image primitives.

Good performance at low SNR and robustness of detection with respect to fast or slow-moving thin overhead clouds is achieved by an event detection model which requires collecting at least 10 images of a particular spatial direction. The method does not degrade due to an accumulation of acquisition artifacts if more images are available.

The track sequence detection method is similar in spirit to LINE [Yanagisawa et al, T JPN SOC AERONAUT S 2012]. The detection is performed by the RANSAC robust method modified for a concurrent detection of a fixed number of tracks, followed by an acceptance test based on a maximum posterior probability classifier. The statistical model of an image primitive track is based on the consistence between the size and the inclination angle of the image primitive, its image motion velocity, and the sidereal velocity, together with a consistence in relative magnitude.

The method does not presume any particular movements of the object, as long as its motion velocity is constant. It can detect tracks without any constraints on their angular direction or length. The detection does not require repeated image transformations (rotations etc.), which makes it computationally efficient. The detection time is linear in the number of input images and, unlike in the LINE proposal method, the number of RANSAC proposals is (theoretically) independent of the number of putative image primitives.

The current (unoptimized) experimental implementation run several hours on a standard two-core CPU architecture. Reliable detection up to the magnitude of 16.5 has been obtained on a test sequence of over 5800 images from the 50 cm TAOS telescope at Lulin Observatory, Taiwan. A comparison with the FPGA Image Stacking, which was the most successful method tested by [Yanagisawa et al, AMOS 2012] shows the proposed method is able to detect 62% more objects of magnitudes 11 – 13.5, 38% more objects of magnitudes 13.5 – 16.5, but only 33% of objects of magnitudes 16.5 – 19. If optimized for speed, the proposed algorithm would be suitable for online detection, assuming an order of 10 or more running images are buffered. The algorithm is not suitable for fast object velocities at which the object typically enters/escapes the field of view during exposure.

I. INTRODUCTION

Object detection is a critical component of any space surveillance system. In this paper we are concerned with detection of orbiting objects in optical images, see Fig. 1a for a typical data example. With methods from the computer vision community at hand we show how they can be utilized to achieve a good balance between detection performance, run-time and memory requirements. The present paper focuses on the detection problem for GEO or near-GEO objects observed in a series of images with not so good signal-to-background ratio. Useful methods must typically collect and process a large amount of data before a detection can be confirmed.

In the literature there are two main approaches to object detection:

- 1) Primitive image objects (events) are detected in input images independently and then linked to tracks. A typical example of this class of methods is the LINE method [8]. It tests all possible track sequences created from all pairs of primitives in subsequent images. The principal advantage of such method is a huge data compression in the first step. This opens the possibility to postpone the linking phase until significantly more data is collected, which is useful in pipeline architectures for on-line data processing. The disadvantage is a lower statistical efficiency of the method, since the primitive object detection cannot use prior information that is available at the level of the track only (the track must be consistent, more details are found in Sec. II-C2). In principle, these methods do not have to limit the speeds and motion directions of the objects to be detected, although they are unsuitable for fast object velocities at which the object typically enters/escapes the field of view during exposure.
- 2) Raw image data (after radiometric corrections) are processed directly [8]. A typical example of this class of methods is the Image Stacking Method [3, 8]. Images are registered in a suitable common image space and the pixelwise activity along a putative (moving) object track is compared against activity along a random track. All directions and object speeds must be explored. In principle, detection is based on a statistical test that compares the hypothesis that the activities come from the same random background process against the alternative hypothesis that they come from different processes. The principal advantage of this method is a superior performance beyond the limiting magnitudes [8]. On the other hand,

this method requires processing a huge amount of data in a single procedure and an exhaustive search over all object motions (a search in 3D space: direction and speed). Hard limits on the search space and image processing on FPGA have to be used to achieve fast processing speeds.

In this paper we attempt to answer the question if prior information on the level of tracks and Bayesian modeling can bring the performance of the primitive object detection and linking methods close to the image stacking methods, as measured by false positive and false negative detection rates in low signal to background (or signal to noise) ratio. We employ a dataset from [8] and our results are directly comparable to those presented in that paper.

II. DETECTION METHOD

A. Image Pre-Processing and Registration

1) *Local Image Artifact Suppression*: Our detection methods need to detect rather faint phenomena. Before the images are used for object detection, local image artifacts need to be suppressed. First, hot and warm pixels need to be masked-out. It is never possible to completely eliminate these artifacts based on local image operations alone but their influence can at least be diminished. Additional processing is then done at the level of detected tracks, as described in Sec. II-D4.

A warm pixel is a normal pixel with an additive stray current in the CCD cell, resulting in an unknown value $\tau(x, y)$ added to the pixel intensity value at position (x, y) . We assume the $\tau(x, y)$ is constant over the whole measurement period. The $\tau(x, y)$ is then estimated as a difference of the pixel value at (x, y) from the median of its 5×5 neighborhood in each image, then taking median over all images. The resulting image $\tau(x, y)$ is subtracted from every input image. After this step, there are still some defective pixels remaining, especially those that exhibit time-variant behavior.

Next, background ambient light changes are reduced by subtracting the median-filtered image (kernel 41×41 px), this operation is applied to each image independently.

2) *Accurate Robust Image Registration*: The input images are registered in a common coordinate system so that locations of stars become fixed. To this end we model the telescope as a projective (pinhole) camera [2]. The camera internal calibration parameters (focal length, pixel size and optical center) are known, represented as the camera calibration matrix \mathbf{K} . When a 3D rotation matrix \mathbf{R} between any two images is known, the images are related by homography whose homogeneous matrix is $\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$ [2]. When warping the source image using \mathbf{H} , one gets the image of the star background as seen in the target image. The homography \mathbf{H} is estimated directly from the input images by the following procedure.

The initial estimate for the rotation matrix \mathbf{R} for each image is given by the angular orientation of the telescope, stored in image meta-data (FITS headers). These rotations are not accurate enough to allow precise image registration, so we refine them by numeric optimization based on pair-wise image correspondences of stars. Using known camera calibration and the initial camera rotations, the pairwise overlap of the fields of view is first estimated. The overlap value defines edge weights of a dense overlap graph. The maximum spanning tree (MST) of the graph is found, giving a set of pairs used in subsequent optimization. With the help of the MST, the homography between any image pair is later obtained by pairwise concatenation.

Stars are detected in each image by a correlation operator with a short streak kernel, followed by a sub-pixel non-maximum suppression and thresholding. The streak kernel is constructed from a zero-mean 2D Gaussian kernel motion-blurred according to the known exposure time and sidereal motion direction. The zero mean guarantees invariance of the detector to dark level shifts of the image. Typically, 1000 stars are detected. For each star, a pair of 2D vectors connecting the star with its two nearest neighbors of lower relative magnitude are found. These vectors, including their orientation, are invariant to changes in intensity caused by changes of atmospheric transmittance, ambient light and sensor gain (exposure). Then for each pair of images in the MST, the vectors of stars are used as descriptors that are matched by thresholded mutual nearest neighbor matching in the descriptor space (One star correspondence originates from one or two descriptor vector matches). The resulting correspondences contain a low percentage of mismatches. These are detected and removed during the subsequent numerical optimization.

For every pair of images (i, j) in the MST we use the total Sampson registration error [2] over the star locations as the numerical optimization criterion. MLESAC algorithm [6] is used to find an initial solution together with outlier (mismatch) identification, then standard gradient minimization of the error is used on the inlier set. The mean correspondence error after registration is about 0.8 px, as measured on the MST edges.

The final common image coordinate system is defined by a virtual camera with a cylindrical retina, where the cylinder radius is the same as the focal length of the original images and its axis is oriented so that registered image rows correspond to the right ascension angle and image columns to the declination angle. In subsequent processing, we work with a 3D stack of registered images, whose third dimension is time.

B. Primitive Event Detection

1) *Dense Normalized Local Image Features*: The visual event detection that will be described in Sec. II-B2 does not work well on raw data, since it is based on finding outliers in long-term temporal pixel activity. We propose a local image feature that proved very stable and efficient in enhancing local contrast and performing local image normalization (offset and gain

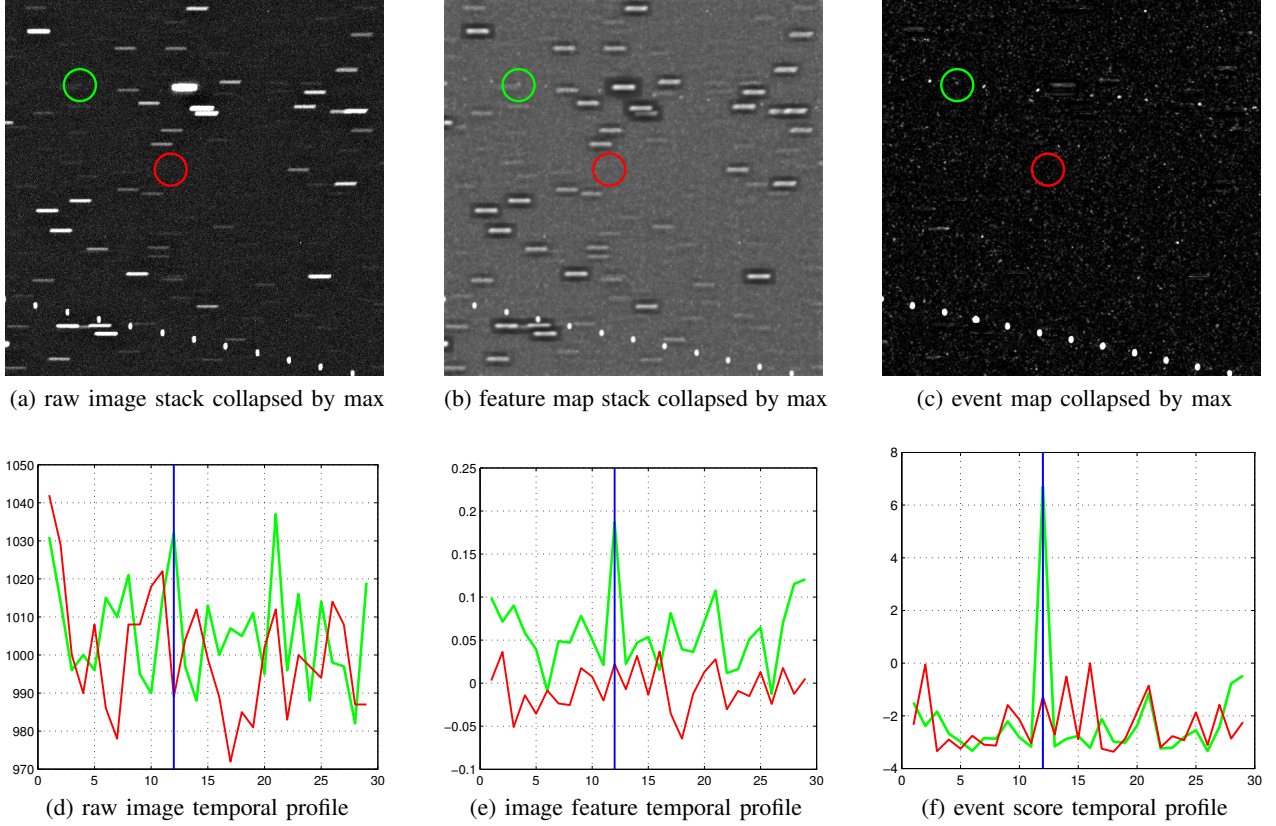


Fig. 1: **Top row:** Raw input image, normalized image feature map, and the event score map, all three collapsed by taking the maximal value over time in the registered image stack. Circled are two locations: One where a high-magnitude event should be detected (green) and a random one (red). We can see that the contrast of the high-magnitude event is improved in the feature map and that the star background is effectively suppressed in the event map in which the second track appears in the upper half of the map. **Bottom row:** Temporal profiles for the two locations. The blue vertical bar identifies the time of 12 at which the green event should be detected. The event score is not only maximal at the correct location, it also stands out prominently over the rest of the green profile (and the entire red profile).

elimination). Since we are mostly interested in geostationary objects and want to detect even faint ones, we use an image model optimized for geostationary objects. The model is the point-spread function of the entire optical path, including the temporal component. Its good approximation is a 3D Gaussian with spatial variance σ_P^2 and temporal variance σ_T^2 (we used $\sigma_P = 3.5$ px, $\sigma_T = 0.4$ px). The Gaussian is represented by a correlation kernel in the registered image stack. The correlation response is normalized by local image variance. The resulting normalized image feature map $r(x, y, t)$ has the range of $[-1, 1]$ and is defined over the domain of the registered image stack. Figs. 1b and 1e show the effect of this step on raw registered data in Figs. 1a and 1d.

2) *Pixelwise Event Detection:* The profile along the temporal dimension of the feature map at location (x, y) contains the temporal activity of that pixel, cf. Fig. 1e. If the pixel (x, y) sees a fixed location on the star background over the entire observation time, there is constant activity, up to background noise (photon noise, thermal noise, electronic noise, etc). If an object happens to fly through the pixel, the activity will feature a local peak.

We took the parametric approach to outlier detection. Its advantage is the simplicity of computation and low noisiness in our sample sizes. Assuming there is at most one event of interest per pixel (x, y) , we can detect it as an outlier in the statistical model of the background temporal activity $r(x, y, t)$. We selected the beta probability distribution for that.

Let $C = (r_1, r_2, \dots, r_n)$ be the activity sample of a given pixel. We are deciding whether the maximal value $\bar{r} = \max_i r_i$ is an outlier in C . We use a modification of Chauvenet's statistic [5] for beta distribution. Let \bar{C} be the set C with \bar{r} excluded. We use a statistic we call the *event score* e

$$e(\bar{r}; \alpha_1, \alpha_2) = -\log \left(n \left(1 - I_{\frac{1+\bar{r}}{2}}(\alpha_1, \alpha_2) \right) \right), \quad (1)$$

where the parameters α_1, α_2 are estimated from \bar{C} by the moment method for beta distribution and $I_x(\alpha_1, \alpha_2)$ is the regularized incomplete beta function (distribution function). The argument of the logarithm in (1) is the expected number of measurements

in the sample of size n that have value of at least \bar{r} assuming the beta distribution model holds for \bar{C} . A high event score identifies \bar{r} as an outlying value. The e is fast to compute, unlike in some non-parametric outlier identification methods. An example of event score map $e(\bar{r}; \alpha_1, \alpha_2)$ in which the time dimension is collapsed by taking the maximal value per pixel is shown in Fig. 1c.

The above model bears some resemblance to the statistical model in the FPGA Stacking Method [7], where it is used for the raw images. The final effect is similar: Enhancing the contrast of image events of interest before the actual detection.

3) *Segmentation of Event Maps To Primitives*: The event map stack is processed, one temporal layer at a time. In each layer, the event score is thresholded. Then contiguous segments are found by a standard binary image segmentation method. The segments are then filtered by several heuristic criteria:

- 1) the segment must contain a 3D regional maximum of the feature map (18-connected neighborhood),
- 2) the segment size must be above 2 pixels and below 500 pixels,
- 3) the maximal feature value r in the segment must be above 0,
- 4) the maximal event score e in the segment must be above 10.

These parameter settings were selected quite conservatively, they only remove the most obvious outliers. We verified that this choice does not affect the performance of the subsequent track detector, it only improves its run time.

Each segment receives a set of attributes: the location of its center in space-time (\mathbf{x}_i, t_i) , its bounding box, the approximate relative magnitude E_i (larger is dimmer), the maximal event score \bar{r} within the segment, and a small raw image within the segment bounding box. We call the resulting elements primitive image objects or *primitives* for short. There were approximately 48 primitives per image detected in our data (see Fig. 5). The primitive detection thus results in a large data compression factor.

C. Bayesian Model for Consistent Track Sequences

Track detection is based on maximum posterior probability. Our model is based on a ratio comparing the probability of the union of a set of linked tracks and the set of remaining loose primitives against the probability that the input set consist of loose primitives only. The logarithm of this ratio will be called *support*. A linked track is an ordered set of mutually similar primitives whose locations are regular in space-time and whose speed is consistent with the primitive shape. There are no prior constraints on the loose primitives that will be called *outliers*.

The support for a collection of tracks (the total support) is given by the sum of individual track supports. Maximization of the total support will be used as the optimization task for the track detector, as will be detailed in Sec. II-D.

The individual track support $s(M, Z)$ consists of several terms

$$s(M, Z) = s_D(M, Z) + s_S(M, Z) + s_F(M, Z) + s_\phi(M, Z), \quad (2)$$

in which s_D is the data support (6), s_S is the consistency support for primitive size and track speed (10), s_F is the ‘temperature’ support (12), s_ϕ is the directional support for geostationary tracks (13), and the M, Z represent the track, as will be explained next.

Suppose, for simplicity, we are modeling a single track. Let a fixed line in space-time be given that corresponds to a thought trajectory of the object. The line intersects the stack of registered images in a fixed discrete set of K space-time locations $Z = (\mathbf{z}_1, \dots, \mathbf{z}_K)$, $\mathbf{z}_i = (\mathbf{x}_i, t_i) \in \mathbb{R}^3$, where the K is given by the intersection of each line with the image data planes, cf. Fig. 2a. In our data the typical value for K is about 300 (the entire registered image stack is spanned). The space-time locations \mathbf{z}_j are given by

$$\mathbf{z}_j = \mathbf{z}_a + (j - j_a) \frac{\mathbf{z}_b - \mathbf{z}_a}{j_b - j_a}, \quad j = 1, \dots, K, \quad (3)$$

where $\mathbf{z}_a, \mathbf{z}_b$ are the locations of two *key primitives* representing the line (the line proposal, see later), and j_a, j_b are their indices.

1) *The Data Model*: The data model must have the same number of terms regardless of the number of detected tracks. Therefore, data is considered as a collection D of all input image primitives, some of which are linked in tracks (inliers), but most of them remain unlinked (outliers, misdetections). Hence $D = T \cup O$, where O are outliers and T are inliers.

The important attributes of a primitive d_i in D include spatial location $\mathbf{x}_i \in \mathbb{R}^2$ in the (registered) event map, location $t \in \mathbb{R}$ in time, and relative brightness $E_i \in \mathbb{R}^1$.

Given Z , a track instance is represented by a matching $M: Z \rightarrow D$. See Fig. 2b. We use a Bayesian generative data model

$$p(D, M, Z) = p(D | M, Z) \cdot p(M, Z) = p(T | M, Z) \cdot p(O) \cdot p(M, Z), \quad (4)$$

where we have chosen uninformed matching prior $p(M, Z) \propto 1$, uninformed outlier distribution $p(O) = p_\omega^{|O|}$ with $p_\omega = \text{const}$, where $|O|$ is the size of the outlier set, and detected track probability

$$p(T | M, Z) = p_\gamma^{K-|M|} \cdot \prod_{(i,j) \in M} p_t(\mathbf{y}_i, E_i | \mathbf{z}_j, E_p), \quad (5)$$

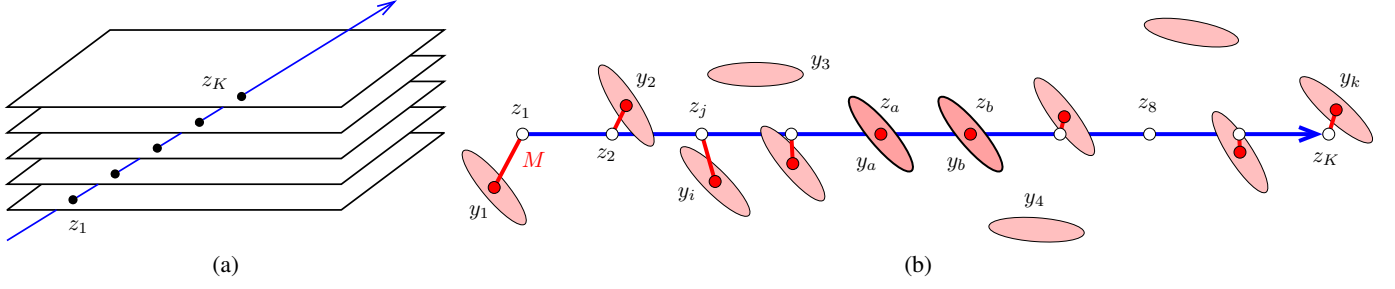


Fig. 2: **Left:** A line in space-time intersects image planes in a discrete ordered set of locations $(\mathbf{z}_1, \dots, \mathbf{z}_K)$. The index of the plane is given by its acquisition time t_i and the position within the plane is given by \mathbf{x}_i (in global coordinates). **Right:** The proposal generating and primitive matching mechanism (collapsed time). Primitive y_i is matched to location z_j , $j = M(i)$ (red). The spacing on the generating line (blue) is defined by the proposal y_a, y_b that (apriori) matches z_a, z_b . A primitive (such as y_3 or y_4) may be left unassigned by M . A position on the generating line (such as z_s) may be left unassigned when there is a missing measurement.

where $\mathbf{y}_i = (\mathbf{x}_i, t_i)$, $|M|$ is the size of matching M , $p_\gamma^{K-|M|}$ is a penalty for unmatched locations in Z , and E_p is the mean image value for the key primitives a, b , ie. $E_p = \frac{1}{2}(E_a + E_b)$.

The decision if the data contain a track (M, Z) or no track ($M = \emptyset, Z = \emptyset$) is based on the log-ratio that compares the two data interpretations

$$s_D(M, Z) = \log \frac{p(D | M, Z)}{p(D | \emptyset, \emptyset)} = (K - |M|) \log p_\gamma + \sum_{(i,j) \in M} \log \frac{p_t(\mathbf{y}_i, E_i | \mathbf{z}_j, E_p)}{p_\omega}, \quad (6)$$

where $K - |M|$ counts locations Z unassigned to primitives.

The $p_t(\mathbf{y}_i, E_i | \mathbf{z}_j, E_p)$ captures two key properties of a good track: Geometric consistence with the proposed track Z and consistence in appearance of the set of matched primitives. These are considered independent. For implementation reasons (the k-d search in Sec. II-D1) we use a Gaussian $p_t(\mathbf{y}_i, E_i | \mathbf{z}_j, E_p) = N(\mathbf{y}_i, E_i; \mathbf{w}_j, \mathbf{S})$ with mean $\mathbf{w}_j = (\mathbf{z}_j, E_p)$ and 4-dimensional diagonal covariance matrix \mathbf{S} . With $V_\gamma = -\log p_\gamma$, $V_\omega = -\log(p_\omega \sqrt{\det(2\pi\mathbf{S})})$ the data support becomes

$$s_D(M, Z) = |M|V_\omega - (K - |M|)V_\gamma - \frac{1}{2} \sum_{(i,j) \in M} (\mathbf{x}_i - \mathbf{w}_j)^\top \mathbf{S}^{-1} (\mathbf{x}_i - \mathbf{w}_j). \quad (7)$$

The term V_ω is a penalty for an outlier primitive, the term V_γ is a penalty for a missing primitive. We have chosen $V_\omega = V_\gamma > 0$ and $\mathbf{S} = \text{diag}(12, 7 \cdot 10^{-4}, 0.25)$.

In summary, given a hypothetical line in space-time represented by Z and a matching from Z to image primitives M then $s_D(M, Z)$ measures data support for the hypothesis, higher $s_D(M, Z)$ is better. For instance, when $K = |M|$ (no missing measurements) and the average error in the last term is smaller than V_ω then the support is positive. If the data support $s_D(M, Z)$ is positive then it is more probable that the data contains the track than it does not. The zero is therefore a decision threshold for the track detector, assuming there was no other information incorporated in the model. A prior model on the track regularity and consistence will just shift the decision threshold, as will be seen in the next subsection.

2) *The Track Speed Model:* The shape of a primitive is either point-like, when it corresponds to a geostationary object or elongated when it does not, since the object makes an apparent motion during the exposure time (this includes stars). The major axis length of the primitive and the object speed relative to a geostationary object are therefore consistent. See Fig. 3a.

At time t the image position of a primitive is $\mathbf{x}(t)$ and its (major axis) length is \mathbf{d}^e . Next time it is observed, it has moved to $\mathbf{x}(t + \Delta_t)$, where the observational interval is $\Delta_t = t_e + t_r$, in which t_e is the exposure time (5.9 s in our data), and t_r is the readout plus the wait time (2.9 s in our data). See Fig. 3b. The relative inter-frame motion vector $\mathbf{d}(t)$ is then

$$\mathbf{d}(t) = \mathbf{x}(t + t_e + t_r) - \mathbf{x}(t) = \frac{\Delta_t}{t_e} (\mathbf{d}^e + \mathbf{d}^s), \quad (8)$$

where \mathbf{d}^e is the oriented length (vector) of the image primitive and \mathbf{d}^s is the sidereal motion vector (the oriented length of the star primitive), both in pixels. This gives a prediction on the oriented length as $\mathbf{d}^e = \frac{t_e}{\Delta_t} \mathbf{d} - \mathbf{d}^s$. We construct a Gaussian distribution with mean $\boldsymbol{\mu}_s = \frac{t_e}{\Delta_t} \mathbf{d} - \mathbf{d}^s$ and covariance matrix $\sigma_s^2 \mathbf{I}$

$$p(\mathbf{d}^e | \mathbf{d}, \mathbf{d}^s, o) = N(o \mathbf{d}^e; \boldsymbol{\mu}_e, \sigma_s^2 \mathbf{I}), \quad \boldsymbol{\mu}_e = \frac{t_e}{\Delta_t} \mathbf{d} - \mathbf{d}^s. \quad (9)$$

The p.d.f. lives on pairs of adjacent measurements in the track. Note that the \mathbf{d} is the property of the track as a whole and \mathbf{d}^s is the property of the data acquisition method, hence $\boldsymbol{\mu}_e$ is fixed for the entire track. Note also that the image motion vector \mathbf{d}^e

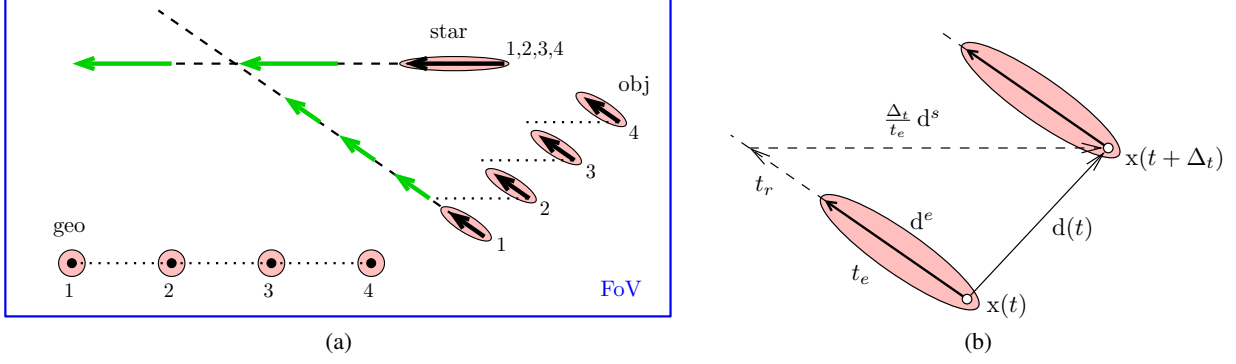


Fig. 3: **Left:** The speed model for a moving/stationary object (telescope w/o sidereal tracking). During exposure the telescope rotates with the Earth and geostationary objects do not move, everything else does. Geostationary objects leave disk-like primitives (geo, bottom), moving objects (obj, center) and stars (top) leave short streaks. The green arrows show primitive locations if there was no image registration. After image registration and collapse of the time dimension the event map contains sequences of dots and streaks. **Right:** Detail of the primitive sequence. After exposure interval t_e , there is a readout and wait interval t_r , at the end of which the star registration process corrects the sidereal motion. Primitives are shown as pink regions.

can be determined up to its orientation sign o . We choose orientation o^* maximizing $o^* = \arg \max_{o \in \{-1, +1\}} p(\mathbf{d}^e \mid \mathbf{d}, \mathbf{d}^s, o)$. Assuming outlier primitive length distribution $p_o(\mathbf{d}^e) = N(\mathbf{d}^e; \mathbf{0}, \sigma_o^2 \mathbf{I})$ with $\sigma_o \gg \sigma_s$, the speed-consistent track support is

$$s_S(M, Z) = \log \prod_{i: (i,j) \in M} \frac{p(\mathbf{d}_i^e \mid \mathbf{d}, \mathbf{d}^s, o)}{p_o(\mathbf{d}_i^e)} \approx |M| \log \frac{\sigma_o}{\sigma_s} - \sum_{i: (i,j) \in M} \frac{\|\mathbf{d}_i^e - o_i \boldsymbol{\mu}_e\|^2}{2\sigma_s^2}, \quad (10)$$

and the sum is over all matched primitives in a track. The approximation by term simplification is sufficient in practice.

We found s_S quite discriminating between correct and incorrect tracks.

3) *A Model for Tracks of Hot Pixels:* Hot pixels behave like exactly geostationary objects. We found our method quite sensitive to them. We have therefore added a prior that lowers the support for tracks of hot pixels. Tracks of hot pixels that lead to false detections have two prominent properties: (1) their “images” are formed by isolated high-value pixels (or small groups of thereof) on a dark background, and (2) the tracks are exactly parallel to sidereal motion direction (direction in which stars move when the telescope is in the passive acquisition mode).

The first component of the hot pixel track model is based on an image feature F . Let \mathbf{v} be the vector of pixel values in the image segment supporting a primitive detection. The image values are first normalized with respect to an offset by subtracting the median, $\hat{\mathbf{v}} = \mathbf{v} - \text{med}(\mathbf{v})$. Let $s(\hat{\mathbf{v}})$ be the sample deviation computed from all values in $\hat{\mathbf{v}}$ with the maximal value $\max \hat{\mathbf{v}}$ excluded. The temperature feature F is then

$$F = \frac{\max \hat{\mathbf{v}}}{1 + s(\hat{\mathbf{v}})}, \quad F > 0. \quad (11)$$

The value of F is high for single-pixel local maxima in relatively dark neighborhood (precisely when the hot pixels lead to artifacts). In true geostationary objects, the PSF of the optical system and the atmospheric turbulence smear the point-like response, which results in a low value of F .

We need a step-like low-pass distribution for identifying the hot pixels. We have chosen $p(F; \theta, \lambda)$ as the complementary error function with a turn-off temperature θ and scaling constant λ . For the ordinary pixels we have chosen an improper uniform distribution $p_o(F) \propto 1$. The ‘temperature’ support for a track is then

$$s_F(M, Z) = - \sum_{i: (i,j) \in M} \log \text{erfc} \left(\frac{F_i - \theta}{\sqrt{2} \lambda} \right). \quad (12)$$

Hot pixel tracks have low s_F .

The second component of the hot pixel track model is based on sidereal motion direction angle ϕ . For the sidereal direction tracks we use a heavy-tailed angular distribution $p(\phi; \phi_s, \gamma)$ derived from Cauchy distribution and for tracks of proper objects we use $p_o(\phi) = \frac{1}{\pi}$ in which ϕ_s is the known sidereal direction (typically $\phi_s = 0$) and γ is the shape parameter (greater γ results in a distribution more focused around ϕ_s). This gives the angular support of

$$s_\phi(M, Z) = - \log \frac{p_o(\phi)}{p(\phi; \phi_s, \gamma)} = - \log \left(\frac{(4\gamma^4 \phi^2 + \pi^2) \tan^{-1}(\gamma^2)}{\pi^2 \gamma^2} \right). \quad (13)$$

The orientation of a track is given by a line fit to the track. Tracks of almost sidereal orientation have low s_ϕ . Although this support term helps, we did not find it particularly discriminative in identifying tracks of hot pixels.

D. Track Detection

Given the segmented event map the goal is to find all tracks. More precisely, the goal is to explain the set of primitives D as 1) a collection of independent tracks T and 2) a set of isolated primitives (outliers) O . Each primitive is therefore ‘explained’ by exactly one final entity. Since the number of tracks is unknown, this is a model selection problem. We avoid the difficulties with Bayesian model selection for discrete problems by adopting a two-phase procedure. In the first phase, the number of tracks k is fixed to a large value (eg. 100). In the second phase the set of k tracks is pruned by a classifier that performs the final track acceptance/rejection decision.

We used a variant of Random Sample and Consensus (RANSAC) sampling, which is a Metropolis Hastings sampler for geometric maximization problems [1]. We modified the sampler so that it finds a set of k best tracks. To this end we added an outlier pool bookkeeping mechanism. We use a priority queue \mathcal{Q} , whose length must not exceed k , and a modification of the acceptance mechanism. The operation $\min \mathcal{Q}$ returns the lowest priority in \mathcal{Q} . If \mathcal{Q} is empty, then $\min \mathcal{Q} = -\infty$. The operation $\text{length } \mathcal{Q}$ returns the number of elements in \mathcal{Q} . Given the set of image primitives D , the sampler works as follows:

- 1) Initialize: The pool of all outliers $O := D$, an empty priority queue $\mathcal{Q} := \emptyset$, an unsuccessful proposal counter $c := 0$.
- 2) Propose a random space-time line and compute the track sequence Z as in Fig. 2a.
- 3) Find the best matching M from Z to the outlier pool O .
- 4) Compute support $s(M, Z)$ using (2).
- 5) Increment the counter $c := c + 1$.
- 6) if $s(M, Z) > \min \mathcal{Q}$ then
 - a) if $\text{length } \mathcal{Q} = k$ then remove the lowest-priority element from \mathcal{Q} and return its set of matched primitives to O ,
 - b) admit (M, Z) to \mathcal{Q} with priority $s(M, Z)$,
 - c) remove the set of matched primitives M from O ,
 - d) reset the counter $c := 0$.
- 7) Repeat Steps 2–7 until the number of unsuccessful proposals c does not exceed a limit $c > c_{\text{stop}}$.
- 8) Return the queue \mathcal{Q} .

The key components of this sampler are the proposal generator from Step 2, the matching procedure from Step 3, and the formula for determining c_{stop} . Subsequent subsections describe these procedures in more detail. The other steps are trivial, except for Step 4, which has been explained in Sec. II-C.

1) *Proposal Generator*: The elementary proposal mechanism takes the pool of open primitives (outliers) O and returns a sequence of space-time locations Z where observable events are expected in the input data. We use two proposals that use different pools. The proposal type is chosen randomly at each iteration.

The first proposal procedure works as follows:

- 1) Select a single primitive $d_a \in O$ by a uniformly distributed random index. Let the d_a have the location-time-energy triple of (\mathbf{x}_a, t_a, E_a) .
- 2) Find primitive $d_b \in O$ whose location-time-energy triple (\mathbf{x}_b, t_b, E_b) is closest to $(\mathbf{x}_a, t_a + \Delta_t, E_a)$, using Mahalanobis distance with covariance matrix \mathbf{S} (see eq. (7)), where Δ_t is the observational interval as in (8) ($\Delta_t = 8.8$ s in our data).
- 3) Find the intersection of the space-time line given by $\mathbf{y}_a = (\mathbf{x}_a, t_a)$ and $\mathbf{y}_b = (\mathbf{x}_b, t_b)$ with all image data planes (cf. Fig. 2)

$$Z = \left\{ \mathbf{y}_a + j \cdot \frac{\mathbf{y}_b - \mathbf{y}_a}{t_b - t_a} \cdot \Delta_t \right\}_{j=j_{\min}}^{j_{\max}}, \quad j_{\min} = \left\lfloor \frac{t_{\min} - t_a}{\Delta_t} \right\rfloor, \quad j_{\max} = \left\lceil \frac{t_{\max} - t_a}{\Delta_t} \right\rceil. \quad (14)$$

in which t_{\min} is the minimal intersection time and t_{\max} is the maximal intersection time.

The minimization in Step 2 of the proposal procedure provides a somewhat restricted set of proposals. The full proposal would involve a random choice among all primitives close to $(\mathbf{x}_a, t_a + \Delta_t, E_0)$, based on Gibbs sampling using the full support model from Sec. II-C.

With the help of k-d tree search in a 4D space scaled by the diagonal scaling matrix \mathbf{S} , finding the closest primitive to a given location is performed in $\log |D|$ time. This makes the proposal a very cheap operation. On a single core of an i7 PC processor, the RANSAC procedure generates about 30 proposals per second on our dataset (data from Day 3, ca. 2200 images, 10^5 primitives in D , see Sec. III-A for data description).

The second proposal is a track re-assembly. This proposal is necessary for the RANSAC sampler not to get trapped in a local optimum. It also helps improve track’s support (priority in the queue) and speeds the search up significantly. The proposal proceeds as follows:

- 1) Select a track in the queue by random uniformly-distributed index. Let the set of matched primitives in the track be M .
- 2) Return the set M to the outlier set O .

- 3) Randomly select primitive $\mathbf{y}_a \in M$. Let $\mathbf{y}_b \in M$ be its adjacent primitive in the direction of increasing time.
- 4) Given the key primitives $\mathbf{y}_a, \mathbf{y}_b$, compute the line sequence Z as in (14).

2) *Matching*: The second key component of the sampler is the matching from the set of observable event locations Z to (a subset of) the detected image primitive pool O . This is illustrated in Fig. 2b. The matching proceeds in two phases. In the first phase, the nearest primitive from O is found by the k-d tree search for each location-time-energy triple (\mathbf{z}_j, E_p) , $j = j_{\min}, \dots, j_{\max}$, where $E_p = \frac{1}{2}(E_a + E_b)$, see also (6). The result is a perfect matching M from Z to O . In the second phase outliers are identified in the matching and removed from it.

The matching problem is the best assignment M^* of primitives to observable locations $M^* = \max_{M \in \mathcal{M}} s_D(D, M)$, where \mathcal{M} is the set of all matchings $M: Z \rightarrow D$ and s_D is the data support from (6). Ideally, matching should maximize the full support (2) but that would not be practical. Note that a RANSAC sampler does not require the proposals to be optimal. The simplification only limits the set of all possible tracks that can be found by the algorithm. The solution to the optimization problem is

$$M_i = \begin{cases} z_j & \text{if } i \in \{a, b\} \text{ or if } p_\omega p_\gamma < p_t(\mathbf{y}_i, E_i \mid \mathbf{z}_j, E_p), \\ \emptyset & \text{otherwise.} \end{cases} \quad (15)$$

The resulting assignment of primitives is not necessarily monotonic in time, but we want to avoid the overheads imposed by dynamic programming that would have to be used. We observed that when the p_t does not allow big deviations the matched sequence is rarely non-monotonic.

- 3) *Stopping Criterion*: Step 7 uses a modified RANSAC stopping criterion based on

$$c_{\text{stop}} = \frac{\log(1 - P)}{\log(1 - w_a \cdot w_b)}, \quad (16)$$

where P is the probability that the last track added to \mathcal{Q} is correct (this is our wish on the algorithm's performance), w_a is the prior probability that a randomly selected primitive in $d_a \in D$ is an inlier (belongs to a track), and w_b is the probability that the second primitive d_b is an inlier assuming the d_a is an inlier too. We used $P = 0.99$, $w_1 = \frac{30}{|D|}$, $w_2 = \frac{1}{5}$. For $|D| = 10^5$ (typical value in our data), we get $c_{\text{stop}} \approx 7.7 \cdot 10^4$.

We assumed that at least one of the restricted proposals for any given track consists of track inliers (correct primitives). We found this restriction work very well in practice. The choice brings a significant speedup of the overall runtime over the standard RANSAC proposals in which both primitives in the initial pair are drawn randomly from O (for this case $w_b = w_a$ in (16), which would result in a 1000-fold slowdown).

4) *Final Detection Decision*: The output from the RANSAC optimization procedure is a list of k best track hypotheses. Some of them are false tracks of hot pixels. To obtain the final detection, we filter all tracks by a classifier. It is based on two criteria:

- 1) temperature support s_F must be above zero,
- 2) tracks shorter than 10 matched primitives must have the total support $s(M, Z) > 25$.

The second rule was learned manually on Day 1 data (one third of input data, see Sec. III-A for data description).

III. EXPERIMENT

A. Data

Images were obtained from 50cm TAOS sensor from Lulin observatory, Taiwan. This is the same data set used for the performance study in [8]. The field of view of the telescope was $1.3^\circ \times 1.3^\circ$, the effective image size 2049(V) \times 2047(H), 16 bit monochromatic. The telescope is pointed at a fixed inertial point near geostationary region. A series of 29 images is taken with 5.9s exposures in 8.8s observation intervals without sidereal tracking or telescope repositioning. Then the telescope is repositioned back at the original inertial point. The sequence is repeated for about seven hours and the observation is repeated for three consecutive nights. Data contains artifacts of varying degree of severity, mostly reflections off clouds and stray light getting in the optical system and reflecting off the inner optical path components, creating various patterns. These are the key factors that limit the maximum detectable object magnitude.

B. Results

We learned the parameters of the individual support terms in (2) independently on data from Day 1. This way we avoided overfitting.

All RANSAC tracks were inspected and obvious false positives were manually labeled. The eight manually identified misdetections include three hot pixel tracks, one track replicated due to a bug in data preprocessing and four false tracks that were found in a region of very high density of artifacts due to stray light patterns in the telescope which could have been eliminated by removing a few images instead of running the method blindly on all the data available.

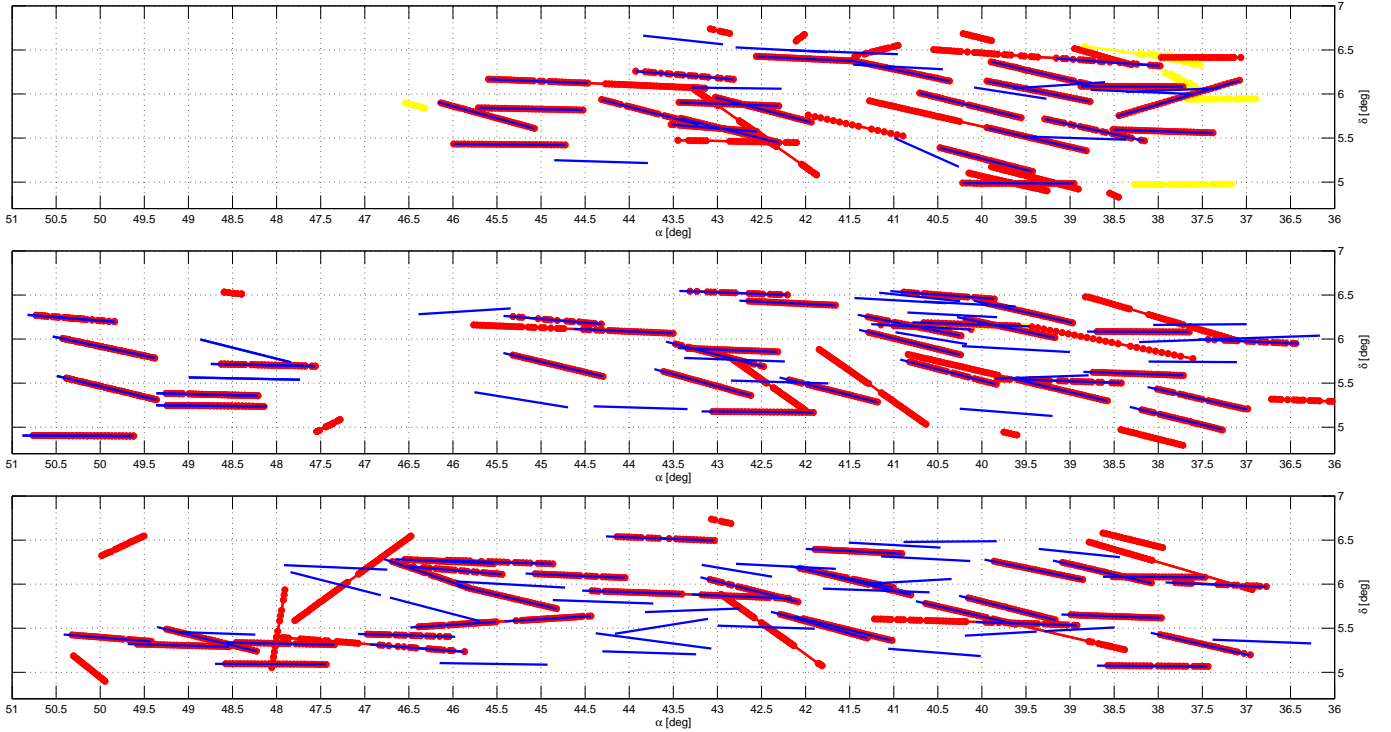


Fig. 4: Detection results for Day 1–3. **Red:** tracks detected by RANSAC (the proposed method), **blue:** tracks detected by the FPGAS [7], **yellow:** RANSAC tracks manually labeled as false positives. In RANSAC the solid dot markers (red, yellow) show locations of primitives matched to the tracks; gaps are mostly due to missing data (the object flew out of field of view during repositioning of the telescope and then returned). The abscissa is the right ascension angle α and the ordinate is the declination angle δ .

We compare our results with the FPGA Stacking Method (FPGAS) that was evaluated in [8]. Detailed track detection results from both methods are shown in Fig. 4. RANSAC detected the red tracks. FPGAS detected the blue tracks. We can see that RANSAC detects some extra tracks at a greater angle. It is possible that the FPGAS did not search over these angles, since most of these objects are of relatively low magnitude that were not of interest in the original study. On the other hand, RANSAC was also able to detect many short tracks. In addition, it detects tracks that continue after a pause (when the telescope was blind due to repositioning) and tracks that entered the field of view in the middle of the data acquisition sequence. A count-based (or median-based) method like FPGAS cannot easily detect the latter three types of tracks.

The table in Fig. 5 shows the number of detected tracks that were manually verified as correct in the RANSAC method (4th column) and the FPGAS method (5th column). The 6th column shows the number of verified tracks that were found by both methods. We can see that in total, RANSAC detected 177 tracks, and FPGAS 146 tracks. In addition, RANSAC detected 8 more false positive tracks that were removed manually and are not included in column 4.

The histogram in Fig. 5 shows the magnitude distribution of detected tracks for the two methods. We can see that RANSAC detects a greater number of low-magnitude objects (mostly those with short tracks or of high track angle). In the range of magnitudes 13.5–16.5, both methods perform similarly. In magnitudes 16.5–18 RANSAC detects about a third of the FPGAS detections. RANSAC fails to detect any object of magnitude over 18. The histogram bin width was selected as optimal by the Scott’s rule [4] for the given number of tracks (146), to obtain a statistically relevant distribution.

IV. CONCLUSION

We have tested the proposed RANSAC method on a dataset sufficiently large for obtaining predictive performance figures. The results were compared with FPGAS, which is the best performing method that was run on the same data. In summary, the proposed RANSAC method is stronger than FPGAS in detecting short tracks and tracks entering the FoV in the middle of the sequence. The FPGAS method is stronger in detecting high-magnitude tracks.

The RANSAC method does not presume any particular movements of the object, as long as its motion velocity is constant. RANSAC can detect tracks without any constraints on their angular direction. The detection does not require repeated image transformations (rotations etc.), which makes it computationally efficient. These traits are similar to LINE [7]. Unlike in the LINE method which could be considered a maximum-likelihood type of method, the Bayesian model behind the RANSAC method allowed us to incorporate additional constraints on the level of the entire track, namely consistency between primitive

data summary			verified track counts by method			
day	images	primitives	RANSAC	FPGAS	both	FP
1	1 584	78 970	35	37	23	5+2(hot)
2	1 980	91 461	42	51	32	1(hot)
3	2 250	105 894	40	58	33	0
tot	5 814	276 325	117	146	87	5+3(hot)

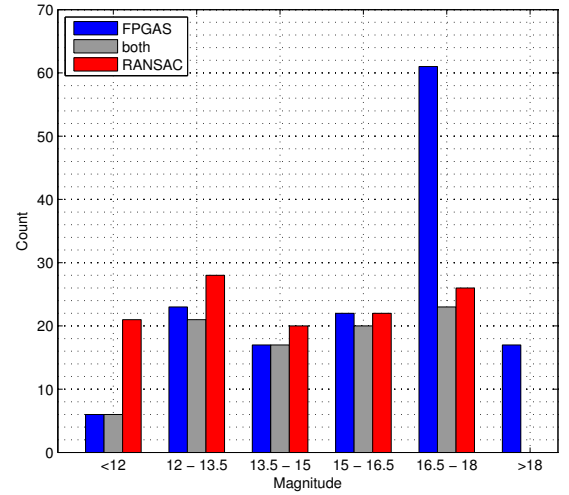


Fig. 5: **Left:** The number of detected tracks per method and data acquisition day. RANSAC: the proposed method, FPGAS: the FPGA Image Stacking method [7], both: detections common to both methods, FP: false positives by RANSAC that were manually removed (these are not included in the RANSAC column). **Right:** The number of detected tracks per apparent object magnitude range. There are no detections by RANSAC for magnitudes greater than 18.

length and the track speed and weak magnitude consistency of the object forming the track.

The LINE method detected 87 objects where the RANSAC detected 117 objects and FPGAS detected 146 objects. The RANSAC detection time is linear in the number of input images and, unlike in the LINE proposal method, the number of RANSAC proposals is (theoretically) independent of the number of detected image primitives. The current (unoptimized) experimental implementation of RANSAC run about 5.5 hours on a standard CPU architecture which seems comparable to LINE, according to [7].

Better results could be obtained by a more rigorous learning. Learning is difficult with the current dataset for which there is no ground truth. Proper learning would probably need simulated data with ground truth.

ACKNOWLEDGMENT

This work was supported by the Missile Defense Agency (USA), contract 201-0528-01, and by the Czech Science Foundation under Project P103/12/1578. Special thanks go to Toshifumi Yanagisawa of JAXA who made their data available to us, assisted us with our initial analysis of the data and who engaged in fruitful discussions about the topic of this paper.

REFERENCES

- [1] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.
- [2] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.
- [3] A. Nakajima, T. Yanagisawa, T. Kimura, T. Isobe, T. Tsuji, M. Yamamoto, T. Hoshino, M. Suzuki, and H. Futami. Space debris observation by ground-based optical telescopes. In *Proc 22nd Int Symp Space Technology and Science*, pages 2055–2060, 2000.
- [4] D. W. Scott, editor. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 1992.
- [5] J. R. Taylor. *An Introduction to Error Analysis*. University Science Books, 1982.
- [6] P. H. S. Torr. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, April 2000.
- [7] T. Yanagisawa and H. Kurosaki. Detection of faint GEO objects using JAXA’s fast analysis methods. *Transactions of the Japan Society for Aeronautical and Space Sciences, Aerospace Technology Japan*, 10(28):Pr_29–Pr_35, 2012.
- [8] T. Yanagisawa, H. Kurosaki, H. Banno, Y. Kitazawa, M. Uetsuhara, and T. Hanada. Comparison between four detection algorithms for GEO objects. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, 2012.