Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 57 (2016) 1607–1623

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Streak detection and analysis pipeline for space-debris optical images

Jenni Virtanen[a,*], Jonne Poikonen[b], Tero Säntti[c], Tuomo Komulainen[c], Johanna Torppa[a,1], Mikael Granvik[a,d], Karri Muinonen[a,d], Hanna Pentikäinen[d], Julia Martikainen[d], Jyri Näränen[a], Jussi Lehti[c], Tim Flohrer[e]

[a] *Finnish Geospatial Research Institute, Geodeetinrinne 2, FI-02430 Masala, Finland*
[b] *Kovilta Oy, Örninkatu 15, FI-24100 Salo, Finland*
[c] *Aboa Space Research, Tierankatu 4B, FI-20520 Turku, Finland*
[d] *Department of Physics, Gustaf Hällströmin katu 2A, FI-00014 University of Helsinki, Finland*
[e] *ESA/ESOC Space Debris Office, Robert-Bosch-Strasse 5, DE-64293 Darmstadt, Germany*

Received 17 November 2014; received in revised form 9 September 2015; accepted 16 September 2015
Available online 25 September 2015

Abstract

We describe a novel data-processing and analysis pipeline for optical observations of moving objects, either of natural (asteroids, meteors) or artificial origin (satellites, space debris). The monitoring of the space object populations requires reliable acquisition of observational data, to support the development and validation of population models and to build and maintain catalogues of orbital elements. The orbital catalogues are, in turn, needed for the assessment of close approaches (for asteroids, with the Earth; for satellites, with each other) and for the support of contingency situations or launches. For both types of populations, there is also increasing interest to detect fainter objects corresponding to the small end of the size distribution.

The ESA-funded StreakDet (streak detection and astrometric reduction) activity has aimed at formulating and discussing suitable approaches for the detection and astrometric reduction of object trails, or streaks, in optical observations. Our two main focuses are objects in lower altitudes and space-based observations (i.e., high angular velocities), resulting in long (potentially curved) and faint streaks in the optical images. In particular, we concentrate on single-image (as compared to consecutive frames of the same field) and low-SNR detection of objects. Particular attention has been paid to the process of extraction of all necessary information from one image (segmentation), and subsequently, to efficient reduction of the extracted data (classification).

We have developed an automated streak detection and processing pipeline and demonstrated its performance with an extensive database of semisynthetic images simulating streak observations both from ground-based and space-based observing platforms. The average processing time per image is about 13 s for a typical 2k-by-2k image. For long streaks (length >100 pixels), primary targets of the pipeline, the detection sensitivity (true positives) is about 90% for both scenarios for the bright streaks ($SNR > 1$), while in the low-SNR regime, the sensitivity is still 50% at $SNR = 0.5$.

Keywords: Space debris; Asteroids; Optical observations; Automatic detection; Image processing

[*] Corresponding author.

E-mail addresses: jenni.virtanen@nls.fi (J. Virtanen), jonne.poikonen@kovilta.fi (J. Poikonen), teansa@asro-space.com, tuomo.komulainen@asro-space.com (T. Komulainen), johanna.torppa@gtk.fi (J. Torppa), mgranvik@iki.fi (M. Granvik), karri.muinonen@helsinki.fi (K. Muinonen), hanna.pentikainen@helsinki.fi (H. Pentikäinen), julia.martikainen@helsinki.fi (J. Martikainen), jyri.naranen@nls.fi (J. Näränen), jussi.lehti@asro-space.com (J. Lehti), tim.flohrer@esa.int (T. Flohrer).

[1] J. Torppa currently at Geological Survey of Finland.

http://dx.doi.org/10.1016/j.asr.2015.09.024
0273-1177/

1. Introduction

During the past decades, the near-Earth space has become densely populated by man-made objects. It has been estimated that our space activities have left behind a population (active satellites as well as debris) of over 700,000 objects larger than 1 cm (from ESA's MASTER model,[2] Flegel et al. (2011)). To have a realistic understanding of this highly dynamic population – its size and orbital distribution, frequency of collisions etc. – constant monitoring of the objects is required. Observations are being carried out with both ground-based and space-based instrumentation. When observing from ground, optical telescopes are best suited for high-orbiting debris (e.g., geostationary orbits, GEO), while radars can detect debris on low-Earth orbit regime (LEO, below 2000 km). However, there is an increasing interest in applying ground-based optical systems also to LEO objects, to complement the use of radars (e.g., Milani et al. (2012)). On the other hand, as ground-based sensors have a detection limit of centimeters (LEO) to decimeters (GEO), space-based monitoring has been considered to assess the population of small debris objects (e.g., Flohrer et al. (2005)). For optical surveys of space debris, different observing scenarios have been applied and suggested (see, e.g., Schildknecht et al. (2004, 2006) and Valtonen et al. (2006)). While specific (object) tracking techniques can be applied to make the objects appear point-like or as short trails in the exposures, e.g., typical for GEO observations, the general survey scenario is always a "track-before-detect" problem, resulting in trailed objects, or streaks, of arbitrary lengths in the acquired images.

We present results from an ESA-funded StreakDet (streak detection and astrometric reduction) activity, which has aimed at formulating and discussing suitable approaches for the detection and astrometric reduction of trailed objects (streaks) in optical observations, as well as developing and evaluating prototype implementations. The main application targets for the processing pipeline were observations from optical LEO surveys, as well as from space-based optical surveys. The study was further restricted to single-image detections, as compared to rather a typical approach with consecutive frames of the same field, to support the detection of LEO objects, which are characterized by short pass duration and short field-of-view (FOV) crossing times which makes obtaining consecutive images (often referred to as a "tracklet") or even images from the same pass challenging. A major focus in the algorithm development has been on the low-SNR processing, i.e., faint and typically, small objects, but also on the efficiency of computation on current and near-future space-based platforms.

The currently available, mature image-processing algorithms for detection and astrometric reduction of optical data cover objects that cross the sensor FOV comparably slowly, that is, nearly point-like objects, or within a rather narrow range of angular velocities, resulting in streaks with predefined lengths. In addition to the routine algorithms for processing point-like objects, several considerations for streak-like objects are also available, including rather complete processing pipelines such as the Apex II image processing package (see, e.g., Kouprianov (2008)), Pan-STARRS Image Processing Pipeline (IPP) and Moving Object Processing System (MOPS), or the TAROT software (Laas-Bourez et al., 2009). However, the existing algorithms are typically restricted by one or more assumptions on the observing scenario (e.g., use of object tracking, i.e., assuming star streaks with predefined length and orientation), on the detected objects (e.g., GEO targets producing streaks with predefined length), or require multiple images of the target. In the case of the survey system currently running at ESA's Optical Ground Station (OGS), the algorithm is not designed and optimized to find faint streak-like features. In particular, some of the challenges identified in the objectives of the StreakDet study have not yet been properly addressed. For the LEO and space-based surveys considered, objects typically produce long, potentially non-linear (curved) and non-uniform (discontinuous) streaks, in short, complex streak morphologies, requiring specific algorithms to identify the streaks in the acquired images.

To meet the objectives in the StreakDet framework, we have considered the processing to consist of two major phases: *segmentation*, robust low-SNR extraction of all necessary information (features) from one image, and subsequently, *classification*, the characterization and efficient reduction of the extracted data. The third and last phase of the basic pipeline, *astrometric and photometric reduction* consist of more routine-like tools for coordinate and magnitude conversions. We discuss the adopted approach in more detail in Section 2.1 and review some existing approaches for the two main phases, some of which have served as a starting point for our algorithm development (Section 2). The prototype pipeline is complemented with potential post-processing steps in terms of orbital validation (Section 3.1) and classification method using training data (Section 3.2). In Section 4, we demonstrate the performance of the prototype using both semisynthetic and real observations of space debris. We give our conclusions in Section 5.

2. Processing pipeline

2.1. About the adopted approach

Segmentation. A common difficulty in all astronomical image analysis is the detection of very low-SNR objects, whether space debris, asteroids, or, e.g., Low Surface Brightness (LSB) galaxies. Solving this problem is the main aim of segmentation: to separate unknown or transient objects from an image containing a large number of known

[2] https://sdup.esoc.esa.int.

static objects (i.e., stars) in order to focus further analysis only on image regions containing potentially interesting targets. The low intensity of the objects to be detected is typically combined with non-uniform image background intensity, which makes it difficult to extract faint objects across the image with a single global set of intensity parameters. Limitations of the CCD camera and optical detection may introduce additional noise to the image to be analyzed. The applicability of different segmentation methods depends on the observing scenario, which has to be considered in the implementation and parameterization of the segmentation.

Best low-SNR feature extraction sensitivity is typically achieved either by assuming multiple, consecutive images of the object (e.g., Stöveken and Schildknecht (2005), Gural et al. (2005), and Yanagisawa et al. (2012)) or by including some a priori assumptions of the motion of the object to be detected (Laas-Bourez et al., 2009; Lévesque and Buteau, 2007). Tagawa et al. (2014) use both information to detect very faint GEO objects, but conclude that the algorithm is too time-consuming to be applied to the LEO region. The Fuzzy Grayscale Hit-and-Miss transform (FHMT) proposed by Perret et al. (2009) was demonstrated to extract LSB galaxies with SNR as low as 1 dB. In order to include both foreground and background pixels inside the structuring elements used to implement FHMT, some prior knowledge on the shape, size, and orientation of the target object is required, or different filter templates have to be used sequentially.

Another interesting approach for feature extraction, the Fourier Phase-Only Transform (PHOT) was proposed by Aiger and Talbot (2010). PHOT provides a computationally very simple method for detecting anomalous local features from otherwise regular (or random) background data. The phase-only content of the Fourier transform was shown to contain significant information, e.g., with respect to edge locations in the image. The method was demonstrated to effectively remove regular as well as random patterns, such as noise, and to preserve localized features with a clearly different phase response. The PHOT algorithm can therefore emphasize image features which do not correspond to overall background texture of the image, i.e., provides a means for novelty detection.

For the StreakDet processing pipeline, the background noise in a telescope image as well as randomly distributed faint point-like stars could be seen to constitute the random background texture of the image, while streak-like structures could be separated by their different response in the PHOT analysis. A 2D FFT-based PHOT transform (Zhang et al., 2013) was initially tested for segmentation, with a goal to extract linear streak features, while removing, at least a majority of, background stars. While the PHOT concept was found to work in principle, sufficient low-SNR performance was not achieved.

The methods developed for the segmentation of low-SNR features in the pipeline are based on conceptually similar area-based grayscale intensity estimation and multiple-window shape analysis than, e.g., in Lévesque and Buteau (2007) or Perret et al. (2009). The proposed algorithms also share similarities to the approach of line extraction from locally-contrast-enhanced low-SNR data proposed in Milani et al. (1996). However, the focus in the implementation is on computational simplicity and use of extensive binary (1-bit) processing for best possible efficiency.

Classification. There are several aspects in the StreakDet objectives, which have led us to consider the subsequent processing of the extracted image features as a classification process. First, as already described, the variety of feature morphologies present in the images is expected to be large, resulting from different observing scenarios (ground- vs. space-based, sidereal vs. object tracking), real object properties (angular velocity, rotational properties), defects of the optical system (blooming and diffraction spikes, ghost and edge effects), as well as observing conditions (seeing variations). Parametrization of targets is the backbone of any classification scheme, where the goal is to come up with measures which can be used to define the final set of parameters defining the classes: parameters that are sufficiently similar within any of these classes and sufficiently distinct from other classes. A recently developed method for automated ice-crystal classification for CCD images from the airborne CPI (Cloud Particle Imager) experiment (Ice-crystal Classification with Principal Component Analysis, IC-PCA; Lindqvist et al. (2012)) has offered a starting point for the geometric and statistical parameterization of the image features.

Second, in the adopted single-image processing scheme and in the low-SNR regime, it is crucial to separate artefacts from the real streaks, as there is no validation available in terms of subsequent images, and the amount of extracted features is expected to become larger when pushing for higher sensitivity. Choice of parameters plays a role, but also finding optimal thresholds, or filters, to reject false detections while keeping (all or most, depending on the set objective) of the real streaks.

Both Lévesque and Lelièvre (2008) and Kouprianov (2008) have discussed a false-alarm rejection algorithm for low-SNR streak detection based on morphological analysis of the streaks and propose parameters, such as the object's elongation or width of the point spread function (PSF), to be used in the rejection. For parametrization of the streaks, Levesque et al. have relied on computation of the moments of inertia, while Kouprianov has developed a PSF-fitting algorithm for elongated objects. Similar method for improving the astrometric accuracy of streak detections was also discussed by Vereŝ et al. (2012). These approaches have served as a basis for the streak characterization and filtering scheme described in Section 2.3.

We put forward three intuitive main classes for streaks based on the assumed observing scenarios, object rotation, and the adopted segmentation scheme (see Section 2.2): point-like (P), linear (L), or curved (C), with detailed classes formed as a combination of the main classes (e.g., varying, discontinuous streak composed of linear features (LL), see Table 1).

Finally, we note that the streak classification scheme implemented in the basic pipeline can be considered as pattern matching, and the complementary Streak-PCA algorithm (Section 3.2) is pattern recognition and requires that a comprehensive training set of images from the observational set-up is available or is being collected during processing.

As illustrated in Fig. 1, processing is carried out in three main parts: segmentation, classification, and astrometry and photometry. The algorithms implemented are described in detail in the following sections.

Table 1
Considered observing scenarios and predefined streak classes.

	Sidereal tracking	Object tracking

Ground-based survey	L = linear, uniform
	LL = linear, varying[a]
		P = (nearly) point-like L, LL

Space-based survey	P, L, LL
	C = curved, uniform
	CC = curved, varying
		(Not considered)

[a] Note: class P is not present, as point-like features are removed in segmentation.

2.2. Segmentation: extracting features

The developed method applies the same general principle that was proposed in Lévesque and Buteau (2007), where a multiple-window filter was used to separate (round) stars from (elongated) streaks. However, in the algorithmic implementation proposed here the actual shape segmentation and star removal process is performed in binary (1-bit black and white; BW) image domain, in order to reduce the amount of computationally intensive grayscale (GS) computation. In Milani et al. (1996), brightest objects in the image, mainly stars, where first removed by a masking operation after which faint features where extracted through median-based local-contrast enhancement. Streak features were then located with a Hough-like optimized line-extraction algorithm. The proposed StreakDet algorithm shares similarities with this approach, in the use of local feature enhancement and streak feature classification. However, the proposed additional binary processing of low-SNR feature data and subsequent object-level classification lead to less false positive detections than indicated by the example of Milani et al. (1996). In Milani et al., the analysis of multiple image frames was assumed to reduce false detections, whereas the StreakDet algorithm should be usable on just a single image.

In Lévesque and Buteau (2007), the extraction of very low-SNR streaks was separated from star removal, and some a priori information was available, allowing the use of a matched filter for faint streak detection. In the

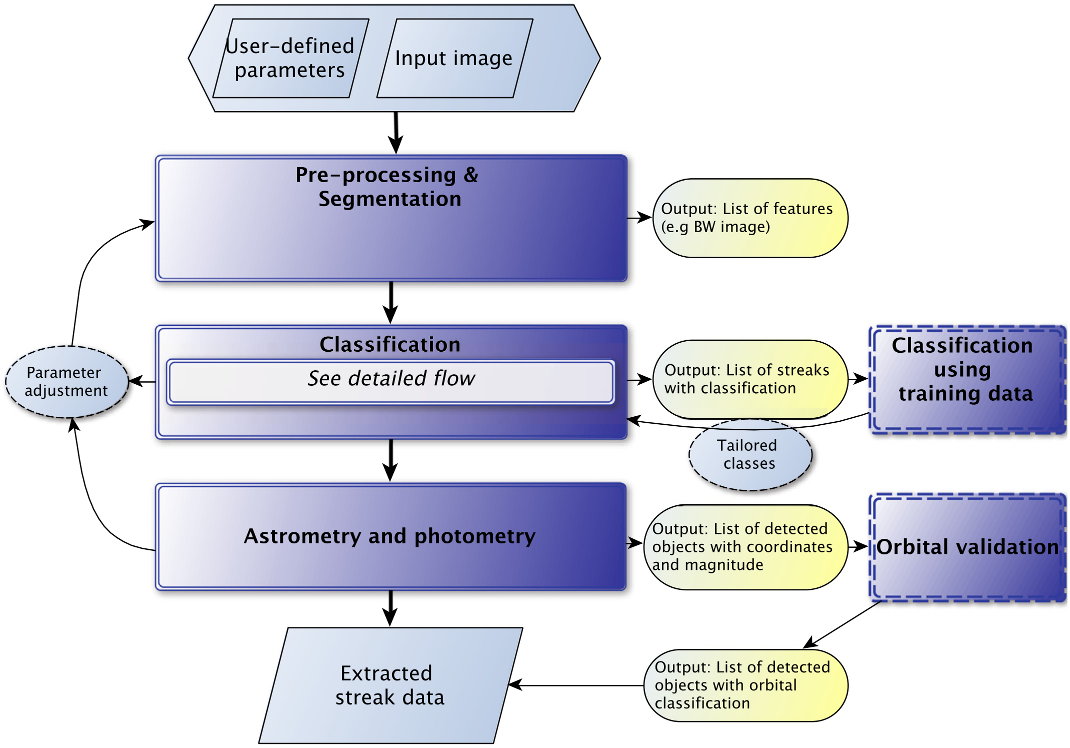
User-defined parameters
Input image
Pre-processing & Segmentation
Output: List of features (e.g BW image)
Classification
See detailed flow
Output: List of streaks with classification
Classification using training data
Tailored classes
Parameter adjustment
Astrometry and photometry
Output: List of detected objects with coordinates and magnitude
Orbital validation
Output: List of detected objects with orbital classification
Extracted streak data

Fig. 1. Current algorithmic flow chart for the prototype developed in the StreakDet project.

StreakDet framework, low-SNR streaks have to be extracted without any a priori information, which also leads to an increase in the number of faint stars to be removed, and is made even more challenging by possibly dense star fields. Reduced amount of data and simpler arithmetic operations offered by a BW-image representation are seen to enable more efficient memory use and low-level computation implementation. BW-image operations can also be more efficiently implemented with high-performance parallel processing platforms, such as Graphics Processing Units (GPUs) or Field Programmable Gate Arrays (FPGAs), than high bit-depth grayscale processing.

Specific attention needs to be given to implementing the GS processing with the simplest possible operations, in order to increase computational efficiency and move to BW-image processing as early as possible. The algorithm is based on obtaining a BW-thresholded image, which contains connected features, i.e., dense groups of pixels, related to low-SNR streak features. First, two GS mean-filtering operations are performed on the original input image. The mean value inside a small (e.g., 3×3) and a large neighborhood kernel (e.g., 21×21) is calculated for each pixel, resulting in two different GS mean images. The difference between the two mean images is then extracted and this difference image is used as input to the segmentation. The goal of the mean-difference operation is to emphasize groups of pixels that differ from their background, which includes stars as well as the desired streak features. Because the local mean-difference computation effectively removes global background gradients or gain differences in subsensors, separate pre-processing steps for image background estimation and flattening are not required. Some artefacts may be generated on the edges of the separate sub-sensor arrays in a mosaic-type imager, as can be seen from the middle image in Fig. 2. Such noise features should be removed by later segmentation and classification stages.

A threshold operation is applied to the GS difference image to extract relevant binary features at low-SNR levels. The SNR sensitivity of feature extraction, with respect to image quality, can be controlled by setting a certain target value to the number of active (binary “1”) pixels that should be extracted by the threshold operation, with respect to the image star density. The star density is estimated from the image prior to the mean-difference operation by evaluating the amount of pixels with intensity above some predefined threshold value, i.e., $N\times$(image mean value), which correspond to bright stars. This helps the algorithm to adapt to image quality, so that, e.g., in the case of a lower star density, objects with a lower-SNR can be extracted while still retaining adequate segmentation efficiency (i.e., minimal number of false positives) in the case of more crowded star fields or with longer integration times.

In practice, the threshold operation results in a very noisy BW image, containing stars, possible streaks, and noise. The image is then processed by iterating a sequence of binary filtering operations to remove all non-streak-like features. This process is based on two assumptions on the streak-related binary features: (1) the pixels belonging to a streak or a streak segment form a connected meaningful feature which has a higher local pixel density than noise features due to background noise, and (2) the features are elongated when looking at the neighborhood of each pixel. The first condition is directly related to the proper image-based selection of threshold values, based on star density, and the second imposes a certain minimum length to a streak that can be practically separated from bright (large) stars or clusters of faint stars close to each other.

Two types of basic filtering operations are used to reduce undesired features, while trying to preserve streaks: mathematical morphology and multiple-window -based pixel-density filtering. The applied morphological operations are basically adapted versions of binary erosion and masked dilation (reconstruction). Binary neighborhood-convolution operations, i.e., pixel counting with square-shaped neighborhood kernels of varying size, are used to remove isolated pixels or very small groups of pixels, which do not have enough active neighbors. This will reduce sparse features, such as random noise, more than denser areas corresponding to streak features. Erosion operations are followed by intermittent binary reconstruction, to strengthen remaining features. This way sparse pixel groups can be gradually removed while retaining dense, meaningful features, such as debris streaks and stars.

After removing noise and very faint (small) stars, remaining field stars are removed from the binary image

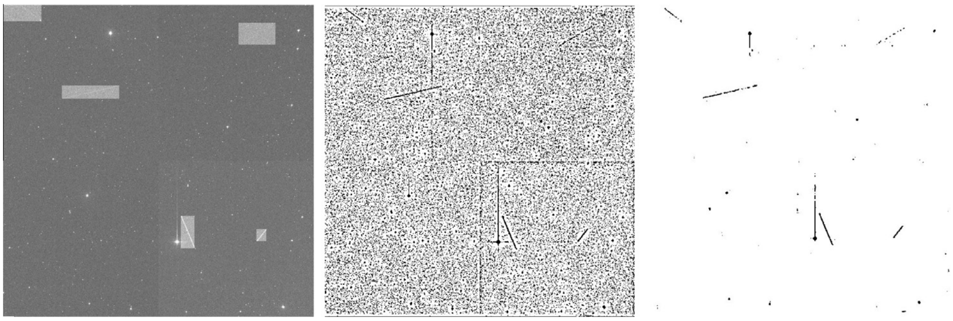

Fig. 2. Segmentation of an example image containing synthetic streaks with the proposed algorithm. Left: original image with streak regions highlighted. Middle: Result of threshold operation after mean-difference grayscale processing. Right: Result of BW streak segmentation.

by applying a multiple-window pixel-density evaluation, where the number of active pixels within two or three differently sized windows are calculated for all active pixels. If the number of active pixels is not seen to grow in an approximately linear fashion as the window size is increased, the pixel is removed. This process can also be iterated and combined with some additional morphological filtering, to gradually remove round stars, while retaining elongated debris features. An example of the segmentation result can be seen in the right-hand side image of Fig. 2. The output of the segmentation algorithm is subjected to a standard Connected Component Labeling (CCL) algorithm, which creates an indexed list of all connected pixel groups (features) in the segmented image. This information along with, e.g., size-based ordering of the CCL-objects is then provided to the feature classification process.

While a sidereal tracking is here considered as the regular operation mode, the StreakDet software can also be used to process images in an object tracking scenario, where field stars appear as linear streaks. In this case, the segmentation can be set to only remove random noise and to preserve both streak-like and round salient features. Because streak-like field stars can be assumed to share a common direction (and length) within the image, they can be removed easily in the classification stages, at CCL-object level. The segmented image should then contain only circular features (tracked objects) and streak-like features with a direction differing from the star field.

2.3. Classification: characterizing features and streaks

In the classification phase, the features extracted by segmentation are analyzed to identify potential streaks. We have put forward a sequential scheme for the StreakDet pipeline, to be applied in three different phases:

1. Characterization of BW features, i.e., pixel groups from segmentation.
2. Characterization of BW streaks, i.e., BW features corresponding to a streak.
3. Characterization of GS streaks, i.e., GS features corresponding to a streak.

Each phase includes characterization and classification steps for the detected features/streaks, and a filtering step to remove uninteresting features/streaks (see Fig. 3).

For all phases, we propose to use an eigenvalue analysis to characterize the features/streaks by computing several parameters, such as width, orientation angle, aspect ratio, and curvature of the feature (see full list of parameters in Table 2). Eigenvalue analysis turned out to give useful parameters to distinguish, e.g., point-like field stars and elongated streaks using the aspect ratio, R; complex features to be forwarded for *unlinking* (see below) using porosity, ρ_1 (i.e., estimating the amount of empty pixels in a feature); artefacts and linear streaks in ground-based survey using curvature, κ.

In **BW characterization**, we want to make sure that all individual and complete streaks have been found. For LEO surveys where long streaks are expected, the likelihood is increasing, on one hand, that a real streak is overlapping with another image feature (real streak, star, artefact), and on the other hand, that we may be detecting brightness variations of the object during the exposure. The resulting morphologies of the features can thus be complex, as either several streaks are extracted as one feature, and need to be *unlinked*, or one streak is extracted as several features in segmentation, and need to be *linked*. Especially for space-based surveys, based on simulations with the ESA PROOF model (Krag et al. (2000); here using version from 2009 with a plugin for outputting optical images), the occurrence of overlapping streaks in the acquired image is highly likely. For unlinking such features, we propose the standard Hough transformation (Hough, 1962), which is used recursively by applying the transformation on the feature (selected for unlinking by its large porosity, ρ_1), and each time identifying and saving the most prominent line as a new feature, before masking it from the analysed feature.

For linking discontinuous streaks, Hough-transformation-based methods were also evaluated, but in the end, *linear predictions* were selected due to the simplicity when dealing with clearly linear features, the main target of the developed pipeline: Starting from the largest feature found in the image, possibly related features are searched for in the proximity of its first eigenvector. An obvious drawback of the selected algorithm is that it does not work for significantly curved streaks (limiting curvature $\sim 9°$).

For **GS parametrization**, to refine the streak parameters, several approaches based on PSF fitting of elongated objects are available in the literature: axisymmetric 2D Gaussian (standard for point-like objects), 1D stretched Gaussian (Kouprianov, 2008), or moving 2D Gaussian (Vereŝ et al., 2012). We have tested the approaches with synthetic streaks, and conclude that moving 2D Gaussian approach gives more accurate and stable results than the 1D stretched Gaussian, and is thus selected for implementation. Fitting is carried out with a non-linear Levenberg–Marquardt least-squares (LS) method, using the following parameterization (see Vereŝ et al. (2012) for details, as well as streak parameters in Table 2): $(x, y)_c$, L, F_{obj}, F_{bg} (i.e., flux of the background sky), α, σ_{PSF}, κ; and constraining $L > 1$, $\sigma > 0.1$, $F_{obj} > 0$. The starting values are taken from the BW streak parameters and the GS area for the fit is extended outside the BW streak-bounding box. In addition, star pixels are removed by applying a rectangular mask to 1% of the brightest pixels in the area. For clearly linear streaks ($L > 100$ px), positions can be fitted with subpixel accuracy. However, some limitations remain, namely for streaks extending close to the image edges (potentially continuing outside the FOV) and for strongly curved streaks, the algorithm does not currently provide stable results, but instead converges to unrealistic parameter

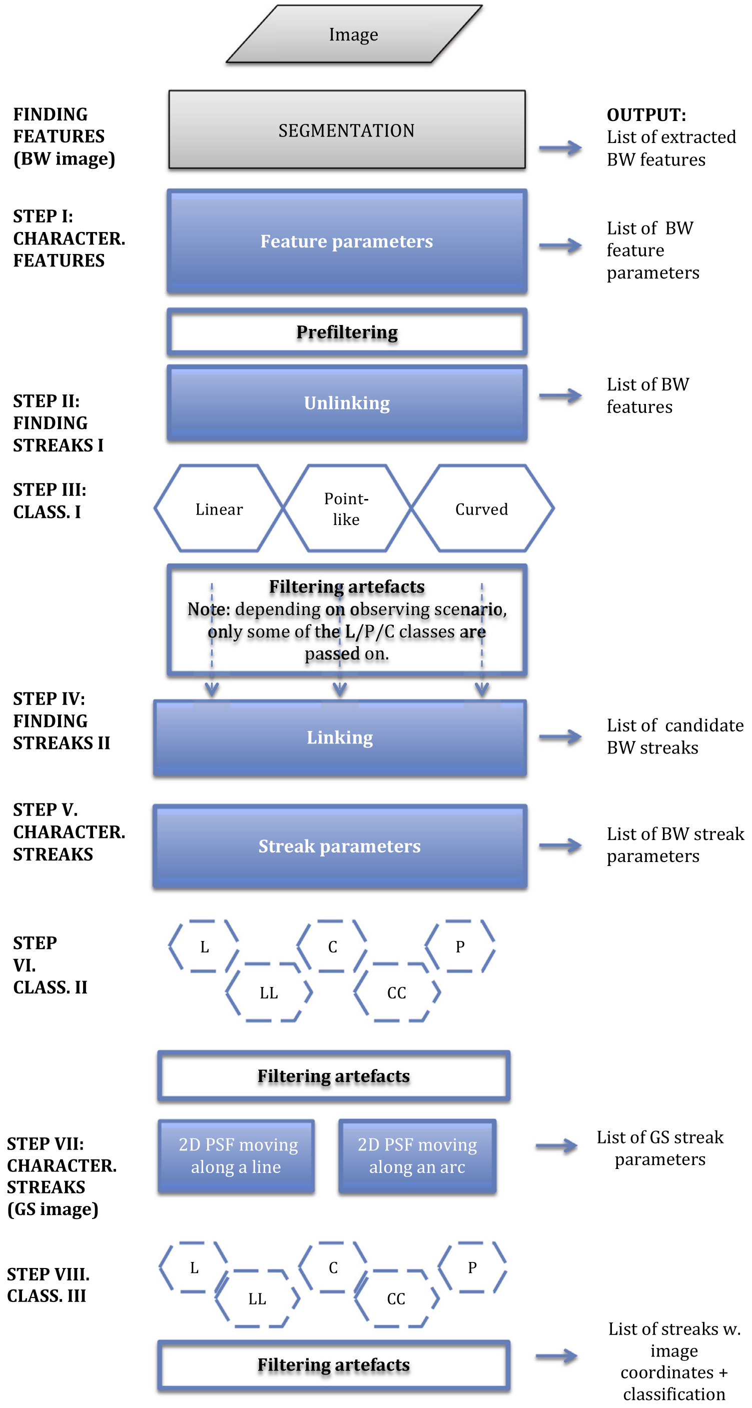
Image
FINDING FEATURES (BW image)
SEGMENTATION
OUTPUT: List of extracted BW features
STEP I: CHARACTER. FEATURES
Feature parameters
List of BW feature parameters
Prefiltering
STEP II: FINDING STREAKS I
Unlinking
List of BW features
STEP III: CLASS. I
Linear
Point-like
Curved
Filtering artefacts
Note: depending on observing scenario, only some of the L/P/C classes are passed on.
STEP IV: FINDING STREAKS II
Linking
List of candidate BW streaks
STEP V. CHARACTER. STREAKS
Streak parameters
List of BW streak parameters
STEP VI. CLASS. II
L
C
P
LL
CC
Filtering artefacts
STEP VII: CHARACTER. STREAKS (GS image)
2D PSF moving along a line
2D PSF moving along an arc
List of GS streak parameters
STEP VIII. CLASS. III
L
C
P
LL
CC
Filtering artefacts
List of streaks w. image coordinates + classification

Fig. 3. Classification flow as described in Virtanen et al. (2014).

values, such as excessive lengths ($\gg FOV$), high curvature, or inconsistent PSF widths (i.e., far from the image PSF). For point-like objects, the axisymmetric 2D Gaussian approach is sufficient, except for field stars observed in sidereal-tracking mode, in which case parameters $(x, y)_c$ are the barycenters of the objects, that is, the weighted mean positions.

Finally, a filtering scheme is put forward, where the features/streaks found are either kept in or discarded from the analysis. This is based on their computed parameters which are used to assign a *streak class* (see Table 1) to each streak, which can then be compared to the classes expected for the observing scenario in question (e.g., ground- or space-based observer). The designing of suitable filters

Table 2
Adopted feature and streak parameters.

Feature/streak parameter	Description	Definition
		BW = for BW image
		GS = for GS image

(x, y), e.g., $(x, y)_c$	Pixel coordinates, e.g., for the center	BW: from centroiding GS: from PSF fitting
L	Length	BW: maximum feature dimension GS: from PSF fitting
σ	Width, perpendicular to L	BW: σ_2, 2nd eigenvalue of the covariance matrix GS: σ_{PSF}, standard deviation from PSF fitting
α	Angle of orientation, measured counterclock-wise from positive x axis	BW: angle of the 1st eigenvector, GS: from PSF fitting
R	Aspect ratio	1 – (2nd eigenvalue/1st eigenvalue)
κ	Angle of curvature	BW: angle $(\alpha_1 - \alpha_2)$ between two halves of the feature GS: from PSF fitting
A_{px}	Area	Pixel area
$A_{2\sigma}$	Area	Area of 2σ eigenellipse
ρ_1	Porosity	$A_{px}/A_{2\sigma}$
F_{obj}	Total flux, i.e., integrated over brightness distribution	BW: flux of BW streak GS: flux of GS streak
unlinked	Logical (yes/no)	BW: from unlinking
N_{feat}	Number of features	BW: from linking
SNR	Signal-to-noise ratio, see Eq. (1)	GS: from PSF fitting

was, in practice, carried out parallel with the pipeline implementation, as it required large-scale testing to find optimal parameters to distinguish between artefacts and real streaks. Also choosing the location of the filters in the processing scheme depended on the performance of the other algorithms, implemented in a sequential manner (see Fig. 3).

We put forward four major filtering steps:

- Prefiltering (within Step I): before any operations, reject
  1. The smallest artefacts: by their pixel area, A_{px}.
  2. Blooming or diffraction features: by their orientation. Currently, we assume only horizontal or vertical features, while diffraction spikes can have other orientations as well. As a side effect, the current filter removes also real streaks if they are very close to the horizontal or vertical axis, but we note, that if blooming is not expected to be present, the filter can be turned off.
  3. Streak-like stars: For object tracking mode, the resulting star streaks can be removed at this stage. Filter is based on the assumption that stars are the most prominent features extracted, and thus recognizing the most dominating values for feature parameter distributions, such as orientation angle, length, and/or aspect ratio, and removing the related features.

- Classification and filtering I (Step III): After unlinking complex features, we have the complete list of individual features and now want to filter out those features not compatible with the underlying observing scenario (see Table 1).
  1. For ground-based observing, we do not expect curved streaks (curvature filter).
  2. For sidereal tracking, we cannot detect point-like streaks, as they cannot be distinguished from stars (aspect ratio filter is tightened).
  3. For object tracking, we primarily expect point-like streaks (aspect-ratio filter is loosened).
  4. Non-compact features, typically star remnants, can be removed based on their porosity (porosity filter). Note that the filters should not be too tight at this point, i.e., before the following linking process, not to loose (too many) features prematurely.

- Classification and filtering II (Step VI): After linking features into potential streaks, i.e., increasing the lengths and the amount of pixel data, we could attempt once more to reduce the number of data to be passed on to the next, more time-consuming step of PSF fitting. For example, aspect ratio or curvature filter could now be tightened and reapplied. Again, the filtering needs to be adjusted to the observing scenario. However, currently, no filtering is performed at this stage.
- Classification and filtering III (Step VIII): After PSF fitting, we introduce one more filtering step to reject streaks with unrealistic fitted parameters.
  1. PSF width: we check for the width of the streak against the background stars, and accept only streaks within some minimum and maximum bounds. For example, some optical artefacts such as blooming and diffraction spikes can also be removed by this filter. Occasionally, also real streaks, typically leaving the FOV may be removed.

2. Curvature: if curved streaks are expected (e.g., space-based observing), and thus curvature is included as one of the PSF-fitting parameters, we can pose a threshold for maximum curvature that is expected a priori.

With the StreakDet software, we provide the user with some example configurations for, e.g., ground-based and space-based surveys. However, we note that, when applying the processing pipeline to a new observing system or survey, some testing with a representative data set will typically be needed to find the optical configuration parameters (see also discussion on Streak-PCA algorithm in Section 3.2).

2.4. Astrometric reduction

In astrometric and photometric reduction, the relative image parameters (pixel coordinates, instrumental flux) resulting from GS streak parametrization are transformed to final output parameters (sky-plane coordinates, magnitude). We note that the emphasis in the algorithm development for the StreakDet software has been in the previous processing phases, thus here we have simply searched for working solutions for these more routine-like processing steps. However, for completeness, we give an overview of the reduction process in terms of coordinate and magnitude conversion, and especially discuss algorithms needed in the automated pipeline, namely automated selection of field stars (both for astrometry and photometry) and automated catalogue matching (to identify the star field). And as the developed software is a prototype, (some of) the algorithms can simply be replaced for an actual operational pipeline.

The selection of astrometric field stars is carried out differently for sidereal (tailored threshold operation to re-extract point-like stars) and object tracking (recognition of linear stars from the segmented features). Matching the observed star field with the star catalogue is carried out with four-star configurations and scale-and-rotation-invariant quad descriptors (Lang et al., 2010).

Coordinate conversion, i.e., plate solution is carried out first with linear mapping, obtained from the descriptor match that covers the largest area in the image, in order to minimize the error throughout the image. This linear mapping is next improved by a polynomial fit including a large number of field stars. For the detected streaks, we give three coordinate pairs (corresponding to the beginning, center, and end positions), but due to the ambiguity in the direction of a streak detected in single image, we give only two time stamps, the center time of the exposure and the exposure time.

To derive brightness estimates for the detected objects, we transform image fluxes into magnitudes as follows. We determine a set of reference stars by selecting 100 random stars from the USNO CCD Astrograph Catalogue (UCAC4; Zacharias et al., 2013) within the area covered by the image FOV, fitting them with 2D Gaussian PSF and then cleaning the data set from outliers (i.e., the stars with highest and lowest background values or total fluxes, typically 16% of the stars). The remaining stars are used to determine the magnitude conversion of the image, i.e., the magnitude zero point fitted by the Levenberg–Marquardt least-squares method, and in turn, the magnitude estimates for the detected streaks. As for the errors in relative photometry, due to several factors they are expected to be large. First of all, in the typical space debris surveys, image calibration is missing or not carried out adequately for photometric purposes. Also, surveys often aim for as high as possible sensitivity of the system and do not use any filters. This leads to intrinsically high inaccuracies in photometry. Second, the star catalogue adopted in the current pipeline, UCAC4, is not a photometric catalogue. In addition, when no filters are used, no color corrections can currently be done, further reducing the accuracy of photometry. From the fitted set of field stars, we get typical rms errors for the magnitude zero point between $V = 0.4 - 1$ mag, which gives a rough estimate for the photometric accuracy of the streak magnitudes.

2.5. Practical implementation

We have put the developed algorithms together into a automated prototype processing pipeline. The StreakDet software is implemented in C++. Multithreading is applied, but currently only to the PSF fitting, as it is the most time-consuming stage. The code has only few external dependencies, such as libraries (OpenCV, CFITSIO, LMFit, libssrckdtree, Boost and JsonBox) and the UCAC4 catalogue for astrometric reduction. All of these are freely available. The resulting software is a command-line tool, but we have also developed a web-based user interface to allow easy manipulation of all the configuration parameters and visualization of the results. Additionally, the browser-based interface supports database integration, which was used to validate the prototype with synthetic input data and comparing the results with the parameters of the synthetic streaks from the database. The command-line version can be scripted to process a large set of input images without user interaction. The software accepts two command-line options, which are the configuration file and the input image in FITS format. We assume that the FITS images have been pre-processed, that is, the basic steps of image calibration (e.g., bias, dark-current and flat-field corrections) have been carried out, removing most artifacts caused by the camera and the optical system. The software further expects that a number of parameters are provided through the FITS header, including sky-plane coordinates of the image center and plate scaling (keywords: RA, DEC, SCALE), epoch of observations and exposure time (EPOMJD, EXPTIME), binning factor used

(XBINNING; YBINNING is assumed to be same) and observatory code (OBSCODE).

StreakDet software can be configured into two different output modes: normal and full output. In normal mode, we give only the final streak parameters both after BW and GS parametrization (see Section 2.3), and the astrometric results in terms of three image and sky-plane coordinate pairs (see Section 2.4). In full mode, we provide more information on what is going on inside the pipeline, at each separate processing stage. For both modes, several output files such as feature/streak parameter listings and visualizations of the detections can be generated based on the user-defined configuration file.

In addition to the basic processing pipeline, two optional modules are provided: GOrb and Streak-PCA (see Section 3).

3. Potential post-processing steps

3.1. Orbital validation

In the single-image processing approach (as compared to multiple images of the same object/field), the verification of detections is subject to additional efforts for the observer. We discuss initial orbital analysis as a first-aid validation tool. The orbit-computation approach routinely utilized within the space-debris community follows the classical approach of celestial mechanics, that is, initial orbit determination with scarce data and differential correction as soon as the amount of astrometry allows for it. The initial orbit determination is often carried out with the methods by Gauss or Laplace or, alternatively, methods that have been developed more recently such as that by Beutler (2005). The common drawback of these classical methods is their inability to correctly portray orbital uncertainty and, in some cases, even guarantee that a 6-dimensional orbital solution is found. The recent advances from asteroid orbit computation can be divided into geometric methods (see, e.g., Milani et al. (2011) and references therein) and Bayesian methods (e.g., Muinonen et al. (2012)). We put forward the novel Markov-chain Monte Carlo version (MCMC; Oszkiewicz et al. (2009) and Muinonen et al. (2012)) of the statistical ranging method (Virtanen et al., 2001) for single-image space-debris observations resulting from our processing pipeline.

The MCMC method for heliocentric orbits was developed and implemented for the data-analysis pipeline of the ESA Gaia mission (Muinonen et al., 2015). We have modified this existing Java code to compute geocentric orbits and accept input data in the form of streaks. The implemented *GOrb* software and its Bayesian approach offers us a complementary statistical tool to estimate the probability whether the detected streaks actually correspond to Earth-bound objects. Tests with simulated streak-like data of the INTEGRAL satellite on a high-Earth Orbit (HEO) shows that the code works properly (Fig. 4) and it may be possible to distinguish between Earth-bound and non-Earth-bound objects (obtained probabilities: HEO 62%, non-Earth-bound 5%, Earth impact 33%).

3.2. Classification using training data

The streak classification in the processing pipeline is complemented with Principal Component Analysis (PCA, Jolliffe (2002)) to study the processed streak data in more detail. What we call the Streak-PCA allows for optional classification of the streak data and is developed from the PCA-based classification method for the shapes (habits) of atmospheric ice particles by Lindqvist et al. (2012). There are two alternative ways to apply Streak-PCA to the streak data: first, it can be applied to the streak parameters available after processing (see Table 1); second, it can be applied to the BW or GS streak images (i.e., the postage stamps of the original images).

For testing Streak-PCA, two data sets of streak images are generated to mimic a real application scenario: first, an independent training set of synthetic streaks with varying parameters, corresponding to the different streak classes (P, L, and C) foreseen for streaks detected in various surveys; and, second, a test set of detected streaks acquired from the implemented processing pipeline when applying it for a selection of semisynthetic images from our database (Section 4). From the training data, we establish the orthonormal eigenvectors in the parameter space to be used when analyzing the actual streak data.

In order for the classification to succeed, there are certain standard procedures that need to be carried out for all individual streak images. A 2 × 2 covariance matrix is

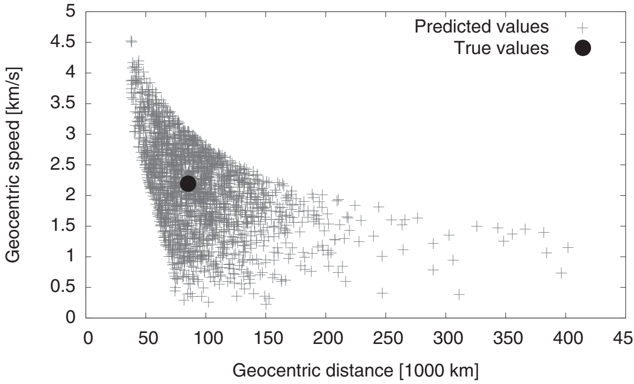

Fig. 4. The cross (+) symbols show the geocentric distance and speed distributions for the INTEGRAL satellite soon after launch based on a single synthetic streak spanning about 20 s. The true distance and speed for the satellite (based on a more extensive set of data) are shown with the black circle. The probability for a high-Earth orbit is about 62%, the probability for Earth impact is 33%, and the probability for a non-Earth-bound orbit is 5%. Note that, due to the 2-body approximation used and the ambiguity in the direction of the object, the probabilities are independent of the assumed direction of the streak and an Earth impact is equivalent to a launch from the Earth. The astrometric uncertainty was assumed to be 1".

computed for the streak in the x, y reference frame of the image. After solving the eigenproblem for the covariance matrix, that is, the eigenvalues describing the axis lengths squared and the eigenvectors giving the directions of the longest and shortest axes, a principal-axes x_p, y_p reference frame is determined.

Before rotation into that reference frame, the curvature of the streak is analyzed by splitting the streak along the minor principal axis and solving the eigenproblem for the two resulting parts. The eigenvectors of the parts indicate the sign of the streak curvature. The streak is then rotated into the principal axes reference frame so that it shows negative second derivative (upward convex).

Finally, the streak is mapped into a standard square window or x_s, y_s reference frame, where both of the coordinates run from -1 to 1. Note that the Streak-PCA is tuned for the classification of shape rather than size, although the size is also available, in a straightforward way, from the absolute values of the eigenvalues described above. In the x_s, y_s reference frame, the silhouettes of the streaks are determined as the minimum and maximum values for the y_s coordinates of illuminated "standard pixels" for each of the standard-pixel x_s coordinates.

The Streak-PCA classification method is completed by the k-Nearest-Neighbor classifier (k-NN classifier, cf. Lindqvist et al. (2012)). After training the classification algorithm, a streak analyzed obtains a class on the basis of the nearest neighbors of the training data set in the Principal Component (PC) subspace chosen. For example, $k = 3$ and $k = 5$ can be utilized in assigning the class. If all of the k nearest neighbors are within the same training class, assigning a class to the streak analyzed is straightforward. If there are several classes present among the k nearest neighbors, procedures need to be established beforehand on how the classification proceeds. Streaks can be left unclassified, for example, for $k = 5$ when there are less than three "votes" for a specific class among the nearest neighbors.

The training data is constructed as "an educated guess". For the training data set, we have generated altogether 3100 streaks for 31 different classes, with 100 streaks in each class: (i) Class 1 consists of streaks with equidimensional shapes (termed point-like); (ii) Classes 2–6 consist of linear streaks with axial ratios 2^{-n} with $n = 1, \ldots, 5$; (iii) Classes 7–11 consist of curved streaks with axial ratios 2^{-1} and angles of curvature $\kappa = (2^m - 1)^\circ$ with $m = 1, \ldots, 5$ (see Table 2); (iv) Classes 12–31 are, in sequences of five, as Classes 7–11 but for axial ratios 2^{-n} with $n = 2, \ldots, 5$. The standard square image is here assumed to be a 51×51 image. Thus, there are 2601 pixels altogether but the silhouettes are described by only 102 values for the y_s coordinates.

In what follows, we discuss the tentative application of the Streak-PCA to altogether 954 streak images output from the Prototype data-processing pipeline that is based on a fully independent simulation. The sizes of the images range from 7 to 366 pixels in the x-direction and from 7 to 1162 pixels in the y-direction. If the number of the pixels in either the x- or y-direction exceeded 401, the images are binned so that the image is scaled to fit into an image of 401×401 pixels. Note that the present application is a blind test where the people running the Streak-PCA have received the simulated streak data without any graphical or other description of the test streaks involved.

These streaks cover a wide range of axial ratios and curvatures. From the training data, we have established the orthonormal eigenvectors in the parameter space, that is, solved the principal components pertaining to the full training data set. Fig. 5 shows the training data in the PC1, PC2 and PC1, PC3 planes. It is clear that the first

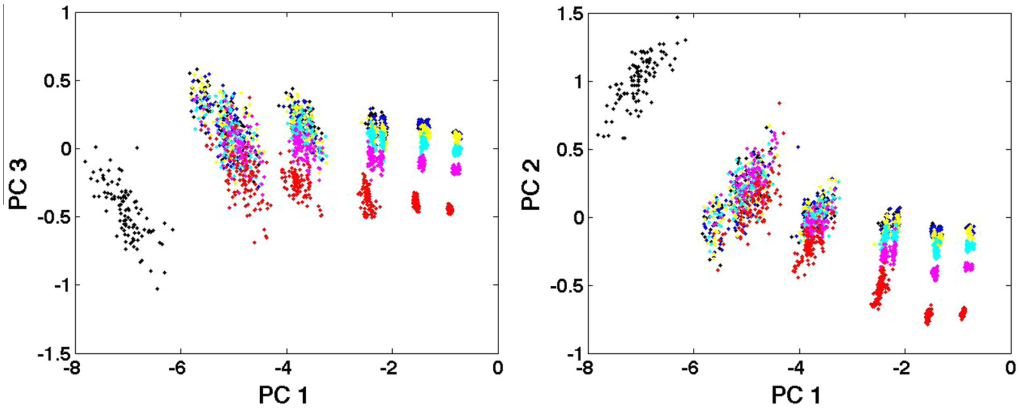

Fig. 5. Principal components PC2 and PC3 plotted against PC1 for the training data of 3100 simulated streaks. The point-like streaks show up as a separate clump to the left in PC1, and the axial ratio is increasing to the right. From the left to the right, the black color corresponds to round (point-like) and linear streaks of increasing axial ratio (mostly hidden below the dark blue points towards the top of the clouds of points). The dark blue, yellow, light blue, pink, and lighter red colors then correspond to streaks with increasing curvature, grouping with the linear streaks of similar axial ratio. Clearly, PC1 is primarily a measure of the axial ratio, whereas PC2 and PC3 characterize the curvature of the streak. Increasing curvature shows up as increasing separation from the black overall trend. For both PC2 and PC3, the curved streaks become well separated for large axial ratios, that is, towards increasing PC1 values. The present example shows that Streak-PCA can be utilized to characterize the streaks directly from the image data.

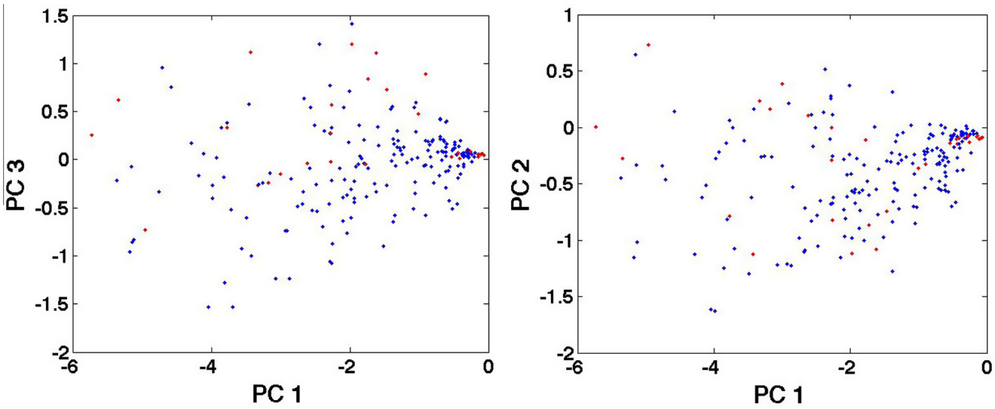

Fig. 6. Prototype test data of 954 streaks plotted in the PC1–3 space defined by the training data (see Fig. 5 and text). The blue bullets describe streaks that obtain one of the training classes with the criterion that, with k-NN and $k = 5$, at least three out of the five nearest neighbors indicate the same class. The red bullets then describe unclassified streaks. First, there are essentially no point-like streaks in the test data. Second, there are substantial numbers of streaks with axial ratios larger than in any of the classes of the training data. Third, there are substantial numbers of strongly curved streaks in the data. Fourth, it is highly probable that the multicomponent and/or porous streaks produce "a diffuse background" in the PC1–3 space. Further studies could be carried out on how to separate such streaks from the compact ones analyzed here.

three principal components (potentially already the first two components) provide a measure of the axial ratios and curvatures of the individual streaks. It is worth emphasizing that no particular features have here been extracted from the images: only the silhouettes of the streaks have been utilized in the analysis.

Fig. 6 shows the location of the 954 test streaks in the PC1, PC2 and PC1, PC3 planes defined by the training data set. Clearly, first, point-like streaks are essentially completely missing from the test data (a completely realistic conclusion based on the segmentation phase). Second, the streaks in the test data tend to have larger axial ratios than the streaks in the training data. Third, there are strongly curved streaks in the test data. Fourth, multicomponent streaks in the test data tend to cause some diffuse background that extends to domains not covered by the training data. Finally, in a real-world application, the training data would next be iterated to better match the data from the observation program, that is, the test data in the present example. Or, alternatively, the observation program should be subjected to a careful analysis giving answers to why rather peculiar streaks are detected by the program.

Applying the method to test data composed of 954 candidate streaks, we have shown that Streak-PCA succeeds in characterizing the test streaks using $k = 3$ or $k = 5$ in the k-NN classifier. In the demonstration, the training data has been generated for clearly separate classes of P, L, and C streaks, allowing us to establish a robust coordinate system for the PCA. In practice, as for the test streaks, the P, L, and C streaks show a continuum of shapes. For the application of the Streak-PCA method, it seems to make general sense to establish a once-and-for-all training system that can be utilized in all streak classification applications. What has been described here can serve as a precursor towards such as system.

The Streak-PCA needs to be tuned for each space-debris observation program individually. That includes the incorporation of specific training data for the program at hand. Once the Streak-PCA has been tuned, it can serve as a valuable tool in the monitoring of the observing system performance. As the streak parameters are constantly computed in the data-processing pipeline, the Streak-PCA can be called for at any particular part of the pipeline to help make decisions for further processing.

4. Performance of the pipeline

4.1. Simulated streak observations

Database of test images. For testing the developed processing pipeline, we have built a representative database for different observing scenarios (see Table 1). The database consists of semisynthetic images created by adding artificial streaks to various real backgrounds. For simulating both ground- and space-based surveys, we have selected a set of optical images observed at several sites, with different pixel scales (0.2–0.8 arcsec/px), star densities[3] (0.2–2.6) and seeing conditions (PSF widths 1–2.5 px). Nearly 20,000 synthetic streaks with different morphological properties were generated into the images by varying several parameters either systematically (length, mean brightnesses, curvature) or randomly (position, orientation, amplitude for brightness variations). For the parametrization of the space-based scenario, we carried out studies with the PROOF software (Krag et al., 2000) to estimate the expected level of curvature and existence of overlapping streaks.

[3] Star density is computed as the percentage of pixels covered by stars, i.e., bright pixels over a certain threshold.

In particular, we have simulated streaks with different signal-to-noise ratios (SNR). In the present study, SNR is defined per streak:

$$SNR = \frac{F_{\text{obj}}}{\sqrt{F_{\text{obj}} + F_{\text{bg}} + \dots}} \tag{1}$$

where F_{obj} is the per-pixel flux from the object averaged over the area covered by the object, and F_{bg} is the per-pixel median flux of the background. While streak SNR values range from 0 to 55, streak parameterization has been applied so as to emphasize the low-SNR regime, i.e., most of the generated streaks have $SNR < 2$.

Case studies. We demonstrate the capabilities of the processing pipeline with selected examples representing the different observing strategies considered:

- Ground-based, sidereal tracking (Fig. 7). Results for detecting linear streaks in both backgrounds with low (< 0.5) and high (> 0.5) star densities show that, while long, bright streaks can be detected in a straightforward way, faint streaks ($SNR < 1$) are typically only detected partially. Short streaks ($L < 50$ px) are more likely to be lost already during the segmentation step.
- Ground-based, object tracking. Fig. 8 for a streak-like star field illustrates how the segmented data can be efficiently reduced during Classification. Currently, some clearly star-related features are included among detected streaks (black boxes), but this can probably be still improved by modifying the star filter to use aspect ration instead of streak length. Also point-like streaks (three synthetic streaks) successfully pass the filtering steps, when the curvature filter is loosened to accept curvatures <60°.

Performance using metrics. To understand the performance of the prototype, we have adopted several metrics:

- Time consumption: average time per image and CPU core is measured as the wall-clock time required to process a pre-defined set of images.
- Sensitivity: the fraction of all streaks that are correctly identified as such (optimum 100%).
- Precision: the fraction of all found streaks that are real streaks (optimum 100%).
- Completeness of the detected streaks: assessed using the relative streak length, defined as the ratio of measured streak length and generated streak length, L/L_{true}.
- Accuracy of photometry and astrometry: RMS of the residuals between the true (generated) values and the measured values for streak center point (astrometry), or streak flux (photometry).
- Accuracy of the orbit solution based on single-image astrometry.

The following image and streak characteristics affect the performance of the methods and need to be taken into account: star density, streak length (L), streak variability (uniform vs. disintegrated streaks), streak curvature (κ; space-based sensors may produce non-linear streaks), and streak SNR (see Eq. (1)). *Sensitivity* α and *precision* β are the metrics that describe the ability of the methods to correctly detect and classify streaks, we estimate them (for precision, change $\alpha \rightarrow \beta$) as a function of the streak SNR and fit an analytical function of the form

$$\alpha = \frac{\alpha_0}{1 + \exp\left(\frac{SNR - SNR_\alpha}{w_\alpha}\right)} \tag{2}$$

where α_0 is the (average) sensitivity for high-SNR streaks and SNR_α is defined as the SNR value where $\alpha = 0.5\alpha_0$, i.e., the sensitivity has dropped to half of the level for high-SNR streaks and w_α is the width of the drop. Examples of sensitivity and precision as a function of SNR for a ground-based sensor are shown in Fig. 9 for long streaks ($L > 100$ px), the best-fit parameters for the functions are, for sensitivity, $\alpha_0 = 91.7 \pm 0.7\%$, $SNR_\alpha = 0.476 \pm 0.007$,

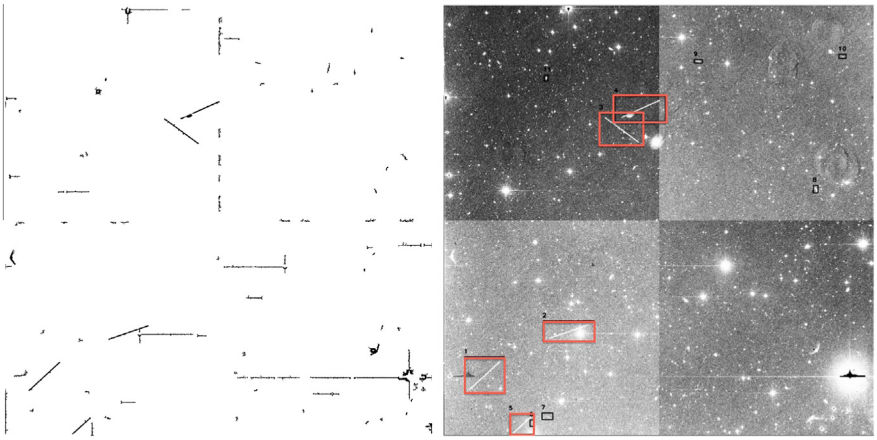

Fig. 7. An example for ground-based, sidereal tracking observing scenario: low star density (0.2), long ($L = 200$ px) and high ($SNR \sim 3$) streaks. Left: segmented image, right: original image with the detected GS streaks (black boxes) and segmented synthetic streaks (red boxes). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

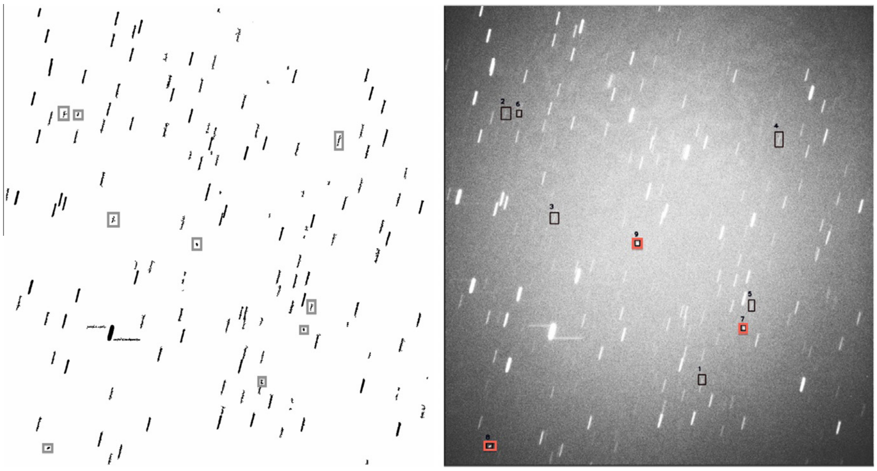

Fig. 8. An example for ground-based, object-tracking observing scenario: Very short, high-SNR streaks ($L = 10$ px). Left: segmented image, right: original image with the detected GS streaks (black boxes) and segmented synthetic streaks (red boxes). Locations of the detected GS streaks are indicated with gray boxes in the segmented image on the left. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

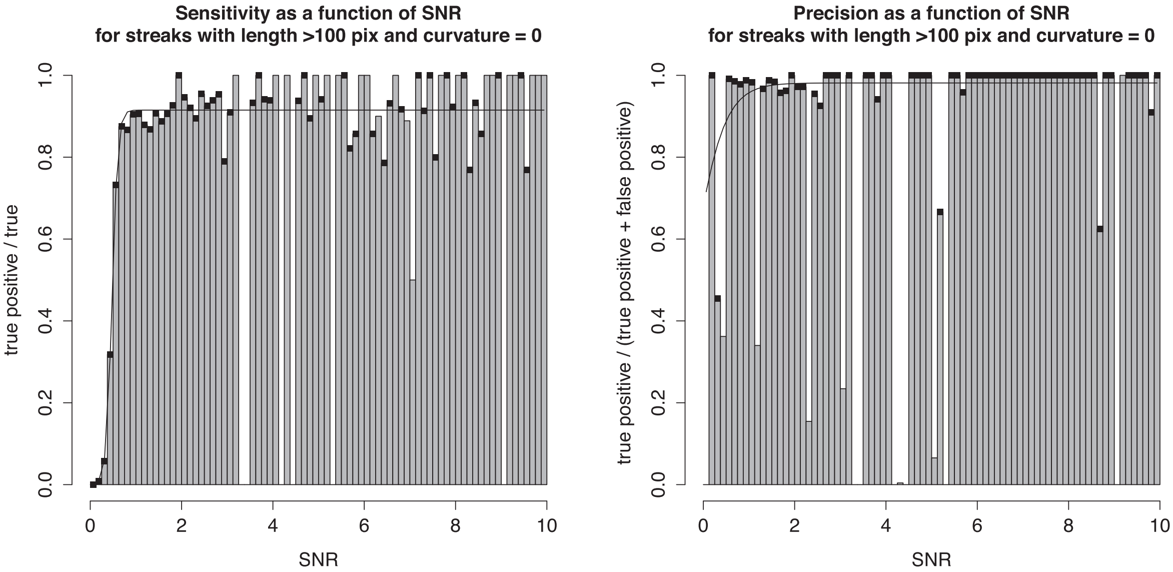

Fig. 9. (Left) Sensitivity and (right) precision for linear, continuous streaks with effective lengths greater than 100 pixels. The black squares refer to data points that contain a statistically meaningful amount of data and are included in the 3-parameter fits described by the thin black line.

and $w_\alpha = 0.053 \pm 0.006$, and for precision, $\beta_0 = 99.0\pm 0.7\%$, $SNR_\beta = 0.288 \pm 0.023$, and $w_\beta = 0.130 \pm 0.025$. We give a summary of sensitivity and precision results for various streak classes in Table 3.

The average time consumption is currently about 13 s for a typical 2k-by-2k image (for a Linux system with a 4-core Intel Xeon 2 GHz with 4 MB cache and 8 GB of memory).

Table 3
Summary of sensitivity and precision results for different streak classes. We give the average values for high-SNR streaks, see Eq. (2). Note that here curved streaks correspond to $\kappa = 16°$.

Streak class	Length L in px	Sensitivity α_0 (%)	SNR_α	Precision β_0 (%)	SNR_β
L	>100	92	0.48	99	0.29
L	<50	63	0.57	94	0.89
LL	>50	68	0.55	n.a.	n.a.
C	>50	89	0.48	97	0.48

We show that the prototype pipeline is capable of detecting at least 50% of the streaks with $0.5 < SNR < 10$. For high-SNR ($SNR > 1$) streaks, the detection sensitivity of the pipeline is over 90% for long (> 100 pixels) streaks, both linear and curved, dropping below 70% for short (< 50 pixels) as well as for discontinuous streaks. In the low-SNR regime, the sensitivity is still 50% at $SNR = 0.5$ for long streaks. The detection precision is in practice 100% for long, high-SNR streaks, and stays constant until the streak length goes below 50 pixels. For the shortest streaks, already around 50% of the detected streaks are false at $SNR = 1$. However, in the adopted single-image approach, tailored for long streaks, lower performance for short streaks is expected.

The streak curvature — potentially arising for a space-based sensor — degrades the sensitivity, as expected. For example, a curvature of $\kappa \sim 2.9^\circ$ leads to a sensitivity of $\alpha_0 = 94.5 \pm 1.0\%$ and $SNR_\alpha = 0.434 \pm 0.009$ whereas $\kappa \sim 16^\circ$ leads to $\alpha_0 = 89.0 \pm 2.6\%$ and $SNR_\alpha = 0.475 \pm 0.023$.

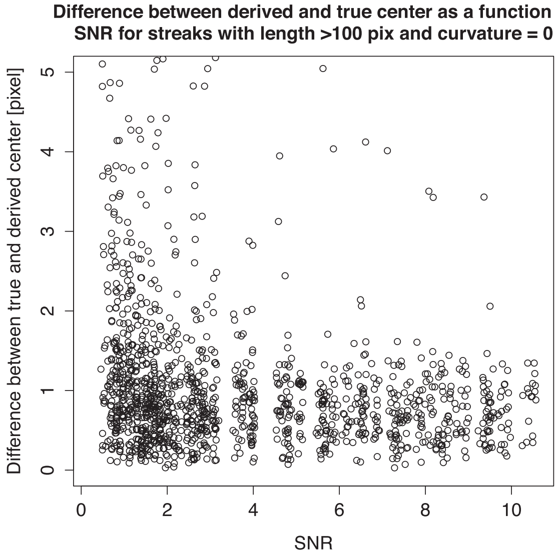

Fig. 10. The Euclidian distance between measured and true streak center coordinates as a function of SNR. Whereas the plot has been cut off at a distance of about 5 pixels, the tail of the distribution extends to about 115 pixels.

The accuracy of astrometry is tested by comparing the true pixel coordinates that were used when generating synthetic streaks to the pixel coordinates estimated by the prototype. We define astrometric accuracy as the RMS of the residuals between the true values and the measured values for streak center point, and relative streak length (cf. rate of motion). For long streaks, the mean coordinates for the measured center point relative to the true center point are -0.002 ± 0.261 pixels and -0.028 ± 0.258 pixels for the x and y coordinates, respectively. While the RMS values are relatively high (10.6 pixels and 10.5 pixels, respectively), we note that they are driven by relatively few but substantial outliers and, in a majority of cases, we can measure the streak center with sub-pixel accuracy (Fig. 10). The accuracy of the streak center is independent of SNR for $SNR > 4$.

The completeness of the detected streaks is assessed using the relative streak length, which we define as the ratio of measured streak length and generated streak length, L/L_{true}. We conclude that the relative length is typically about unity for long, high-SNR streaks. Indeed, a more detailed analysis reveals that the average relative streak length is 1.007 ± 0.004 for $SNR > 1$ and 0.837 ± 0.020 for $SNR < 1$. The fraction of streaks with $0.9 < L/L_{true} < 1.1$ is 71% for $SNR > 1$ and 12% for $SNR < 1$. Also the photometric accuracy is relatively high for long, linear, uniform streaks: the fraction of such streaks with $0.9 < F_{\rm obj}/F_{\rm obj,true} < 1.1$ is 92% for $SNR > 3$ and 33% for $SNR < 3$.

4.2. Real ground-based observations

We show an example of the processing of a real streak observed at the ESA Optical Ground station (OGS) at Tenerife (Fig. 11). The long, real streak (no 1) is correctly detected, but also four other candidate streaks remain after BW processing (right-most figure), while GS processing further reduces two of the streaks as artefacts based on their unrealistic PSF width. Thus, the final detection is two streaks (no's 1 and 6).

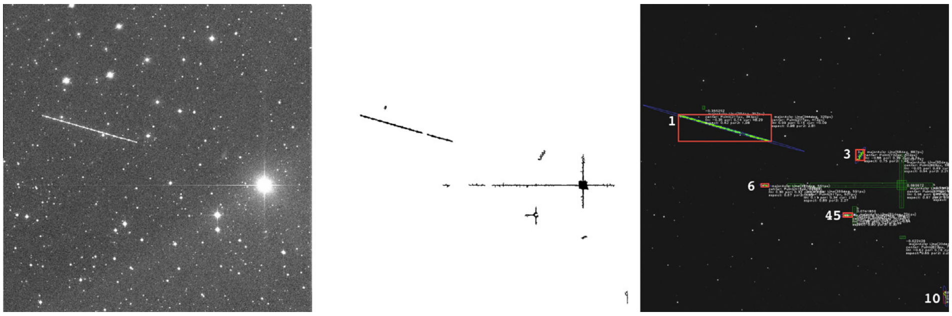

Fig. 11. An example for a real streak observation of a long, varying streak. Left: original image, middle: segmented image, right: original output image from the pipeline with the detected BW streaks (red boxes), including numerical information on streak parameters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusions

We have discussed the implementation of a prototype pipeline for streak detection and astrometric reduction. We have developed algorithms for single-image detections of streaks, where particular attention needs to be paid to the process of extraction of all necessary information from one image (segmentation), and subsequently, to efficient reduction of the extracted data (classification). The pipeline consists of three main modules: segmentation (Section 2.2), classification (Section 2.3), and astrometric and photometric reduction (Section 2.4). In addition, we have implemented two independent modules that can be used optionally: MCMC orbital analysis (Section 3.1) for orbital validation of the detected streaks, and Streak-PCA (Section 3.2) for classification analysis using training data.

We consider that the most important aspect of the prototype pipeline is the possibility for tuning the algorithms, such as the filtering scheme for different observing scenarios and optical setups. While the performance of the processing pipeline can and should be further improved with comprehensive testing using more varied and extensive real-life test image data, we provide the user with several configuration parameters and default thresholds for ground-based and space-based scenarios, based on the step-wise analysis and performance evaluation (Section 4) of the prototype using our database of semisynthetic images.

We have shown that the prototype pipeline is capable of detecting at least 50% of the streaks with $0.5 < SNR < 10$. For long streaks (length > 100 pixels), the primary targets of the pipeline, the detection sensitivity (true positives) is about 90% for both ground- and space-based scenarios for the bright streaks ($SNR > 1$), while in the low-SNR regime, the sensitivity is still 50% at $SNR = 0.5$. For short streaks, the performance of the pipeline is considerably weaker (sensitivity 63%). However, as the tailoring of the prototype was carried out for long streaks, there is room for improvement in the algorithms for short streaks.

Moreover, as we are following the single-image approach (as compared to multiple images of the same object/field), the verification of detections is nevertheless subject to additional efforts for the observer. As a first-aid tool, we offer the orbital validation module described in Section 3.1. The orbital analysis may be used to distinguish between Earth-bound and non-Earth-bound objects, but prior to entering routine operations the approach requires further studies which are outside the scope of this paper.

As described in Section 3.2, we have proposed the Streak-PCA tool as means to carry out more extensive analysis of the feature data resulting from a specific observing scenario. Streak-PCA with the *k*-NN classifier could be called from different parts of the prototype data-processing pipeline to yield information on the particular space-debris streak or streak candidate.

Although in this paper, we have only considered space-debris observations, the developed algorithms and the pipeline can naturally be applied to asteroid detection, and we foresee possibilities for detection of any kind of extended objects in astronomical images.

Acknowledgment

The work was carried out under the ESA contract No. 4000108373/13/D/SR. Observational data was made available for the StreakDet project by the following telescopes: ESA's Optical Ground Station (courtesy of ESA/D. Koschny), Zimmerwald Observatory (courtesy of the University of Bern) and University of Hawaii 2.2-m telescope (courtesy of University of Hawaii).

References

Aiger, D., Talbot, H., 2010. The phase only transform for unsupervised surface defect detection. In: IEEE Conference on Computer Vision and Pattern Recognition, 13–18 June 2010, San Francisco, USA. pp. 295–302.

Beutler, G., 2005. Methods of Celestial Mechanics I and II. Springer-Verlag, Berlin Heidelberg.

Flegel, S., Gelhaus, J., Mckel, M., Wiedemann, C., Kempf, D., Krag, H., 2011. Maintenance of the ESA MASTER Model. Final Report of ESA contract 21705/08/D/HK, M09/MAS-FR. June 2011.

Flohrer, T., Peltonen, J., Kramer, A., et al., 2005. Space-based optical observation of space debris. In: Danesy, D. (Ed.), Proceedings of the Fourth European Conference on Space Debris, ESA SP-587. ESA Publications Division, Noordwijk, pp. 165–170.

Gural, P., Larsen, J., Gleason, A., 2005. Matched filter processing for asteroid detection. Astron. J. 130, 1951–1960.

Hough, P., 1962. Method and means for recognizing complex patterns. U. S. Patent 3069654.

Jolliffe, I.T., 2002. Principal component analysis. In: Springer Series in Statistics, second ed. Springer, NY.

Kouprianov, V., 2008. Distinguishing features of CCD astrometry of faint GEO objects. Adv. Space Res. 41 (7), 1029–1038.

Krag, H., Beltrami-Karlezi, P., Bendisch, J., Klinkrad, H., Rex, D., Rosebrock, J., Schildknecht, T., 2000. PROOF—The extension of ESA's MASTER model to predict debris detections. Acta Astron. 47 (2–9), 687–697.

Laas-Bourez, M., Blanchet, G., Boer, M., Ducrotte, E., Klotz, A., 2009. A new algorithm for optical observations of space debris with the TAROT telescopes. Adv. Space Res. 44, 1270–1278.

Lang, D., Hogg, D.W., Mierle, K., Blanton, M., Roweis, S., 2010. Astrometry.net: blind astrometric calibration of arbitrary astronomical images. Astron. J. 37, 1782–2800.

Lévesque, M.P., Buteau, S., 2007. Image processing technique for automatic detection of satellite streaks. DRDC Canada, TR 2005-386.

Lévesque, M.P., Lelièvre, M., 2008. Image processing technique for automatic detection of satellite streaks. DRDC Canada, TR 2006-587.

Lindqvist, H., Muinonen, K., Nousiainen, T., Um, J., McFarquhar, G. M., Haapanala, P., Makkonen, R., Hakkarainen, H., 2012. Ice-cloud particle habit classification using principal components. J. Geophys. Res. 117, D16206 (12 pages).

Milani, A., Villani, A., Stiavelli, M., 1996. Discovery of very small asteroids by automated trail detection. In: Rickman, Valtonen (Eds.), Worlds in Interaction: Small Bodies and Planets of the Solar System. Springer Netherlands, pp. 257–262.

Milani, A., Tommei, G., Farnocchia, D., Rossi, A., Schildknecht, T., Jehn, R., 2011. Correlation and orbit determination of space objects based on sparse optical data. MNRAS 417, 2094–2103.

Milani, A., Farnocchia, D., Dimare, L., et al., 2012. Innovative observing strategy and orbit determination for low Earth orbit space debris. Planet. Space Sci. 62, 10–22.

Muinonen, K., Granvik, M., Oszkiewicz, D., Pieniluoma, T., Pentikäinen, H., 2012. Asteroid orbital inversion using a virtual-observation Markov-chain Monte Carlo method. Planet. Space Sci. 73 (1), 15–20.

Muinonen, K., Fedorets, G., Pentikäinen, H., Pieniluoma, T., Oszkiewicz, D., Granvik, M., Virtanen, J., Tanga, P., Mignard, F., Berthier, J., Dell'Oro, A., Carry, B., Thuillot, W., 2015. Asteroid orbits with Gaia using random-walk statistical ranging. Planet. Space Sci. (submitted for publication)

Oszkiewicz, D., Muinonen, K., Virtanen, J., Granvik, M., 2009. Asteroid orbital ranging using Markov-Chain Monte Carlo. Meteorit. Planet. Sci. 44 (12), 1897–1904.

Perret, B., Lefevre, S., Collet, Ch., 2009. A robust hit-or-miss transform for template matching applied to very noisy astronomical images. Pattern Recognit. 42, 2470–2480.

Schildknecht, T., Musci, R., Ploner, M., et al., 2004. Geostationary Orbit Objects Survey. Final Report of ESA Contract No. 11914/96/D/IM.

Schildknecht, T., Musci, R., Serra Ricart, M., et al., 2006. Geostationary Transfer Orbit Survey. Final Report of ESA Contract No. 12568/97/D/IM.

Stöveken, E., Schildknecht, T., 2005. Algorithms for the optical detection of space debris objects. In: Danesy, D. (Ed.), Proceedings of the Fourth European Conference on Space Debris ESA SP-587. ESA Publications Division, Noordwijk.

Tagawa, M., Hanada, T., Oda, H., Kurosaki, H., Yanagisawa, T., 2014. Detection algorithm of small and fast orbital objects using faint streaks; application to geosynchronous orbit objects. In: 40th COSPAR Scientific Assembly, August 2–10, 2014, Moscow, Russia. Abstract PEDAS.1-7-14.

Valtonen, E., Peltonen, J., Riihonen, E., Eronen, T., Flohrer, T., Schildknecht, T., Stoeveken, E., Wokke, F., Kramer, A., van Benthem, R., 2006. Space-based optical observation of space debris. Final Report of ESA Contract No. 17140.

Vereŝ, P., Jedicke, R., Denneau, L., Wainscoat, R., Holman, M.J., Lin, H., 2012. Asteroid astrometry and photometry with trail fitting. Publ. Astron. Soc. Pacific 124 (921), 1197–1207.

Virtanen, J., Muinonen, K., Bowell, E., 2001. Statistical ranging of asteroid orbits. Icarus 154 (2), 412–431.

Virtanen, J., Torppa, J., Näränen, J., Muinonen, K., Granvik, M., Poikonen, J., Lehti, J., Säntti, T., Komulainen, T., 2014. Streak detection and astrometric reduction. Final Report of ESA Contract No. 4000108373/13/D/SR.

Yanagisawa, T., Kurosaki, H., Banno H., et al., 2012. Comparison between four detection algorithms for GEO objects. In: Proceedings of the 2012 AMOS Conference, September 11–14, 2012, Hawaii, USA. p. 91. <http://www.amostech.com/TechnicalPapers/2012.cfm>.

Zacharias, N., Finch, C.T., Girard, T.M., Henden, A., Bartlett, J.L., Monet, D.G., Zacharias, M.I., 2013. The fourth U.S. naval observatory CCD astrograph catalog (UCAC4). Astron. J. 145, 44.

Zhang, Q., van Beek, P., Yuan, C., Xu, X., Seo, H-J, Li, B., 2013. Efficient defect detection with sign information of Walsh Hadamard transform. In: Proc. SPIE 8661, Image Processing: Machine Vision Applications VI, 86610A, March 6, 2013.