Planetary and Space Science 182 (2020) 104847

Contents lists available at ScienceDirect

Planetary and Space Science

journal homepage: www.elsevier.com/locate/pss

Advances in the meteor image processing chain using fast algorithms, deep learning, and empirical fitting

Peter S. Gural

Gural Software and Analysis LLC, 351 Samantha Drive, Sterling, VA, 20164, USA

A R T I C L E I N F O

Keywords:
Meteors
Techniques: image processing
Methods: numerical

A B S T R A C T

The processing pipeline for both video meteor detection and track analysis has evolved to embrace several new algorithms, which have improved the efficiency and performance of various steps in the meteor image processing chain. With the advent of larger pixel count digital sensors, the image processing techniques have needed to keep up with the computational load by not only employing higher end processors, but developing faster thresholding, clustering, and tracking algorithms for detection. In addition, machine learning methods employing both recurrent and convolutional deep neural networks have helped remove the human-in-loop false alarm mitigation step inherent in many meteor collection processing streams. The application of a matched filtering algorithm has helped to refine the measurement positional accuracy of propagating meteor tracks for post-detection analysis. The use of improved multi-site track aggregation has dramatically reduced the occurrence of mis-associating unrelated tracks during the combination into a single trajectory. When coupled with an improved minimization metric in the multi-parameter fitting method for trajectory estimation, this yields better meteor orbital solutions. Finally, proposed concepts in using a convolutional neural network as a meteor detector and performing trajectory fitting with an empirically based propagation model, show promise for more robust meteor image processing and analysis in the near future.

1. Introduction

The meteor imaging community has begun migrating towards multi-megapixel, progressive-scan digital sensors, and away from the traditional analog cameras with less than half-million pixels per frame. While this has been beneficial from the lowered noise, image quality, higher bit depth, spatial coverage, and angular resolution perspectives, the significantly larger image sizes have stressed processing loads. For example, full high-definition (HD) 1080p imagery possesses what amounts to approximately six times the pixel count of NTSC or PAL video. The computational issue has been partially offset by both Moore's law as well as employing larger capacity CPUs, but this can be limiting in multi-camera systems without resorting to high cost PC systems or custom GPU implementations. In addition, the advent of innovative meteor collection systems for tracking meteors for fragmentation studies and spectroscopy that require real-time responsiveness of these short-lived events, has resulted in a re-examination of the entire image processing chain to increase computational efficiency, while also maintaining detection and analysis robustness.

Finally, with the exploding growth of low-cost meteor camera deployments, meteor collection networks have tried to automate as many steps in the processing pipeline as possible, specifically those steps which were traditionally addressed by having a human-in-the-loop (HIL) perform the process. Thus, several algorithms have been undergoing new development and are being incorporated into various meteor imaging systems and their processing pipelines. This paper addresses a broad base of those enhancements and will highlight the significant improvements in algorithms, providing references for those algorithms described in greater detail in the published literature.

The paper is sub-divided as follows: section 2 sets the stage by covering the general image processing pipeline employed by most meteor collection and analysis systems. Section 3 examines the meteor detection phase of the process including options for various up-stream thresholding methods and a very fast clustering/tracking technique as an alternative to classical streak detection methods. Section 4 summarizes the results of employing neural networks to address automating the meteor classification task for false alarm mitigation. Section 5 considers the potential extension of using a convolutional neural network for detection, rather than as a post-detection screener. Section 6 describes an algorithm in meteor positional measurement refinement using matched filtering techniques. Section 7 covers multi-site aggregation of tracklets, and section 8 considers the improvements possible for the multi-parameter fit

E-mail address: pgural@gmail.com.

https://doi.org/10.1016/j.pss.2020.104847
Received 25 October 2019; Received in revised form 17 December 2019; Accepted 15 January 2020
Available online 20 January 2020

version of an atmospheric trajectory estimator.

2. Image processing pipeline

The typical meteor image processing pipeline can be exemplified by the Cameras for All-sky Meteor Surveillance's (CAMS) sequence of operations ranging from image capture to orbital parameter estimation (Jenniskens et al., 2011) as generically depicted in Fig. 1.

The specific processing steps and their general descriptions are as follows:

- **Capture** – Ingest of video frame imagery from a camera into computer memory or local storage, via digital interface or analog digitization device (frame grabber).
- **Compression/Storage** – Optional reduction of imagery bandwidth for compact transmission and/or hard drive archival storage. Examples of compressed imagery storage are the H.264 video compression standard or the CAMS compressed FF files format detailed below after the bullet points.
- **Detection** – Identification and positional measurement of meteor traces employing pre-processing clean-up, background estimation and tracking, thresholding for pixel exceedances, multi-frame propagating streak detection algorithms, and finally centroiding or leading-edge estimation of the detected propagating line segment per frame. Detection algorithm examples include clustering/tracking (Gural, 2016), small kernel convolution (Molau and Gural, 2005), and the Hough transform (Gural, 1999a),
- **Calibration** – Obtain an astrometric solution to map positional measurements from focal plane row and column to celestial coordinates (e.g. polynomial warping, radial dependency, all-sky barrel distortion). This step also includes photometric calibration that maps the spatially integrated intensities minus background to apparent magnitude.
- **Confirmation/Classification** – Optional HIL based manual review or an automated machine learning scan of potential candidate meteors, to cull false alarms from real meteors.
- **Aggregation** – Combination of multi-camera, multi-site meteor tracklet measurements through space-time coincidence and meteor specific geometric constraints, to associate several measured tracklets to a commonly observed meteor between cameras.
- **Trajectory** – Estimation of the 3-dimensional atmospheric track to include both the radiant direction and the motion dynamics including an entry velocity with deceleration profile. Example algorithms in use are intersecting planes, least means squares, multi-parameter fit, and Monte Carlo (Ceplecha, 1987; Borovicka, 1990; Gural, 2012; Vida et al., 2019a respectively).
- **Orbit** – Estimation of the Keplerian orbital elements via analytic and numeric techniques through transformations of the estimated trajectory parameters into geocentric, then heliocentric coordinates (Ceplecha, 1987; Jenniskens et al., 2011; Clark and Wiegert, 2011; Jansen-Sturgeon et al., 2018).

Most of these steps are automated in current imaging system deployments with the confirmation and aggregation steps having been the most difficult to achieve the removal of the HIL from the processing pipeline (see sections 4 and 7 respectively for detailed solutions).

An example of a simple incremental efficiency improvement in a pipeline was seen in the speed and quality enhancement of the 64:1 image compaction implementation for the CAMS compression algorithm (Jenniskens et al., 2011). A compression technique that retains on a per pixel basis, the temporal maximum across 256 frames, frame number of the max, the temporal mean and standard deviation that both exclude the maximum value. Changes were 1) a simple restructuring of array storage allocation, that placed various tracked pixel characteristics in closely adjacent memory locations, resulted in more efficient memory accesses on modern day L1/L2 caching CPUs, and 2) by tracking the highest four values in time per pixel, the revised compression algorithm eliminated fireball ghosting in the multi-frame mean estimation (Gural, 2016). More significant algorithmic improvements applicable to the general meteor image processing flow follow in the next sections.

3. Fast meteor detection

Improving the speed of detection is critical for many modern meteor capture systems given the increased pixel count, higher frame rates, and for some systems a need for real-time responsiveness. Modern HD and mega-pixel sensor arrays are representative of the high pixel throughput requirement for the first two items, whereas camera tracking applications that try to follow a meteor or fireball before fading out or entering dark flight, falls into the class of instrumentation that requires rapid-response with no latency. There are several functional components to the detection process that have recently shown significant gains in performance. These include speeding up the thresholding step for finding pixel exceedances above a pseudo-stationary background estimation, and the use of fast clustering rather than line finding algorithms for detecting meteor streaks.

With respect to the processing step of fast thresholding of each pixel in a video frame sequence, various options have been explored in a recent paper (Gural, 2019a). A comparison was made of several new and existing algorithmic methods comprising the fastest and/or most robust approaches. Ten different algorithms were explored whose general attributes considered were:

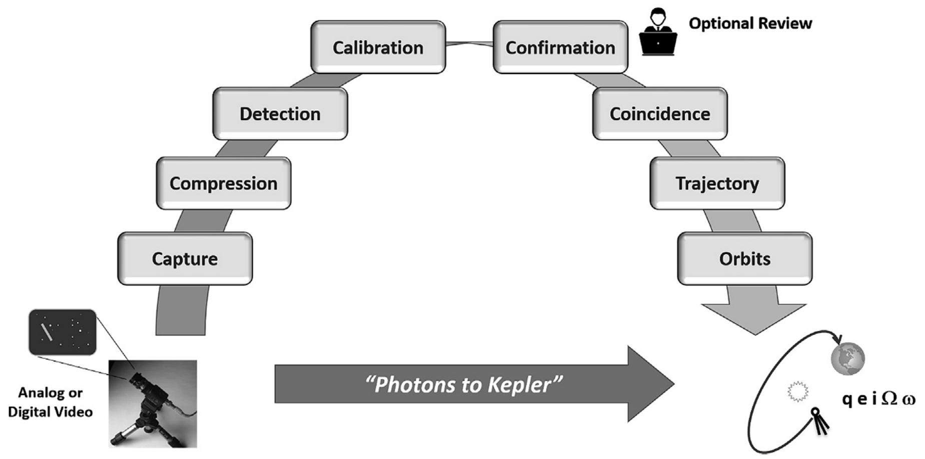

Fig. 1. An end-to-end video meteor processing pipeline from video frame capture to Keplerian orbital parameter estimation.

- Use of global versus local per-pixel estimates for the background statistics
- Estimation and tracking filter options for the mean and noise variance of the pseudo-stationary background
- Thresholding formulation and pixel desensitization aspects of the various methods
- Latency of the algorithms as to when a detection in a frame is declared after the capture
- Timing performance results on a high definition (HD) sized imagery example.

While global thresholding methods are well known for their highly efficient processing due to limited array allocation requirements and thus smaller number of memory fetches, a new localized pixel algorithm was formulated that produced the same timing results as the fastest global methods. The algorithm makes two very simplifying assumptions to avoid storing and tracking the background mean and variance, thus minimizing the time to read and write to memory which can be a significant contributor to runtime costs. The algorithm assumes that a past frame "P" (either adjacent in time or further back in the past) is a good approximation for both the mean and variance for the current frame's background statistics. The threshold for finding pixel exceedances is then defined as $T = P + k * P^{1/2}$ where k is a user defined standard deviation multiplying factor and the square root is rapidly performed via a lookup table when P in comprised of integers.

The new algorithm does handle the variability in the mean and variance across the focal plane on a per pixel temporal basis, albeit not as robustly as a tracked mean and variance, but avoids the global method's assumption of uniform background response. For the fastest processing in a high pixel count scenario and/or real-time responsive instrument, this is the most efficient thresholding algorithm found to date. On a i7-4770 single processing core and implemented in C, the algorithm processed a HD sized 1920 x 1080 frame in 1.6 ms. Table 1 reproduces a summary table of the various algorithmic options examined for fast thresholding.

Once thresholded and an exceedance list of pixel positions is obtained, the next step in the detection phase is to look for linearly propagating line segments across sequential frames. There are many line detection algorithms in the image processing literature and some have been applied to meteor detection. Four that have been successfully implemented involve MetRec's small 5x5 kernel convolution method, MeteorScan's pixel-pairing Hough transform algorithm, UFOcapture's basic thresholding (all three summarized in Molau and Gural, 2005), and the MAIA project's use of Canny edge detectors (Koten et al., 2014). More recently however, a very fast clustering or blob detection technique was demonstrated to perform as well as the Hough transform but with a dramatic 40x improvement in runtime costs (Gural, 2016).

The fast clustering algorithm employs a hierarchical mapping from pixel rows and columns that have been obtained from a threshold pixel exceedance list, and aggregates those exceedances into NxN sized counting cells as shown in Fig. 2.

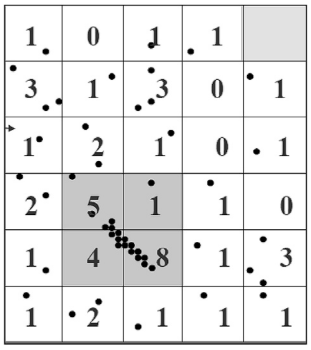

Fig. 2. Exceedance pixel clustering (block dots) accumulation into NxN cells (small squares with numbered counts) that employs fast integer indexing arithmetic. Overlapped 2x2 blocks of cells are combined in macro-cells for detection statistics and initial centroid estimation.

N is given by the maximum meteor motion in pixels per frame derived from the maximum apparent angular velocity of a meteor (Gural, 1999b) such that $N = V \sin D/q/f_s/\rho$ for the highest entry speed V = 72 km/s, lowest expected distance q = 70 km from the sensor, D = 90° from a radiant, the video frame rate f_s in frames per second, and the sensor's angular resolution per pixel ρ given in radians. For N > 32, a cascade of decimation levels is often called for to avoid missing long meteor tracks spanning across a frame, thus a hierarchical approach may be called for. The NxN sized cells contain accumulated exceedance counts that are further 2x2 binned into macro-cells and then threshold detected. Each "detected" macro-cell has its centroid computed on all the demeaned gray level pixels within the macro-cell and fed to a multi-frame, multi-track, α−β Kalman filter (Blair, 1992). The filter spawns, updates, and trims multiple tracks given the cluster centroids, and provides the final detection discriminator by tracking through and reporting on linear motion over a sufficient number of frames. Due to its high efficiency of processing and detection, the clustering and tracking modules are now incorporated into all CAMS daily processing operations, having replaced the slower pixel-pairing Hough transform method.

It was noted above that a decimation step may be required if the along-track pixel spread of a meteor per frame, exceeds the binning size of the clustering cells. The reason is the clustering approach is more of a blob detector than a line segment detector, and becomes less efficient when detecting long thin streaks. To alleviate this short-coming, particularly on narrow field-of-view camera systems, the imagery is broken into spatially resampled processing sub-streams. For example, a full resolution image sequence can be parallel processed along with 2x2 and 4x4 decimated image streams using a customized resampling algorithm. The fast decimation algorithm developed for use upstream of the clustering module, consists of taking the mean of the M highest pixel values in an MxM block to avoid SNR loss from straight averaging of the M^2

Table 1
Comparison of attributes for various fast thresholding algorithms listed in timing performance order for an HD 1080p sized image. Options 4 through 10 track image statistics on a per pixel basis. FOR = first order response filter. For more detailed explanation of the table contents see the WGN article (Gural, 2019a). This table was reproduced from the article by permission of the International Meteor Organization.

ID	Method	Tracking CONOPS		Mean μ	Std Dev σ	Threshold Operation	Desensitized Threshold	Latency #Frames	2.1 σ msec
1	Global Threshold	Past Frame μ & σ		μ_{Global}	σ_{Global}	$T = \mu_G + k\,\sigma_G$	No	1	1.6
2	Global Difference	Current - Past Frame		~Zero	σ_{Global}	$T = k\,\sigma_G$	No	1	2.7
3	Global Histogram	Current - Past Frame		No	No	%Exceed, No IFs	No	1	3.1
4	Past for Mean & σ	No Tracking of μ & σ		P	$P^{½}$	$T = P + k\,P^{½}$	No	1	1.6
5	Mean, Root-μ for σ	Pixel μ	$FOR\ \alpha = 2^2$	μ	$\mu^{½}$	$T = \mu + k\,\mu^{½}$	Yes	1	1.9
6	2 Frame Difference	Pixel σ	$FOR\ \alpha = 2^5$	~Zero	σ	$T = k\,\sigma$	Minimal	2	2.6
7	MaxFilter	Max Pixel	$FOR\ \alpha = 2^Q$	No	No	FOR Maxpixel	Yes	1	3.0
8	Past Frame Difference	Pixel σ	$FOR\ \alpha = 2^5$	~Zero	σ	$T = k\,\sigma$	Minimal	1	4.1
9	Track Mean & σ	Pixel μ & σ	$FOR\ \alpha = 2^5$	μ	σ	$T = \mu + k\,\sigma$	No	1	5.3
10	CAMS Compression	Pixel Level	N Frames	μ	σ	Maxpixel, Maxframe	Yes	N	8.9

pixels (Gural, 2016). This permits the clustering algorithm to detect long streaks per frame as the decimation process foreshortens the tracks but maintains roughly the same signal-to-noise.

4. Classification via deep learning

Automation of the entire meteor processing pipeline has been a recent focus of development efforts in the meteor community. One particular aspect involves the meteor classification step (also referred to as confirmation) to determine if detected meteors are in fact meteors and not false alarms such as aircraft, satellites, clouds, birds, bats, and bugs. Conventional techniques have either relied on human review of each detection's video snippet or the classification step is skipped entirely. The latter can be done in the hope that false alarm mitigation can be achieved when multi-site tracks are space-time aggregated into trajectories. However, when large numbers of false alarms are present, they can easily induce mis-alignments and false associations generating a significant percentage of bad orbits. This had occurred when the HIL had been removed from the CAMS confirmation and/or aggregation process in an attempt to embrace greater autonomous operations.

Machine learning techniques, specifically deep learning (DL) recurrent neural networks (RNNs) and convolutional neural networks (CNNs), have been applied to automating the meteor classification problem. A very detailed and descriptive paper on RNNs and CNNs successfully developed for the CAMS meteor collection system, has been published (Gural, 2019b), so only a summary of the results contained in that paper will be outlined herein. RNNs work by learning the temporal correlations in time series measurements and were found to be well suited to reviewing historical CAMS measurements, where the imagery was not readily available and only the measurement centroids and integrated intensities could be accessed off a central processing server. CNNs on the other hand, learn to exploit the spatial or space-time correlations in image sequences, so were found more appropriate to use remotely at each meteor collection station's site, where the stored imagery sequences for CAMS actually resides. Since far more spatial information content is available in imagery versus time series measurements of track centroids, the CNN was found to outperform the RNN on the same data set.

Deep learning requires labeled data sets to train a set of weights and biases embedded into either a long-short term memory (LSTM) RNN or deeply layered CNN architecture. It was found that of the 200,000 human analyst labeled meteors and false alarms used for both neural network training and testing, that the human had labeled the data set with effectively a 99.0% recall. Recall is defined as the number of true meteors classified correctly as meteors. The "MeteorNet" CNN was trained and tested on maxpixel (temporally collapsed images of the maximum pixel value in time) using small image chips centered on the meteor trace. This was possible because the upstream detector cues the location of the meteor in the larger field of view and the important features can be spatially localized. Extracting 20,000 meteors and false alarms for post-training independent testing, the CNN yielded 99.94% recall on post-detection CAMS space-time imagery that had used the cluster/tracker detector upstream. In contrast, the RNN achieved 98.1% recall, which is still extremely good given the limited spatial content of the centroid measurement data. See Table 2 for both network's full set of metric measures where for meteor classification, the recall and leakage are the significant measures of interest.

Table 2
Performance metrics for the two trained networks: percentage of true meteors classified correctly (recall or sensitivity), false alarms classified as meteors (leakage), false alarms classified correctly (specificity), percent of both classes estimated correctly (accuracy), and the standard machine learning F1 score.

Metric	RNN on Time Series	CNN on Imagery
Recall	98.1%	99.94%
Leakage	2.1%	0.4%
Specificity	97.9%	99.6%
Accuracy	98.0%	99.7%
F1	0.980	0.998

In particular, the RNN was shown to sufficiently clean up the false alarms upstream of the aggregator/trajectory analysis modules, so that the majority of orbits generated, were of good quality as visualized in Figs. 3 and 4. Clearly the poorly estimated orbits with unrealistic eccentricities and geocentric velocities were eliminated almost completely by the automated neural net removal of false alarm tracks. Remaining poor orbits can be traced back to either very noisy measurements or there were only very short tracks for all the contributing cameras, making velocity estimation difficult.

An important point to make is that when the imagery is available, the CNN has been shown to outperform a human in classification performance, and the computer algorithm is not subject to fatigue or cognitive attention issues. For more details on the RNN and CNN architectures, data pre-conditioning requirements, and hyper-parameters used, see the detailed published article (Gural, 2019b). The CNN MeteorNet classifier will be deployed to CAMS camera sites for post-detection false alarm mitigation.

5. CNN as a meteor detector

Since the application of a deep learning CNN algorithm was so successful on the meteor classification problem, it has been conjectured that the trained CNN could also be used as an upstream meteor detector directly on the ingested video imagery, rather than as a post-detection screener. In its current configuration, the feed forward part of the CNN is computationally efficient enough for processing small image chips, but when scaled up to applications involving HD 1080p sized images to be processed at a frame rate of 30 Hz, the runtime costs on a CPU are high. For example, a 64x64 input to the "MeteorNet" CNN takes 13 ms for classification whereas one full HD 1920x1080 sized image takes 8 s on the same Intel i7 class CPU. Several avenues of investigation exist to mitigate the runtime issue. Use of a GPU is one, but one must consider the I/O loading of pushing video frame-rate data onto the GPU platform and availability of GPUs on amateur PC systems. Another approach involves techniques such as employing deep compression (Han et al., 2016) to remove the low activation kernels from the neural network architecture and switch to integer arithmetic within the convolutional layers.

If a significantly faster feed forward portion of the network can be developed, then one can use the trained MeteorNet CNN on much larger sized input imagery. This would be achieved by replacing the final fully-connected layer with a convolution kernel having stride equal to the final maxpooling product dimension. Multiple softmax layer outputs would thus be obtained representing a mosaic tiling of sub-region probabilities across the original image, where each sub-region can then be classified independently. Alternatively, using a simple and fast image segmentation algorithm, one could identify localized areas of interest and then apply the smaller 64x64 sized MeteorNet to a very limited subset of small but high-interest regions. Examples of fast segmentation algorithms are Faster R–CNN (Ren et al., 2017), YOLO (Redmon et al., 2016), attention focusing methods, or one may simply train a simpler two-layer version of a CNN that reports on regional probabilities without in-depth feature generation. The latter holds promise since in MeteorNet, the first layer has a low computational load compared to the later layers that have to convolve across a deep set of input features and generate a deep set of output features. Developing the CNN as a video steam meteor detector is an ongoing area of future investigation, as is using the same techniques for radar head echo signature discovery using complex arithmetic in the CNN.

6. Leading edge position refinement

Once meteors have been detected, measurements need to be made for the position of the track as a function of time (frame number) in focal plane coordinates. These row and column measurements are later

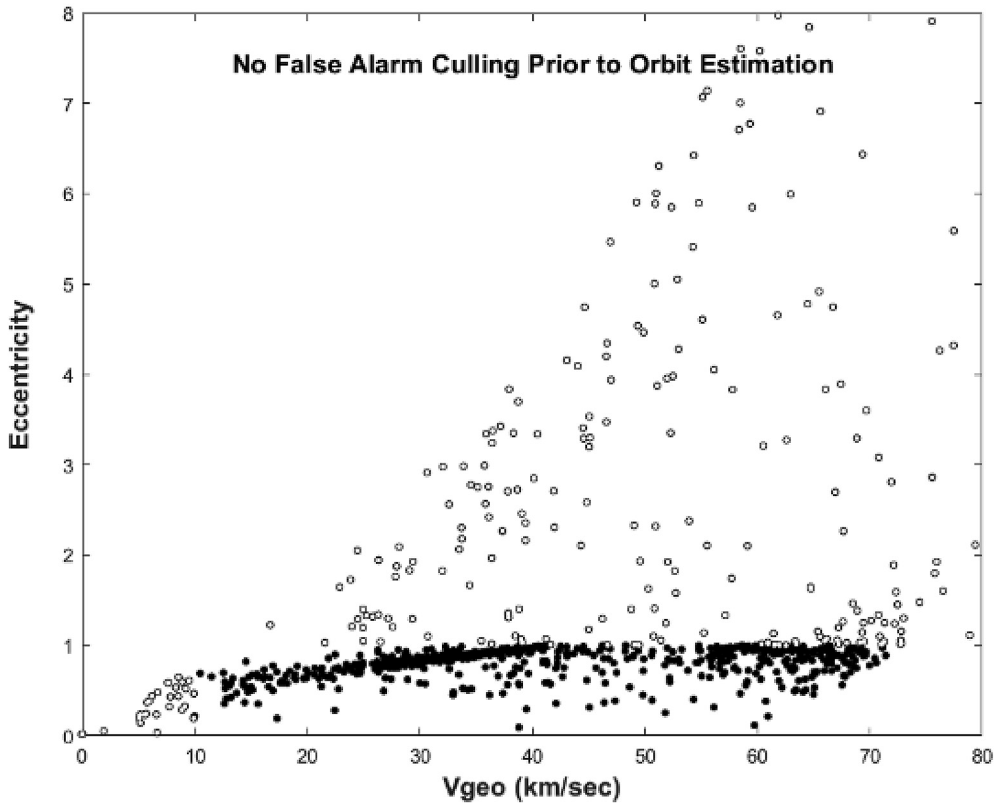

Fig. 3. Orbital parameters of eccentricity and geocentric velocity for estimated tracks where no false alarm removal was done prior to track aggregation and trajectory solution. Open circles are tracks with Vgeo outside the range 12–72 km/s and/or eccentricity >1.

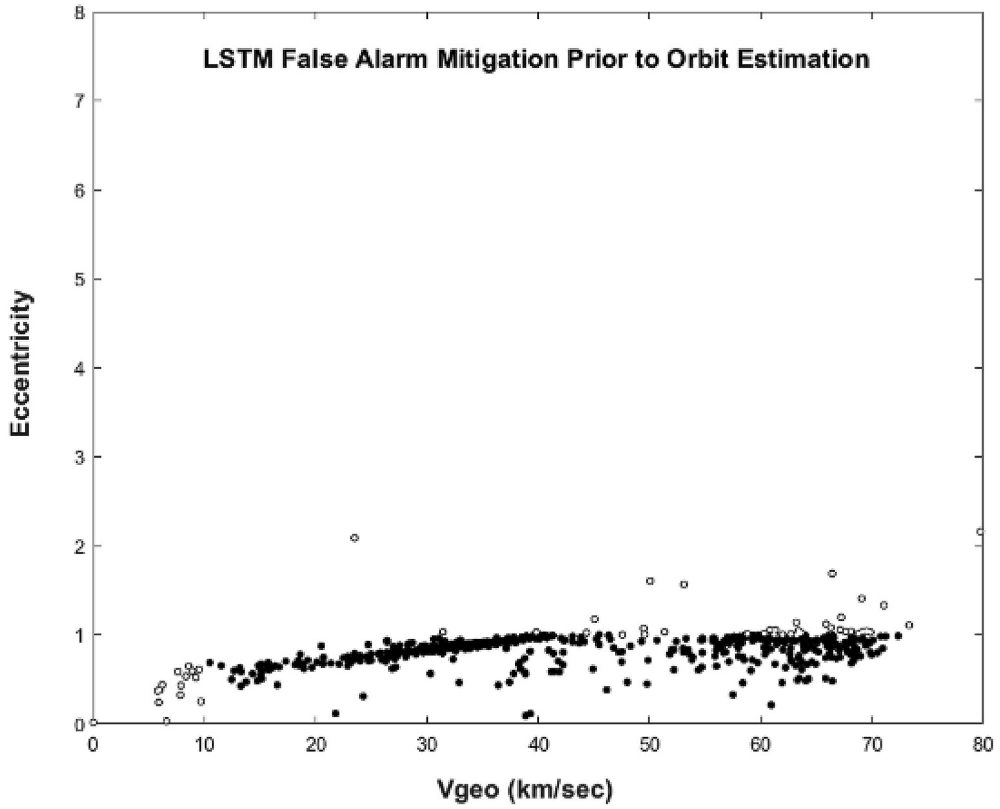

Fig. 4. Orbital parameters of eccentricity and geocentric velocity for estimated tracks using a LSTM based automated false alarm removal RNN. Open circles have the same meaning as in Fig. 3.

converted to inertial coordinates for trajectory estimation. The simpler measurement method is centroid estimation of the meteor line segment in the frame, but that is subject to bias from meteor intensity changes during the frame collection period, or wakes and long-lasting trains visibly trailing behind the meteor. A better technique is to estimate the leading-edge position of the meteor accounting for the point spread function of the imaging system. But this has usually been done via HIL-based manual pick-point estimation to obtain reliable results.

For the University of Western Ontario's electron multiplying charge coupled device (EMCCD) cameras, the detection system produces at least one meteor per minute making it infeasible to find leading edge pick points manually in the data volume generated. Instead a highly reliable automated fitting algorithm was developed that uses matched filtering (MF) or template matching (Gural, 2017) in the following scenario. The EMCCD processing pipeline employs an upstream detector, in this case the hierarchically decimated cluster/tracker modules described in section 3. That module cues a matched filter refinement step for improved leading-edge pick-point estimates. After detection of the meteor streak, an initial guess of starting position and velocity state vectors are perturbed using a particle swarm optimizer (Shi and Eberhart, 1998; Tsoulos and Stavrakoudis, 2010; Eslami et al., 2012). A maximum likelihood estimate (MLE) for the matched filter (Mohanty, 1981) is used as the minimization cost function as shown in Equation (1) where S = raw imagery, <S> = mean imagery, R = covariance matrix, T = hypothesized motion template.

$$\text{MLE} = \log_{10} \{ \; 0.5\ \Sigma\ [\ (S - <S>)\ R^{-1}\ T^{t}\ ]^{2} / \Sigma\ T\ R^{-1}\ T^{t}\ \} \tag{1}$$

The algorithm was found to work well, except in a few rare cases when a constant velocity motion hypothesis did not fully account for geometric foreshortening/lengthening effects, as well as deceleration changes in apparent angular velocity across the focal plane. Adding a simple acceleration term scaled by the projected velocity vector handles both effects to the level of human accuracy in the picking of the leading-edge position. The mathematical expression for the motion is given in Equations (2) and (3) for the column "X" and row "Y" coordinates of the focal plane respectively, where X_0, Y_0, V_x, V_y, and A are the five MF fitting coefficients, time "t" is relative to the time of the starting position (X_0, Y_0), and $V = \text{sqrt}(V_x{*}V_x + V_y{*}V_y)$.

$$X = X_0 + V_x\, t + (V_x / V)\, A\, t^2 / 2 \tag{2}$$

$$Y = Y_0 + V_y\, t + (V_y / V)\, A\, t^2 / 2 \tag{3}$$

The implemented template generation utilizes a two-dimensional Gaussian point spread function estimate that is convolved with the MF line segment motion hypothesis across the focal plane, and accounts for intensity variations of the meteor track from one frame to the next. It assumes a fixed intensity model intra-frame which captures the light curve signature of most meteors at the video rate of 17 frames per second used by the EMCCD systems, and only mis-matches cases of extreme meteor flaring during the frame. In those rare cases, the average intensity during the frame must suffice otherwise a model of intra-frame flaring would need to be introduced adding to the dimensionality of the matched filter fitting. An example of a geometrically-induced non-linear motion stretching away from a constant apparent angular velocity hypothesis is shown in Fig. 5, showing the before and after images for the temporal second-order term upgrade (a.k.a. acceleration term).

7. MULTI-SITE aggregation of tracks

To perform trajectory estimation between multiple cameras contributing measurements from multiple sites, one needs to combine camera tracks together that are associated with the same meteor. In the case of two stations with a camera track contributing from each site, the pairing association is quite straightforward using temporal coincidence and simple geometry constraints. However, when there is a mix of more than two tracks contributing from several sites, plus the potential for a meteor to span several cameras at one site, plus track contamination from clouds and aircraft false alarms, plus the lack of camera timing synchronization for sites not employing GPS or internet timing (which rely solely on unregulated/drifting PC clocks), these effects can make the space-time combinatorics and logic very challenging. For example, until recently CAMS employed a coordinator analyst to visually review each aggregated trajectory's triangulation solution, and manually try to cull outlier tracks to obtain the desired single meteor atmospheric trajectory with only the true meteor tracks contributing to the solution.

In August 2017, an attempt was made to automate the CAMS aggregation process with machine learning methods, but that continues to remain a work in progress. However, at the same time a different approach was worked on in parallel. The aggregation module's algorithmic steps were re-examined to improve the combinatorics and constraints (Gural, 2017). In the bulletized list below are outlined the processing steps now used in the CAMS aggregation algorithm, where the new capabilities and constraints are indicated as *italicized* bullet points. The intersecting planes (IP) method plays a key role in the first pass of constraints where the IP algorithm was detailed in a paper by Ceplecha, but had been described by many authors (Davidson, 1936; Porter, 1942; Whipple and Jacchia, 1957; Wray, 1967) prior to the seminal paper (Ceplecha, 1987). Note that the steps below include typical values used in the CAMS processing pipeline and those values should be carefully reconsidered before their use on other camera systems.

- Identify all temporally coincidence tracks that fall within a specified timing tolerance relative to a primary reference track, those then constitute a track set. The tolerance is given by the degree of temporal synchronization between cameras.
- Enforce that the site separation of the most disparately distant cameras in the set is sufficiently large (typically 5 km), thus avoiding data sets containing tracks from only one site.
- Cull any camera tracks that have been previously aggregated into a trajectory
- Cull tracks based on a user specified minimum count (typically 4 measurement positions) for a given site's camera track, as extremely short duration tracks contribute to poor radiant and velocity estimation (use number of field measurements for interleaved cameras)

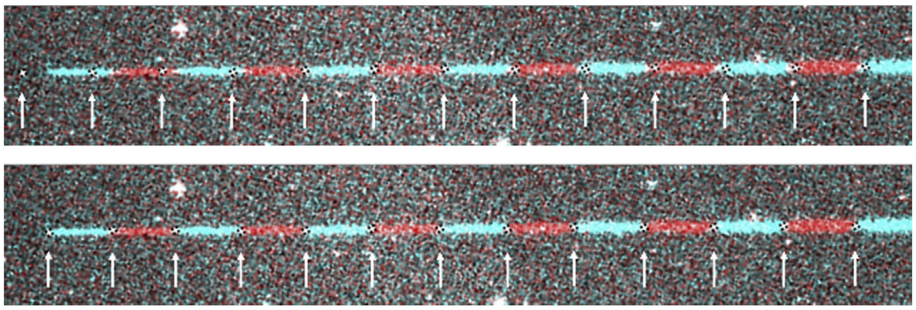
Fig. 5. Misalignment of the constant apparent angular velocity model's automatic pick-points (top) due to the geometric perspective shortening of a meteor across the field of view and fully accounted for with a second order temporal term (bottom). The red and cyan color traces represent sequential odd and even frame numbers of the multi-frame meteor trace of the temporally combined image. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

- *Use the intersecting planes algorithm (IP) to obtain pairwise trajectory radiant direction solutions between the primary track and each of the other tracks in the set independently. This is followed by a LMS velocity estimate assuming no deceleration. This early pass culls any track that falls outside a minimum or maximum user specified velocity or has an inconsistent along-track velocity between the independent measurements of the track pair. Sufficient tolerance outside the Solar System bound orbit velocities allows for hyperbolic meteors at the high velocity end (typical cutoff* <80 km/s*), and using a careful selection at the low velocity end excludes low Earth orbit artificial satellites (typical cutoff* >8 km/s*).*
- The IP convergence angle must be greater than a user specified minimum value of typically 2°.
- The IP radiant position relative to the camera pointing direction must fall within a user specified maximum angle of typically 180°
- The IP begin height must be greater than the end height
- The IP begin/end heights must within user specified upper/lower limits. Typical values are the begin height be greater than 70 km and the end height be below 120 km.
- *Look for radiant direction consistency between all IP pairings and cull outliers using a fairly coarse user specified angular tolerance of 10 degrees*
- Calculate the trajectory solution of the set's remaining aggregated tracks based on the multi-parameter fitting technique (Gural, 2012) with either a constant or exponential dynamic velocity model.
- *Automate the analyst review step to remove the HIL by performing three-dimensional co-alignment and co-linearity tests between all projected tracks given the single trajectory solution. The constraints are user specified with typical values of a maximum of 3 degrees in track crossing angles and 0.*5 km *offset between tracks respectively. This is similar to how a human reviewer would visually screen for good trajectories looking for highly overlapped projected measurement plots in latitude/longitude and height/downrange, color coded for each station's tracks.*
- Perform a final test to ensure the begin height is below the end height. Can be relaxed in the case of very rare grazer meteors.

Aggregation and trajectory estimation in the automated "coincidence" module of CAMS, now typically operates with less than 1% of the orbits resulting in poor quality which possess entry velocities outside the Solar system bound orbit speeds and eccentricities much greater than unity. This is currently considered an acceptable false alarm rate given the ability to run fully hands-off operations. Adding a machine-learning based meteor classification or confirmation step upstream, has helped improve quality in general, since far fewer contaminating tracks feed the automated coincidence aggregator. Adding a neural network process to the aggregation step as well is a work in progress.

8. Atmospheric trajectory estimation

There are now four generally used trajectory estimation algorithms in the meteor community. The first is the simple and commonly used IP technique previously referred to. The second was an improvement over IP that used a least mean squares (LMS) approach (Borovicka, 1990), which more seamlessly allows more than two tracks in the solution without ad hoc weighting. But these solvers only generated the radiant direction, and a second independent step of solving for the velocity along the track was required. By solving for the radiant direction and velocity state vectors simultaneously, an algorithm called multi-parameter fitting (MPF) was developed which showed improved performance at small convergence angles due to an implicit extra degree of freedom in the fitting (Gural, 2012). However, the MPF requires the use of a closed form mathematical model of the meteor's motion, which if it does not match the actual velocity dynamics profile, results in mis-estimation of entry velocities (Vida et al., 2019b). The Monte Carlo (MC) method (Vida et al., 2019a) tries to avoid that issue by aligning lags but reverts back to an independent radiant direction calculation followed by a separate velocity estimation approach, and also cannot deal with tracks that do not overlap spatially or temporally.

Borrowing from the work done for MC, the MPF has had its cost function reformulated so the particle swarm optimizer tries to minimize both the measurement ray angular separation to the straight-line trajectory solution along the radiant direction, as well as minimize the misalignment between measurement points from each camera along the solved 3D track to refine the velocity portion of the solution. The method shows promise as improving performance in some difficult cases for MPF and analysis is ongoing.

However, this particular MPF modification does not address the primary issue that MPF employs a velocity model that is hard wired to a specific mathematical function. This mathematical model does not always represent the actual meteor dynamics when one employs the functional forms of the model published to date (constant, linear, and exponentially changing velocity in time). That is, the deceleration of meteors does not lend itself to a closed form mathematical model. To address this, work is underway to use an empirically fit velocity profile obtained from the actual measurements, iteratively refined from a steadily improving trajectory solution. Thus, at periodic computational break-points in the MPF particle swarm update, a monotonic and smooth curve will be fit for the velocity profile, and then fixed while updating the position and velocity parameters in the trajectory solution. This would run iteratively until convergence to a final solution is reached. An exploration is underway to determine the best smoothing algorithm and monotonic fitting approach to the velocity profile, thus generating a dynamical solution not tied to a mathematical model. The algorithm would solve for all the trajectory parameters simultaneously and also seamlessly handle non-overlapping tracks.

Author statement

As single author I am the sole inventor, developer, analyst, and writer of this paper and its algorithmic and performance results contents.

Declaration of competing interest

As single author there is no conflict of interest with other authors nor with past or ongoing work.

Acknowledgements

The author would like to thank Dr. Peter Jenniskens of the SETI Institute, Dr. Peter Brown and Denis Vida of the University of Western Ontario, and Damir Segon of the Croatian Meteor Network for many fruitful and stimulating discussions over the years, as these techniques have been developed in coordination with various ongoing research programs at their respective organizations.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.pss.2020.104847.

References

Blair, W.D., 1992. NSWC Dahlgren Division 1992 Technical Report 297, p. 1.
Borovicka, J., 1990. Bull. Astron. Inst. Czechoslov. 41, 391.
Ceplecha, Z., 1987. Bull. Astron. Inst. Czechoslov. 38, 222.
Clark, D., Wiegert, P., 2011. MAPS 46 (8), 1217.
Davidson, M., 1936. J. Br. Astron. Assoc. 46, 292.
Eslami, M., Shareef, H., Khajehzadeh, M., Mohamed, A., 2012. Res. J. Appl. Sci. Eng. Technol. 4 (9), 1181.
Gural, P.S., 1999a. MeteorScan Documentation and User's Guide – (Sterling VA, USA), Version 2.2.
Gural, P.S., 1999b. WGN. J. IMO. 27, 111.
Gural, P.S., 2012. MAPS 47 (9), 1407.
Gural, P.S., 2016. Proceedings of the IMC 2016. Egmond, Netherlands, p. 96.
Gural, P.S., 2017. Proceedings of the IMC 2017. Petnica, Serbia, p. 56.
Gural, P.S., 2019a. WGN. J. IMO. (in press).
Gural, P.S., 2019b. MNRAS, vol. 489, p. 5109.

Han, S., Mao, H., Dally, W., 2016. In: Conference Proceedings of the ICLR, 2016.
Jansen-Sturgeon, T., Sansom, E., Bland, P., 2018. MAPS 54 (9), 2149.
Jenniskens, P., Gural, P.S., Dynneson, L., Grigsby, B.J., Newman, K.E., Borden, M., Koop, M., Holman, D., 2011. Icarus, vol.216, p. 40.
Koten, P., Pata, P., Fliegel, K., Vitek, S., 2014. Proceedings of the IMC. Poznan, Poland, p. 53.
Mohanty, N.C., 1981. IEEE Trans. PAMI. 3, 606.
Molau, S., Gural, P.S., 2005. WGN. J. IMO. 33 (1), 15.
Porter, J.G., 1942. Memoirs of the BAA, 34, 37.
Redmon J., Divvala S., Girschick R., Farhadi A., 2016, 10.1109/CVPR.2016.91, p. 779.
Ren, S., He, K., Girschick, R., Sun, J., 2017. IEEE Trans. on PAMI 39 (6), 1137.
Shi, Y., Eberhart, R., 1998. In: 1998 Proceedings of Congress on Evolutionary Computation, p. 79.
Tsoulos, I., Stavrakoudis, A., 2010. Appl. Math. Comput. 216, 2988.
Vida, D., Gural, P.S., Brown, P.G., Campbell-Brown, M.D., Wiegert, P., 2019a. MNRAS. https://doi.org/10.1093/mnras/stz3160.
Vida, D., Brown, P.G., Campbell-Brown, M.D., Wiegert, P., Gural, P.S., 2019b. MNRAS. https://doi.org/10.1093/mnras/stz3338.
Whipple, F.L., Jacchia, L.G., 1957. Smithson. Contrib. Astrophys. 1, 183.
Wray, J.D., 1967. The Computation of Orbits of Doubly Photographed Meteors. The Univ of New Mexico Press, Albuquerque, p. 5.