

Generic method for real-time satellite detection using optical acquisition system

Anca Ciurte, Adrian Soucup, Radu Danescu
Department of Computer Science
Technical University of Cluj-Napoca
Romania

Email: Anca.Ciurte@cs.utcluj.ro, Adrian_Scp@outlook.com, Radu.Danescu@cs.utcluj.ro

Abstract—Space debris has a tremendous increase in the last decade, arousing the attention of the experts in the field. The surveillance of the space is a first step in monitoring the traffic of floating objects and has several applications such as the correction of orbit coordinates for satellites or avoidance of collisions. An improved and flexible framework for real-time detection of satellites using a cheap optical surveillance system is proposed in this paper. The detection method is based on the Radon Transform for lines. The satellite candidates resulted after processing the Radon space are validated by imposing constraints over the satellites length and brightness, and over the stereo matching. We additionally propose a parallel approach for Radon transform on GPU in order to fulfill the real-time conditions. We test our method on a large and variate data set, containing satellites from different orbit ranges, namely medium and high orbits. A high accuracy over 95% was obtained in average for real time satellites detection with minimal false positives.

I. INTRODUCTION

A. Context and motivation

Space surveillance is a topic of great interest nowadays because of the exponential increase of space debris that will soon become a critical problem. Schildknecht presents a detailed survey of space debris in [1] and states that "the space debris population in certain orbital regions will become so dense that space operations will become severely hampered after a few decades, if space operations are continued to be conducted in the same manner as today". Intensive studies were performed lately in order to implement solutions to these problems. Some ideas are to de-orbit heavy space object and/or to perturb the small ones in order to force them re-enter atmosphere, but there are however great risks and high implementation costs.

In this context, the satellite detection problem is of great interest. Some of its major applications are to avoid satellite collisions with nonfunctional floating space objects, to update the satellites coordinates in order to correct the deviations from the orbit or to detect non catalogued space objects and update existing catalogs.

There are several types of surveillance systems that are used for space object detection, mainly radar and optical based. Radar systems are generally used in the case of low orbit, while the optical systems are preferred for higher orbits because they have a better detection rate to power requirement ratio. In our case, we use an optical system and address our detection problem for a large range of orbits, from medium to high orbits. According to [2], earth orbits are classified into three categories:

- Low Earth Orbits (LEOs) with orbits below 2000 km,
- Medium Earth Orbits (MEO) with altitudes around 20000 km,
- Geostationary Earth Orbits (GEOs) around 36000 km from the ground.

GEO orbits and the ones of higher altitude are also known as High Earth Orbits (HEO). The MEO orbit is used mainly by the navigation satellites (e.g. GPS, Glonass, Galileo), and communication, while GEO orbit by satellites for telecommunication, broadcasting and meteorological services [3].

B. Related work

To this end, there are few studies in literature on this topic of satellite detection in astronomic images. The most related with our work are as follows.

Stoeveken et. al presents in [4] several methods for space debris objects detection. They consider different observation strategies for image acquisition process, whitening the star tracking - the same strategy as ours. For this strategy, they eliminate the stars in the image based on their orbital coordinates extracted from a star catalog. Then, they eliminate the remained background artifacts, such as those induced during the acquisition process by using a median image and find the satellites candidates in the difference image between the current frame and median one. They also introduce a speed-up detection process by reducing the search space based on the predicted trajectories of the target objects from previous frames.

Wallace proposes in [5] another method for satellite streak detection. He presents an approach for background noise estimation, based on which he determine the threshold used to remove the background. Then, the satellites streaks are differentiated from other objects (such stars) using the image moments.

Levesque et al. presents in [6], [7] a method for satellite detection used to update their corresponding orbital coordinates. The image acquisition strategy is based on star tracking, ensuring a linear aspect of the satellite in the image. In a pre-process step they remove the background and the stars in the image, and then use oriented filters for the satellite streak detection. They benefit of an estimation of the satellite trajectory based on orbital coordinates of the satellites, restricting thus the searching space and consequently the satellite candidates.

An improvement of this method is further proposed by L evesque and Leli evre in [8]. This new approach consists in

extracting every possible object candidate and then eliminating the false positives through a set of rules on the object features. The rules concern a quantification of the object features such as the signal-to-noise ratio and the moments of inertia. Simulated images, generated with controlled parameters were used to establish the best thresholds for satellite streaks detection. A highly precise detection was confirmed on real images too, even in the case of very faint satellite streaks.

Oniga et al. presents a new detection approach in [9] for low orbit satellites. They subtract the background which is estimated based on previous frames, under the assumption that stars are fixed points in images. Satellite streaks are then detected in the obtained image based on several features such as area, major axis length, minor axis length, eccentricity, equivalent diameter, perimeter and solidity. This work was extended in [10] where a stereo system is used for image acquisition, that allows further processing such as 3D position estimation of the detected satellite.

In a previous work [11] we have introduced a new method satellite streak detection. The method is based on Radon transform for line detection and is able to detect satellites in single astronomic image, without having any prior information like satellite coordinates or any predictions from the previous frames. The method was successfully validated for some image sequences of MEO satellites.

C. Contribution of our work

In this paper we present new extensions of [11]. We propose a new approach for Radon Transform, designed for GPU implementation, that considerably improved the speed of the detection process. Additional image processing steps of contrast increase between the satellite streak and background is introduced in the detection method, leading to a higher detection rate, especially in the case of faint satellites. Finally, the satellite candidates are extra validated based on stereo matching principles that improves the results by eliminating the false positives. Moreover, an exhaustive validation study was performed on a large and variate dataset containing satellites from either medium and high orbits.

In summary, the contributions of this paper are:

- a flexible framework for satellite streak detection, adapted to any sky region,
- speed-up improvements of the satellite detection method [11] based on CUDA architecture,
- a new more accurate criteria for candidate satellite validation using stereo matching,
- and a high detection accuracy of over 95% in average for an extent dataset that contains MEO and HEO satellites.

1) *Paper structure*: The context and the motivation of this problem along with a detailed state of the art and contributions of this work are presented in Section I. The theoretical background and the proposed method are presented in Section II. The dataset and detection results are presented in Section III. The paper concludes with some discussion and future research directives in Section IV.

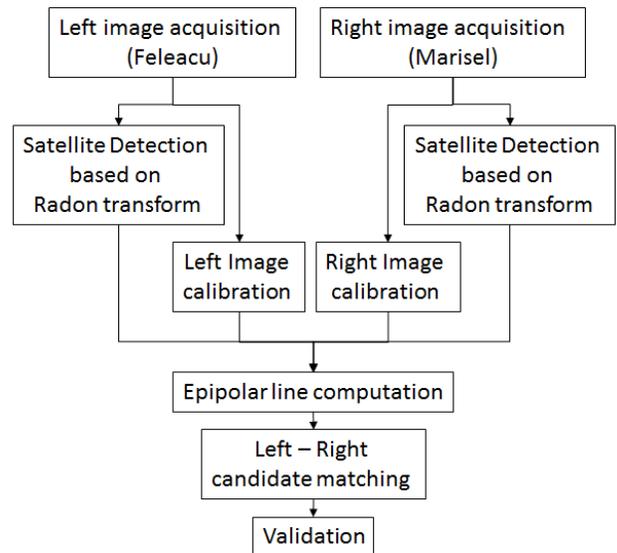


Fig. 1: The block diagram of the system

II. METHOD

An overview of the proposed method is illustrated in Fig. 1 as a block diagram.

First, pairs of *stereo images are simultaneously acquired* using an optical surveillance system. Then, the satellites are detected in each image from the left-right pair using a *detection method based on Radon Transform*. A new GPU approach of this method is introduced in II-B that considerably improve the computational performance. In order to extra-validate the detection results and reduce the false positives, a *stereo matching step* follows, where, for each detected satellite in left image, we search for its correspondent in right image along the *epipolar line*. The satellites are *finally validated* only if exists a *left-right candidate matching*.

Each of these steps are detailed in the following subsections.

A. Overview on the acquisition system

The following two optical systems were used for astronomic image acquisition. They differ by the type and performance of the telescope. We will further refer as *Optical System 1* and *Optical System 2*. Both of them are designed to acquire synchronized stereo pairs of images and each of them contain two optical sensors placed in different physical location (Marisel and Feleacu, Romania) with a baseline of 37 km.

The components of each sensors are as follows:

Optical System 1 (Fig. 2a):

- Newtonian 150mm, f/5 telescopes (D=150mm, F=750mm),
- DSLR Canon EOS 50D (2352 x 1568 pixels) camera,
- Equatorial tracking mount, type Celestron CG5,
- GPS based trigger for acquisition synchronization;

Optical System 2 (Fig. 2b):

- Meade RCX400 12 (D=300mm, F=2400mm),
- DSLR Canon EOS 50D (2352 x 1568 pixels) camera.

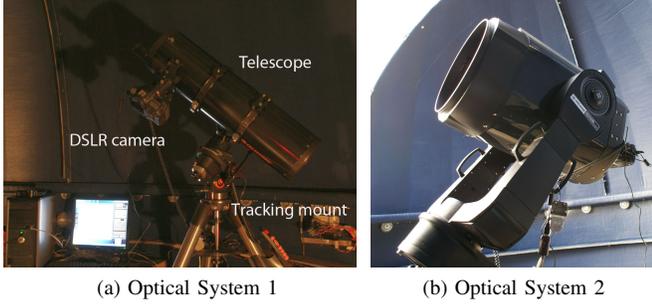


Fig. 2: The optical sensors.

Optical System 1 deals very well with MEO satellites (see Fig.3a), but for satellites of higher orbits, the level of noise in images will be considerably higher (see Fig. 3b). The system description and some results on these kind of images were presented in [12]. *Optical System 2* overcome these limitations and manage to acquire high quality images for high orbits too (see Fig. 3c)

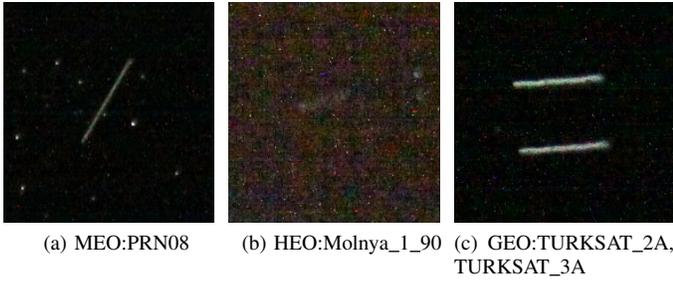


Fig. 3: Example of different satellites appearance, differing by their orbit type and acquisition optical system.

The images are acquired simultaneously, synchronized by a GPS-controlled external signal, with a certain time exposure (around 5 seconds), depending on the orbit type: MEO or HEO. Consequently, the satellite will appear in images as line segments of certain length that can be estimates as a function of time of exposure, distance from orbit, satellite velocity, angular field and image resolution.

B. Satellite detection based on Radon Transform

The satellite detection algorithm should find all the line segments from the input image. Each line segment has a slightly higher intensity than the rest of the background color.

The goal is to find where are located the satellite streaks and this is achieved by using Radon Transform. The Radon Transform will be calculated on subimages of varied size, depending on the length of the expected satellite streak in image. In our tests we used a varying size for image sequences $w \in \{151, 201, 251\}$. The final result is the reunion of the intermediary results found for all subimages. For each subimage, this approach takes all the possible lines from the

image space and computes the integrals over those lines:

$$R\{I\}(L) = \int_L I(x, y) ds, \quad (1)$$

where:

- I is the input image,
- $R\{I\}$ is the Radon Space,
- L is a line from the input image,
- ds is the differential element on the line L .

The output $R\{I\}(L)$ is called Radon Space and we use this to compute which line has higher intensity pixels. An improvement over the classical Radon Transform is using:

$$R\{I\}(L) = \int_L I(x, y)^\alpha ds. \quad (2)$$

where α is an exponent less than 1. It can be shown that if $\alpha < 1$ the radon space will have higher peaks for longer lines with lower intensity pixels instead of shorter lines with high intensity pixels, eg: stars.

After computing the Radon Space we search for local maxima to find the best candidate lines (see Fig. 4c). After finding k candidate lines, we test each line for the probability of being a satellite. The idea is to traverse the image in the direction of the line and compute the length of the color intensity peak. If the length is greater than a threshold u , which represents the satellite minimum length we have a hit. Because of the image noise the line can be interrupted and the solution we use is filtering the signal with a Gaussian filter (see Fig. 4d). If all the conditions are fulfilled, then the candidate is validates as satellite (see Fig. 4e).

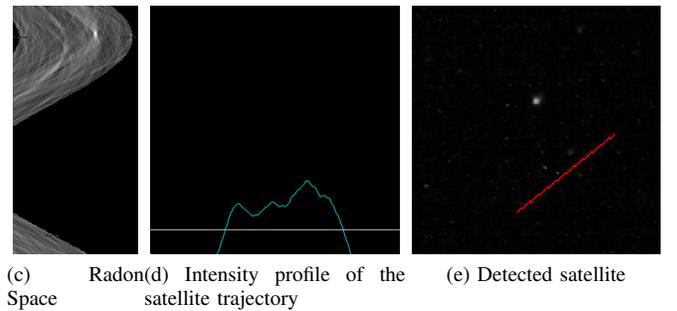
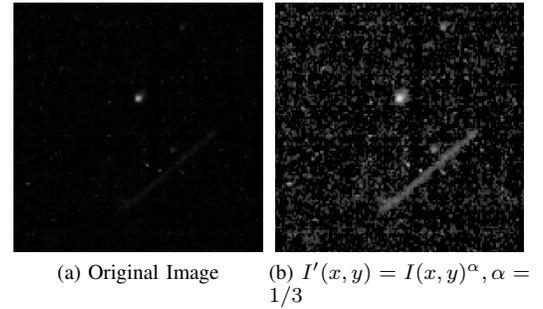


Fig. 4: Intermediary steps of the proposed detection method on single subimage.

Amdahl's law [13] states that the system's performance depends on the component that is used most of the time.

According to this law, we considered to optimize the most expensive step, which is the Radon Transform.

1) *Radon Transform. A GPU approach:* For a better understanding of the addressed problem, we first detail the classical approach of the Radon Transform for line detection. The main idea is to compute, for each pixel from the input image all the lines that pass through that pixel and accumulate its intensity into the corresponding lines from the Radon Space. The obtained sums are then normalized by the line length.

The pseudocode that calculates the *serial Radon Transform* is as follows:

```

function SERIAL_RADONTRANSFORM( $I$ )
   $h \leftarrow$  HEIGHT( $I$ )
   $w \leftarrow$  WIDTH( $I$ )
   $dh \leftarrow \lfloor \sqrt{w^2 + h^2} + \sqrt{2} \rfloor$ 
   $R\{I\} \leftarrow 0$ 
  for  $i \in [0, h)$  do
    for  $j \in [0, w)$  do
      for  $\theta \in [0, AngleDivisions)$  do
         $r \leftarrow \lfloor i \cos \theta + j \sin \theta + 0.5 \rfloor$ 
        if  $r \geq 0 \wedge r < dh$  then
           $R\{I\}(\theta, r) \leftarrow R\{I\}(\theta, r) + I(i, j)$ 
        end if
      end for
    end for
  end for

  return  $R\{I\}$ 
end function

```

where $R\{I\}$ is the resulted Radon Space of a given image I .

The image data stream is acquired by the system with 8 second difference between two consecutive images and exposure time of 5-6 seconds. Real-time in this context means any method that computes the satellite position within the time frame of two consecutive acquisitions, thus any algorithm faster than 8 seconds.

In order to make the algorithm run faster there are two parallelization approaches: on the CPU by executing *Serial_RadonTransform* for each subwindow in parallel and GPU by parallelizing of the Radon Transform algorithm itself. We use both methods for maximum performance.

The parallelization on the CPU is straightforward, we just run the algorithm for different subwindows on separate threads. The speed-up was proportional with the number of processors the machine had. This method had been improved further by pre computing for each $R\{I\}(\theta, r)$ the list of pixels that will accumulate there (the line itself) and sort them such that they preserve cache locality. By this method, the algorithm didn't need the branch instruction inside the inner loop thus improving the branch prediction. The speed-up of this approach was considerable, almost as fast the GPU algorithm which will be presented next.

The technological choice for speeding up the algorithm on the GPU is NVIDIA's CUDA parallel computing platform. According to [14], the best practices for maximizing execution speed with CUDA are:

- Parallelize the sequential code

- Find the right values for launch configuration
- Check the global memory access patterns and minimize the number of accesses
- Organize the global memory accesses to be coalesced
- Minimize the thread divergence within the same warp

The Radon Transform algorithm falls in the category of the histogram primitive which is not easily parallelizable.

The proposed method is implemented on a NVIDIA Fermi architecture [15]. The GPU is composed of multiple Streaming Multiprocessors (SM) and each SM has 32 CUDA cores. Each core from the same SM executes the same instruction for different data.

From a logical point of view the execution is divided into thread blocks and threads. Each thread will execute the same kernel function for different data. The difficulty when parallelizing the Radon Transform is having to accumulate in a memory location that might be accumulated by other threads. The proposed solution uses atomic operations in a way that collisions are probabilistically avoided. Another approach used accumulation in parallel for each $R\{I\}(\theta, r)$ but was slower because of the random access pattern in global memory.

The following pseudocode represents the *Radon Transform in parallel*, the naive and fast solution.

```

function PARALLEL_RADONTRANSFORM( $I$ )
   $h \leftarrow$  HEIGHT( $I$ )
   $w \leftarrow$  WIDTH( $I$ )
   $dh \leftarrow \lfloor \sqrt{w^2 + h^2} + \sqrt{2} \rfloor$ 
   $R\{I\} \leftarrow 0$ 
  for  $\{i, \theta\} \in [0, h) \times [0, AngleDivisions)$  do in parallel
    for  $j \in [0, w)$  do
       $r \leftarrow \lfloor i \cos \theta + j \sin \theta + 0.5 \rfloor$ 
      if  $r \geq 0 \wedge r < dh$  then
        ATOMICADD( $R\{I\}(\theta, r), I(i, j)$ )
      end if
    end for
  end parallel for
  return  $R\{I\}$ 
end function

```

The CUDA setup is as follows: h is the number of thread blocks and *AngleDivisions* is the number of threads. The rationale of this work distribution is giving different memory locations for threads from within a thread block to perform atomic operations. Collisions happen when r gets the same value for the same θ and different $\{i, j\}$ pairs.

This implementation has a relatively coalesced memory access pattern for global memory reads and tries to minimize collisions of atomic_add operations.

C. Candidate validation using stereo matching

The proposed detection method might produce a number of false positives because of the high noise and background stars. For each of the left and right input images we have a number of satellite candidates that will be further validated through a stereo matching step. The false positives will be thus eliminated.

The stereo matching is performed using the method presented in [10], by computing the epipolar line. This way, the searching space for the satellite correspondent in the synchronized image pair is reduced to a line $ax + by + c = 0$. The epipolar lines can be determined by intersecting the image plane with the plane formed by the optical centers of the two optical systems and the target 3D point, like in Fig. 5.

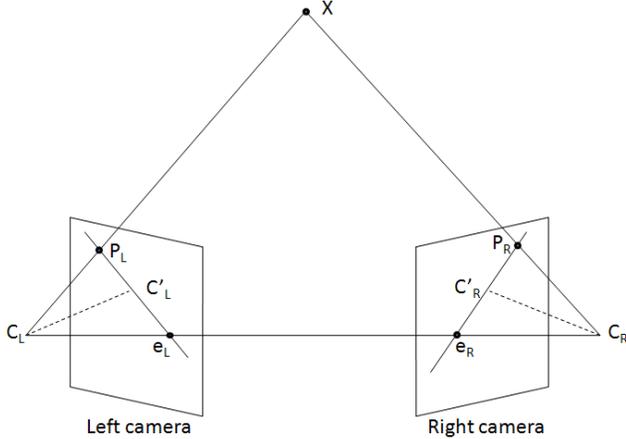


Fig. 5: The epipolar geometry [10]

The parameters of the right epipolar line are determined based on the coordinates of the left point P_L as:

$$[a \ b \ c]^T = FP_L,$$

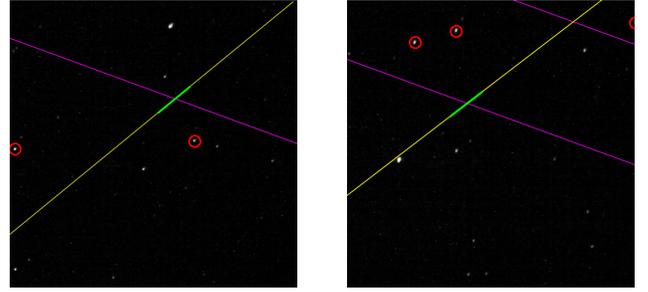
where F is the fundamental matrix computed based on the intrinsic and the extrinsic camera parameters. The extrinsic parameters, namely the translation vector and the rotation matrix are found through the calibration process, where, a set of 3D points in the scene of known coordination are mapped into the image space. For this specific type of astronomic images, we consider the stars as fixed points in the sky with known coordinates, and use them for the calibration process. The star's real coordinates are taken from an online catalog *The Sky* [16] and overlapped them over the image plane. The matching is then performed by minimizing the Sum of Absolute Differences.

For satellite candidate validation, we compute the mass center of the candidates in one image, calculate the epipolar lines of these points and try to intersect them with every candidate from the other image. A match is where the epipolar line and a candidate from the stereo pair image intersect (see Fig. 6), case when we proceed forward with the triangulation. The great advantage of using the stereo matching is that the false positives, such stars, will be considerably minimized. A final validation step comes after triangulation, when we simply invalidate the objects that have positions outside the field of view of the cameras.

III. RESULTS

A. Dataset

The proposed method was tested and validated on a dataset containing image sequences with medium and high orbit satellites. The image sequences were acquired at different moments



(a) The purple line is the epipolar line and it validates the satellite (green line) by intersecting with it. The red circles are some of the calibration stars.

(b) The other epipolar line is not intersecting the satellite and it corresponds to a false positive. The result is invalidated by this step.

Fig. 6: Stereo matching

of time with either one of the acquisition system described in Section II-A, summing up a total number of 1061 pairs of synchronized images. More details on the image sequences contained in the dataset can be seen in Table I.

B. Speed-up results

We begin by testing the performance of the proposed speed up process. All the presented approaches in Section II-B1 were tested on an Intel i7 CPU and GTX 480 GPU configuration. The running times along the approach summary are presented in Table II. Tests prove best results for Multithreaded GPU approach with atomics, with a total time of 3.9s.

C. Detection results

We test further the accuracy of the proposed detection method in comparison with [11]. Figure 7 shows some qualitatively detection results for the HEO image sequence, where in the first line are crops of the original images and in the second line crops of the results with proposed method. Detected satellites are marked in green.

Some quantitatively detection results such as true positives, false negatives and false positives for the whole dataset are presented in Table III. By

- true positives we refer to the satellites that were detected as satellites,
- true negative - satellites that were not detected,
- false positives - non-satellites that were detected as satellites.

Some examples of true positives detected after applying the detection on single image are shown in Fig. 8. Due to the benefits of stereo matching features, the proposed method manages to tremendously minimize such detection errors.

There were however few cases where our method did not detect the satellites (see Fig. 9). This happened only in the case of the HEO sequence, where, because of the orbit height and acquisition system less performing for such long distance, the image gain needed to be very high, and consequently, the acquired images were very noisy. Besides, the satellites in this sequence are rotating, meaning that they are represented with variable intensity values in different frames; in some frames, they are not even visible.

Orbit type	Acquisition system	#Pair of images	#Pair of satellites	Observed satellites
MEO	System 1	89	65	PRN08, PRN10, 733, 738
HEO	System 1	46	44	Cosmos_2447, Molniya_1_90
GEO	System 2	926	567	TURKSAT_XX, ASTRA_XX, HELLAS-SAT_X, EUTELSAT_XX, etc.
TOTAL		1061	676	

TABLE I: Dataset

Algorithm	Initial / Subsequent Calibration	Satellite detection in one image	Total time	Summary
Single Threaded Radon Transform	30s / 1s	25s	51s	The detection runs with the naive CPU version of the Radon Transform.
Single Threaded with CUDA Radon Transform	30s / 1s	6.5s	14s	The detection runs with the CUDA version of the Radon Transform. Everything else is done on the CPU
Multi-threaded (optimized) CPU	9s / 1s	2s	5s	Process each sub-window on a different thread. The Radon transform is done on the CPU. The optimization is done by saving for each pixel from the Radon Space the corresponding pixels from the input image. The array is sorted to improve the cache locality and there are no branch instructions when computing the accumulator.
Multi-threaded (optimized) GPU (no atomics)	9s / 1s	3.3s	7.6s	Process each sub-window on a different thread. The Radon transform is done on the GPU. The optimization is done by saving for each pixel from the Radon Space the corresponding pixels from the input image. The array is sorted again. On the GPU it turns out that the algorithm is slower due to random memory access and low bandwidth
Multi-threaded (optimized) GPU (with atomics)	9s / 1s	1.45s	3.9s	Process each sub-window on a different thread. The Radon transform is done on the GPU with atomicAdd. The optimization is made by not computing the frequency of each pixel in the Radon Kernel. We run that kernel exactly once at the beginning of the application and we only compute the accumulator with CUDA.

TABLE II: Speed-up results

IV. CONCLUSION

Several challenges of satellites detection in the night sky were discussed and addressed in this paper. The most important of them are the detection accuracy and real-time conditions. A new GPU approach was introduced in this paper in order to reach real-time performance of our system. The proposed technique proved a significant improvement in terms of computational time. The new improvements on the detection confer the method a higher robustness to the image noise and low contrast images. It also manage to reduce the false candidates

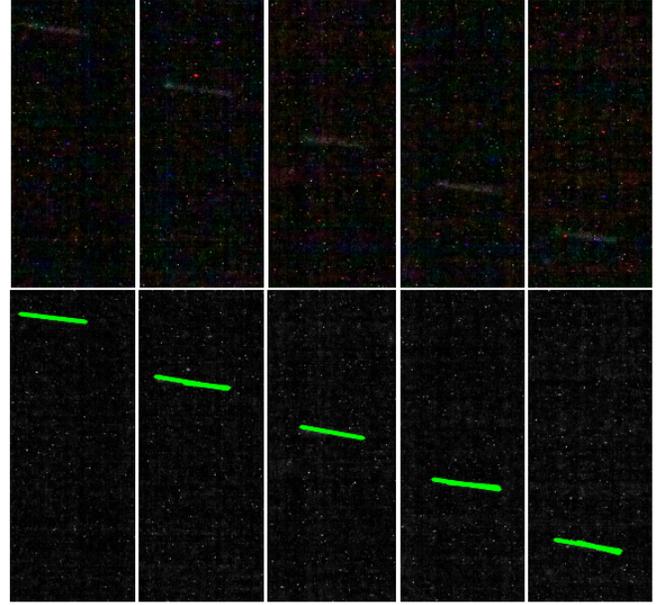


Fig. 7: Detection results on consecutive frames of a HEO image sequence

Sequence	MEO	HEO	GEO
<i>True positives:</i>			
Method [11]	65 (100%)	37 (84%)	567 (100%)
Proposed method	65 (100%)	38 (86%)	567 (100%)
<i>False negatives:</i>			
Method [11]	0	7	0
Proposed method	0	6	0
<i>False positives:</i>			
Method [11]	139	125	784
Proposed method	0	4	0

TABLE III: Quantitatively detection results

introduced by the background noise and stars. The validation of our method on a large and variate dataset prove the robustness of the proposed detection method and makes it invariant to different acquisition conditions.

The functionality of the system can be further improved by setting up the conditions for non-stop surveillance, such as a complete automatic setting of the system parameters, achievement of storage resources, implementation of a database with known satellites and the possibility of updating new ones.

ACKNOWLEDGMENT

This work was supported by the AMHEOS grant (*Automatic Medium and High Earth Orbit Observation System Based on Stereovision*) of the Romanian National Authority

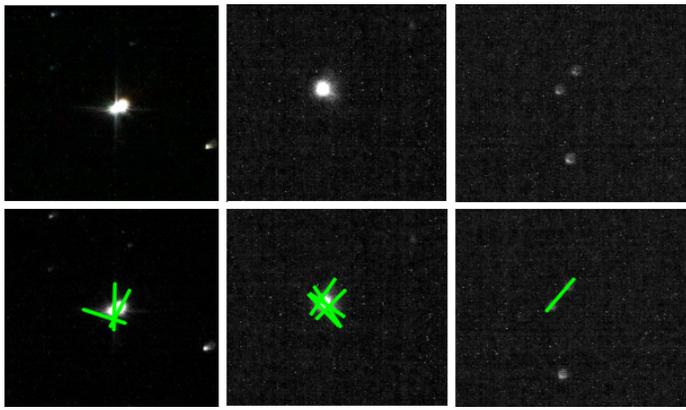


Fig. 8: Example of false positives after satellite detection in single image.



(a) Cosmos_2447 (b) Cosmos_2447 (c) Molniya_1_90

Fig. 9: Worst case: example of undetected satellites.

for Scientific Research, CNDIUEFISCDI, project code PN-II-PT-PCCA-2011-3.2-0651, contract number 210/2012.

REFERENCES

- [1] T. Schildknecht, "Optical surveys for space debris," *The Astronomy and Astrophysics Review*, vol. 14, pp. 41–111, 2007.
- [2] M. Capderou, *Satellites Orbits and Missions*. France: Springer-Verlag, 2005.
- [3] V. Agapov, I. Molotov, and Z. Khutorovsky, "Analysis of situation in geo protected region," in *Advanced Maui Optical and Space Surveillance Technologies, Maui, Hawaii, 2009*, pp. 180–189.
- [4] E. Stoveken and T. Schildknecht, "Algorithms for the optical detection of space debris objects," in *Proceedings of the 4th European Conference on Space Debris, 2005*, pp. 637–640.
- [5] B. Wallace, "The drdc ottawa space surveillance observatory," in *Advanced Maui Optical and Space Surveillance Technologies, Maui, Hawaii, 2007*, pp. 12–20.
- [6] M. Lévesque and S. Buteau, "Image processing technique for automatic detection of satellite streaks," *DRDC Valcartier, Canada, Tech. Rep. DRDC Valcartier TR 2005-386*, 2007.
- [7] M. Lévesque, "Automatic reacquisition of satellite positions by detecting their expected streaks in astronomical images," in *Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Hawaii, 2009*, p. E81.
- [8] M. Lévesque and M. Lelièvre, "Improving satellite-streak detection by the use of false alarm rejection algorithms," *DRDC Valcartier, Canada, Tech. Rep. DRDC Valcartier TR 2005-386*, 2008.
- [9] F. Oniga, M. Miron, R. Danescu, and S. Nedevschi, "Automatic recognition of low earth orbit objects from image sequences," in *IEEE International Conference on Intelligent Computer Communication and Processing, 2011*, pp. 335–338.
- [10] R. Danescu, F. Oniga, V. Turcu, and O. Cristea, "Long baseline stereovision for automatic detection and ranging of moving objects in the night sky," *Sensors*, vol. 12, no. 10, pp. 12 940–12 963, 2012.
- [11] A. Ciurte and R. Danescu, "Automatic detection of meo satellite streaks from single long exposure astronomic images," in *9th International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, 2014*, pp. 538–544.
- [12] R. Danescu, A. Ciurte, and V. Turcu, "A low cost automatic detection and ranging system for space surveillance in the medium earth orbit region and beyond," *Sensors*, vol. 14, no. 2, pp. 2703–2731, 2014.
- [13] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," *IBM Sunnyvale, California*, 1967.
- [14] NVIDIA. (2014, Feb.) *CUDA C BEST PRACTICES*. [Online]. Available: http://docs.nvidia.com/cuda/pdf/CUDA_C_Best_Practices_Guide.pdf
- [15] P. N. Glaskowsky, "Nvidia's fermi: The first complete gpu computing architecture," *Sep. 2009*.
- [16] Bisque. (2014, May) *Theskyx serious astronomer edition (3/21/2012)*. [Online]. Available: <http://www.bisque.com/sc/pages/TheSkyXProfessional-Edition.aspx>