METEORIX

A new processing chain for detection and tracking of meteors from space

M. Millet[1,2], N. Rambaux[3], A. Petreto[1,2], F. Lemaitre[1], L. Lacassagne[1]

[1]LIP6, Sorbonne Université, CNRS

[2]LERITY - Alcen

[3]IMCCE, Observatoire de Paris, PSL Université, Sorbonne Université, CNRS

International Meteor Conference

September 26th, 2021

The project :

- Universitary CubeSat mission of Sorbonne University[1]
- Several purposes :
  - **Educational** Enroll students in a space mission
  - **Astronomical** Estimate the flux of meteoroids and space debris entering the atmosphere
  - **Technological** Show the feasibility of a *real-time* processing chain embedded in a nanosatellite.
- Step A : proof of concept validated
- Step B : definition and prototyping of the payload (in progress)

The CubeSat :

- CubeSat 3U : 3 cubes of $10cm^3$.
- Payload :
  - Camera
  - Processing chain
- **Strong energy constraint** : -10W available

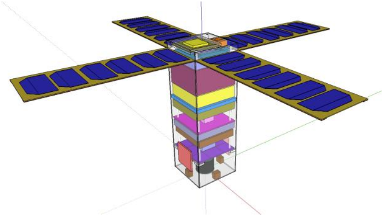


Figure 1 – View of Meteorix's CubeSat[2]

State of art

- MeteorScan (1995) [3], MetRec (1999) [4], UFOCapture (2004) [5]
  - Old but still used and popular
  - Probability of detection in real-time greater than 80%.
- Ground camera networks
  - FRIPON[6] (by IMCCE) with 250 cameras
  - Wide sky coverage
- RPi Meteor Station [7]
  - Croatian Meteor Network
  - Designed for low power computer (Raspberry Pi)
- Processing chains with neural networks
  - Probability of detection between 96%[8] and 99.9%[9].
  - Require hardware[10][8] that can't be embedded in a nanosatellite.
  - Too few images of meteors from space.

State of art

Image processing techniques used for meteor detection are not suited for space detection.

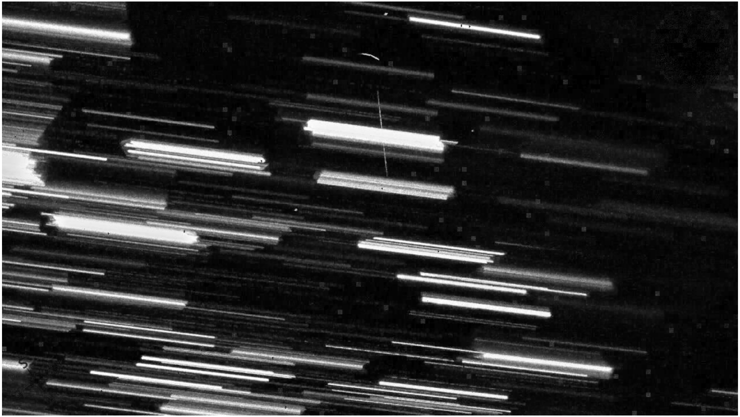


Figure 2 – Example from RMS : Which lines are meteors ? Which lines are cities ?

⇒ **No processing chain for space detection**

Processing chain

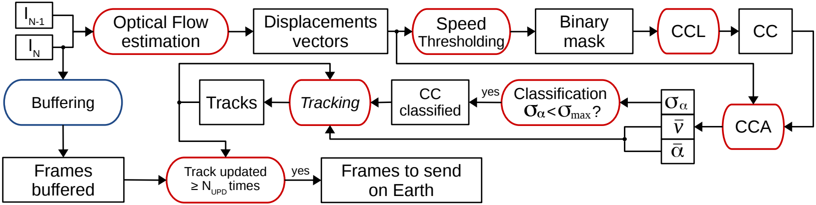


Figure 3 – CC = Connected Components, CCL = Connected Components Labelling, CCA = Connected Components Analysis, $\bar{v}$ = average speed, $\bar{\alpha}$ = mean angle, σ_α = angular standard deviation, $\sigma_{max} = 30\,\text{deg}$, $N_{MAJ} = 3$

Step 1 : Optical flow

- Optical flow estimation = apparent movement estimation
- Horn & Schunck [11]
  - Iterative algorithm
  - Pyramidal algorithm [12]
- Field of displacement vectors
- Speed in px/frame

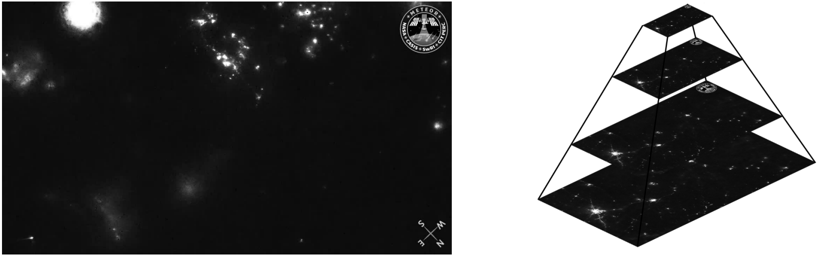


Figure 4 – Frame 119 from sequence v86

Step 1 : Optical flow

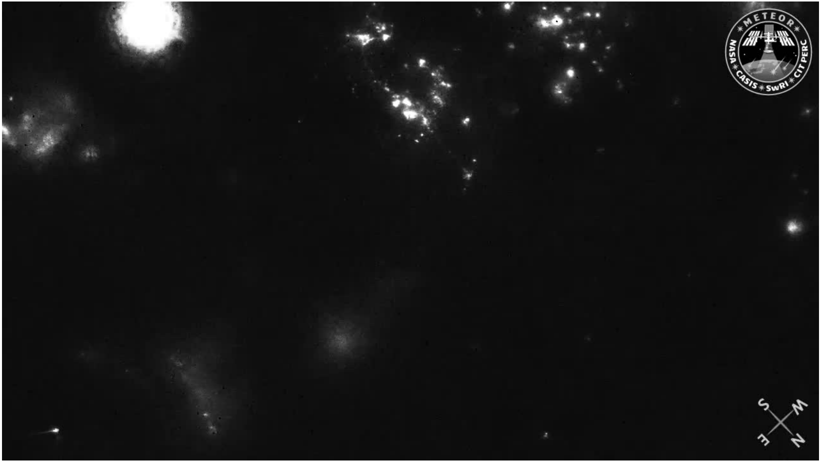


Figure 5 – Example : Frame 119 of sequence v86

Step 1 : Optical flow

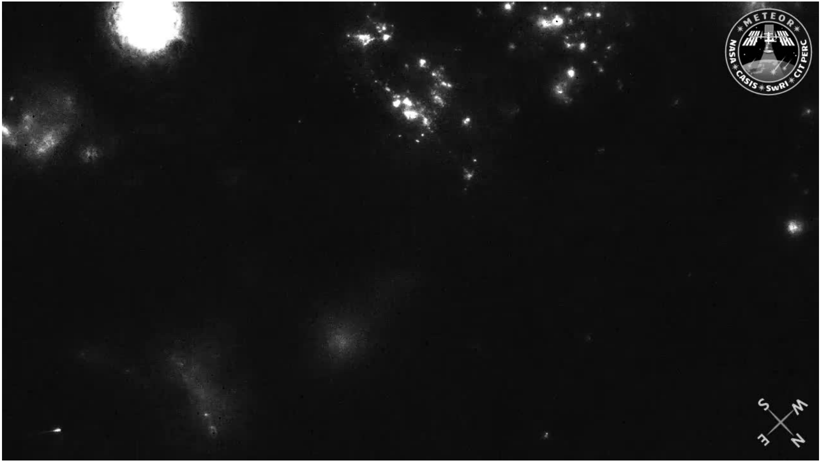


Figure 6 – Example : Frame 120 of sequence v86

Step 1 : Optical flow

Figure 7 – Example : optical flow between frame 119 and 120

Step 2 : Speed thresholding

- Meteors, space debris and lightnings are faster than Earth.
- Threshold set at 2.5px/frame.

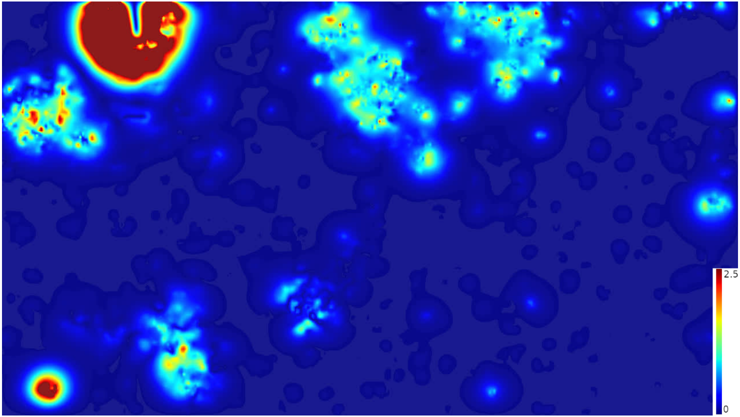


Figure 8 – Example : norm of displacements vectors

Step 2 : Speed thresholding

- Mathematical morphology : opening and closing.
  - To remove pixels without neighbors.
  - To regroup nearby cluster of pixels.

Figure 9 – Example : Binary mask of the fastest pixels.

Steps 3 and 4 : Connected Components Labelling and Analysis

- Pixels representation → objects representation.
- Compute statistical features on each object.
  - Average speed $\bar{v}$.
  - Average angle $\bar{\alpha}$ and its standard deviation σ_α.

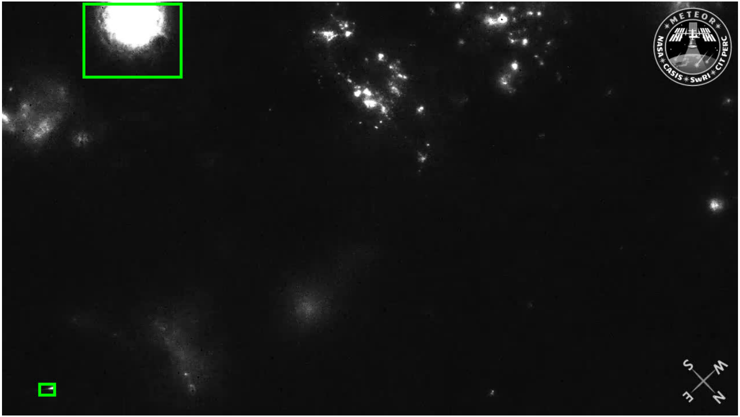


Figure 10 – Example : connected components with their bounding box.

Step 5 : Classification by σ_α

- How to differentiate the connected components ? (= fastest objects)
  - The trajectory of the meteor is rectilinear.
  - Low angular standard deviation ⇒ displacement vectors have a similar direction.
- A lightning spreads in all directions inducing a high angular standard deviation.

Figure 11 – Example : Angles

Step 6 : Tracking

- Simple algorithm based on a state machine
- The position of a meteor can be extrapolated in case of temporary loss.
- 2 purposes :
  - To confirm that it is a meteor (3 detections in less than a second).
  - To group frames into a sequence.

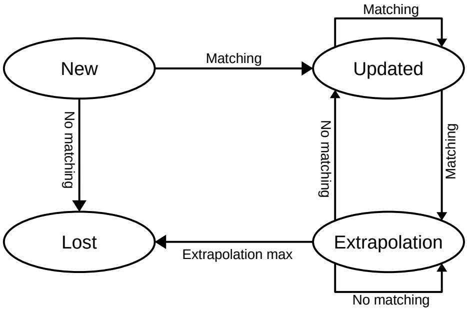


Figure 12 – State machine

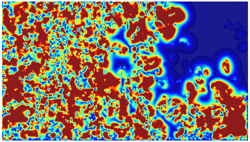
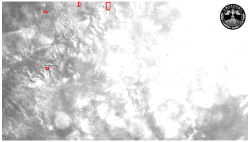
Validation bench

⇒ Qualify the processing chain

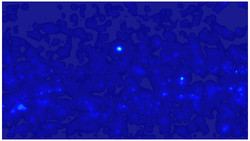
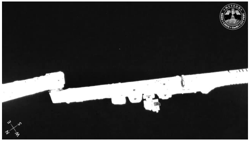
- ISS Meteor mission of Chiba University (Japan) [13].
  - High resolution camera filming Earth from the ISS [14].
  - 150 video sequences containing 50 meteors.
- Manual analysis to build a ground truth for each meteor.
- What is a valid detection ?
  - The supposed meteor is on the ground truth's trajectory.
  - The supposed meteor progresses on this trajectory in the right direction.
- Three scores :
  - Meteor detected or not.
  - Ratio between number of frames containing a meteor and number expected from the ground truth.
  - Number of false positives

Results

- 48 out of 50 meteors detected.
  - Probability of detection : 96%.
  - 70% of frames containing meteors are labelled as such.
  - Not really comparable to state of art results.
- Some issues
  - Cloudy and brightly sequences $\Rightarrow$ false positives

  - Fixed speed threshold $\Rightarrow$ false negatives

- Bonus : detection seems work on Earth (with stationary camera).

Conclusion

- Proposition of a new processing chain for meteors detection from space.
- Validation bench to qualify the chain.

Next steps :

- Adaptive threshold on speed.
- Algorithm optimizations.
  - Quality [15]/consumption [16]/speed [17] trade-off.
  - Simple and/or efficient algorithms [18, 19].
- Consider other hardware architectures (FPGA ?) [20].

Thanks

Références I

[1] N. Rambaux et al. Meteorix : a cubesat mission dedicated to the detection of meteors. In *42nd Assembly of Committee on Space Research (COSPAR)*, 2018.

[2] N. Rambaux et al. Meteorix : A cubesat mission dedicated to the detection of meteors and space debris. In *ESA NEO and Debris Detection Conference- Exploiting Synergies -ESA/ESOC, Darmstadt, Germany*, 2019.

[3] P. Gural. An operational autonomous meteor detector : Development issues and early results. *WGN, Journal of the International Meteor Organization*, 25 :136–140, 1997.

[4] S. Molau. The meteor detection software MetRec. page 131, 1999.

[5] SonotaCo. SonotaCo UFOCapture website.

[6] F. Colas et al. Fripon : a worldwide network to track incoming meteoroids. *Astronomy and Astrophysics (A & A)*, 644 :1–23, 2020.

[7] D. Vida, D. Zubović, D. Segon, P. Gural, and R. Cupec. Open-source meteor detection software for low-cost single-board computers. 2016.

[8] Y. Galindo and A. Lorena. Deep transfer learning for meteor detection. pages 528–537, 2018.

[9] P. Gural. Deep learning algorithms applied to the classification of video meteor detections. *Monthly Notices of the Royal Astronomical Society*, 489, 2019.

[10] D. Cecil and M. Campbell-Brown. The application of convolutional neural networks to the automation of a meteor detection pipeline. *Planetary and Space Science*, 186 :104920, 2020.

[11] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[12] E. Meinhardt-Llopis, J. Sánchez, and D. Kondermann. Horn-schunck optical flow with a multi-scale strategy. *Image Processing on line*, 20 :151–172, 2013.

[13] Chiba. Meteor project `http://www.perc.it-chiba.ac.jp/project/meteor`.

[14] T. Arai, M. Kobayashi, M. Yamada, T. Matsui, and Cometss Project Team. Meteor Observation HDTV Camera Onboard the International Space Station. In *Lunar and Planetary Science Conference*, page 1610, 2014.

[15] T. Romera, A. Petreto, F. Lemaitre, M. Bouyer, Q. Meunier, and L. Lacassagne. Implementations Impact on Iterative Image Processing for Embedded GPU. In *European Signal Processing Conference (EUSIPCO)*, Dublin, Ireland, 2021.

Références II

[16] A. Petreto, A. Hennequin, , T. Koehler, T. Romera, Y. Fargeaix, B. Gaillard, M. Bouyer, Q. L. Meunier, and L. Lacassagne. Energy and execution time comparison of optical flow algorithms on SIMD and GPU architectures. In *IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–6, 2018.

[17] S. Piskorski, L. Lacassagne, S. Bouaziz, and D. Etiemble. Customizing CPU instructions for embedded vision systems. In *Computer Architecture, Machine Perception and Sensors (CAMPS)*, pages 59–64. IEEE, 2006.

[18] L. Lacassagne and B. Zavidovique. Light Speed Labeling : Efficient connected component labeling on RISC architectures. *Journal of Real-Time Image Processing (JRTIP)*, 6(2) :117–135, 2011.

[19] F. Lemaitre, A. Hennequin, and L. Lacassagne. How to speed connected component labeling up with SIMD RLE algorithms. In *ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)*, pages 1–8, 2020.

[20] Ilias Bournias, Roselyne Chotin, and Lionel Lacassagne. FPGA Acceleration of the Horn and Schunck Hierarchical algorithm. In *International Symposium on Circuits and Systems (ISCAS)*, 2021.

[21] H. Ye, L. Lacassagne, D. Etiemble, L. Cabaret, J. Falcou, and O. Florent. Impact of High Level Transforms on High Level Synthesis for motion detection algorithm. In *IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–8, 2012.

[22] H. Ye, L. Lacassagne, J. Falcou, D. Etiemble, L. Cabaret, and O. Florent. High level transforms to reduce energy consumption of signal and image processing operators. In *IEEE International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 247–254, 2013.

[23] L. Lacassagne, D. Etiemble, A Hassan-Zahraee, A. Dominguez, and P. Vezolle. High level transforms for SIMD and low-level computer vision algorithms. In *ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)*, pages 49–56, 2014.

[24] A. Hennequin, I. Masliah, and L. Lacassagne. Designing efficient SIMD algorithms for direct connected component labeling. In *ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)*, pages 1–8, 2019.

Références III

[25] L. Lacassagne and A. B. Zavidovique. Light Speed Labeling for RISC architectures. In *IEEE International Conference on Image Analysis and Processing (ICIP)*, pages 3245–3248, 2009.

[26] L. Cabaret, L. Lacassagne, and D. Etiemble. Parallel Light Speed Labeling for connected component analysis on multi-core processors. *Journal of Real-Time Image Processing (JRTIP)*, 15(1) :173–196, 2018.

[27] L. Cabaret and L. Lacassagne. A review of world's fastest connected component labeling algorithms : Speed and energy estimation. In *IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–8, 2014.

Example

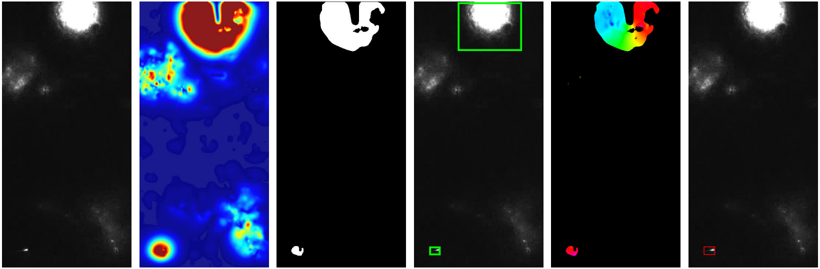


Figure 13 – Complete example for frame 119

How to speed up detection ?

Our goal is to reach real-time rate (25 frame/s).

- High Level Transforms
  - reducing time and energy consumption [21, 22, 23].
- Code parallelization
  - Multi-threading
  - SIMD (data parallelism) [24]
- Quality trade off

Connected Component Labeling

- Performed with Light Speed Labeling algorithm [25].
  - Available in a parallel version [26].
  - Very fast and energy efficient [27].