

# METEORIX

High performance computer vision application for Meteor detection from a CubeSat

M. Millet<sup>1,2</sup>, N. Rambaux<sup>3</sup>, A. Cassagne<sup>1</sup>, M. Bouyer<sup>1</sup>, A. Petreto<sup>1</sup>, L. Lacassagne<sup>1</sup> and Meteorix team

<sup>1</sup>LIP6, Sorbonne University, CNRS

<sup>2</sup>LERITY - Alcen

<sup>3</sup>IMCCE, Observatory of Paris, PSL, Sorbonne University



## General Context: Meteor detection

- ▶ A meteor is a luminous phenomena occurring when a meteoroid enters the Earth's atmosphere
  - ▶ Provides information on the history of the solar system
  - ▶ Allows to estimate the matter flow entering the atmosphere
- ▶ Same phenomena when an artificial satellite re-enters the atmosphere
  - ▶ Useful detection to plan safe space activities
  - ▶ Useful detection to space debris
- ▶ Meteors detection can be achieved on Earth with ground-camera networks like FRIPON [1]
- ▶ Chiba University showed the feasibility of a space observation of meteors with ISS Meteor experiment[2, 3]
  - ▶ 2 advantages : absence of weather constraint and wide coverage



Figure 1: An example of meteor from ISS Meteor[2] (Chiba Univ.)

- ▶ University CubeSat 3U mission of Sorbonne University[4] for meteors detection and characterization
- ▶ The payload is composed by a camera [5] and a computer-vision processing chain [6]

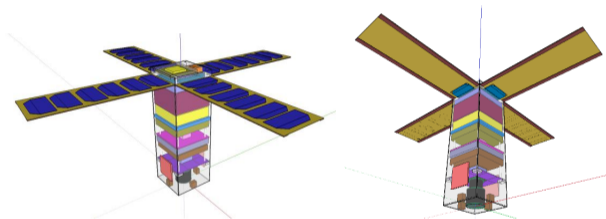
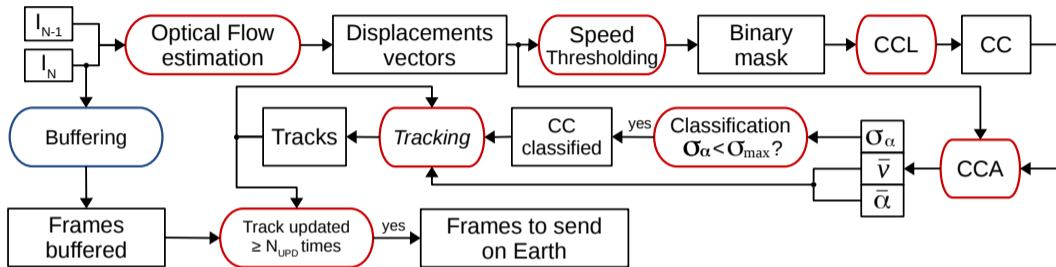


Figure 2: Views of Meteorix's CubeSat[7]

- ▶ This presentation focuses on the processing chain which has 2 main constraints :
  - ▶ Low power consumption: **7W available** for processing images
  - ▶ Real-time detection: processing 25 frames per second (i.e. 40ms/frame)

**Goal = to demonstrate the feasibility of a real-time processing chain embedded in a CubeSat**

# Meteorix Computer Vision Processing Chain



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

- ▶ validated thanks to the video sequences of the ISS Meteor[8] experiment of Chiba University
- ▶ 47 out of 50 meteors from these sequences

Reminder: We have to reach the real-time detection (40ms/frame, 25frames/s)

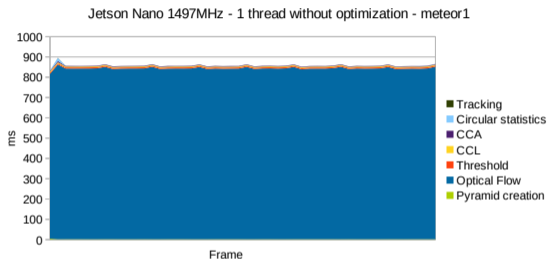
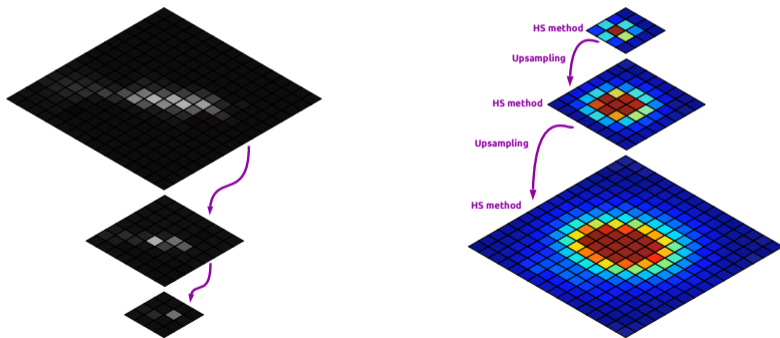


Figure 4: Execution time for each frame of the sequence meteor1 from ISS Meteor. The processing chain takes more than 800ms to process one frame on a Nvidia Jetson Nano board

- ▶ The optical flow algorithm takes the most of the time (> 95 %)
- ▶ Far from real time... a  $\times 20$  speedup is required !

⇒ Implementation should be optimized

- ▶ Optical flow estimation is an estimation of the apparent movement
- ▶ Horn & Schunck [9] method used in the processing chain
  - ▶ Iterative algorithm more suited to algorithm optimizations
  - ▶ Pyramidal version to estimate larger movements



**Figure 5:** The first step of the pyramidal H&S method is the creation of a pyramid by downsampling the source image. The second step is the computation of the flow and then its upsampling for the next scale.

- ▶ To master ARMv8 architecture in order to exploit all its features (already used in other CubeSat)

## High Level Transforms [10, 11, 12]

- ▶ **Operator Fusion**: to reduce memory accesses
- ▶ **Operator Pipeline**: to maximize persistence of data within CPU caches
- ▶ **Operator Factorisation & Reduction**: to reduce amount of computation
- ▶ **Algorithm (parameter) Tuning**: to adapt complexity to quality of detection

## Architecture optimizations

- ▶ **Thread Level Parallelism**: OpenMP loop parallelism to distribute work on CPU cores
- ▶ **Instruction Level Parallelism** SIMD data parallelism [13] for regular computation within a core

## Arithmetical Optimizations

- ▶ **Reciprocal estimate**:  $a \times \frac{1}{b}$  instead of  $\frac{a}{b}$  is faster (but less accurate)
  - ▶ Division : 34 cycles
  - ▶ Multiplication + Reciprocal :  $5 + 5 = 10$  cycles
- ▶ **Fused Multiply-Add**: compute  $a \times b + c$  in one instruction
- ▶ **Floating point precision**: simple (32-bit) or half (16-bit) to increase SIMD parallelism ( $8 \times 16$  vs  $4 \times 32$ )

## System Optimization

- ▶ **Frequency tuning** of CPU cores and memory to reduce power consumption

## Results : Execution time and algorithm tuning

- ▶ Optical flow computed with 3 scales
  - ▶ without optimization : 841 ms for optical flow a 1497 MHz
  - ▶ with optimizations : 80ms for optical flow a 1497 MHz  $\Rightarrow \times 10$ , but not enough
- ▶ 3 iterations scheme has been tested to reduce execution time:
  - ▶ 6, 1 or 0 iteration at largest level

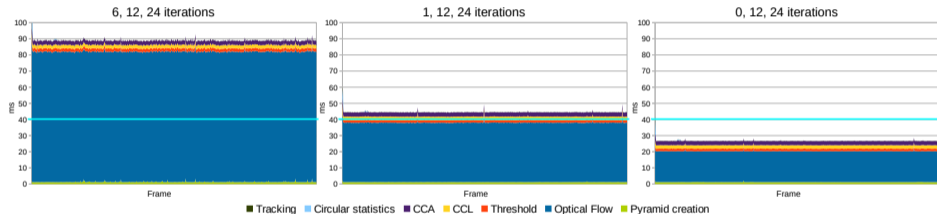


Figure 6: Jetson Nano 1497MHz, 4 threads

$\Rightarrow$  Real-time reached by reducing iterations on the bigger scale of the pyramid !

Iteration scheme	[6, 12, 24]	[1, 12, 24]	[0, 12, 24]
Detected meteors	47	47	47

$\Rightarrow$  Reducing iterations does not change the number of detected meteor !  
(optical flow is still enough accurate)



## Results : Power consumption vs frequency & execution time

In order to respect the power limitation, we looked for the highest frequency respecting the execution time constraint

Frequency	1497MHz	1224MHz	1036MHz	921MHz	825MHz
Execution time (ms/frame)	26.8	31.4	35.5	38.8	42.3
Power consumption (W)	4.9	4.2	3.9	3.7	3.6

- ▶ 825MHz: beyond the execution time constraint
- ▶ 921MHz: under the execution time constraint but close  $\Rightarrow$  need a safety margin
- ▶ 1036 MHz: seems like a good compromise, offering safety margins for both constraints

## $\tau$ -Herculids shower (end of May 2022)

- ▶ The processing chain is adapted to airborne camera embedded to aircraft
- ▶ Same algorithms [14, 15] but the Optical Flow is replaced by an image registration algorithm

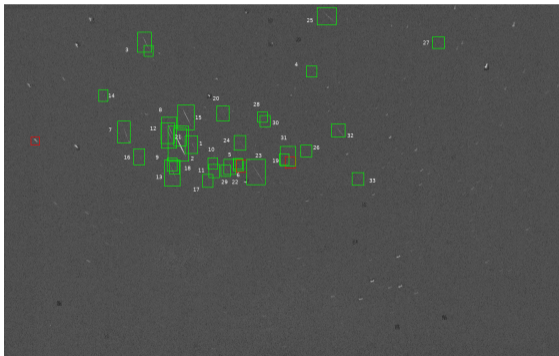


Figure 7: Composite of a 33 meteor cluster [16], 100% of detection

⇒ These detections increase the Technology Readiness Level of the algorithm to 5

- ▶ A new processing chain for meteor detection from space has been designed and validated with a dataset
- ▶ Algorithmic, architectural, arithmetical and system optimizations have been applied to:
  - ▶ reach **real time processing** (40 ms) while enforcing
  - ▶ **power consumption** constraint and
  - ▶ **quality** of detection
- ▶ Next step: load balancing = optical flow on GPU [17] or FPGA [18] (more suited to such a kind of computing)

Thank you !

And a special thanks for :



- [1] F. Colas et al. Fripon: a worldwide network to track incoming meteoroids. *Astronomy and Astrophysics (A & A)*, 644:1–23, 2020.
- [2] Chiba. Meteor project <http://www.perc.it-chiba.ac.jp/project/meteor>.
- [3] 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.
- [4] N. Rambaux et al. Meteorix: a cubesat mission dedicated to the detection of meteors. In *42nd Assembly of Committee on Space Research (COSPAR)*, 2018.
- [5] N. Rambaux, J. Vaubaillon, S. Derelle, M. Jacquart, M. Millet, L. Lacassagne, A. Petreto, P. Simoneau, K. Baillie, J. Desmars, D. Galayko, and R. Chotin. Meteorix camera tests for space-based meteor observations. *WGN, Journal of the International Meteor Organization (IMO)*, 49(5):1–3, 2021.
- [6] M. Millet, N. Rambaux, A. Petreto, F. Lemaitre, and L. Lacassagne. Meteorix — a new processing chain for real-time detection and tracking of meteors from space. *WGN, Journal of the International Meteor Organization (IMO)*, 49(6):1–5, 2022.
- [7] 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.
- [8] T. Arai, M. Kobayashi, M. Yamada, H. Senshu, K. Wada, S. Ohno, K. Ishibashi, R. Ishimaru, K. Maed, and T. Matsui. International Space Station-Based Meteor Observation Project: Initial Results. In *Lunar and Planetary Science Conference*, 2017.
- [9] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] L. Lacassagne and B. Zavidovique. Light Speed Labeling for RISC architectures. In *IEEE International Conference on Image Analysis and Processing (ICIP)*, pages 3245–3248, 2009.
- [15] 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.
- [16] J. Vaubaillon et al. Detection of a 2022  $\tau$ -herculid meteor cluster. 2022.
- [17] 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.
- [18] R. Chotin I. Bournias and L. Lacassagne. FPGA Acceleration of the Horn and Schunck Hierarchical algorithm. In *International Symposium on Circuits and Systems (ISCAS)*, 2021.

# Example of processing

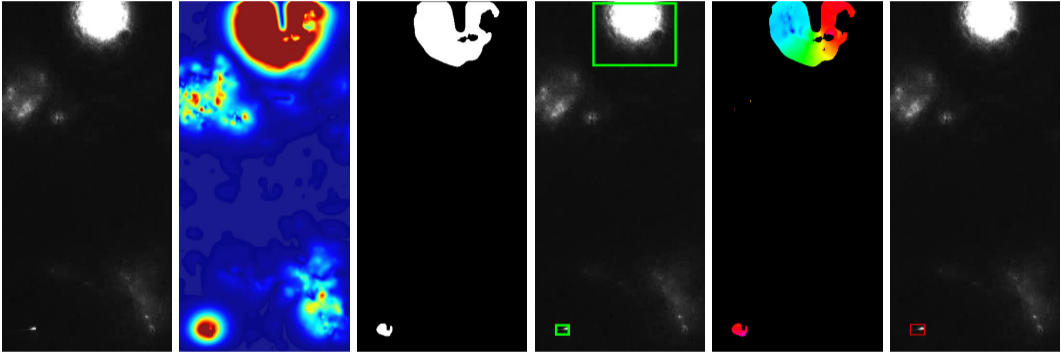


Figure 8: The different steps of the processing chain

## The main goals of the mission

- ▶ **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.



# What is SIMD ?

## ▶ SIMD : Single Instruction Multiple Data



A	1.1	2.2	3.3	4.4
			x	
B	2.0	4.0	3.0	0.0
			=	
C	2.2	8.8	9.9	0.0

Figure 9: 128-bit SIMD registers and 32-bit floating point multiplication

- ▶ CPU 4-core ARM Cortex 57
- ▶ 4GB RAM
- ▶ GPU 128-core Maxwell



Figure 10: NVIDIA Jetson NANO developer kit

- ▶ 4 solar panels
- ▶ Li-Ion battery (48kWh)
- ▶ Power board
- ▶ Magnetorquer board
- ▶ Reaction wheel
- ▶ S-band transmitter with two antennas (for scientific mission)
- ▶ UHF/VHF transceiver with one antenna
- ▶ Interface board
- ▶ On-board computer