

CARS: A Photogrammetric Pipeline for Global 3D Reconstruction using Satellite Imagery

David Youssefi¹, Valentine Bellet¹, Yoann Steux², Mathis Roux², Cédric Traizet², Marian Rassat², Tommy Calendini²

¹ Centre National d'Etudes Spatiales (CNES), 18 avenue Edouard Belin, Toulouse, France
(david.youssefi, valentine.bellet)@cnes.fr

² CS GROUP, 5 rue Brindejonc des Moulinais, Toulouse, France
(yoann.steux, mathis.roux, cedric.traizet, marian.rassat, tommy.calendini)@cs-soprasteria.com

Keywords: Image processing, Photogrammetry, Stereovision, Digital Surface Model, High Performance Computing

Abstract

We present CARS, a multiview stereo pipeline developed by CNES, in its 1.0 release. This pipeline will be integrated into the processing chain of the CO3D (French acronym for "Constellation Optique 3D") mission. The goal of this mission is to generate a 3D model of the Earth in less than four years. Because this is an operational mission involving massive production, particular attention has been paid to ensuring that the software is robust, efficient and includes a set of advanced automatic processing features. The paper will provide a comprehensive overview of all the features developed since its creation to achieve this goal.

1. Introduction

The CO3D (French acronym for "Constellation Optique 3D") mission, launched successfully at the end of July 2025, consists of four satellites orbiting in pairs to map all land areas in 3D. This collaborative mission between CNES, the French space agency, and Airbus, a private aerospace company, leverages photogrammetry by capturing images from multiple angles of the same site. Due to their synchronized positioning in the same orbit, satellites can freeze moving objects, enhancing the accuracy of their 3D reconstructions (Lebègue et al., 2020).

In this context, CNES develops the open-source software CARS¹ (Youssefi et al., 2026) (French acronym for "Chaîne Automatique de Restitution Stéréoscopique") with the support of CS GROUP. This Multi View Stereo Pipeline (MVS) enables Digital Surface Models (DSM) to be produced from pairs of stereoscopic images. It is integrated into the operational CO3D image processing pipeline called CODIP (Melet et al., 2020). CARS is open-source software: the CNES 3D team chooses to release this tool prior to the mission's launch to gather user feedback and improve the quality of final CO3D products. Currently, most users are familiar with this software through Pléiades or WorldView stereo images.

The main challenge of the CO3D mission is to generate a model of the Earth's surface in less than four years, requiring CARS software to be highly efficient and scalable. This paper will describe how CARS software addresses these challenges by leveraging advanced algorithms and parallel processing techniques to ensure accurate 3D reconstructions.

2. Algorithm design

In the field of 3D reconstruction, rendering methods such as Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) and Gaussian Splatting (Kerbl et al., 2023) are being extensively studied by the scientific community specializing in image processing. Inspired by this research, the remote sensing community is trying to adapt these methods to address the lack

of images from the same season (Derksen and Izzo, 2021), to sensor modeling (Marí et al., 2022) or scaling (Billouard et al., 2025). Although it will enable multiple acquisitions to be made at the same site for site monitoring, the CO3D mission is primarily a coverage mission: the 3D model will therefore generally be derived from two images; as a result, the new rendering methods are not currently suitable for use in the operational workflow. For the time being, CNES continues to use and adapt these methods for research purposes.

MVS pipelines generally perform best with a small number of images; in particular, they typically use epipolar geometry resampling to limit the search for corresponding points to a single direction. Several MVS software existed prior to the development of the CARS software (Beyer et al., 2018), (Krauß et al., 2013), (Rupnik et al., 2017), (Qin, 2016), (de Franchis et al., 2014), but none of them met the licensing requirements necessary for collaboration with Airbus for CO3D, were stable enough to scale up without artifacts or were designed to be parallelized for processing large satellite images.

The development of CARS began in mid-2019. The two articles published in 2020 describe the algorithmic steps (Michel et al., 2020) as well as the use of Dask (Youssefi et al., 2020) to link the processing steps. During the past six years, the initial CARS workflow described in these papers has been revised to simplify the user interface and improve stability, calculation time, and quality.

The following subsections first review the basic pipeline, which will be used several times in the multi-resolution approach presented in the next section. Since photogrammetry performs poorly for water areas and the CO3D DSM should not contain any nodata values, a subsection is dedicated to the algorithms used to fill these holes. Finally, the last subsection presents the use of a global land cover map and the parameters which should be adjusted depending on the area's classification on the aforementioned map.

2.1 Surface modeling key steps

CARS processes a set of image pairs to generate 3D point clouds, which are then concatenated and projected onto a regu-

¹ CARS: <https://github.com/cnes/cars>

lar grid to produce the final DSM. In this section, we focus on surface reconstruction, defined as pairwise processing of images from the sensor level to the DSM described in Figure 1.

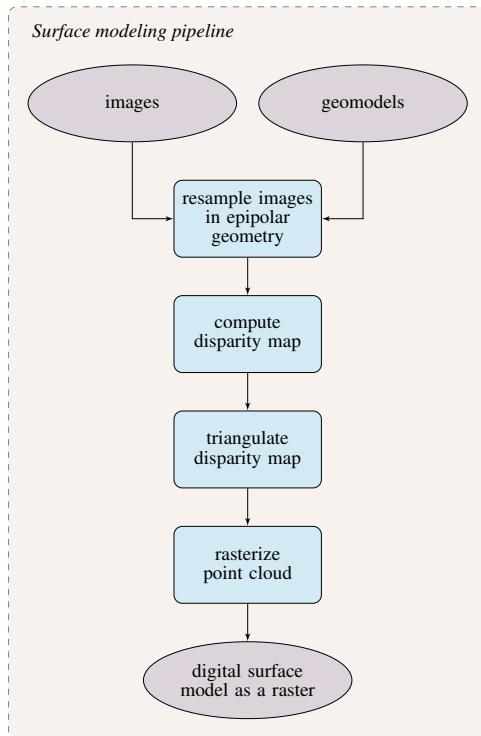


Figure 1. CARS pipeline used at each resolution.

Images and geomodels. It is important to note the following before going through these steps; for the processing to work, the input images must be sensor-geometry images accompanied by the geometric model. Unlike orthorectified images, sensor images are not aligned with a specific coordinate system. CNES has a library called LibGEO that allows physical models to be read directly (notably those from CO3D), which CARS is able to utilize as a plugin (see subsection 3.3). This library is not open-source, which is why CARS can use Shareloc², also developed by CNES. This library is simpler because it does not read physical models, but rather the widely used and standardized models Rational Polynomial Coefficients (RPC) to convert from image space to ground space.

Resample images in epipolar geometry. The steps are those of a classic MVS. The first step is to resample the images using epipolar geometry. The matching algorithms used in the next step can be used to track moving objects within a scene. In the specific case of photogrammetry, the aim is to reconstruct 3D models of objects that are stationary within the scene but appear to move between two viewpoints: our eyes work in much the same way. In this case, with two images taken from two different viewpoints, all possible positions of a point in the left-hand image lie on a line in the right-hand image known as the epipolar line locally straight with satellite images (de Franchis et al., 2014).

Compute disparity map. Once the images have been resampled in epipolar geometry, CARS calls Pandora³, a stereo matching framework. For each pixel in the left image, it finds

the pixel in the right image with the most similar neighborhood. The Pandora framework is inspired by the taxonomy of (Scharstein et al., 2001). In particular, it allows the use of several similarity metrics: the well-known Census metric (Zabih and Woodfill, 1994), as well as deep learning-based metrics such as MC-CNN (Zbontar and LeCun, 2016). Finally, the offsets obtained from these measurements are smoothed using the well-known algorithm Semi Global Matching (SGM) (Hirschmuller, 2008).

Triangulate disparity map. The result generated by Pandora is a disparity map showing the column offset for each pixel in the left image (row i and column j). These offsets $\delta_j(i, j)$ are then converted into a map of point positions in the left epipolar image: $P_{\text{epipolar left}}(i, j) = (i, j)$ and in the right epipolar image: $P_{\text{epipolar right}}(i, j) = (i, j + \delta_j(i, j))$. These positions are mapped back to the original sensor geometries, allowing RPC models to be used to draw sight lines for the corresponding pixels. The intersection of these sight lines (the point closest to both lines) then gives the calculated position of the point $P_{\text{cloud}}(i, j) = (X_{ij}, Y_{ij}, Z_{ij})$ composing the final point cloud.

Rasterize point cloud. This point cloud is projected onto a regular grid. We define a regular grid that covers the generated point cloud of origin $(x_{\text{start}}, y_{\text{start}})$ and size $(x_{\text{size}}, y_{\text{size}})$ with constant resolution. For each cell of center (c_x, c_y) , we consider the subset of points p_k contained in the disk D centered on this cell. Then, the altitude assigned to the cell $z(c_x, c_y)$ is a Gaussian weighted average (standard deviation σ) of the distance d_k from its center.

$$z(c_x, c_y) = \frac{\sum_{p_k \in D} z_k e^{-d_k^2/2\sigma^2}}{\sum_{p_k \in D} e^{-d_k^2/2\sigma^2}}$$

By doing this for each cell in the grid, we obtain the final DSM as a raster.

2.2 Epipolar alignment and disparity ranges

The steps outlined above summarize the key and standard steps of photogrammetric reconstruction, while ignoring two limiting factors: geometric models may be imperfect, and the correlation step even when limited to a single line is time-consuming and memory-intensive.

Epipolar alignment. Epipolar resampling is performed using geometric models. If these models are biased, the corresponding pixels may no longer be on the same line as expected. This is why CARS performs a standard sparse matching step using Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) prior to surface modeling, a pipeline called tie points. This step is performed directly on the images resampled in epipolar geometry using the raw geometric models. Thus, for a list of matches $P_{\text{epipolar left}}(i, j)$ and $P_{\text{epipolar right}}(i, j)$, the residual offset is simply calculated by taking the difference between the row positions $\delta_i(j)$ of the matched points. This offset is then reapplied to the right model so that the images in epipolar geometry can be aligned correctly.

Disparity ranges. The step pipeline that takes the longest in terms of computation time and also requires the most memory is the correlation step performed by Pandora. It is the longest step because the computational complexity of SGM (Hirschmuller, 2008), for an image of size (W, H) (respectively

² Shareloc: <https://github.com/cnes/shareloc>

³ Pandora: <https://github.com/cnes/pandora>

width and height) and a range of disparities of D , is $\mathcal{O}(WHD)$. Moreover, memory consumption is also high because the algorithm requires storing a cost volume measuring $W \times H \times D$. These constraints can only be relaxed by adjusting the disparity range. To attempt to reduce this disparity range, resampling in epipolar geometry uses an initial elevation as the default altitude. Thus, the disparity is zero if the pixels compared are indeed at the initial altitude. Additionally, the latest version of CARS includes a multi-resolution approach described in the following subsection.

2.3 A multi-resolution approach

Among the developments in the pipeline, the CARS software now adopts a multi-resolution approach to generate 3D models (see Figure 2). Each sensor image is subsampled in multiple resolutions and the surface modeling pipeline is applied to each resolution. This makes it possible to drastically reduce the calculation times and improve the quality of the output DSM by refining the altitude search intervals for the Pandora correlator.

Subsample images. The input images are subsampled to several resolution levels. As mentioned in the previous subsection, the computational complexity and memory usage in the correlation process are directly related to the size of the input images: the smaller the images, the lower the computational time and memory usage. Furthermore, although the disparity range is expressed in pixels, it corresponds to a range of possible heights. Reducing the size of the images, therefore, reduces the disparity range to be explored. In this multi-resolution approach, the surface modeling pipeline can, for example, be run three times (corresponding to the requested resolution levels): first at very low resolution to roughly estimate the overall ground elevation. Next, a second iteration can be used to roughly estimate the elevation of above-ground features (trees, buildings, etc.). A final iteration will be used to actually calculate the full-resolution DSM.

Create minimum and maximum elevations. Each step in the multi-resolution pipeline generates a DSM. This DSM, which must be smoothed as much as possible to avoid image distortion (particularly by lowering its resolution), is used to resample images and thereby reduce the disparity range. Furthermore, in order to reduce this disparity range, the raw low resolution DSM generated at the previous resolution is used to generate two minimum and maximum elevations that correspond to the elevation ranges explored at the next resolution.

2.4 Filling algorithms

Sometimes, it is difficult to determine the altitude of certain areas using simple photogrammetric processing, particularly in cases of clouds, occlusion, or poor matches detected during the Pandora correlation stage (such as water or shadow areas). As a result, these areas are typically assigned a nodata value. However, since nodata values can be challenging for end users to manage, we have developed specific processes in CARS to fill in these pixels.

To address this, CARS uses a land cover map as input to apply different fillings based on the type of land cover. To generate the land cover map, we employ the open-source software SLURP⁴ (Tanguy et al., 2024), developed by CNES. This tool processes four-band satellite images (red, green, blue, near-infrared) and applies several machine learning algorithms to

⁴ SLURP: <https://github.com/cnes/slurp>

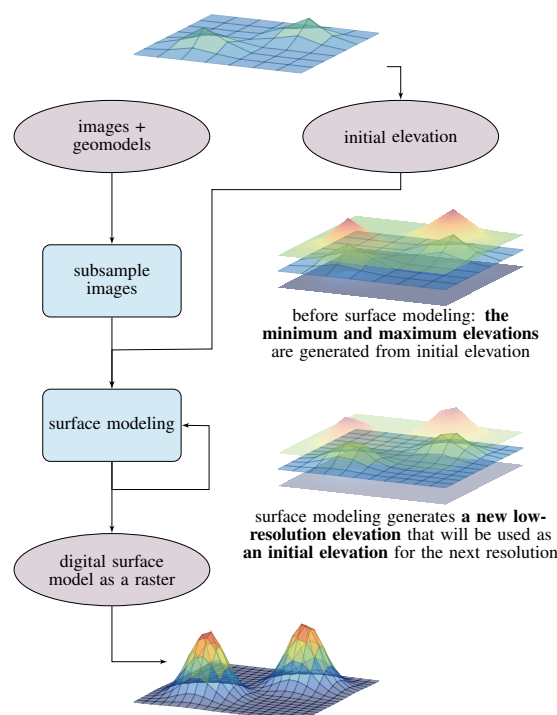


Figure 2. CARS Multi-resolution. The initial elevation allows to estimate the elevation range to be explored (minimum and maximum elevations); this estimate is refined through successive iterations until the final DSM is obtained.

create multiple simple masks (water, shadow, low/high vegetation, built-up areas). These masks are then combined and refined into a six-class classification (water, low/high vegetation, built-up areas, bare soil, undefined). SLURP algorithms rely on pre-calculated primitives such as NDVI, NDWI, and texture, and also incorporate external data sources such as Global Surface Water (Pekel et al., 2016)⁵ (a dataset that maps global surface water dynamics over 37 years using satellite observations) or the World Settlement Footprint 2019⁶ (a global map of human settlements derived from Sentinel-1 and Sentinel-2 satellite data using temporal analysis and machine learning).

For water surfaces, different treatments are required, depending on the type of water body: sea, lake, or river. Since these areas have different characteristics, filling varies accordingly. SLURP can generate a water mask categorized into these three types (sea, lake, river) using an external data source: the Copernicus Water Body Mask⁷. For sea, filling involves setting the values to zero at the geoid level. For lakes, the altitude of the lake's edges is used, outliers are removed, and a constant value is then applied across the entire lake. For rivers, filling is done using an endogenous Digital Terrain Models (DTM) calculated by the open-source software Bulldozer (Lallement et al., 2023) developed by CNES, as Bulldozer extracts a DTM from a DSM.

⁵ Global Surface Water:

<https://global-surface-water.appspot.com/>

⁶ World Settlement Footprint 2019:

<https://download.geoservice.dlr.de/WSF2019/>

⁷ Water Body Mask:

<https://dataspace.copernicus.eu/explore-data/data-collections/copernicus-contributing-missions/collections-description/COP-DEM>

The same filling, endogeneous filling, as for rivers, is applied to occluded and shadowed areas. For clouds, similar to water areas, a cloud mask can be provided as input to CARS. We suggest using the filling with an exogenous lower-resolution DSM. Figure 3 summarizes the filling methods for each case.

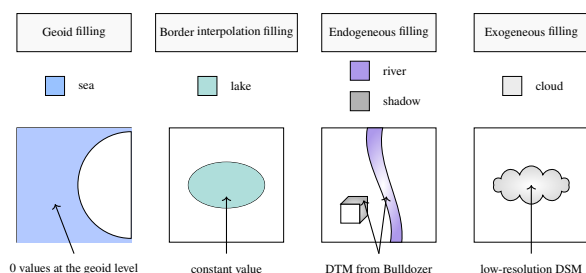


Figure 3. Overview of the different filling methods. From top to bottom: the type of filling, the filled elements, and a diagram showing the filled objects and the filling values.

2.5 Global land management

Finally, to address the challenge of generating a global DSM, CARS will carry a global land cover map (the ESA WorldCover at 10 m resolution⁸) to adapt the settings to the reconstructed area of the world (see Figure 4).

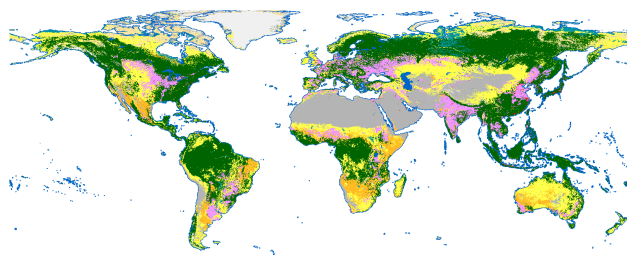


Figure 4. Global cover map integrated into the CARS software. One color/value per setting.

Depending on the type of landscape (urban, rural, desert, mountainous, forested, etc.), the CARS pipeline may require relatively different settings. In particular, the correlation settings, which significantly impact the performance of the software, could be adjusted, for example, to smooth out the disparity maps in natural landscapes.

Another approach worth considering, again with the aim of striking a balance between quality and computation time, would be to apply specialized processing in urban areas to reconstruct buildings more accurately: to achieve this, one could use different resolutions depending on whether the landscape is natural or not, or even implement an optimization strategy based on the algorithm SGM and guided by semantic segmentation of buildings (Dumas et al., 2022).

3. Implementation

3.1 Data model

As shown in Figure 1, the surface modeling pipeline consists of four main steps; these collectively enable the generation of the DSM from a set of sensor images and their associated geomodels. To manage and structure the data throughout this process,

⁸ ESA WorldCover: <https://doi.org/10.5281/zenodo.7254221>

and because no suitable solution existed previously, an internal CARS data structure has been introduced. CARS datasets can be represented either as Xarray Datasets or Pandas DataFrames, and their primary purpose is to efficiently organize and handle tiled data.

The rasters processed within CARS can reach very large dimensions, making a tiled approach necessary. The CARS data structure facilitates both the storage and the merging of these tiles.

In terms of construction, a CARS dataset is similar to an Xarray Dataset or Pandas DataFrame: each variable must have its specified dimensions, and relevant attributes should be assigned. Additionally, it allows the specification, for each tile, of the corresponding window, the global raster profile (including dimensions, transform, and related metadata), as well as the overlap regions used as processing margins.

This data structure is therefore well suited for parallelization, as the tiling strategy allows independent processing of each tile, while the overlap regions ensure continuity and consistency between neighboring tiles.

3.2 Parallelisation

Given the scale of the images and the computational intensity of stereo-matching, CARS employs a tiled processing strategy. This approach ensures that memory overhead remains manageable while enabling parallel execution to optimize CPU utilization relative to the system's memory constraints.

CARS was initially designed around the Dask⁹ framework, as explained in (Youssefi et al., 2020). The implementation relies on low-level Dask primitives (Delayed and Futures) rather than high-level tiled structures. While parallelized objects like Dask-backed Xarray or Pandas would be more efficient, the intricate geometry transformations in our pipeline currently prevent us from adopting these optimized abstractions. The current reliance on low-level Dask primitives in CARS can lead to non-deterministic performance issues. Specifically, inefficient memory management may cause Dask workers to terminate or entire cluster nodes to fail due to memory exhaustion.

Given that the CO3D ground segment requires a reliable automated workflow, these performance issues were unacceptable. Consequently, we shifted toward a more predictable execution model using Python's native multiprocessing capabilities. To improve scalability, CARS now utilizes an Orchestrator to manage parallelization. This framework consists of two main pillars:

- **Clusters:** An abstraction of Dask's execution model. It allows seamless switching between Dask, Sequential, or a custom Multiprocessing backend. The latter enhances performance through task graph factorization and optimized temporary file handling.
- **Registries:** These automate the lifecycle of CarsDatasets, managing tile storage, and standardized file output (e.g., .tif).

By decoupling the processing logic from the execution engine, the Orchestrator enables developers to focus on feature growth without needing to master the complexities of distributed computing.

⁹ Dask: <https://www.dask.org/>

3.3 Plugins

CARS adopts a modular approach to its applications and pipelines, continuing to support and extend its plugin system. Any Python package can register itself as a CARS plugin at installation time through environment variables and is automatically imported when CARS is executed. Upon initialization, any class inheriting from the Application, Pipeline, or Geometry templates is registered accordingly, enabling seamless use through configuration files.

In addition to Shareloc, the architecture supports the integration of alternative geometry libraries as plugins. This modularity ensures compatibility with the unique sensor characteristics and physical modeling requirements of various missions: this is used in the CO3D mission with the LibGEO plugin.

Recent work has focused on improving the quality of the dense matching step by incorporating monocular geometry estimation derived from MoGe2 (Wang et al., 2025). In particular, an edge map (classifying pixels as belonging to vertical slopes, such as building facades) was introduced and is used during the dense matching step. Along with this functionality, an edge detection plugin has been developed, allowing users to generate such auxiliary data directly within CARS (see Figure 5). In addition, the raw depth and normal maps produced by MoGe2 can be exported. However, these outputs currently exhibit artifacts from tiling and will require further work to ensure proper alignment between adjacent tiles and at the DSM global level. A tile ID map is provided to indicate pixel-level coherence.

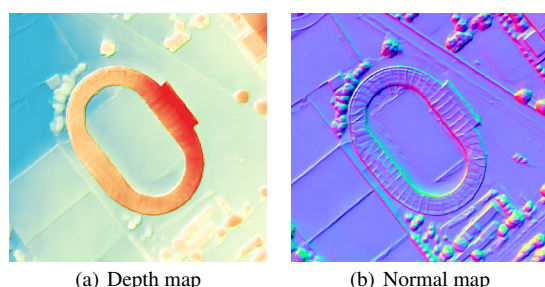


Figure 5. The outputs of the CARS edge detection plugin on a Pléiades Image (Toulouse, France).

Pléiades © CNES 2013, Distribution CNES PWH

Beyond stereo reconstruction, CARS can serve as a foundation for pipelines targeting different objectives. For example, CS GROUP has developed a pipeline dedicated to urban mesh generation¹⁰. Using the native parallelization capabilities of CARS, using depth maps and building classification data, this plugin produces lightweight city-scale meshes (see Figure 6). More recently, and in connection with the edge detection plugin, support for LOD2 mesh generation has been introduced.

The ongoing work also aims to facilitate the integration of alternative tools within the stereo reconstruction pipeline at a lower level, particularly for sparse and dense matching steps.

4. Results

Since the CARS pipeline is designed to be integrated into an operational pipeline, we created a database using Pléiades imagery

¹⁰ CARS Point Cloud to Mesh plugin:

<https://github.com/CS-SI/cars-point-cloud-to-mesh>

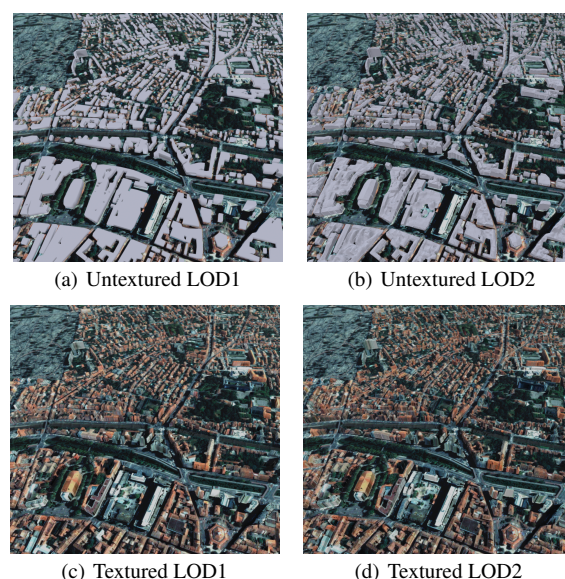


Figure 6. The outputs of the CARS mesh generation plugin, textured and untextured, on Pléiades data (Toulouse, France).

Pléiades © CNES 2013, Distribution CNES PWH

data called “CARS Technical Control” to monitor the quality of the DSM and, above all, to track the pipeline’s processing times.

Figure 7 shows a subset of the CARS Technical Control database. The sites have been chosen to benchmark CARS in different types of landscapes. For example: mountains with Ecrins, cliffs with Cassis, canyons with Ardèche, islands with Ouât, high buildings and dense urban areas with Paris.

Table 1 shows the production times obtained from the dataset. Productions have been launched on the CNES high performance computing cluster. Each production used 32 cores. The processing speed given in milliseconds per pixel considers the time required by CARS to process one pixel in sensor geometry with one worker (CPU).

Location	Sensor Size (px)	DSM Size (px)	Time	Speed (ms per px)
Ardèche	40000x36692	21300x20278	3h33	0.27
Cassis	40000x72280	21319x37793	7h03	0.28
Colmar	40000x43380	20651x21983	2h36	0.17
Ecrins	40000x109692	22007x56101	10h53	0.28
Lyon	40000x17964	23716x10013	1h11	0.19
Marseille	40000x20868	23594x12155	1h50	0.19
Ouat	40000x28072	20223x14297	2h04	0.21
Paris	40000x48196	20569x24264	2h49	0.16

Table 1. Production time on CARS Technical Control Dataset

The two slowest productions are Cassis and Ecrins. The Ecrins site is located in the Alps, with peaks reaching an altitude of more than 3500 m. Mountainous areas are more likely to take longer to be processed, because the altitude ranges to explore are higher. The Cassis site is less mountainous, but contains a high cliff like Cap Canaille, which is the highest cliff in France (400 m). Cliffs are hard to process with CARS because even with the multi-resolution approach, the altitude range to explore at the final resolution is necessarily the height of the cliff (400 m).

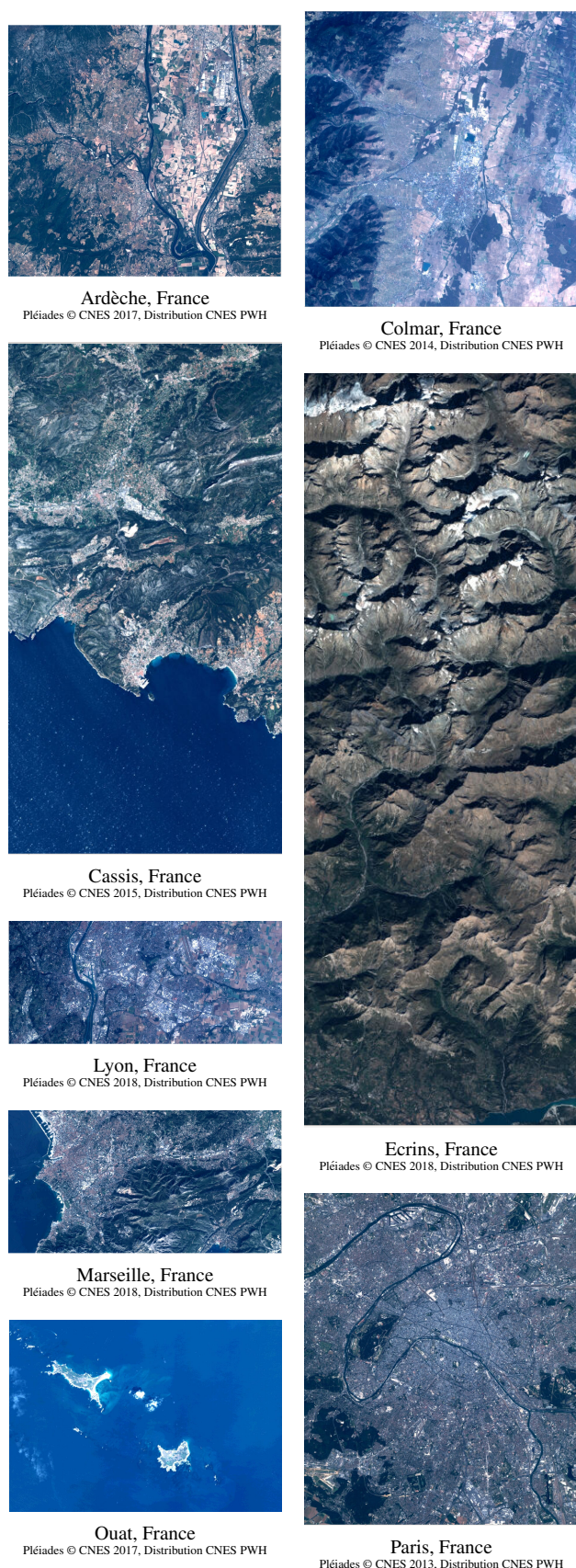


Figure 7. Preview of Pléiades images used for benchmark.

However, the processing times remain stable across different scenes thanks to the multi-resolution approach. In fact, the processing of downsampled images is relatively negligible, and the creation of minimum and maximum elevations allows for relatively similar disparity intervals regardless of the scene being processed. Thus, on average, CARS processes a pair at a speed of $0.2 \text{ ms per px} \pm 0.05$.

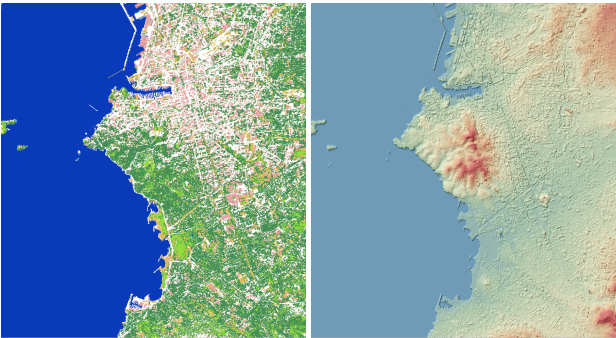
And because the process must be fully automated, this section presents examples of fully automated water filling (see Figure 8). For each example, the filled area is shown in blue on the classification map. The type of filling used is determined by the subclass of the classification of water (sea, river, or lake), as described in section 2.4. The classification of other pixels (ground, vegetation, etc.) is not used by CARS during the filling process.

Although there is no guaranty that the elevation will be exactly as calculated, the water filling provides a realistic representation of areas not reconstructed by photogrammetry and simplifies the use of these data, which will consist solely of valid data. Note that along with these products, CARS transfers the classification provided as input and also generates a `filling.tif` file, allowing the user to see how the software filled the final result.

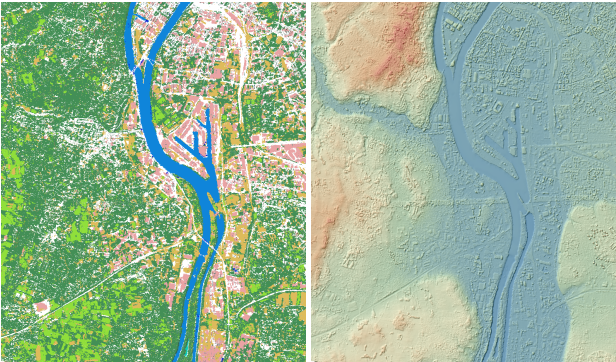
5. Conclusion and perspectives

During the first months In-Orbit Test (IOT) phase, CARS has already demonstrated its ability to generate DSM from data acquired by the constellation's satellites: Figure 9 represents the DSM over the "Sea of Ice" area (valley glacier located on the northern slopes of the Mont Blanc massif, in the French Alps) computed by CARS from a pair of CO3D stereo images. Figure 10 shows a DSM of Toronto, the host city of the XXV ISPRS Congress in 2026.

After several years of development, CARS is ready to generate Earth in 3D thanks to major modifications detailed in this paper. The first stable version 1.0 available at the beginning of 2026 will be tested during the qualification phase following the current IOT phase. Because it is open source, this software, already used by a large community of users and in new projects at CNES, will hopefully find new users who can count on the support of the development team.



Geoid filling on Marseille (France) DSM (sea)
Pléiades © CNES 2018, Distribution CNES PWH

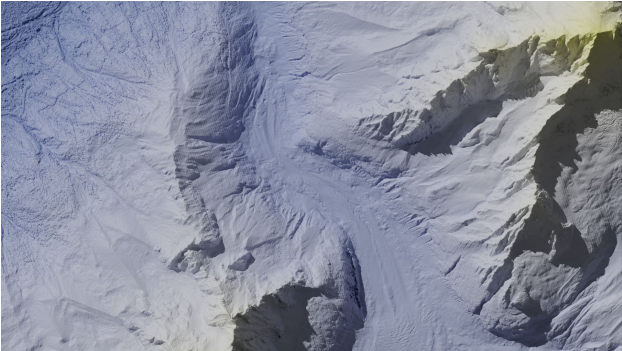


Endogenous filling on Lyon (France) DSM (river)
Pléiades © CNES 2018, Distribution CNES PWH

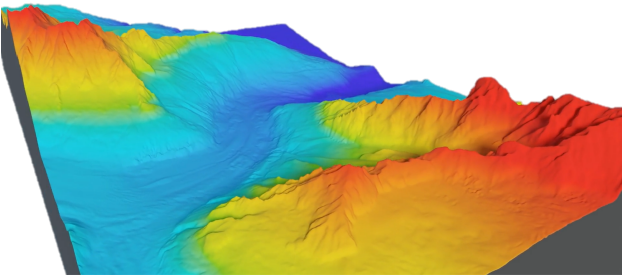


Border interpolation on Geneva (Switzerland) DSM (lake)
Pléiades © CNES 2017, Distribution CNES PWH

Figure 8. Filling method results on DSM produced from Pléiades images (classification on the left, DSM on the right).

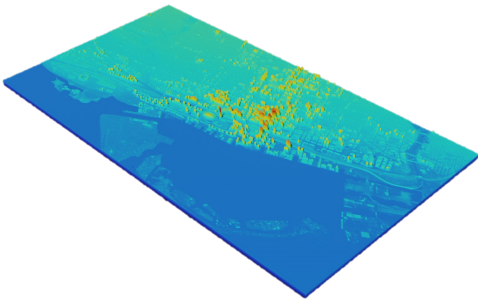


(a) Above view

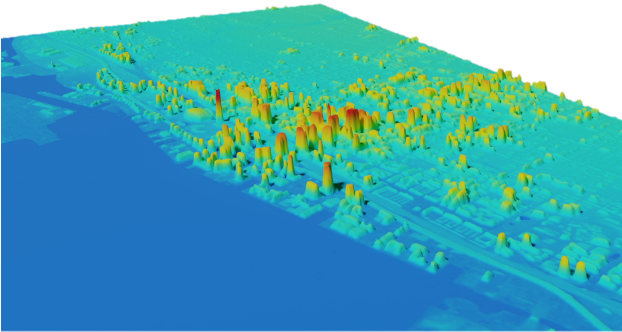


(b) Side view

Figure 9. "Sea of Ice" (France) DSM calculated by CARS from CO3D stereo images colored according to altitude.



(a) Overview of a DSM generated from a full acquisition (5x7 km)



(b) Zoomed in version

Figure 10. Toronto (Canada) DSM calculated by CARS from CO3D stereo images colored according to altitude.

References

- Beyer, R. A., Alexandrov, O., McMichael, S., 2018. The Ames Stereo Pipeline: NASA's Open Source Software for Deriving and Processing Terrain Data. *Earth and Space Science*, 5(9), 537–548.
- Billouard, C., Derksen, D., Constantin, A., Vallet, B., 2025. Tile and slide : A new framework for scaling nerf from local to global 3d earth observation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*.
- de Franchis, C., Meinhardt-Llopis, E., Michel, J., Morel, J.-M., Facciolo, G., 2014. An automatic and modular stereo pipeline for pushbroom images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, 49–56.
- Derksen, D., Izzo, D., 2021. Shadow neural radiance fields for multi-view satellite photogrammetry. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 1152–1161.
- Dumas, L., Defonte, V., Steux, Y., Sarrazin, E., 2022. Improving Pairwise DSM With SGM: A Semantic Segmentation For SGM Using An Automatically Refined Neural Network. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2022, 167–175.
- Hirschmuller, H., 2008. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 328–341.
- Kerbl, B., Kopanas, G., Leimkuehler, T., Drettakis, G., 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4).
- Krauß, T., d'Angelo, P., Schneider, M., Gstaiger, V., 2013. The Fully Automatic Optical Processing System Catena At DLR. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W1, 177–183.
- Lallement, D., Lassalle, P., Ott, Y., 2023. Bulldozer, a free open source scalable software for DTM extraction. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4/W7-2023, 89–94.
- Lebègue, L., Cazala-Hourcade, E., Languille, F., Artigues, S., Melet, O., 2020. CO3D, a worldwide one one-meter accuracy DEM for 2025. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B1-2020, 299–304.
- Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2), 91–110.
- Marí, R., Facciolo, G., Ehret, T., 2022. Sat-NeRF: Learning multi-view satellite photogrammetry with transient objects and shadow modeling using RPC cameras. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1310–1320.
- Melet, O., Youssefi, D., L'Helguen, C., Michel, J., Sarrazin, E., Languille, F., Lebègue, L., 2020. CO3D mission digital surface model production pipeline. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020, 143–148.
- Michel, J., Sarrazin, E., Youssefi, D. et al., 2020. A New Satellite Imagery Stereo Pipeline Designed For Scalability, Robustness And Performance. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020, 171–178.
- Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., Ng, R., 2021. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1), 99–106.
- Pekel, J.-F., Cottam, A., Gorelick, N., Belward, A., 2016. High-resolution mapping of global surface water and its long-term changes. *Nature*, 540.
- Qin, R., 2016. RPC Stereo Processor (Rsp) – A Software Package For Digital Surface Model And Orthophoto Generation From Satellite Stereo Imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-1, 77–82.
- Rupnik, E., Daakir, M., Pierrot Deseilligny, M., 2017. MicMac – a free, open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*, 2(1).
- Scharstein, D., Szeliski, R., Zabih, R., 2001. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, 131–140.
- Tanguy, Y., Raille, C., Buffe, F., 2024. Smart land use masks : A simple and robust approach to produce low/high vegetation masks from a single high resolution satellite image. *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, July 7-12, 2024*, IEEE, 4164–4168.
- Wang, R., Xu, S., Dong, Y., Deng, Y., Xiang, J., Lv, Z., Sun, G., Tong, X., Yang, J., 2025. MoGe-2: Accurate Monocular Geometry with Metric Scale and Sharp Details.
- Youssefi, D., Bellet, V., Steux, Y., Roux, M., Traizet, C., Rassat, M., Calendini, T., 2026. CNES/CARS: a satellite multi view stereo framework. *Zenodo*.
- Youssefi, D., Michel, J., Sarrazin, E. et al., 2020. CARS: A photogrammetry pipeline using dask graphs to construct a global 3d model. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, 453–456.
- Zabih, R., Woodfill, J., 1994. Non-parametric local transforms for computing visual correspondence. J.-O. Eklundh (ed.), *Computer Vision — ECCV '94*, Springer Berlin Heidelberg, Berlin, Heidelberg, 151–158.
- Zbontar, J., LeCun, Y., 2016. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17, 1–32.