The Journal of Open Source Software

virtughan: A Virtual Computation Cube for EO Data Using On-the-Fly Tiling Computation

Kshitij Raj Sharma [1¶] **and Upendra Oli** [1]

1 Paris Lodron University of Salzburg, Austria ¶ Corresponding author

DOI: 10.21105/joss.08078

Software

- Review
- Repository
- Archive

Editor: Rachel Wegener

Reviewers:

- @jabhalla
- @rvg296

Submitted: 09 February 2025
Published: 24 June 2026

Summary

virtughan is a Python-based geospatial data pipeline designed for on-the-fly computations on raster tiles. By leveraging Cloud-Optimized GeoTIFFs (COGs) and SpatioTemporal Asset Catalog (STAC) endpoints (*SpatioTemporal Asset Catalog (STAC) Specification*, n.d.), virtughan enables on-demand data processing across multiple zoom levels and time dimensions. virtughan focuses on on-demand tile computation, dynamically computing results as needed while minimizing data transfers and infrastructure overhead for entire scenes (*mercantile*, n.d.; *rio-tiler*, n.d.). This cost-effective approach ensures that only the necessary tiles (i.e., bounding boxes of interest) are retrieved, with computations applied directly during data access (*Sentinel-2 L2A COGs on AWS*, n.d.). The framework supports user-defined band math, multi-temporal analyses, and partial reads from cloud-hosted COG archives, along with a caching mechanism to optimize repeated requests (*FastAPI*, n.d.). The current implementation supports Sentinel-2 Level-2A imagery (queried via Element84's Earth-Search STAC API (*earth-search*, n.d.) and read directly from the AWS Open Data Sentinel-2 L2A COG bucket (*Sentinel-2 L2A COGs on AWS*, n.d.)) and Landsat 8 Collection 2 Level-2 imagery (queried via the Microsoft Planetary Computer STAC API (Microsoft, n.d.)); additional sensors are listed under Future Directions. The architecture is conceptually similar to existing STAC/COG tile services such as TiTiler (Development Seed, 2024), but virtughan emphasizes the integration of STAC querying, tile-level computation, temporal aggregation, overlap filtering, and AOI export within a lightweight Python workflow. Ultimately, virtughan provides an open-source framework for lightweight EO visualization and analysis. Rather than replacing EO data cubes, virtughan is intended to complement them in workflows where users need dynamic access to derived EO products without downloading complete scenes. This lightweight approach makes the satellite imagery processing more accessible for researchers, analysts, and developers.

Statement of Need

Big Earth Data, with its high-resolution, multi-temporal satellite imagery, poses growing challenges for storage and real-time processing, often exceeding traditional workflows' capacities (Sudmanns et al., 2020). As EO data volumes expand, efficient management strategies are needed to address rising computational and storage demands. Data cubes have emerged as a structured approach to managing large-scale EO datasets, facilitating efficient data access and analysis through precomputed storage architectures (Giuliani et al., 2019). Some data cube implementations store pre-aggregated or processed data layers, which can lead to increased storage requirements and may not support frequent or dynamic data updates effectively. In such cases, the storage of multiple processed layers can extend the storage burden, and pre-computation performed at the scene level rather than per tile may allocate resources to areas that are not always relevant to a given analysis. More recent developments, including STAC-based services and semantic, on-demand EO data cubes, address several of these limitations by deferring computation or querying multimodal data without precomputed

product layers (Kröber et al., 2025). While cloud-based EO platforms like Google Earth Engine offer scalable solutions, they require significant infrastructure, limiting accessibility. Also, downloading images within a user's area of interest is often time-consuming, as platforms like Copernicus Browser typically require downloading the entire image.

The architecture of virtughan is conceptually similar to existing STAC/COG tile services such as TiTiler (Development Seed, 2024), but it emphasizes the integration of STAC querying, temporal aggregation, overlap filtering, and AOI export within a lightweight Python workflow. To address the challenges of storage and precomputation in EO analysis, virtughan retrieves and computes only the image tiles required for a requested area of interest. By performing computations on demand rather than storing precomputed products for complete scenes, it reduces storage needs and computational overhead while enabling efficient processing across multiple zoom levels. Consequently, virtughan provides a lightweight and cost-effective framework that complements traditional EO data cubes and cloud-based processing platforms in workflows requiring dynamic access to derived EO products.

Implementation

Tile Requests and Partial Reads

A user or front-end requests map tiles via `(z, x, y)` coordinates (along with an optional date/time range and custom band math). Using mercantile's approach(*mercantile*, n.d.), virtughan determines the tile's bounding box. It then queries the configured STAC API to identify scenes covering that region: Element84's Earth-Search (*earth-search*, n.d.) for Sentinel-2 L2A scenes hosted on AWS Open Data, or the Microsoft Planetary Computer (Microsoft, n.d.) for Landsat 8 Collection 2 Level-2 scenes. Windowed reads of the underlying Cloud-Optimized GeoTIFFs then fetch only the portion of the image corresponding to the requested tile (*rio-tiler*, n.d.), with Planetary Computer asset URLs signed at request time.

On-the-Fly Computation

Once partial reads are loaded, virtughan applies user-defined formulas or filters (e.g., NDVI, cloud masking) per pixel. Because all computations occur at tile-creation time, the framework can flexibly incorporate new formulas or data corrections without reprocessing entire scenes. For requests spanning multiple acquisition dates, intersecting scenes are filtered and aggregated using the selected reduction method (e.g., median, max, or min). For requests involving bands with different native spatial resolutions, virtughan resamples bands to a common grid before computation. The current implementation does not perform radiometric harmonization between overlapping scenes acquired at different dates.

Tiling and Caching

Tile requests are handled using the standard Web Mercator tile matrix (z, x, y). The processed tiles (e.g., PNG or JPEG) can be cached. If an identical tile request recurs, virtughan serves it directly from the cache; improving performance and lowering bandwidth usage. As zoom levels shift, the system adjusts how the partial reads are resampled, ensuring minimal repeated data access. The current implementation uses a local in-memory cache maintained separately for each running process.

Download Images within Area of Interest

virtughan allows the users to download the images within their area of interest instead of downloading the whole image. The area of interest can be a polygon, rectangle, or a map window. The original Level-2 images (Sentinel-2 L2A or Landsat 8 Collection 2 L2) can be downloaded by filtering bands, date, and cloud cover. If multiple images fall within the selected time range and area, all relevant images captured at different times are retrieved. These

downloaded images can further be visualized and analyzed using tools like QGIS and other geospatial applications.

Performance Comparison

To illustrate the practical differences between workflows, we compared virtughan with the Sentinel Image Downloader QGIS plugin for generating Sentinel-2 indices (e.g., NDWI) for selected Areas of Interest (AOI).

Area of Interest	Area (km^2)	Scenes	virtughan time (s)	Workers	Sentinel Downloader time (s)	Speedup
Fewa Lake, Nepal	25.48	7	84	1	1283	15.3×
Kathmandu, Nepal	91.31	12	219	1	1744	8.0×
Salzburg, Austria	52.36	3	50	1	559	11.2×

Table 1: Workflow time required to generate a visualizable Sentinel-2 NDWI product for three AOIs. Both workflows process the same set of Sentinel-2 scenes (Scenes column). Speedup is computed as the Sentinel Downloader time divided by the virtughan time. Across these AOIs, virtughan ran 8.0–15.3× faster (mean 11.5×) using a single worker; a four-worker virtughan run further reduced the Fewa Lake time to 64 s. The Sentinel Downloader workflow requires full-scene transfer, local storage, and processing within QGIS before the requested index can be visualized, whereas virtughan retrieves and processes only the AOI tiles.

This comparison should be interpreted as a workflow-level comparison rather than a direct computational benchmark. For AOIs located in scene overlaps (e.g., Fewa Lake), the Sentinel Image Downloader workflow may additionally pull duplicate overlapping scenes from two overpasses, which further increases its transfer and processing time. In contrast, virtughan processes only the required image data for the requested area.

The evaluation was performed on a system with an AMD Ryzen 9 5900HX processor (3.30 GHz), 32 GB RAM, Windows 11, and a 15 Mbps internet connection. The reported measurements exclude authentication time (e.g., CDSE login) for the Sentinel Image Downloader workflow and polygon extraction time for both workflows.

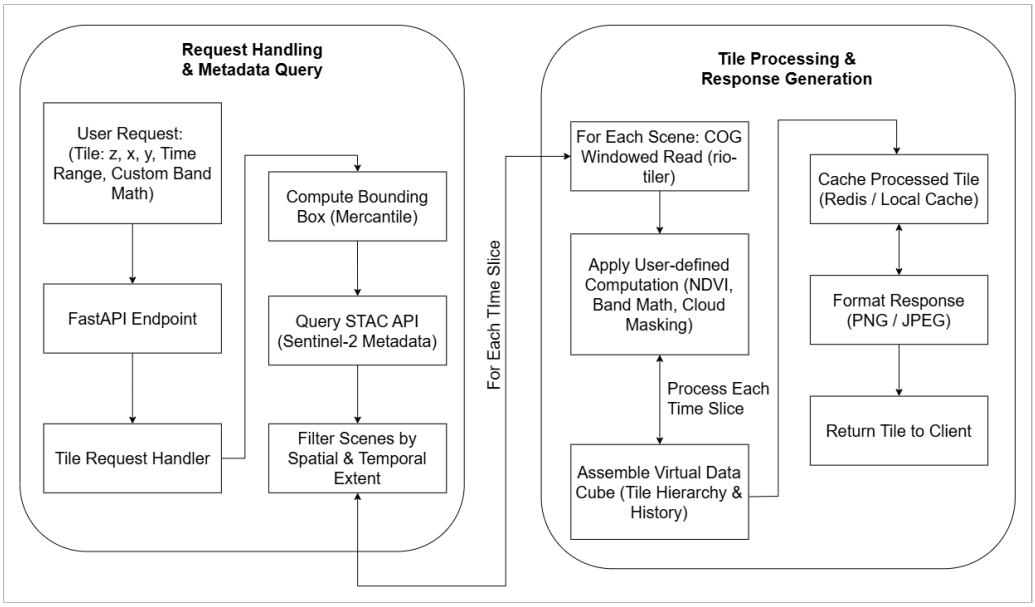

Figure 1: End-to-end flow of a virtughan tile request: a (z, x, y) tile with optional time range and band-math is resolved against the configured STAC API, rio-tiler performs COG-windowed reads of only the needed pixels per scene, user-defined computation (e.g., NDVI, cloud masking) is applied per time slice, and the assembled tile is cached and returned to the client.

Application Areas

virtughan enables on-the-fly geospatial data processing for various Earth observation applications. It helps monitor environmental changes like deforestation, glacial lake expansion, urban heat islands, and wildfires without requiring extensive data storage. In disaster response, it provides rapid analysis of floods, landslides, cyclones, and earthquakes. Urban planners can analyze land use, infrastructure growth, and air quality. AI integration on virtughan obtained datasets can support automated land classification, object detection, and biodiversity tracking. It also aids security efforts, including border monitoring and conflict damage assessment. As an open-source platform, virtughan enhances accessibility for citizen science, environmental advocacy, and academic research.

Future Directions

virtughan already enables users to retrieve and process only the necessary image tiles on demand, reducing storage and computational overhead. To further enhance its capabilities, several key improvements are planned, enabling virtughan to integrate more data sources, and support advanced analytics while maintaining its lightweight framework.

- **Mosaicking**: Automating multi-scene merges for larger coverage areas.
- **Additional Sensors**: Adding Sentinel-1 and other Landsat missions beyond the currently supported Sentinel-2 L2A and Landsat 8 Collection 2 Level-2.
- **Plugins and ML Integration**: A QGIS plugin has been developed to support AOI selection, temporal filtering, and custom band calculations. Future work will extend it with additional sensors, machine-learning inference and advanced band math.
- **Distributed Caching**: Supporting scalable deployments for high-traffic or cluster-based environments.

These future developments will increase virtughan's flexibility, efficiency, and usability, making Earth observation data processing more accessible across a wider range of applications.

Acknowledgments

We extend our gratitude to the maintainers of foundational GIS libraries such as Rasterio (*Rasterio Documentation*, n.d.), rio-tiler (*rio-tiler*, n.d.), Mercantile (*mercantile*, n.d.), and FastAPI (*FastAPI*, n.d.). Additionally, we are grateful to our colleagues and mentors at the Copernicus Masters in Digital Earth program, co-funded by the European Union and hosted at Paris Lodron University of Salzburg, for their valuable insights and support. A special thanks to Dirk Tiede and Martin Sudmanns for their guidance on existing data cube methodologies.

References

Development Seed. (2024). *TiTiler: Dynamic tile server for COG and STAC assets*. https://developmentseed.org/titiler/

earth-search: Earth-Search STAC API for Landsat and Sentinel-2 COGs. (n.d.). https://github.com/Element84/earth-search.

FastAPI: A modern, fast web framework for Python. (n.d.). https://fastapi.tiangolo.com.

Giuliani, G., Camara, G., Killough, B., & Minchin, S. (2019). Earth Observation Open Science: Enhancing Reproducible Science Using Data Cubes. *Data*. https://doi.org/10.3390/data4040147

Kröber, F., Sudmanns, M., Abad, L., & Tiede, D. (2025). On-demand, semantic EO data cubes – knowledge-based, semantic querying of multimodal data for mesoscale analyses anywhere on Earth. *ISPRS Journal of Photogrammetry and Remote Sensing*, *228*, 552–565. https://doi.org/10.1016/j.isprsjprs.2025.07.015

mercantile: A Python library for working with map tiles. (n.d.). https://github.com/mapbox/mercantile.

Microsoft. (n.d.). *Microsoft Planetary Computer STAC API*. https://planetarycomputer.microsoft.com/api/stac/v1.

Rasterio documentation. (n.d.). https://rasterio.readthedocs.io.

rio-tiler: Rasterio plugin for reading and creating mosaics. (n.d.). https://github.com/cogeotiff/rio-tiler.

Sentinel-2 L2A COGs on AWS. (n.d.). https://registry.opendata.aws/sentinel-2-l2a-cogs/.

SpatioTemporal Asset Catalog (STAC) specification. (n.d.). https://stacspec.org.

Sudmanns, M., Tiede, D., Lang, S., Bergstedt, H., Trost, G., Augustin, H., Baraldi, A., & Blaschke, T. (2020). Big Earth data: disruptive changes in Earth observation data management and analysis? *International Journal of Digital Earth*, *13*(7), 832–850. https://doi.org/10.1080/17538947.2019.1585976