University of California
Santa Barbara

Towards the twilight of file-centricity

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Environmental Science and Management

by

Niklas Fabian Griessbaum

Committee in charge:

Professor James Frew, Chair
Professor Jeff Dozier
Professor Amr El Abbadi

December 2022

The Dissertation of Niklas Fabian Griessbaum is approved.

Professor Jeff Dozier

Professor Amr El Abbadi

Professor James Frew, Committee Chair

December 2022

Towards the twilight of file-centricity

by

Niklas Fabian Griessbaum

Acknowledgements

The past 6 years have been an exhilarating ride. Never in my (adult) life have I had the opportunity to absorb as much knowledge, learn so many things, and experience so many life changes and mental growth. Besides the fantastic lectures on all sorts of topics one can attend at a university, the most important experiences have been the constant interactions with absolutely brilliant and kind-hearted people. First and foremost, I am forever thankful for the hundreds of hours I spent in the office of my advisor Frew, debating anything from language, databases, two-stroke engines, and rocket science. Considering that I came from quite a different educational background, I cannot even begin to express how thankful I am for all the patience and for all the things I learned (and for him forgiving me for all the things I refused to learn). Having to miss those weekly roundups after completing my degree certainly will leave a big hole in my life. Even though I, unfortunately, spent significantly less time with the rest of my committee members, Amr El Abbadi and Jeff Dozier, I am incredibly thankful for the unequivocally kind and helpful feedback and support they have provided.

I want to thank the good folks and my dear colleagues at Bayesics LLC, Kwo-Sen Kuo and Mike Rilee, Mike Bauer, and Dai-Hai Ton That, and at OPeNDAP, James Gallagher and Nathan Potter. The various projects we have worked together on over the past years have not only been a whole lot of fun but taught me so many invaluable things. I am incredibly thankful for their constant support and all of the knowledge they shared with me. Also, I want to thank Kuo and Mike, in particular, for giving me the opportunity to pursue a Ph.D. in triangles. Who would have figured?

Huge appreciation also goes Jeff Dozier, Ned Bair, and Timbo Stillinger for patiently explaining the beautiful colors of snow, providing ideas and feedback, and skiing with me. I particularly thank Jeff for (trying to) teach me that *heat* is not a noun.

The constant help and support from the excellent staff at Bren and UCSB made my life worryless and allowed me to focus on my work without distractions. Without fail, Steve Miley, Kristine

Duarte, Satie Airamé, Brad Hill, Mike Colee, and Aaron Martin promptly solved any problem I could possibly throw at them. I am grateful for the learning experience I had with my cohort, fellow Ph.D. students, and Professors at Bren, as well as the many students I had the chance to interact with. They sparked so many good ideas and helped me solidify my knowledge. Particularly, I would like to thank Gabriella Alberola, Prof. Mark Buntaine, Prof. Roland Geyer, and Patrick Hunnicutt for providing incredibly interesting use cases.

Possibly the most important group of people I want to thank for making my academic voyage over the last years absolutely unique are my friends and roommates. Spending endless hours debating science and life with Jason, Sam, Paul, Eva, Jonathan, Alan, Andre, Este, Will, Connor, Blaine, and my choosen family Jeff, Ishany, and Matthew has been not only a huge blast at any given time but also an incredible learning experience. Being surrounded by brilliant and kind people from all sorts of disciplines that are constantly driven to dissect facts, interpret data points, and establish and rebut theories and models is not only the best possible pastime but offered so much to learn. To the same extent, my friends back home, Eric, Malte, Alex, Franzandra, and Johannes, have been nothing but supportive over the last years and the best foothold one could wish for. Finally, my parents; my dad Dieter for always having been curious; my mom Helga for always being there for me; my son Isach for making it all worthwhile.

I deeply appreciate the various funding I received from the National Science Foundation (NSF), the National Aeronautics and Space Administration (NASA), the Earth Science Information Partners (ESIP), and Bren throughout the past year. I appreciate the funding through NSF Information and Intelligent Systems (IIS) grant 1302212[1] and from the 2018 ESIP lab, having allowed me to work on data citation, NASA Advancing Collaborative Connections for Earth System Science (ACCESS) 17 program (Award number: 80NSSC18M0118)[2] having allowed me to work on spatiotemporal indexing (and provided me with my adurious companion `schiss`), and the Bren fellowship having allowed me to focus on my research without any distractions.

[1] `https://www.nsf.gov/awardsearch/showAward?AWD_ID=1302212`

[2] `https://www.earthdata.nasa.gov/esds/competitive-programs/access/stare`

Curriculum Vitæ

Niklas Fabian Griessbaum

Education

2022	Ph.D. in Environmental Science and Management (Expected), University of California, Santa Barbara.
2012	Dipl.-Ing. in Mechanical Engineering, Karlsruhe Institute of Technology.

Publications

Michael L Rilee, Kwo-Sen Kuo, Niklas Griessbaum, et al. "A Portable Approach to Integrating Diverse Geo-Science Data Using Stare-Aware Databases and Transitioning to Cloud". In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, July 2021, pp. 2428–2431. ISBN: 978-1-6654-0369-6. DOI: 10.1109/IGARSS47720.2021.9553975. URL: https://ieeexplore.ieee.org/document/9553975/

Michael L. Rilee et al. "STARE into the future of GeoData integrative analysis". In: *Earth Science Informatics* Stanford 2017 (Jan. 2021). ISSN: 1865-0473. DOI: 10.1007/s12145-021-00568-8. URL: http://link.springer.com/10.1007/s12145-021-00568-8

Kwo-sen Kuo et al. "Towards A Moving Object Database for Geophysical Phenomenon Episodes Using STARE". in: February. 2021. ISBN: 0000000205

Michael Rilee, Niklas Griessbaum, et al. "STARE-based Integrative Analysis of Diverse Data Using Dask Parallel Programming Demo Paper". In: *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, Nov. 2020, pp. 417–420. ISBN: 9781450380195. DOI: 10.1145/3397536.3422346. URL: https://dl.acm.org/doi/10.1145/3397536.3422346

Niklas Griessbaum, James Gallagher, et al. "Solving science use cases with STARE (Demo Paper)". In: *Proceedings of ACM Sigspatial conference (SIGSPATIAL'20). ACM, New York, NY*. 2020

Michael L Rilee, Kwo-Sen Kuo, James Frew, et al. "Stare Towards Integrative Analysis with Minimized Data Wrangling Hassle". In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, Sept. 2020, pp. 901–904. ISBN: 978-1-7281-6374-1. DOI: 10.1109/IGARSS39084.2020.9323859. URL: https://ieeexplore.ieee.org/document/9323859/

Michael Rilee, Kwo-Sen Kuo, James Gallagher, et al. *STARE for scalable unification of diverse data within Earth, Space, and Planetary Science*. Dec. 2019. DOI: https://doi.org/10.1002/essoar.10501446.1. URL: https://doi.org/10.1002/essoar.10501446.1

Michael Karl Löffler and Niklas Griessbaum. "Storage devices for heat exchangers with phase change". In: *International Journal of Refrigeration* (2014). ISSN: 01407007. DOI: 10.1016/j.ijrefrig.2014.04.016

Abstract

Towards the twilight of file-centricity

by

Niklas Fabian Griessbaum

File-centricity is a paradigm in which files are the smallest unit of data. File-centricity has two significant advantages: 1) Files package data and thus allow data to be stored and distributed agnostic of their content. 2) Files provide a natural identity and even an identifier (the file-name) to data, allowing us to reference and de-reference data. However, file-centricity leaves it to the individual data user to interpret the structure of file contents and align diverse data during extract, transform, and load (ETL) processes.

My thesis is that the content-structure agnostic nature of files causes unnecessary bottlenecks in the flow from data to knowledge in environmental sciences. Unblocking those bottlenecks requires moving data processing paradigms away from file-centricity and towards data-centricity. In my dissertation, I address the "twilight of file-centricity" and technologies required to transition from file-centricity to data-centricity.

Moving towards data-centricity requires replacing files with individual observations as the smallest unit of data. In practical terms, this means storing data in a predefined schema in some form of database. However, this requires 1) the ability to identify data (rather than files). 2) data to be aligned, meaning attributes and dimensions have to be harmonized across datasets, allowing data comparison and association.

This dissertation presents solutions to these two challenges: 1) With the web service Open-source Project for a Network Data Access Protocol (OPeNDAP) Citation Creator (OCCUR), I demonstrate how data queried through OPeNDAP servers can get assigned identities that can be referenced and de-referenced. 2) The Spatio-Temporal Adaptive-Resolution Encoding

(STARE) software collection enables data-centric science. The collection contains software to spatiotemporally align data by using the universal spatiotemporal representation STARE. The collection further contains software to perform geospatial analysis and various storage backends. 3) In a science use case, I explore how spatiotemporal alignment of data can help simplify and improve environmental data science and demonstrate how analysis in a data-centric world can be carried out.

Summarizing, this thesis provides solutions to central requirements to move towards data-centricity and into the twilight of files.

Contents

Glossary

ACCESS Advancing Collaborative Connections for Earth System Science. v, 97, 101, 156

AMSR Advanced Microwave Scanning Radiometer. 78

APA American Psychological Association. 26

API Application programming interface. 11, 18, 19, 21, 22, 28, 30, 44, 45, 50, 54, 55, 62, 71, 75, 94, 99, 100

ARM Atmospheric Radiation Measurement. 26

ATBD Algorithm Theoretical Basis Document. 124, 130

ATMS Advanced Technology Microwave Sounder. 78

AVIRIS Airborne Visible / Infrared Imaging Spectrometer. 108

AWS Amazon Web Services. 79

BRDF Bidirectional Reflectance Distribution Function. 83, 87

CBMI Center for Biomedical Informatics. 27

CCCA Data Centre at the Climate Change Centre Austria. 27

CCL Connected-component labeling. 89

CRREL Cold Regions Research and Engineering Laboratory. xii, 139

CSL Citation Style Language. 21, 22, 27, 29

CSV comma-separated values. 55

CUD Create, Update, and Delete. 13, 26

CUES Cold Regions Research and Engineering Laboratory (CRREL)) and University of California, Santa Barbara (UCSB) Energy Site. 139, 148–150

DAAC Distributed Active Archive Center. 25, 36, 95, 97

DAP Data Access Protocol. 28

DAS Dataset Attribute Structure. 17, 21, 22, 29

DDS Dataset Descriptor Structure. 17

DEM Digital Elevation Model. 94

DGG Discrete Global Grid. 75

DMR Dataset Metadata Response. 28

DNB Day Night Band. 85, 87

DOI Digital Object Identifier. 11, 12, 21, 22, 25–29

EAD Envoced Archival Description. 26

ECEF Earth-centered, Earth-fixed coordinate system. 51, 94, 100

EODC Earth Observation Data Centre. 27

ESIP Earth Science Information Partners. v, 29

ETL extract, transform, and load. vii, 2–4, 37, 38, 45, 101

ETM Enhanced Thematic Mapper. 107

FAIR Findable, Accessible, Interoperable, and Reusable. 4, 9

fSCA fractional snow-covered area. 6, 104, 106, 107, 109, 113, 114, 121, 124, 127, 130, 131, 133, 137, 139, 140, 142, 143, 145–156

GIS Geographical Information System. 37, 60, 61, 65

GMI Global Precipitation Measurement. 78

GPM Global Precipitation Measurement. xiii, 78, 89

GUI Graphical User Interface. 26

HDF Hierarchical Data Format. 17, 55, 66, 67, 97

HID Hierarchical Triangular Mesh (HTM) Spatial Identifier. 48, 49

HNR Handle.Net Registry. 13

HTM Hierarchical Triangular Mesh. xiii, 5, 34, 43, 44, 48–50, 63, 67, 74, 82, 100, 104, 121

HTTP Hypertext Transfer Protocol. 17

IFOV Instantaneous Field of View. 38, 43, 56, 57, 60, 62, 63, 70, 71, 73, 75, 77–79, 84, 85, 105, 110, 112, 113, 115, 121, 123, 124, 126, 129–131, 139, 140, 142, 144–146, 149, 152, 155

IIS Information and Intelligent Systems. v

IMGERG Integrated Multi-satellitE Retrievals for Global Precipitation Measurement (GPM). 89, 90, 93

ISIN integerized sinusoidal. 39, 94–96

IUPHAR International Union of Basic and Clinical Pharmacology. 26

JDDCP Joint Declaration of Data Citation Principles. 16

JPSS Joint Polar Satellite System. 38

JSON JavaScript Object Notation. 21, 22, 29

L2G Gridded Level-2. 110

LAADS Level-1 and Atmosphere Archive & Distribution System. 95, 97

LAP Light absorbing particle. 109, 137

LSID Life-Science Identifier. 12

MAE mean absolute error. 104, 142, 143, 146, 147

MEMSCAG Multiple Endmember Snow-Covered Area and Grain Size. 108

MESMA Multiple Endmember Spectral Mixture Models. 108

MHS Microwave Humidity Sounder. 78

MLA Modern Language Association. 26

MODIS Moderate Resolution Imaging Spectroradiometer. 6, 25, 38, 39, 41, 57, 58, 60, 94, 99, 101, 104–106, 108, 110–113, 122, 124–126, 129–132, 135, 136, 139, 140, 142, 144–147, 149, 152–155, 159

MODSCAG MODIS Snow-Covered Area and Grain size. 108, 110

NASA National Aeronautics and Space Administration. v, 35, 41, 61, 78, 92, 97, 101, 156

NDSI Normalized Difference Snow Index. 106–108, 113

NDVI Normalized Difference Vegetation Index. 35, 99, 106, 108, 113, 116–118

NetCDF Network Common Data Form. 17, 55, 60, 97

NITRC Neuroimaging Informatics Tools and Resources Clearinghouse. 26

NOAA National Oceanic and Atmospheric Administration. 38

NPP National Polar-orbiting Operational Environmental Satellite System Preparatory Project. 38

NSF National Science Foundation. v, 29

OCCUR OPeNDAP Citation Creator. vii, 4, 5, 8, 17–23, 28–30, 157, 158

OPeNDAP Open-source Project for a Network Data Access Protocol. vii, xv, 4, 5, 8, 11, 17–21, 23, 28–30, 39, 95, 97, 102, 156, 157

ORNL Oak Ridge National Laboratory. 25

PEP Python Enhancement Proposal. 55

PID Persistent Identifier. 11, 15, 16, 25, 26, 30

PODS Parallel Optimized Data Stores. 67, 75, 77–79

PPS Precipitation Processing System. 78

PyPI Python Package Index. 55

QTM Quaternary Triangular Mesh. 47

RDA Research Data Alliance. 8, 10, 15, 16, 27, 30

RDBMS Relational Database Management System. 26

RDF Resource Description Framework. 25

REST Representational State Transfer. 18, 19, 21, 22, 30

RIS Research Information Systems. 10, 29

RMSE root-mean-square error. 142, 143

ROI Region of Interest. 35, 36, 38–40, 57, 67, 70, 75, 78, 80, 81, 84–86, 89, 97, 118, 122, 123, 135, 136, 139–141, 155

RTD Read the Docs. 55, 63, 65–68, 71–73, 80

SDSS Sloan Digital Sky Survey. 48, 74

SID STARE Spatial Identifier. 49–52, 54, 55, 57, 58, 60, 62, 63, 65, 67, 68, 70–75, 79, 80, 94, 97

SPIReS Snow Property Inversion from Remote Sensing. 104, 108–110, 113, 127, 130, 133, 139, 142, 143, 145, 155

SQL Structured Query Language. 55

SSMIS Special Sensor Microwave Imager/Sounder. 78, 79

STARE Spatio-Temporal Adaptive-Resolution Encoding. vii, viii, xvi, 5, 6, 34, 43–46, 49–52, 54, 55, 57, 60, 62, 63, 65–67, 70, 71, 73–76, 78–83, 85, 86, 89–95, 97, 100, 101, 121–123, 134–138, 142, 143, 145, 149, 152, 155, 156, 158, 159

SWE Snow Water Equivalent. 35, 105

SWIG Simplified Wrapper and Interface Generator. 54

SWIR Short Wave infrared. 106

TM Thematic Mapper. 106

UCSB University of California, Santa Barbara. xii, 139

UNF Universal Numerical Fingerprint. 13, 18–20, 23, 25, 30

URL Uniform Resource Locator. 13, 17–19, 21, 23, 157

URN Uniform Resource Names. 13

UUID Universally Unique Identifier. 25

VAMDC Virtual Atomic and Molecular Data Centre. 27

VIIRS Visible Infrared Imaging Radiometer Suite. 5, 38, 41, 57, 83, 85–87, 92, 101, 104–106, 123, 129, 139, 140, 145–147, 149, 152–155, 159

WCS Web Coverage Service. 4, 11, 29, 39

WGDC Working Group on Data Citation. 8, 10, 15, 16, 27, 30

WGS World Geodetic System. 50, 56, 65, 66

WKB Well-known binary. 74, 94

WKT Well-known text. 95

WMS Web Map Service. 11, 29

Chapter 1

Introduction

1.1 Motivation

Environmental informatics is the application of information technology to environmental sciences (J. E. Frew and Dozier, 2012). As such, it addresses the information infrastructure that environmental scientists leverage to obtain knowledge from environmental data.

Environmental data is traditionally collected, archived, and distributed in computer files. Alike their real-world counterpart, computer files are containers holding content (i.e., the data) and have intelligible labels (i.e., filenames). Since files are mere containers, there is no prescribed structure for their content. Reading files, therefore, requires contextual knowledge (aka metadata) to interpret what the content means.

Packaging data into files generalizes the tasks of archiving and distributing data. The discretized nature of files makes it easy to reason about their identity: We may state where a file is located, what its name is, evaluate its size, and even compute checksums. Further, files can be archived and distributed without considering the structure of the content or how the content is to be interpreted. While this simplifies the task for data repositories, it pushes the responsibility of acknowledging the data's structure to the users, who have to extract, transform, and load (ETL) data prior to extracting knowledge (Michael Lee Rilee, K.-S. Kuo, et al., 2016; Alexander S. Szalay and Blakeley, 2009).

> **My thesis is that the content-structure-agnostic nature of files causes unnecessary bottlenecks in the flow from data to knowledge in environmental sciences. Unblocking those bottlenecks requires moving data processing paradigms away from file-centricity and towards data-centricity. In my dissertation, I address the "twilight of file-centricity"[1] and technologies required to transition from file-centricity to data-centricity.**

[1] German: Dateidämmerung [d̥a'taɪ̯'dɛmə ʁʊŋ]

1.2 File-Centricity

The issue with file centricity is that it incurs redundant work and prohibits data processing at the point of storage: A system agnostic of the structure of the data it holds is incapable of performing computations on the data. It can merely act as a point of preservation and distribution. Data, therefore, has to be moved (more accurately: copied) to the point of computation (Gray's 3rd Law, (Alexander S. Szalay and Blakeley, 2009)). However, data movement is undesired since it results in uncoordinated and unstructured data duplication and storage waste[2]. But it is not merely the duplication of the data that is problematic. Moreover, every user that copies data must redundantly acknowledge the data's structure during ETL.

During ETL, users align data from various sources. Aligning data means harmonizing attributes and dimensions across datasets, allowing us to associate and compare data. In other words, alignment means that a common concept to address coincidence throughout all datasets exists. In the environmental sciences, characterized by a prevalence of spatiotemporally resolved data from observations and models, it is often of interest to associate spatiotemporally coincident data and thus to spatiotemporally align data.

However, spatiotemporal alignment is cumbersome and challenging. There is a multitude of concepts, file formats, and referencing scheme in which spatiotemporal data is stored and expressed: To name a few, locations may be conceptualized as continuous fields or as discrete features, expressed through affine transformations or as lists of coordinates, stored as projected image files or in relational databases. Time, and more so time duration, may be expressed by many calendrical formats or as offsets of epochs with or without consideration of leap sec-

[2]Further, we face an increasing disparity between network speeds and compute power. User bandwidth speeds have been following Nielsen's Law (Nielsen, 1998) and grew annually by 50 % over the last 36 years, while compute power has been following Moore's Law (Moore, 2006) and grew annually by 60 % for the last 40 years, making it less and less attractive and ultimately infeasible to move data to the point of computation as data volumes grow (Hey, Tansley, and Tolle, 2009). While researchers may choose to copy gigabytes worth of data for their analysis, copying petabytes will not be an option within the near future (A. Szalay and Gray, 2006). The predefined package size of files makes matters worse: If the package size does not exactly equal the area of interest for an analysis (A file might, for example, contain bands, areas, or periods not needed for a given analysis.), a transfer overhead is incurred (Gray et al., 2002).

onds. Consequently, spatiotemporal alignment is typically an error-prone tailormade process involving a multitude of compromises and much work.

1.3 Towards Data-Centricity

Contrary to file-centricity, data-centricity requires data to be co-aligned. Data alignment voids the necessity of (or at least simplifies) ETL and makes computation at the point of storage possible[3]. In practical terms, this means storing data in some kind of database[4].

I am addressing three questions arising in the file-twilight of environmental sciences:

How do we handle data identity in a data-centric world? Citations help to make data Findable, Accessible, Interoperable, and Reusable (FAIR) (Wilkinson et al., 2016). In less abstract terms, data citations provide identity to data, which allows referencing and de-referencing. Provision of identity is a critical challenge in the twilight of file-centric workflows since a natural addressable identity of data is lost as soon as files as a package of data are abandoned.

Technologies such as the Web Coverage Service (WCS) and the Open-source Project for a Network Data Access Protocol (OPeNDAP) (Gallagher, Potter, et al., 2007) play a vital role in the twilight of file-centricity. Their ability to seamlessly provide access to data rather than to files provides an ideal starting point for theoretical and practical excursions on how to address identity and citations in a data-centric workflow.

With the development of the web service OPeNDAP Citation Creator (OCCUR)[5] (chapter 2), I am exploring an approach for assigning identity and citations to dynamic data. OCCUR is a web

[3]The ultimate goal of data-centricity is voiding the need (or at least reducing) data movement. Data alignment is hereby crucial: Having data aligned allows for improved data sharding/placement. I.e., spatiotemporally coincidental data can be stored in physical proximity in, e.g., shared-nothing architectures.

[4]"A structured set of data held in computer storage and typically accessed or manipulated by means of specialized software." (Oxford English Dictionary, 2022).

[5]`http://occur.duckdns.org`. `https://github.com/NiklasPhabian/occur`

service that allows users to assign and store identities for data retrieved from OPeNDAP queries. OCCUR creates and stores identifiers for identities which can later be resolved through OCCUR, whereby OCCUR will verify that the data has not changed since the identity assignment. OCCUR further brokers identities by ensuring that identical data shares the same identity.

How do we spatiotemporally align data to enable data-centric environmental science? Time and space are the most prevalent (and, simultaneously, most challenging) dimensions that must be aligned in the environmental sciences. We, therefore, require a universal method to express space and time.

I am addressing data alignment with the development of the Spatio-Temporal Adaptive-Resolution Encoding (STARE) software collection (chapter 3). STARE is a spatiotemporal referencing and indexing schema that is built upon a Hierarchical Triangular Mesh (HTM) quadtree (K.-S. Kuo and Michael Lee Rilee, 2017) and provides a common concept to evaluate spatiotemporal coincidence for environmental data. The STARE software collection is a first step towards true data-centricity since it allows all datasets required for a given spatiotemporal data analysis to be stored in the same spatiotemporal (database) schema. The STARE software collection contains software to convert conventional files containing spatiotemporal data into the STARE schema and provides a variety of data-centric storage backends.

How do we perform environmental science in a data-centric world? Jim Gray's 4th Law (Hey, Tansley, and Tolle, 2009; Alexander S. Szalay and Blakeley, 2009) postulates that a data engineering challenge should be approached by determining the 20 most important questions a researcher may want a given data system to answer. Following this spirit, I solved a set of science use cases to drive the development of the STARE software collection. Chapter 3 contains two smaller undertakings: In section 3.5.1, I generate time series of night lights from Visible Infrared Imaging Radiometer Suite (VIIRS) data for a set of administrative areas to determine the characteristics of night light intensity drop and recovery during and after natural

disasters. In section 3.5.2, I track precipitation events and extract spatiotemporal incident data from various sensors. Finally, chapter 4 provides an approach to improve the accuracy of fractional snow-covered area (fSCA) retrievals. I here exploit the relatively high spatial accuracy of Moderate Resolution Imaging Spectroradiometer (MODIS) geolocations and use STARE to align irregularly spaced observations.

All three use cases demonstrate how STARE allows for integrating inhomogeneous data from various sensors at different spatial and temporal resolutions by providing a harmonized schema for time and space and thus allowing for data-centric workflows.

Chapter 2

OCCUR - An automated data citation system for OPeNDAP resources

Abstract

The Open-source Project for a Network Data Access Protocol (OPeNDAP) Citation Creator (OCCUR) is a web service that creates identifiers and citations for data served by OPeNDAP servers. As a partial implementation of the Research Data Alliance (RDA) Working Group on Data Citation (WGDC) guidelines, it addresses the need to identify arbitrary subsets of revisable datasets. OCCUR creates identifiers from a combination of an OPeNDAP query and timestamp and saves a hash of the query's result set. OCCUR can then dereference these identifiers to access the data via OPeNDAP and generate human-readable citation snippets. When accessing data via an identifier, OCCUR compares the saved hash with the hash of the retrieved data to determine whether the data has changed since it was cited. OCCUR uses CiteProc to generate citation snippets from the identifier's OPeNDAP query, timestamp, dataset-level metadata provided by the OPeNDAP server, and optionally the query result set hash.

2.1 Introduction

2.1.1 Why Data citations

(Hey, Tansley, and Tolle, 2009) coin the term *fourth paradigm* as "using computers to gain understanding from data created and stored in our electronic data stores." The fourth paradigm adds the exploration of data collected from instruments and simulations to the traditional empirical, theoretical, and computational approaches to scientific research. In this context, data collection and assembly are themselves significant research activities (J. E. Frew and Dozier, 2012).

A crucial step into the *fourth paradigm* is acknowledging data as first-class research products. As such, data must be persistently available, documented, citable, reusable, and possibly peer-reviewed (Callaghan et al., 2012; Kratz and Strasser, 2014) - a process summarized as making data "Findable, Accessible, Interoperable, and Reusable (FAIR)" (Wilkinson et al., 2016). Data citation is one of the required building blocks to achieve this goal, and widespread adoption of data citations is expected to benefit the progress of science (CODATA-ICSTI Task Group on Data Citation Standards and Practices, 2013; Data Citation Synthesis Group, 2014).

However, there is no consensus on what data publication means (Kratz and Strasser, 2014) nor on how data citation mechanisms are to be implemented (Costello, 2009). The lack of data citation standards was criticized more than a decade ago by (Altman and G. King, 2007). Years later, (Altman, Borgman, et al., 2015) and (Tenopir et al., 2011) find that even though required by publishers, researchers still too often do not make data publicly available nor cite the data consistently. There are both cultural and technical reasons for this.

(Lawrence et al., 2011) find that traditionally only conclusions are valued; little attention is given to the fitness of the data for *re-* interpretation. This reduces the motivation for data production and publishing. Even more so, (Tenopir et al., 2011) stresses that researchers may be motivated to purposely withhold data to retain their own ability to publish findings.

On the technical side, robustness, openness, and uniformity in data publication still need to be improved (Starr et al., 2015; Koltay, 2016). Cost, not so much for the storage but for curation efforts, is another reason preventing data publication (Gray et al., 2002). (Tenopir et al., 2011) states that a significant reason for data withholding is the effort required to publish data. Additionally, (Belter, 2014) finds that data citation practices are inconsistent even when used. (Assante et al., 2016) illustrates citation practices ranging from exporting a formatted citation snippet (or a generic format such as Research Information Systems (RIS) or BibTex[1]) to embedding links to the data to sharing data on social media.

To address the need for more standards and uniformity, The Research Data Alliance (RDA) Working Group on Data Citation (WGDC) released 14 recommendations to enable automized and machine-actionable identification and citation of evolving and subsettable datasets (Rauber, Asmi, Uytvanck, et al., 2015; Rauber, Asmi, Van Uytvanck, et al., 2015). (Rauber, Gößwein, et al., 2021) describes several (partial) implementations of those recommendations.

(Silvello, 2017) provides an exhaustive review of data citations' current state in terms of motivations and implementations. Based on a meta-study, the author identified six motivations for data citations: Attribution, connection, discovery, sharing, impact, and reproducibility. We simplify these to identity, attribution, and access:

Identity

Citations provide an identity to data, enabling referencing and reasoning about data (Bandrowski et al., 2016) even in its absence (e.g., no longer extant or inaccessible behind a paywall). Identifying data also allows for evaluating its usage, relevance, and impact (Honor et al., 2016).

[1] `http://www.bibtex.org/`

Attribution

Citations attribute data to authors, allowing them to take professional credit for it and providing accountability to the sponsors of the data's collection/creation and publication. This provides an incentive for sharing (Niemeyer, Smith, and Katz, 2016; Callaghan et al., 2012; Kratz and Strasser, 2014).

Access

A citation provides information on how to retrieve the cited material (e.g., the journal, year, and pages). Persistent access to data is essential to enable reusability and reproducibility (Starr et al., 2015).

2.1.2 What is data citation?

Citations provide identity, attribution, and access mechanisms to cited material. Data citations differ from citations of printed material in that the cited content (i.e., the data) may evolve and in that meta-information such as authorship or provenance may vary within a continuous dataset (Buneman, Davidson, and J. Frew, 2016). Further, data citations cannot be statically generated for subsettable data unless the number of possible subsets is trivially small. This is specifically true when data, rather than files, are accessed through, e.g., Application programming interfaces (APIs)[2]. Data citations, therefore, have to be machine-actionable, both in terms of dynamic creation (as a function of time and subsetting parameters) and in terms of resolving citations to the cited material (Assante et al., 2016; Altman, Borgman, et al., 2015; Buneman, Davidson, and J. Frew, 2016).

Data citations often use actionable Persistent Identifiers (PIDs) such as Digital Object Identifiers (DOIs). However, actionable PIDs blur the distinction between identity and access

[2]E.g. Web Map Service (WMS), Web Coverage Service (WCS), or Open-source Project for a Network Data Access Protocol (OPeNDAP) (Gallagher, Potter, et al., 2007)

(Federation of Earth Science Information Partners (ESIP), 2012). In this context, (Buneman and Silvello, 2010) emphasizes that DOIs should be considered a part of, but not a substitute for, data citations. Identity and access remain two distinct facets of a citation, and there is utility in data identity regardless of whether or not the data can be accessed or even still exists. (Parsons and Fox, 2013) criticize another aspect of DOI use in data citations: DOIs are misunderstood to provide imprimaturs and persistence. However, a DOI cannot provide persistence and should solely be understood as a locator and identifier, which is required long before an imprimatur can be issued.

In the following, we address questions related to data citation:

1. How are datasets and their subsets identified?
2. How is fixity assured?
3. How are revisable datasets handled?
4. How do citations facilitate access to data?
5. How are human-readable citation snippets/strings generated?

Answers to these questions vary widely depending on the scientific domain, as well as particular dataset characteristics such as complexity (tables, arrays, graphs), volume, update frequency, and the repository's services, specifically regarding subsetting.

Identity

A unique identity can be represented by any arbitrary unique string (CODATA-ICSTI Task Group on Data Citation Standards and Practices, 2013). In some contexts, already established identifiers such as filenames (Buneman, Davidson, and J. Frew, 2016) or accession numbers (Bandrowski et al., 2016) may serve this purpose. In practical terms, (Altman and G. King, 2007) suggests that the identity should double as a handle to the data by associating it with a naming resolution service, e.g., through the use of, e.g., a DOI, Life-Science Identifier (LSID),

Uniform Resource Names (URN), Handle.Net Registry (HNR)[3], or Uniform Resource Locator (URL).

Contrary to traditional publications, data may be queried to produce a potentially unlimited number of subsets from a single source (Davidson et al., 2017; CODATA-ICSTI Task Group on Data Citation Standards and Practices, 2013). It is, therefore, necessary to reference a dataset and every possible subset of a dataset. (Altman and G. King, 2007) coin the term "deep citation" to describe the ability to reference subsets of data. Further, data may evolve, which opens the discussion of how to identify the varying states of a dataset (Huber et al., 2015).

This further prompts the question of at which granularity a unique identity should be assigned. (Buneman and Silvello, 2010), therefore, introduce the concept of a "citable unit": An object of interest such as a fact stated in a scientific paper or a subset within a dataset.

Fixity

Datasets may change unintentionally through malfunctions and malicious manipulation or intentionally due to Create, Update, and Delete (CUD) operations. Data fixity is the property of data to remain unchanged, and fixity checking verifies that data has not changed. If fixity checking is included in a data citation system, it can verify if a data object is identical to the referenced data.

(Altman and G. King, 2007; Rauber, Asmi, Van Uytvanck, et al., 2015; Crosas, 2011) suggest including Universal Numerical Fingerprints (UNFs) into data citations to allow fixity checking.

[3] `http://www.handle.net/`

Though mentioned in most theoretical discussions about data citations[4], few data citation system implementations have addressed fixity so far.

Revisability

Datasets may evolve through updating, appending, or deletion (Klump, Huber, and Diepenbroek, 2016). We will refer to these datasets as *revisable data* to distinguish them from datasets changing due to errors or malicious manipulation.

Literature frequently intermixes the term "fixity" and the ability to cite revisable datasets. A revisable dataset is anticipated and intended to change its *state* over time. A citation system consequently has to be able to distinguish between the states of a revisable dataset (Rauber, Asmi, Van Uytvanck, et al., 2015; Klump, Huber, and Diepenbroek, 2016). However, to achieve this, merely the abstract state a citation is referencing has to remain fixed (i.e., there cannot be an ambiguity of the referenced state). This is true independently of the ability to dereference a citation to the referenced state (i.e., the actual state being fixed). Identifying and referencing a dataset's ephemeral state is required for data citation. However, the ability to persistently retrieve this state is a data publication, not a data citation challenge. It is up to the publisher to choose an apt level of zeal:

- **Pessimistic:** data is assumed to be ephemeral; consequently, citations cannot ever be dereferenced.
- **Optimistic:** data is assumed to be fixed. Citations always dereference to the current state of the data.

[4](Pasquetto, 2020; Schubert, Seyerl, and Sack, 2019; ESIP Data Preservation and Stewardship Committee, 2019; Davidson et al., 2017; Silvello, 2017; Alawini2017; Buneman, Davidson, and J. Frew, 2016; Prakash et al., 2016; Altman, Borgman, et al., 2015; Starr et al., 2015; Rauber, Asmi, Uytvanck, et al., 2015; Ball and Duke, 2015; Huber et al., 2015; Rauber, Asmi, Van Uytvanck, et al., 2015; Kratz and Strasser, 2014; Pröll and Rauber, 2013; Bechhofer et al., 2013; Callaghan et al., 2012; CODATA-ICSTI Task Group on Data Citation Standards and Practices, 2013; Federation of Earth Science Information Partners (ESIP), 2012; Lawrence et al., 2011; Crosas, 2011; J. Frew, Janée, and Slaughter, 2011; Buneman and Silvello, 2010; Costello, 2009; J. Frew, Metzger, and Slaughter, 2008; Altman and G. King, 2007)

- **Opportunistic:** data is assumed to remain fixed for some time. Citations can be dereferenced only until the data changes.
- **Pedantic:** every data state is saved; consequently, citations can always be dereferenced to the referenced state.

Versioning can be used to identify states of revisable data. We acknowledge that the term *version* is often used synonymously with the term *state* (of a dataset or a single record). However, in the following, we will use the term *version* as a policy-prescribed reference to a state.

A concern in data versioning is how to reach "[a] consensus about when changes to a dataset should cause it to be considered a different dataset altogether rather than a new version."[5] This question is futile from a pure identity perspective since every state needs to be identifiable. The distinguishing between version versus state, therefore, is mainly connected to the nature of PIDs (and the costs associated with minting them) (Klump, Huber, and Diepenbroek, 2016), as well as the notion of hierarchical association and provenance.

As mentioned above, it is open to debate whether reproducibility is a hard requirement for data citations of a revisable dataset. If not, resolving citations could be deprecated altogether (pessimistic) or only allowed until the data has changed (opportunistic). The opportunistic approach could, e.g., be implemented by timestamping modifications (e.g., by comparing the file systems' last modification date) or through fixity checking. The RDA WGDC (Rauber, Asmi, Van Uytvanck, et al., 2015) elaborates on this approach: A dataset (or a subset) should be given a new identity when the data has changed since the last time the dataset (or subset) was requested. This is recommended to be implemented using a normalized query store and checksums (Ball and Duke, 2015).

[5] `https://www.w3.org/TR/dwbp/#dataVersioning`

Access

There is a common agreement that data citations should use actionable PIDs as an access mechanism. E.g., (Altman and G. King, 2007) suggests a citation to contain an identifier that can be resolved to a landing page (not to the data itself), a requirement also specified by the Joint Declaration of Data Citation Principles (JDDCP) (Data Citation Synthesis Group, 2014; Altman, Borgman, et al., 2015). The landing page, in turn, should contain a link to the data resource. The advantage is that the identifier can be resolved regardless of whether the data is behind a paywall or does not exist anymore (at all or in the referenced state).

The question of data access is connected to reproducibility and on what granularity level changing states of a revisable dataset should be stored.

Citation texts

Data citation systems should use metadata standards (CODATA-ICSTI Task Group on Data Citation Standards and Practices, 2013) and be capable of generating human-readable citation snippets to facilitate the use of data citations and lower the boundaries for data citation (Buneman, Davidson, and J. Frew, 2016; Rauber, Asmi, Van Uytvanck, et al., 2015). An advantage of citation texts, including, e.g., title, author, and date, is that they allow the reader to quickly assess the relevance, quality, and concurrency of the cited material (Buneman and Silvello, 2010). The ability to automatically create citation texts is also recommendation 11 for data citations of the RDA WGDC.

2.2 OCCUR Implementation

The OPeNDAP Citation Creator (OCCUR) enables automated citation creation and dereferencing for data served by OPeNDAP servers. It is implemented as a web service that allows users to:

1. Assign and store identities to data
2. Create identifiers for identities
3. Resolve identifiers while verifying that the data has not changed since the creation of the identifier.
4. Broker identities, i.e., ensuring that identical data shares the same identity.
5. Generate formatted citation snippets for OCCUR identifiers and any arbitrary OPeNDAP query.

OCCUR is built as a third-party web service without needing to modify data repositories.

2.2.1 OPeNDAP Primer

The OPeNDAP[6] simplifies access to remote data. It is widely used (but not limited to) to access remote Hierarchical Data Format (HDF) and Network Common Data Form (NetCDF) files. An OPeNDAP server allows an OPeNDAP client to request the structure and metadata of data and query subsets of the data. The client retrieves those requests through Hypertext Transfer Protocol (HTTP) `GET` requests to an OPeNDAP URL served by the OPeNDAP server. The structure and metadata are retrieved by requesting the Dataset Descriptor Structure (DDS) and the Dataset Attribute Structure (DAS) of the data. The former describes the shape of the data (such as the dimensions), while the latter contains metadata populated by the data provider. Queries for data subsetting are specified through constraint expressions in the URL query strings.

[6]here, referring to the protocol, not to the company of the same name developing the protocol

2.2.2 Retrieving identities

In OCCUR, data identity is described by:

a) the query (i.e., the full OPeNDAP URL) that produced the data, and

b) a UNF of the result set of the query (i.e., the data).

Though not strictly necessary, the time of the identity creation (which serves as a proxy for the query execution time) is used as a third attribute to describe an identity to increase convenience and human readability.

An identity's identifier is the concatenation of the OPeNDAP URL and the identity creation timestamp. (See class diagram in Figure 2.1). Identities and their corresponding identifiers are permanently stored within OCCUR upon user request.

A user can retrieve a data identifier by submitting the OPeNDAP URL used to produce the data to OCCUR with the following Representational State Transfer (REST) API call:

```
GET $OCCUR_HOST/store/?dap_url=$DAP_URL
```

When a user requests to retrieve the identity of data, OCCUR first fetches the queried data from the OPeNDAP server and creates a UNF (by default, an MD5[7] hash) of the current state of the data. OCCUR will then verify if an identity to the same query has already been stored. If not, OCCUR creates and stores a new identity (consisting of the query, the UNF, and the current timestamp). The user is then provided with the newly created identifier.

If one or more identifiers to the same OPeNDAP URL already have been stored, OCCUR compares the UNF of the current data state to the UNFs of the stored identities. If an identity with the identical query and UNF is found, it is assumed that the data is currently in the same

[7]MD5 is a cryptographically broken algorithm. However, we here utilize it merely to verify data integrity (i.e., to identify unintentional corruption). The hash function may be replaced in the future, e.g., by SHA-256, to verify against malicious data corruption.

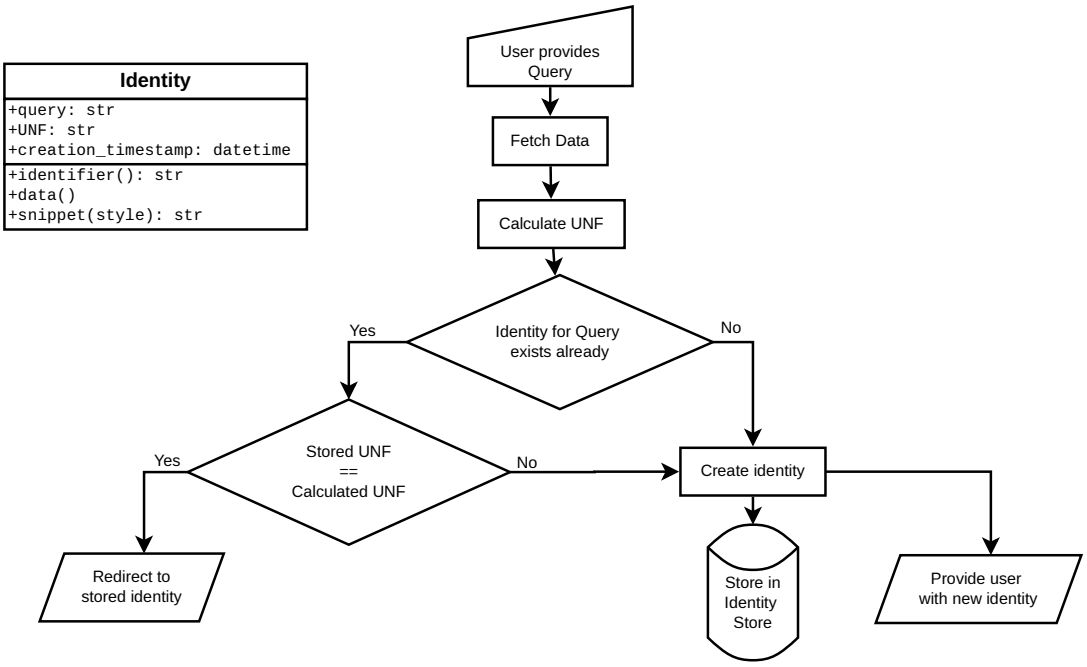


Figure 2.1: Class diagram of the data identity and flow for retrieving an identity.

state as when this stored identity was created. Their identity is, therefore, identical, and the user is provided with the identifier of this stored identity.

If the UNF of the current state differs from all stored UNFs to the same OPeNDAP URL, a new identity is created, and the user is provided with the newly created identifier. Figure 2.1 schematically illustrates the flow for retrieving an identity.

2.2.3 Dereferencing identities

OCCUR follows an opportunistic approach for dereferencing identities: An identifier can only be dereferenced during the time interval between identity creation and the time the state of the referenced data changes. A user can dereference an identifier by submitting the following REST API call:

```
GET $HOST/dereference/?identifier=$IDENTIFIER
```

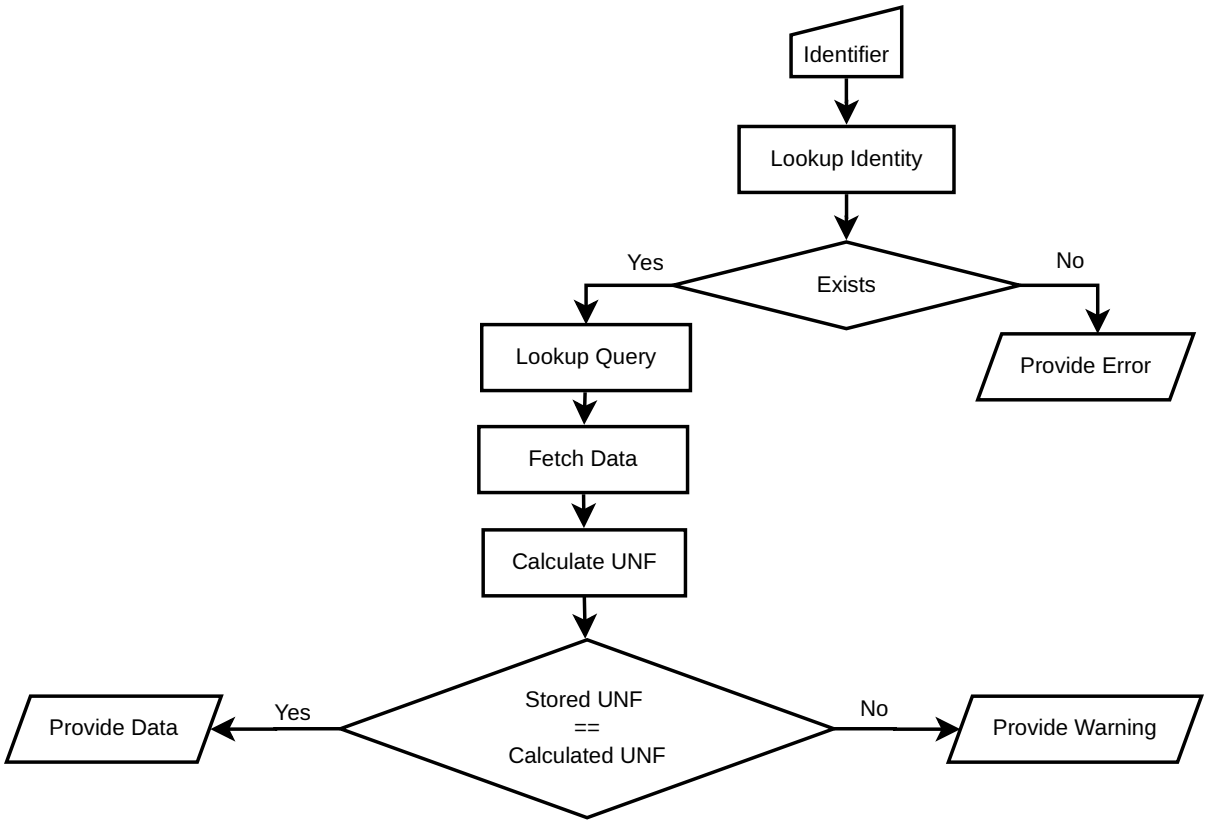


Figure 2.2: Flow for the dereferencing an identity.

OCCUR first verifies if an identity for the submitted identifier exists/is stored. If so, OCCUR will look up the associated OPeNDAP query and UNF. OCCUR then executes the stored query to retrieve the current state of the data. For this state, the UNF is calculated and compared to the stored UNF. If the UNFs match, it can be assumed that the data is in the same state as it was during the identity creation. The user thus can be redirected to the (landing page of the) data.

If the UNFs don't match, the current state can be assumed to differ from the state of the data at the identity creation. The user, thus, will be redirected to a landing page containing a warning indicating that the referenced state of the data is not accessible anymore. It then is up to the user whether or not to retrieve the data in its current state. The flow of the dereferencing is illustrated in figure 2.2.

2.2.4 Formatting citations

OCCUR allows the creation of human-readable formatted citation snippets from both OCCUR identifiers and plain OPeNDAP URLs. This service can be understood as the OPeNDAP pendant to `www.crosscite.org`, which allows the creation of citation snippets from DOIs.

A user specifies the (e.g., journal) formatting style[8] together with the identifier or the OPeNDAP URL. Besides human-readable snippets, a user can choose to retrieve the raw bibliographic information in BibTeX or Citation Style Language (CSL) - JavaScript Object Notation (JSON) format. In its simplest form, a user can request a formatted citation snippet with the following REST API calls:

To format an identifier:

```
GET $HOST/format/identifier=$IDENTIFIER&style=$STYLE
```

Or, for a plain OPeNDAP URL:

```
GET $HOST/format/dap_url=$DAP_URL&style=$STYLE
```

For the first case, OCCUR will first verify if an identity with the specified identifier exists. If it exists, OCCUR will look up the according OPeNDAP query and derive the URL of the according DAS. The DAS is then fetched, and OCCUR will extract metadata from its "global" section[9]. OCCUR will hereby extract any legal citeproc-py keyword (see appendix 2.4.2). This metadata is converted into the citeproc-csl JSON format. In case the DAS contains a DOI, OCCUR will query `https://www.doi.org` for the CSL JSON representation of this DOI[10] to retrieve additional metadata.

[8] OCCUR supports any of the formats defined in the official repository for CSL citation styles `https://github.com/citation-style-language/styles`.

[9] the part that satisfies: `/(? <= global.* (?=)/*ims*`

[10] OCCUR uses content negotiation to query doi.org with the header `Accept: application/vnd.citationstyles.csl+json` to retrieve a CSL-JSON metadata response

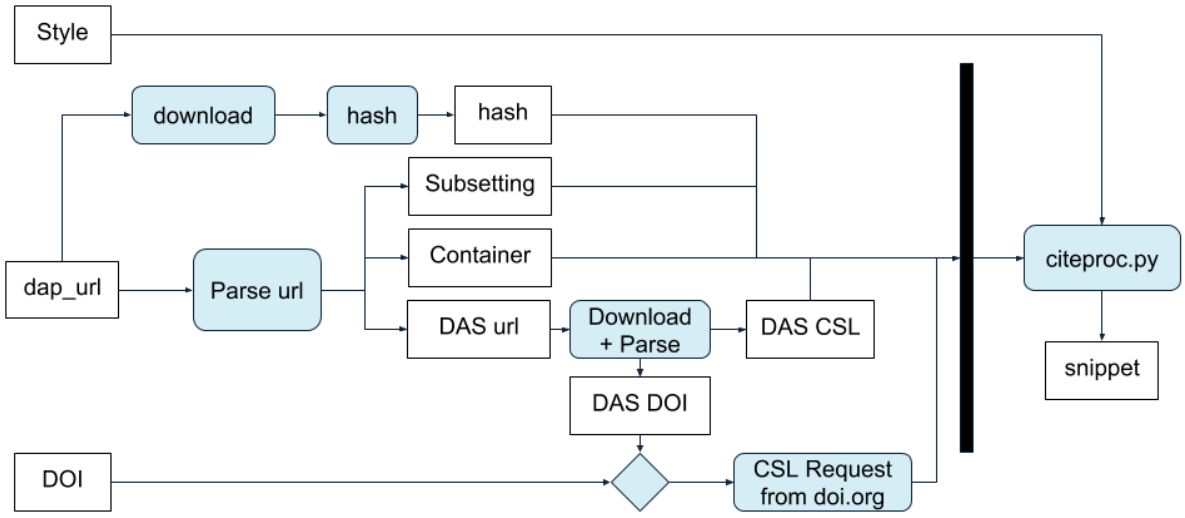


Figure 2.3: Formatting of citations.

The CSL JSON of the DAS and the CSL JSON of the DOI are then merged. Together with the style definition, they are fed into citeproc[11] to produce a formatted citation snippet, which is then served to the user.

In case the DAS does not contain a DOI or the user wants to overwrite the DOI specified in the DAS, the user can choose to add a DOI to the snippet request, i.e., with the following REST API call:

```
GET $HOST/format/identifier=$IDENTIFIER&style=$STYLE&doi=$DOI
```

The flow for creating the citation snippet is illustrated in figure 2.3.

2.2.5 Web frontend and landing pages

OCCUR includes a web API: The web-service endpoints default to resolve to landing pages rather than to the identifiers or data itself. The reference/identifier endpoint resolves to a

[11] OCCUR uses citeproc-py (`https://github.com/brechtm/citeproc-py`)

landing page that allows the user to inspect the stored identity (specified by the URL, the UNF, and identity creation timestamp), access the referenced data, as well as to format the identity to a human-readable citation snippet.

Seminally, the dereference/data endpoint resolves to a landing page of the data rather than the data itself. Those landing pages provide the user with links to the identifier landing page and the referenced data. If the referenced state of the data is no longer available, the landing page additionally contains a warning informing the user that the data has changed. The user may now choose to access the newer state of the data or create a new identity.

2.2.6 Use Example

A schematic timeline for using OCCUR might look as follows: User A queried data from an OPeNDAP server. The user now wants to create an identifier for this data to include in a reference. They thus take the OPeNDAP URL used to query the data to OCCUR and request an identifier. OCCUR resolves the OPeNDAP URL, fetches the data, computes the UNF, and stores it together with the OPeNDAP URL and the current timestamp as a new identity. The user then is provided with the identities' identifier. They then may use this identifier to create a human-readable citation snippet. Later, another user, B, may receive the reference (e.g. from a publication), including the OCCUR identifier, and uses OCCUR to dereference it to the referenced data. OCCUR will first verify if the referenced state is still available. If so, the user is provided with the data. If not, the user is given a warning. They now have the option to either access the newer state of the data or create a new identity.

Both the retrieval of an identity and the dereferencing of the identifier are illustrated in figure 2.4.

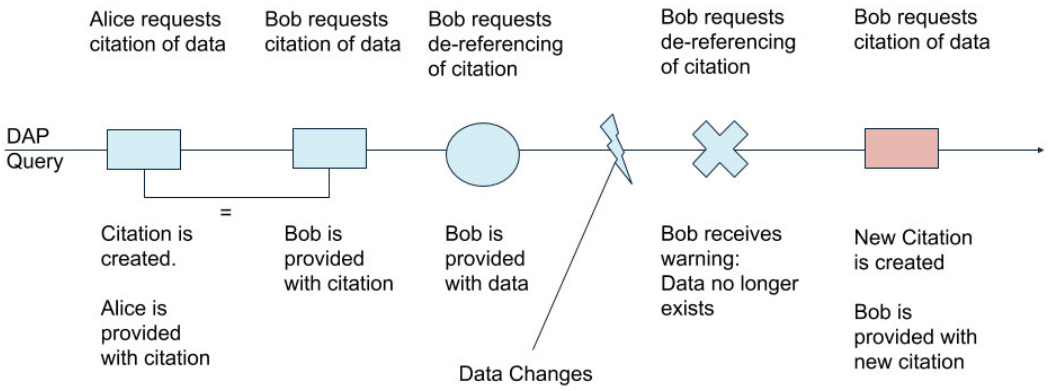


Figure 2.4: Timeline of identity creation and retrieval.

2.3 Related work

The following is a non-comprehensive summary of services for automized data citation creation that have been presented in the past:

MatDB[12] implements a data publication and citation service for engineering materials. Datasets are made citable by enforcing minimal discipline-specific metadata and minting DataCite DOIs. Fixity is assured by snapshotting the dataset at the time of DOI minting. Revisability of data is made possible through policy-enforced versioning (Austin, 2016).

(Alawini et al., 2017) created a citation service for the Resource Description Framework (RDF) eagle-i database. Since this database itself does not version its data (only the most recent version is available), the authors implemented an external service that versions eagle-i data to provide users access to revised data. The service tracks and stores every change in the original dataset. The authors note that this approach is viable for eagle-i since the dataset changes very slowly.

The dataverse networks software (Crosas, 2011) aggregates data in "studies." Studies may contain several datasets, and each study shares a common persistent identifier. A citation to a dataset (and subsets) is implemented as the combination of the studies' PID appended with the UNF of the cited data.

(Cook et al., 2016) presents the data product citations at a Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center (DAAC). The DAAC assigns a DOI per dataset, which may contain between one and tens of thousands of files. A single file within a dataset can be identified by appending the file's Universally Unique Identifier (UUID) to the DOI (using the `urlappend` functionality of the DOI resolver). The DAAC also provides a Moderate Resolution Imaging Spectroradiometer (MODIS) subsetting service. The user can request a citation to the subset, comprised of the dataset's citation appended with a textual description of the temporal

[12] `http://doi.org/10.17616/R3J917`

and/or spatial subsetting. The citation can be requested as a formatted snippet and in BibTex format.

A very similar approach is implemented for the Atmospheric Radiation Measurement (ARM) Data Archive (Prakash et al., 2016). Upon data order fulfillment, the user is provided with both the data and a citation that contains a citation, including a textual description of the temporal and/or spatial subsetting. The ARM Data Archive also hosts a citation creator, a Graphical User Interface (GUI), that allows the creation of a data citation subject to a fully qualified dataset stream name and optionally manually specified subsetting parameters. The user hereby can choose between the custom ARM citation style, American Psychological Association (APA), Modern Language Association (MLA), and Chicago.

(Honor et al., 2016) describes a reference implementation for data citations of a database holding neuroimaging (the specific use case is the Neuroimaging Informatics Tools and Resources Clearinghouse (NITRC)). The citable units for their use-case are individual images aggregated in studies/projects. Upon data upload, hierarchically, each image and each study is assigned a DOI. The authors recognize that this implementation may result in the generation of many DOIs; however, they evaluate the solution feasible for their use case.

(Pröll and Rauber, 2013) present a reference implementation for data citations to data managed by a Relational Database Management System (RDBMS). The system is based on the premise that a timestamped SELECT query can correctly identify data. To allow revisability, the system timestamps every CUD operation and acknowledges the validity time ranges of records rather than allowing modification of records. When a user wants to create a citable subset, the system will store the according timestamped `SELECT` query, calculate a hash for the result set, and assign a PID to the query.

(Buneman and Silvello, 2010) describes an approach for citing digital archives described by the Envoced Archival Description (EAD) and for the International Union of Basic and Clinical Pharmacology (IUPHAR) database. They hereby focus on the concept of citable units and

their application on hierarchical/structured and revisable datasets.

(Rauber, Gößwein, et al., 2021) details the experiences of several adapters of the 14 RDA WGDC recommendations. Among those are the Center for Biomedical Informatics (CBMI), Virtual Atomic and Molecular Data Centre (VAMDC), Data Centre at the Climate Change Centre Austria (CCCA), and the Earth Observation Data Centre (EODC). All adapters were able to improve their data distribution infrastructure through the adaptations of the recommendations. However, each adapter's individual implementations and the required work vastly varied. A common challenge can be identified as enabling data versioning and timestamping.

Apart from those fully integrated data citation systems, we want to mention the crosscite DOI Citation formatter[13]. The crosscite DOI Citation formatter generates citation snippets from metadata retrieved from DOI landing pages. The citation snippets are formatted through citeproc subject to styles defined through the CSL[14].

[13] `https://citation.crosscite.org/`

[14] `https://citationstyles.org/`

2.4 Discussion and Outlook

We demonstrated how a third-party data citation system can be built for data accessible through the RESTful data access protocol OPeNDAP. We define a *citeable unit* as any result set of a `SELECT/GET` request. Identities and the corresponding identifiers to citable units can be created and retrieved upon user request. The identities are defined and permanently stored as a combination of the query, the result set's hash, and the identity creation timestamp. Identifiers can be opportunistically dereferenced as long as the referenced identity is still available. The availability is verified by comparing the hash of the referenced identity and the hash of the query result set at the time of dereferencing. It is hereby irrelevant if the data changed intentionally subject to a revision or unintentionally due to rot or malicious manipulation. It shall be noted that the used hashes stay opaque to the user as OCCUR handles the result set verification internally.

OPeNDAP allows users to request data in a delivery container format that differs from the storage container format. OCCUR calculates hashes in the delivery container rather than in the storage format.

OCCUR requires fetching the complete result set to calculate the hashes, which may inflict high transfers on both OCCUR and the OPeNDAP resource. To avoid these high transfers, hashes of result sets should be calculated at the point of storage and exposed/queried through an API. Data Access Protocol (DAP) - 4 already provides the option of including CRC32 checksums in the Dataset Metadata Response (DMR)[15], which could be used for this purpose.

Identities and identifiers are created, stored, and resolved within OCCUR. This comes with the advantage of low cost. However, these identifiers cannot be considered persistent. It is necessary to use an external service such as DOI, identifiers.org[16], or name2thing[17] to mint persistent identifiers to OCCUR identities.

[15] `https://docs.opendap.org/index.php/DAP4:_Specification_Volume_1#Checksums`
[16] `https://identifiers.org/`
[17] `https://n2t.net/`

OCCUR identifiers can be converted into human-readable citation snippets as well as machine-readable bibliographic entries in BibTex, RIS, and CSL - JSON format. OCCUR hereby exploits that OPeNDAP resources have a designated location for dataset-level metadata: the DAS. The metadata extracted from the DAS is combined with the hash, the OPeNDAP query literal, and the identity creation time to create snippets through citeproc. We recognize that the DAS might be a duplicate metadata location since a dataset-level DOI may have already been registered in many cases. In these cases, we encourage merely the inclusion of the dataset DOI into the DAS. OCCUR will parse this DOI and resolve it for the corresponding metadata.

OCCUR is meant to be a demonstration that can be adapted to other RESTful data access services such as WMS and WCS. It could be envisaged to remain a centralized third-party system or be integrated into the data services.

Acknowledgments

The development of OCCUR was supported by an National Science Foundation (NSF) grant (Award Number: 1302212)[18] and the Earth Science Information Partners (ESIP) federation as a 2018 ESIP lab project. We would like to thank for the feedback of we received at the 2017 Bren Ph.D. Symposium and the 2018 ESIP summer meeting; particularly James Gallagher, Ted Habermann, and Mark Parsons.

Competing interests

The authors have no competing interests to declare.

[18] `https://www.nsf.gov/awardsearch/showAward?AWD_ID=1302212`

Appendix

2.4.1 RDA WGDC recommendation compliance

OCCUR implements several of the 14 RDA WGDC recommendations (Rauber, Asmi, Van Uytvanck, et al., 2015). It stores queries with metadata, PID, and timestamps (R7, R8, R9). It also uses UNFs to achieve result set verification (R6). Since OPeNDAP normalizes queries by alphabetically sorting the constraint expressions, OCCUR can also fulfill the recommendation to ensure query uniqueness and that a single subset is not referred to by more than one identifier (R4). Further, when accessing files through OPeNDAP, a stable sorting of the result sets can be guaranteed (R5). OCCUR is machine actionable through its REST API (R12) and resolves to human-readable landing pages (R11), both of which allow the creation of formatted citation snippets (R10). Since OCCUR is a third-party service, it does not address data versioning (R1) and timestamping (R2), as we see those as the repositories' responsibility. Additionally, a technology migration (R13) and migration verification (R14) will only be possible to be carried out in collaboration with the according repository.

Figure 2.5 evaluates the compliance of the current state of OCCUR with these recommendations.

R1	Data Versioning	Repository responsibility
R2	Timestamping	Repository responsibility
R3	Query Store Facilities	Yes
R4	Query Uniqueness	Not yet
R5	Stable Sorting	Not yet
R6	Result Set Verification	Yes
R7	Query Timestamping	Yes
R8	Query PID	Yes
R9	Store Query	Yes
R10	Automated Citation Texts	Yes
R11	Landing Page	Yes
R12	Machine Actionability	Yes
R13	Technology Migration	Not yet (Only manually so far)
R14	Migration Verification	Not yet

Figure 2.5: Evaluation of WGDC compliance.

2.4.2 Legal citeproc keywords

```
NAMES = ['author', 'collection_editor', 'composer',
         'container_author', 'editor', 'editorial_director',
         'illustrator', 'interviewer', 'original_author',
         'recipient', 'translator']
DATES = ['accessed', 'container', 'event_date', 'issued',
         'original_date', 'submitted']
NUMBERS = ['chapter_number', 'collection_number',
           'edition', 'issue', 'number', 'number_of_pages',
           'number_of_volumes', 'volume']
VARIABLES = ['abstract', 'annote', 'archive',
             'archive_location', 'archive_place',
             'authority', 'call_number', 'citation_label',
             'citation_number', 'collection_title',
             'container_title', 'container_title_short',
             'dimensions', 'DOI', 'event', 'event_place',
             'first_reference_note_number', 'genre',
             'ISBN', 'ISSN', 'jurisdiction', 'keyword',
             'language', 'locator', 'medium', 'note',
             'original_publisher', 'original_publisher_place',
             'original_title', 'page', 'page_first', 'PMCID',
             'PMID', 'publisher', 'publisher_place',
             'references', 'section', 'source', 'status',
             'title', 'title_short', 'URL', 'version',
             'year_suffix']
```

Chapter 3

A Software Collection to enable STARE-based geospatial analysis of remote sensing data

Abstract

Geospatial analysis is predicated on the ability to evaluate geospatial coincidence between georeferenced objects. The sheer volume of remotely sensed data and their irregular spacing are a disabling roadblock for scientists, currently only circumventable by spatiotemporal discretization and sampling of observations. While spatial discretization simplifies the evaluation of geospatial coincidence, it decreases the data fidelity. The alternative geospatial referencing and indexing schema, the Spatio-Temporal Adaptive-Resolution Encoding (STARE), built on top of a Hierarchical Triangular Mesh (HTM), allows performant spatial coincidence evaluation of undiscretized observation. We present the software collection built around STARE that enables scientists to process remote sensing data at the actual sensor geolocation accuracy and resolution. It contains methods to read conventional geospatial data and to convert legacy representation into STARE representation. It further contains STARE based geoprocessing methods and storage backends for STARE indexed data.

3.1 Introduction

Geospatial data analysis extends conventional data analysis by introducing the special attribute of geolocation. This allows us to associate geolocalized observations with other geospatial objects and data, providing insights into phenomena that would otherwise appear unrelated. In the historic textbook example, Dr. John Snow graphically superimposed the locations of cholera clusters with the locations of water wells to identify the source of an 1854 cholera outbreak in London. In contemporary uses, we might associate whale injuries with shipping lanes, car accidents with road conditions, blackout locations with local public policies, Normalized Difference Vegetation Index (NDVI) measurements with plots of land, or estimations of Snow Water Equivalent (SWE) with watersheds. The ability to spatially associate datasets is predicated on the ability to evaluate spatial relations. We want to associate data that fulfill some spatial criteria, e.g., within a distance, intersecting, or containing. As humans, we use spatial intuition to evaluate the relations of spatial objects: we may say: "Santa Barbara is in California," enabling us to associate data located in Santa Barbara with California. Computationally, we represent locations as geometric objects in a coordinate system, allowing us to compute spatial relations of arbitrary spatial objects.

At the beginning of any spatial analysis, we use the ability to evaluate coincidence to find and extract relevant data: We define a spatiotemporal Region of Interest (ROI) and use this definition in a search query to extract data that (for example) intersects our ROI. We then convert the extracted data to spatial objects convenient for analysis and apply spatial (and other) operators to these objects.

Unfortunately, these steps are seldom seamless in real-world analysis. In the 1992 sci-fi novel *Snow Crash* (Stephenson, 1992), Neal Stephenson envisages an information system called *CIC Earth*, a central user interface to access "every bit of [spatial] data". There are some attempts to mimic this concept in, e.g., multiple virtual globe or Digital Earth projects (e.g., Google Earth, National Aeronautics and Space Administration (NASA) WorldWind, Microsoft TerraServer,

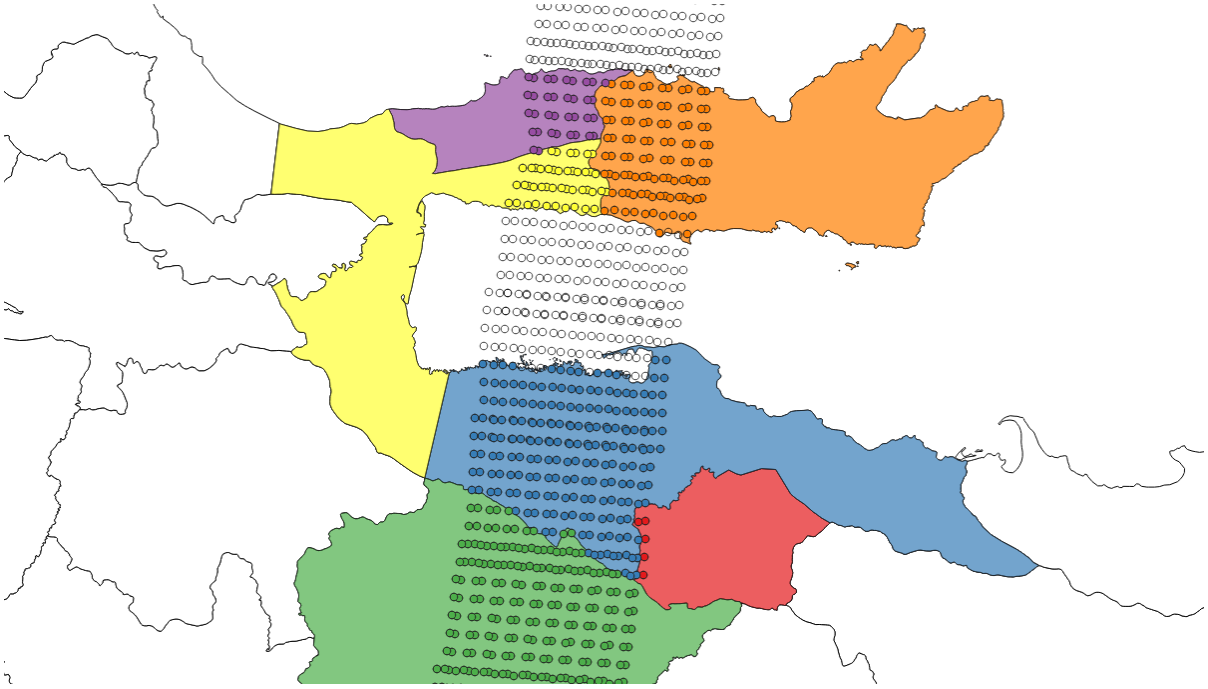


Figure 3.1: Observations (circles) associated with political boundaries

VGIS (Faust et al., 2000)), as well as in data repository interfaces such as NASA's Distributed Active Archive Centers (DAACs). In reality, there is no single querying and access mechanism for *all* spatial data. Instead, geospatial data is distributed by many actors across many repositories. Since no universal method exists to represent locations (specifically for locations having a spatial extent), repositories use idiosyncratic mechanisms for spatial data discovery and extraction. Some repositories may allow querying canonical or standardized place names (such as countries and states or survey site identifiers). Others may allow specifying the spatial ROI as rings, bounding boxes, or even polygons. Things are more complicated if the spatial ROI is dynamic, i.e., changing over time. Examples of evolving spatial ROIs are events such as storms or wildfires.

Since there is no unified representation of data in general and geolocation in particular, repositories will deliver data in many possible container formats (*files*) and geospatial representations. Locations may be expressed in spherical or ellipsoidal geographic coordinate systems or in two-

dimensional projected (Cartesian) coordinate systems at varying resolutions.

Consequently, users must step through an iterative process called extract, transform, and load (ETL) rather than simply searching for, retrieving, associating, and analyzing data. During ETL, data first has to be discovered, then extracted, subsequently transformed into a unified ("harmonized") representation, and finally loaded into a system in which the analysis operations can be carried out (i.e., a desktop Geographical Information System (GIS), such as QGIS[1], ArcGIS[2]; a database, such as PostGIS[3]; or objects in a programming language, such as MATLAB, Python, or R). The harmonization process is typically time-consuming, computationally expensive[4], and bespoke to a particular analysis environment. The analyst has to balance computational performance and data fidelity: An appropriate geographic or projected coordinate system, as well as a spatiotemporal resolution, has to be chosen, both of which have an implication on the preservation of spatial properties (areas, distances, shapes) and computational performance. While any commonly used geospatial relation test can theoretically be computed in both geographic and projected coordinate systems, operations are cheaper and thus more performant in projected 2D coordinate systems. Further, none of the commonly used desktop GISs (ArcGIS, QGIS) or programming libraries (GeoPandas[5] or r-spatial's[6] `sf`) supports spherical or ellipsoidal computations. Only within recent years technologies that gracefully allow performing geospatial analysis in geographic coordinate systems (S2geometries[7], PostGIS geographies[8], SphereGIS[9]) have become available. Thus, geographic information systems and their users will typically harmonize data by projecting all relevant data into a single (locally) appropriate coordinate system. Note that there may not actually be a single locally appropriate

[1] `https://www.qgis.org/`

[2] `https://www.arcgis.com/`

[3] `https://postgis.net/`

[4] Anecdotally, consuming the majority of the time of geospatial analysis.

[5] `https://geopandas.org/en/stable/`

[6] `r-spatial` (`https://r-spatial.org/`) is not to be confused with `rspatial` (`https://rspatial.org/`)

[7] `https://s2geometry.io/`

[8] `http://postgis.net/workshops/postgis-intro/geography.html`

[9] `https://github.com/NiklasPhabian/SphereGIS`

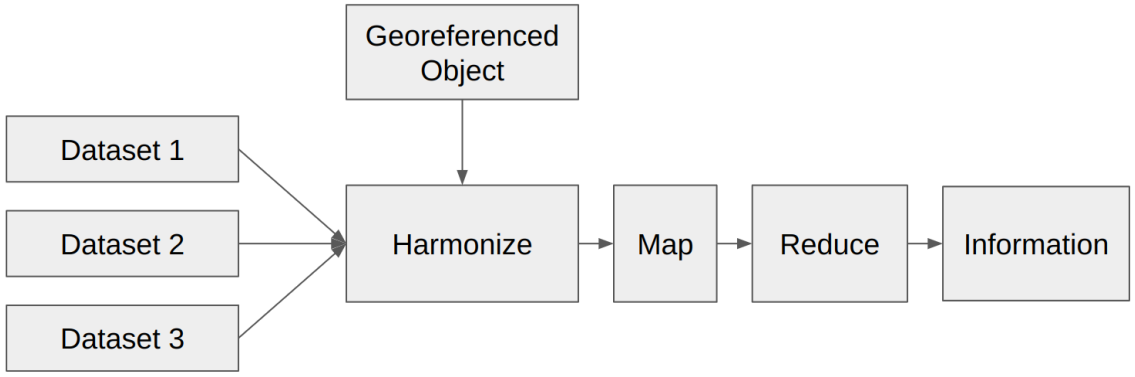


Figure 3.2: Flow data to information during geospatial analysis

projected 2D coordinate system if the ROI has a sufficiently large spatial extent, requiring yet even more adurous efforts and compromises.

In the analysis of remote sensing data, two factors complicate the ETL process and the association (map) operations. Firstly, remote sensing observations are overwhelming in volume: A single remote sensing instrument will register thousands of individual observations (Instantaneous Field of Views (IFOVs)) per second. Since mission durations often span multiple years, a single space-born sensor may accumulate trillions of observations during its lifetime. (Even at its coarsest spatial resolution Moderate Resolution Imaging Spectroradiometer (MODIS) aboard Terra has made over 6 trillion individual observations (observation is here to be understood as the registration of a spectrum) within its 22 years of operation. The Visible Infrared Imaging Radiometer Suite (VIIRS) aboard the Joint Polar Satellite System (JPSS) satellites Suomii National Polar-orbiting Operational Environmental Satellite System Preparatory Project (NPP), National Oceanic and Atmospheric Administration (NOAA) -20, and JPSS -2 each register about 1 trillion observations per year).

Secondly, remote sensing data is inherently irregular: Subject to the nonlinearity in the dynamics of the space- or aircraft trajectory, the optical properties of the atmosphere, and the topography of the Earth's surface, remote sensing observations are irregularly spaced. Thus

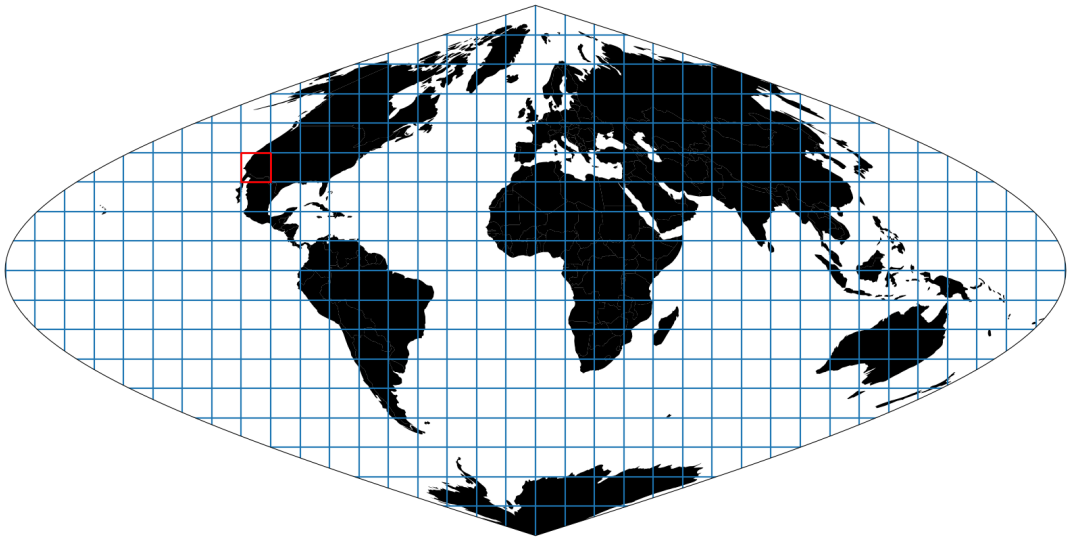


Figure 3.3: The MODIS integerized sinusoidal (ISIN) tile grid divides the Earth into 460 nonoverlapping "tiles" of 10°x10°. MODIS observations are spatially binned into those tiles. This makes accessing data for a given region simple: A user only once has to evaluate which tiles cover the ROI and then request data for those tiles. Each tile contains 2400x2400 cells (at 500 m resolution), into which individual observations are further binned.

every single observation has to be considered to have a unique location. The irregular spacing is only exacerbated for sensors with wide scan angles in which successive observations are registered under constantly changing viewing angles leading to intravariablity within a single set of observations.

The combination of irregularity and volume makes it impossible for data repositories to allow users to query and extract individual observations subject to arbitrarily defined regions of interest[10]; it is infeasible to perform trillions of spatial relation tests ad-hoc using conventional technologies. Therefore, repositories bin observations into spatial grid tiles (c.f. figure 3.3) and/or temporal chunks (aka granules). The choice of the size of grid tiles and temporal chunks is relatively arbitrary and may differ vastly for individual products. Rather than actually querying observations, users thus query bins that intersect their region of interest. For the

[10]Exceptions may be technologies like Open-source Project for a Network Data Access Protocol (OPeNDAP) or Web Coverage Service (WCS), which are mainly used as middleware and not (yet) designed for end-users.

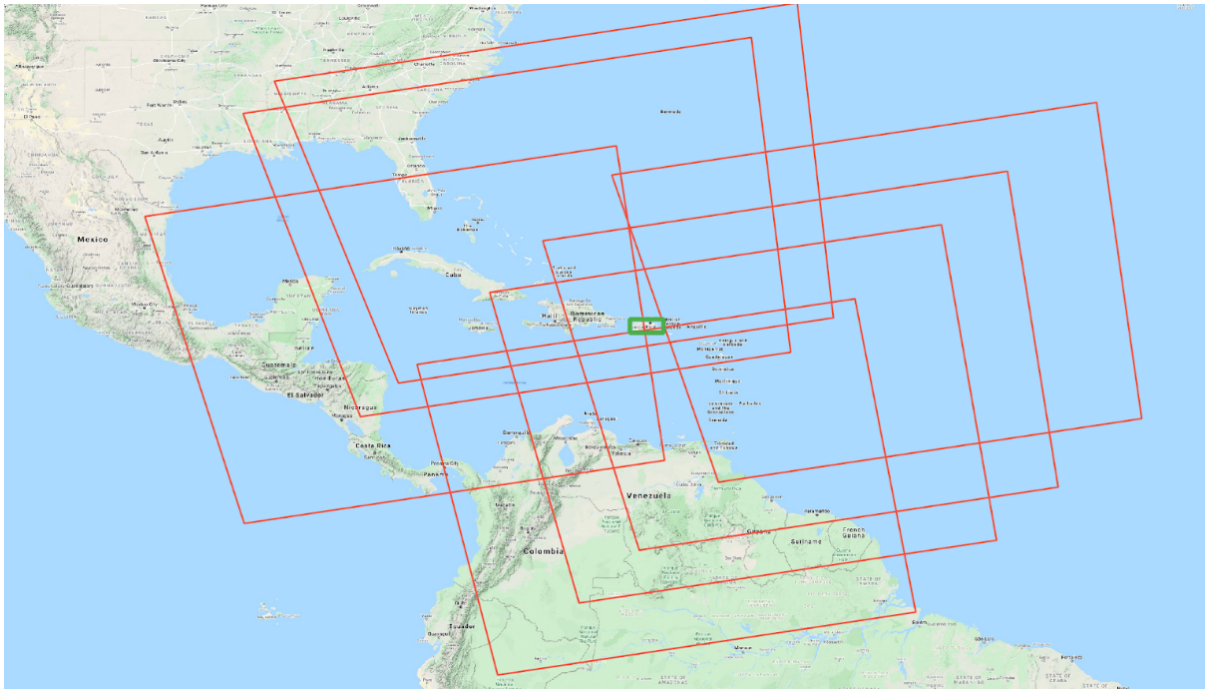


Figure 3.4: Footprints of granules (red) and a region of interest (green). Note that the granule footprints are orders of magnitude larger than the ROI.

repositories, this has the advantage of performing far fewer spatial relation tests. However, it coarsens the level of spatial queryability. Rather than extracting the intersection of the data of interest and the ROI, users extract bins that intersect the ROI. This, in turn, means that users extract far more data than they need (leading to a transfer overhead) and requires the users to extract the intersecting data from the bins.

Though the user will have extracted only a small subset of the entire data by extracting only the intersecting bins, they are still faced with having to execute potentially trillions of spatial relation tests both to spatially subset and extract the data and to map the data to other geospatial objects (c.f. figure 3.4). In most cases, this remains impossible. Even seemingly simple tasks such as clipping or cropping quickly become challenging in conventional geographic information systems and libraries such as QGIS, ArcGIS, PostGIS, r-spatial, or GeoPandas. Since repositories are aware of these limitations, they offer datasets in which the irregularly spaced locations of the observations are discretized and sampled/aggregated into a (two-dimensional)

grid and. In NASA terminology, those datasets are referred to as Level 3 products. The grid dimensions now function as an index and proxy to the geolocation of the observations. While it is often the only possible way for users to process remote sensing data that have been spatially discretized, this approach comes at a cost:

1. The precision of the geolocation of remotely sensed observations may be orders of magnitude higher than the selected discretized grid resolution (e.g., the MODIS geolocation accuracy is approximately 50 m at nadir (R. E. Wolfe et al., 2002), whereas data is gridded into 500 m cells). Thus, discretization introduces a significant loss of precision in geolocation.
2. Since the repository performs discretization, the sampling function may be obscure, if not opaque to the users. This voids transparency and complicates the provenance trace. In some cases, users may favor a sampling function that differs from the one used to generate the gridded data. (A classic example may be that the MOD09GA sampling algorithm favors snow-free observation, being counterproductive if, e.g., snow-covered areas are investigated).
3. Not all remote sensing observations are available as gridded datasets (e.g., MODIS thermal anomalies MO/YD14*, VIIRS Day/Night band VNP02DNB), leaving those datasets inaccessible to users solely able to process gridded datasets.
4. The gridded data is a redundant representation of the observations, meaning that information is duplicated and, thus, storage space wasted.

In summary: The goal of geospatial data analysis is to extract information about spatial objects, which is achieved by evaluating spatial relations, requiring spatial representations to be harmonized. However, data harmonization is tedious and non-trivial because of the multitude of container formats and data representations. Finally, evaluating spatial relation tests on large volumes of irregularly spaced data is a disabling bottleneck, currently only circumventable by discretizing locations, resulting in decreased observation fidelity.

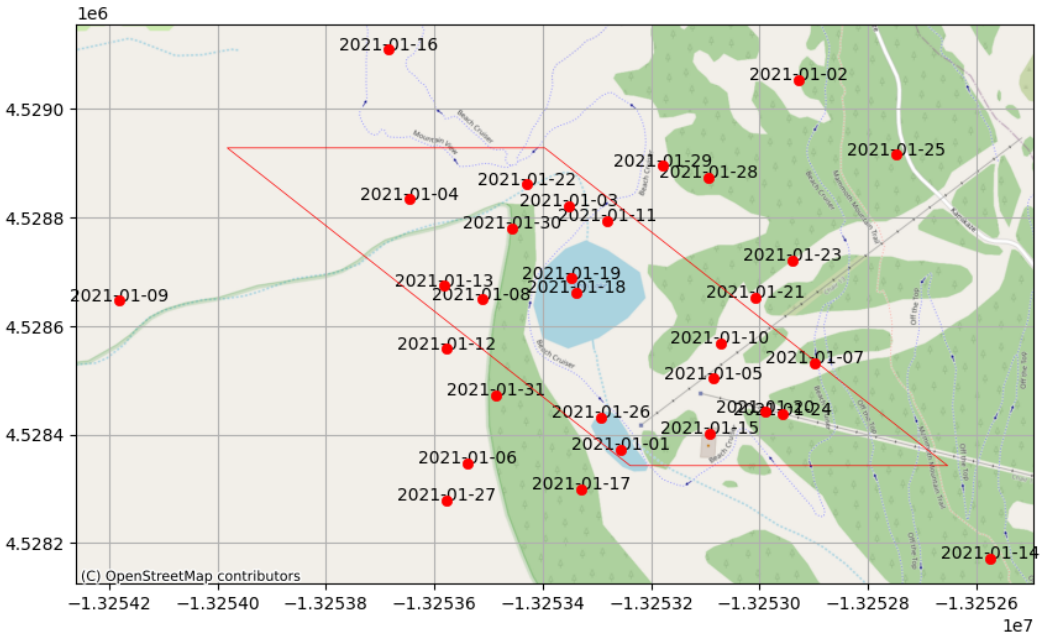


Figure 3.5: Geolocations of observations (red dots) vs. the discretized location (red parallelogram)

3.2 Towards a solution

The conflict between processibility and fidelity can only be resolved through technologies that can **performantly** determine geospatial coincidence of extensive collections of irregularly spaced locations without needing data discretization. In this paper, I present a collection of software built around the Spatio-Temporal Adaptive-Resolution Encoding (STARE) (K.-S. Kuo and Michael Lee Rilee, 2017; Michael Lee Rilee, K.-s. Kuo, et al., 2018; M. Rilee, K.-S. Kuo, Gallagher, et al., 2019; Michael L Rilee, K.-S. Kuo, J. Frew, et al., 2020; Michael L. Rilee et al., 2021), which drastically simplifies remote sensing data processing and empowers scientists to utilize the full fidelity of the data. STARE serves as a harmonizing location representation allowing for cheap spatial relation tests between arbitrarily shaped geospatial objects.

STARE is a universal geolocation encoding that obviates the need for gridding and sampling data. It is a geospatiotemporal indexing and representation scheme based on a Hierarchical Triangular Mesh (HTM) (Kunszt, Alexander S Szalay, and A. R. Thakar, 2001; Dutton, 1996; Goodchild and Shiren, 1992; Fekete and Treinish, 1990), which recursively subdivides the Earth's surface into nested quadtrees, allowing triangular regions ("trixels") as small as $0.01\,\mathrm{m}^2$ to be identified with a single integer value. The nesting properties of STARE trixels are an elegant solution for aligning multi-resolution Earth science data. The geospatial coincidence between two trixels can be evaluated by comparing their paths in the STARE tree structure. STARE allows not only to express the location of a point/location (including its spatial uncertainty or resolution) but also of arbitrarily shaped areas (polygons) through a trixel tessellation.

The resolution is encoded into the STARE index, making it possible to evaluate spatial coincidences of data of differing and/or varying resolutions. This is a required feature to express the locations of, e.g., IFOVs of wide-scan swath data, which are characterized by considerable variations of the footprints of pixels at nadir vs. the footprint of pixels at the end of a scan. Data represented with STARE thus becomes interoperable without sampling and gridding. (Michael

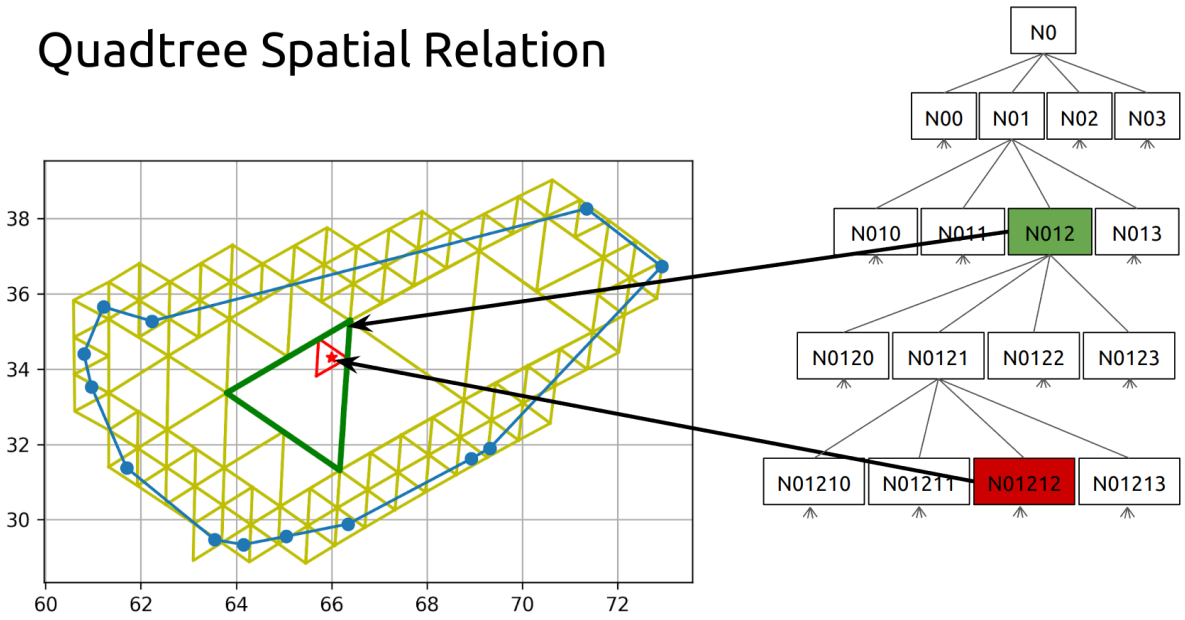


Figure 3.6: Evaluating the spatial coincidence of two HTM trixels is achieved by comparing their paths in the tree structure.

L. Rilee et al., 2021) describes the underlying principles of STARE and the reference implementation of the STARE Application programming interface (API), which is exposed through Python bindings in the PySTARE library.

STARE has the potential to be used as a unifying representation for any geospatial data. However, more than STARE as a concept is needed to enable a researcher to work with diverse data seamlessly. Instead, a collection of software and abstractions are required to make STARE an opaque technology hidden in the backend of software and services. Thus, we created a collection of software that enables STARE - based geospatial analysis, allowing users to scale analyses in variety and volume. To enable a complete STARE - based geospatial workbench, we identified the need for the following capabilities:

- Represent features, such as points, lines, (convex and nonconvex) rings, polygons, and collections thereof, with STARE.
- Determine spatial relations between spatial objects in STARE representation to perform

spatial subsetting and spatial associations.

- Associate locations and attributes in a unified data model.
- Use an API with functions and idioms similar to familiar geoprocessing software.
- Visualize.
- File I/O methods to read and convert commonly used feature- and raster data packages (shapefiles, geopackages, GeoTIFF, PostGIS) to STARE representations.
- Persist datasets in STARE representations.

Specifically for working with remote sensing data where data volumes and data variety makes ETL a significant burden, we additionally identified the following needs:

- Create STARE representations from conventional geolocations of remote sensing products and store them. I.e., making data STARE-ready.
- File I/O methods to read remote sensing data files (granules).
- Subset extensive local collections of remote sensing granules.

With the increased importance of cloud computing, we further addressed the need to store and access remote sensing data in cloud buckets and designed a parallel optimized data storage that allows aligning geospatial data in storage.

The remainder of the paper will discuss the following topics:

1. A brief summary of STARE and its capabilities for evaluating spatial coincidence.
2. The STARE base library and its Python bindings PySTARE, which stand at the bottom of our software stack.
3. Software to convert collections of remote sensing data from legacy/conventional representations to STARE representations (STAREMaster).
4. A high-level data abstraction and processing API (STAREPandas).
5. Storage mechanisms and backends for geospatial data in STARE representation (STARE-Lite, PostSTARE, STAREPods).

6. A set of use cases that demonstrate the opportunities in STARE-based geospatial analysis.
7. An outlook on future developments of STARE software collection and services.
8. Conclusion.

3.3 STARE

Using hierarchical data structures to represent or index geospatial data has been explored for decades (Dutton, 1996; Samet, 1988). Hierarchical data structures are based on the recursive decomposition of an initial planar or solid and can, for example, be implemented as quadtrees (Samet, 1988).

The initial applications of quadtrees in the geospatial domain have mainly focused on the representation of two-dimensional data in terms of visualization and image processing and were typically based on the tessellation of squares (Lugo and Clarke, 1995).

An early example of the use of quadtrees to represent the globe three-dimensionally is (Dutton, 1984), who proposed the establishment of a Geodesic Elevation Model in which locations of elevation measurements are encoded or indexed in a quadtree. (Dutton, 1989) suggests using this quadtree for general indexing of planetary data and envisages a replacement of coordinates in geospatial data with quadtree-based "geocodes", given that the community could agree on a standard method to generate "geocodes".

In parallel efforts (Fekete, 1990; Fekete and Treinish, 1990), and (Goodchild and Shiren, 1992) (and later also (Lugo and Clarke, 1995)) implemented the Quaternary Triangular Mesh (QTM) initially suggested by (Dutton, 1984). While (Goodchild and Shiren, 1992) used an octahedron as the initial regular solid, (Fekete, 1990; Fekete and Treinish, 1990) used an icosahedron. The resulting structures allow us to geospatially index every feature object on the planet. (Fekete, 1990; Fekete and Treinish, 1990; Goodchild and Shiren, 1992; Lugo and Clarke, 1995) tessellate each triangle into four triangles, allowing them to store each tessellated triangle with two bits. (Goodchild and Shiren, 1992) point out that the length of a trixel address (i.e., the index), which corresponds to the level/depth in the hierarchy, indicates the size (or spatial uncertainty) of the indexed object.

(Dutton, 1996) explored the tradeoffs of the choices of the initial solid (Tetrahedron, Octahedron, icosahedron) and, with this, empathizes the advantages of an octahedron for practical

reasons: It is straightforward to orient an octahedron so that cardinal points (the poles) occupy vertices and cardinal lines (equator, prime meridian) align with edges; and when so oriented, most vertices lie in oceans or sparsely populated land areas.

The idea of indexing spherical data with a quadtree was picked up again by (Barret, 1995) and further adapted by (Kunszt, Alexander S Szalay, Csabai, et al., 2000; Kunszt, Alexander S Szalay, and A. R. Thakar, 2001; Alexander S Szalay et al., 2005), who developed an indexing schema for the Sloan Digital Sky Survey (SDSS), which would later be implemented into the SkyServer[11] (Alexander S. Szalay, Gray, et al., 2002; A. Thakar et al., 2003). The authors coined the term HTM.

All nodes of the HTM quadtree are spherical triangles. The quadtree is created by recursively dividing the triangles into four new ones (quadfurcation) by using the parent-triangle corners and the midpoints of the parent triangle sides as corners for the new ones. The name of a new node (triangle) is the concatenation of the name of the parent triangle and an index 1 through 4. Thus, node names increase in length by two bits for every level. The authors distinguish between HTM names and the HTM Spatial Identifiers (HIDs), which is the 64-bit-encoded integer of the HTM name. (Kondor et al., 2014) use and extend the HTM implementation to tessellate complex regions on the Earth's surface.

(Doan et al., 2016) emphasize the importance of indexes on geospatial database performance as they govern data placement alignment and suggests HTM as a promising approach.

(Michael Lee Rilee, K.-S. Kuo, et al., 2016) advanced the HTM implementation from right-justified mapping to left-justified mapping:

In a right-justified mapping, trixels in proximity but at different quadfurcation levels are mapped to separate locations on the number line. For example, the trixel **S0123** has a binary HID of `1 00 01 10 11` and thus an HID of 283, while the trixel **S01230** has a binary HID of

[11]The SkyServer was built by Tom Barclay, Jim Gray, and Alex Szaley from the TerraServer (Barclay, Eberl, et al., 1998; Barclay, Gray, and Slutz, 1999) source code. The latter was a project demonstrating the real-world scalability of Microsoft SQL Server and Windows NT Server.

`1 00 01 10 11 00` and thus an HID 1132. The IDs are far from each other on the number line, while both trixels share the same first 5 digits in their name prefix (and thus are contained in each other).

Left-justified mapping respects geometric containment by right-padding binary HIDs with zeros. The quadfurcation level (in right-justified mapping implicitly given by the length) is specified by the last digits of the name. In left-justified mapping, the two trixels above would be named **S0123004** and **S01230005** (the last digits of the names (`4` and `5`) indicate level 4/5), which would translate to the binary HIDs of `1 00 01 10 11 00 00 100` and `1 00 01 10 11 00 00 101` and thus HIDs 36 228 and 36 229.

Therefore, co-located indexes are in similar index ranges regardless of the level. (K.-S. Kuo and Michael Lee Rilee, 2017) extend the implementation with a temporal component and name the resulting universal geoscience data representation the STARE. In the following, we will refer to left-justified IDs as STARE Spatial Identifiers (SIDs).

HTM Name	Binary HID	HID	STARE Name	Binary SID	SID
S0123	`1 00 01 10 11`	283	**S0123004**	`1 00 01 10 11 00 00 100`	36 228
S01230	`1 00 01 10 11 00`	1132	**S0123005**	`1 00 01 10 11 00 00 101`	36 229

3.4 STARE Software collection

3.4.1 STARE library API and PySTARE

We implemented the left-justified HTM encoding, extensively described in (Michael L. Rilee et al., 2021; Michael L Rilee, K.-S. Kuo, J. Frew, et al., 2020; M. Rilee, K.-S. Kuo, Gallagher, et al., 2019; Michael Lee Rilee, K.-S. Kuo, et al., 2016; K.-S. Kuo and Michael Lee Rilee, 2017) in the STARE C++[12] base library, which stands at the base of the STARE software stack. Its functionality encompasses the following:

1. Lookup of SIDs for points and regions.
2. Methods for interrogating and manipulating SIDs.
3. Conversion of SIDs to trixel node locations (edges and center points).
4. Perform intersection tests between SIDs and sets of SIDs.

The STARE library has no file I/O capabilities, nor does it directly support geographic objects or collection of geographic objects. It is thus agnostic to data formats. Geographic locations are passed as arrays of pairs of floating point World Geodetic System (WGS) 84 longitudes and latitudes, while STARE index values are passed as arrays of 64-bit integers.

In STARE, two notions of location exist: points and contiguous regions. A point is represented as a single SID, while a contiguous region is represented as a set of SIDs. A single SID simultaneously encodes a location and a level of uncertainty or resolution. A SID directly corresponds to a trixel, having three vertices, a center point, and a calculable area. The conversion of a point described as a single (latitude, longitude) pair is achieved by finding the trixel (at the specified quadfurcation level) that intersects the point. The lookup of the set of SIDs corresponding to a contiguous region is achieved by finding all trixels (of a specified quadfurcation level) with at least one vertex within the contiguous area.

[12] `https://github.com/SpatioTemporal/STARE`

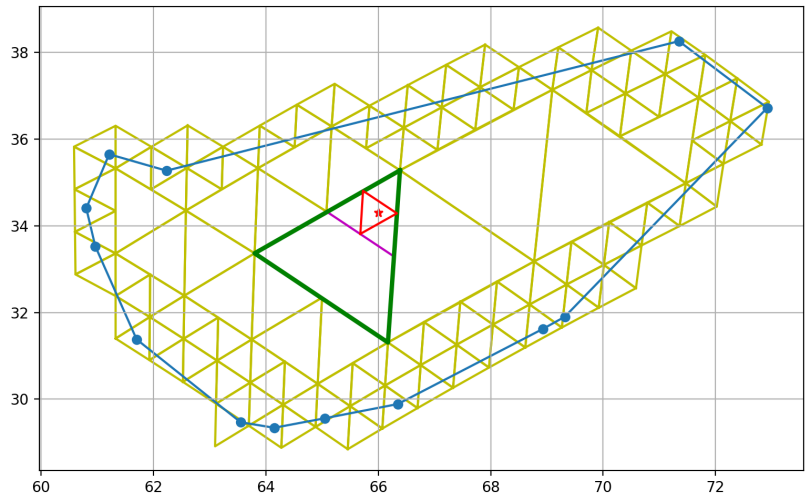


Figure 3.7: A polygon (blue) and its representation as a set of trixels that cover it (yellow). A point (red star) is represented by a single trixel (red triangle) at a chosen STARE level.

STARE can look up SIDs for two types of contiguous regions: convex hulls and nonconvex rings. For convex hulls, each edge of the hull is treated as a great circle (represented as its normal vector) that constrains the hull. A (trixel-) vertex (represented as an Earth-centered, Earth-fixed coordinate system (ECEF) vector) is found to be inside a hull if it is inside all constraints. For a hull with n edges, n dot products are thus required to determine if a vertex is inside the hull. In a recent addition, STARE can now also look up the SIDs of (nonconvex) rings. The algorithm is based on SphereGIS'[13] spherical ray-casting point-in-polygon (more accurately: point-in-ring) tests which are adapted from (Bevis and Chatelain, 1989; Chamberlain and Duquette, 2007): Each of the ring's edges is a great circle segment, which is represented as triplets of great circles: One being the circle-edge norm vector representing the edge's line and direction, one for the 'left' terminator and one for the 'right' terminator. To test if a point is inside the ring, we cast a ray from the point to another random point of the sphere.

The ray itself, thereby, is a great circle. Since the great circle ray wraps around the sphere, it will intersect the ring's edges either not at all or an even number of times. We, therefore, cannot merely count the number of intersections but instead have to distinguish how many times a ray

[13] `https://github.com/NiklasPhabian/SphereGIS`

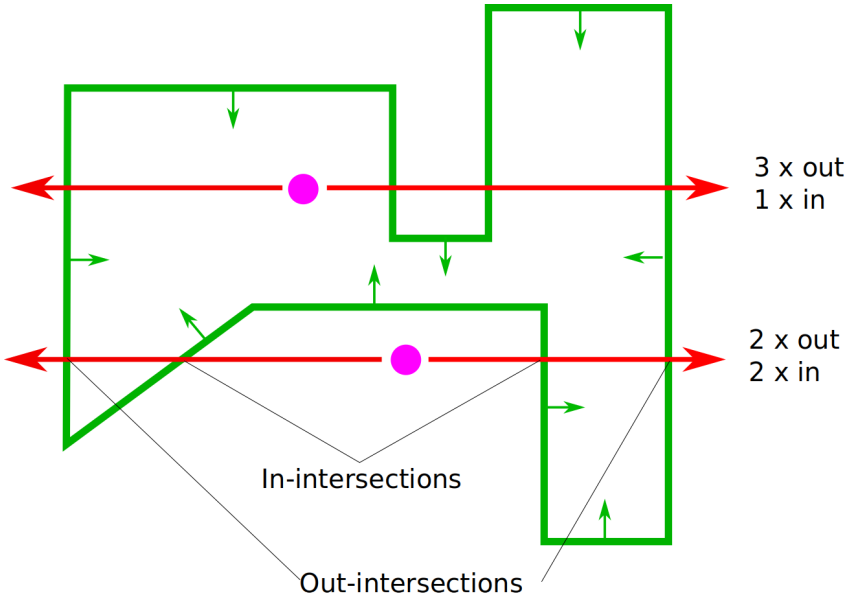


Figure 3.8: Spherical ray-casting point-in-ring tests: We compare the number of times a ray enters the ring vs. how many times it exits it. If the ray exits the ring more often than it enters it, the point is inside it.

enters (or, alternatively *exists*) the ring. We know a ray intersects an edge if the intersection of the ray with the edge's great circle is between the edge's terminators. The terminators are great circles perpendicular to the edge and the nodes of the edge. Since the ray does not have a direction, we determine if a ray *enters* (rather than exits) the ring by first determining on which side of the edge the point in question is (i.e., calculating the dot product of the edge's great circle norm vector and the point). If the point is on the side of the edge's hemisphere (i.e. their dot-product is positive), the ray exits the ring when it crosses the edge. If a point is inside the ring, the ray will exit it more often than it enters it. For a ring with n edges, 3 n cross products (one to produce the norm vector of the edge's great circle, one to produce each terminator great circle) and 3 n dot products must be performed to determine if a given vertex is within a ring.

For convenience, STARE can further look up the sets of SIDs that cover a circular region around a given latitude and longitude, subject to a radius. STARE also allows for interrogation and manipulation of the STARE level of individual SIDs and calculating the area that the trixel

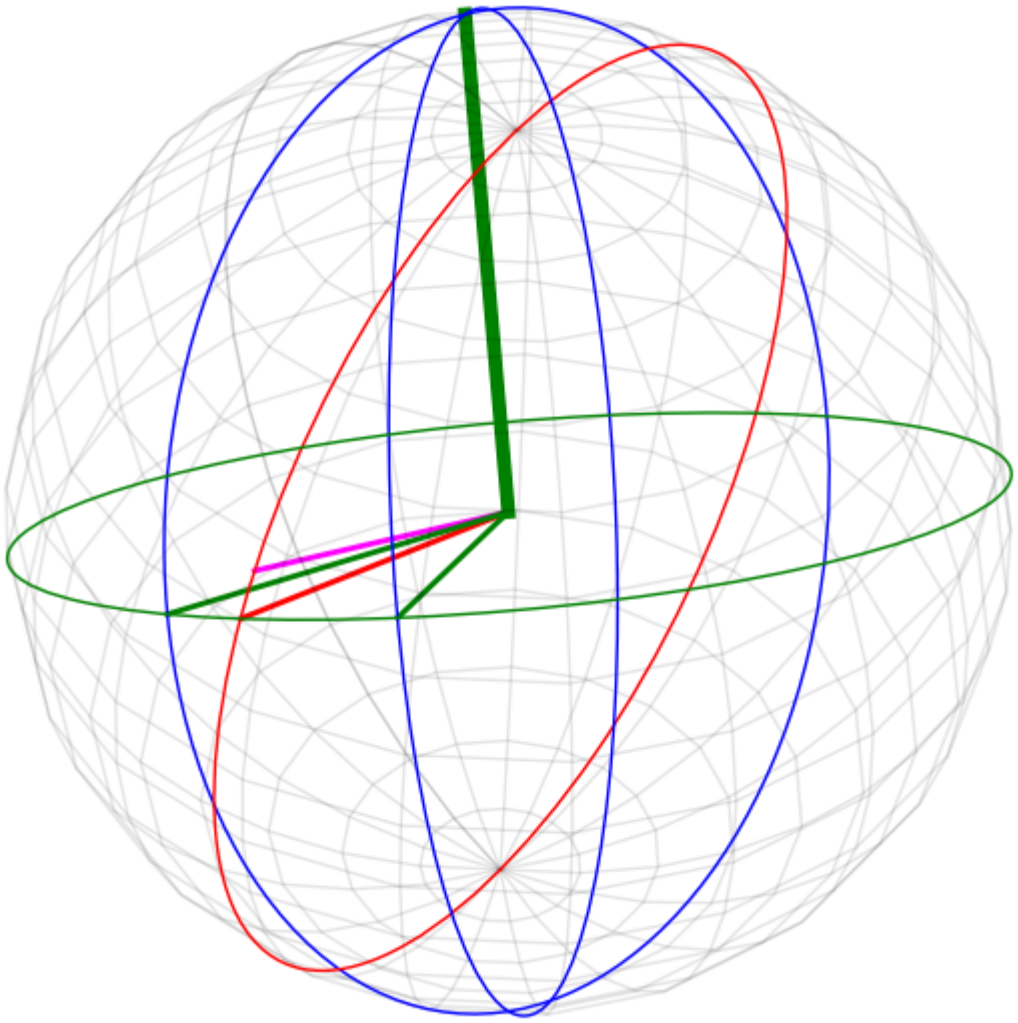


Figure 3.9: Spherical visualization of the point-in-ring test: The green great circle is the edge of a ring, with its two terminators (thin green vectors) and thick green normal vector. The point in question is the magenta vector. The ray is the red great circle which intersects the edge's great circle at the red vector. The intersection appears between the two terminators. Therefore we declare the ray to intersect the edge. Since the point is on the edge's hemisphere (positive dot product between magenta and thick green vector), we declare the ray to exit the ring at the intersection.

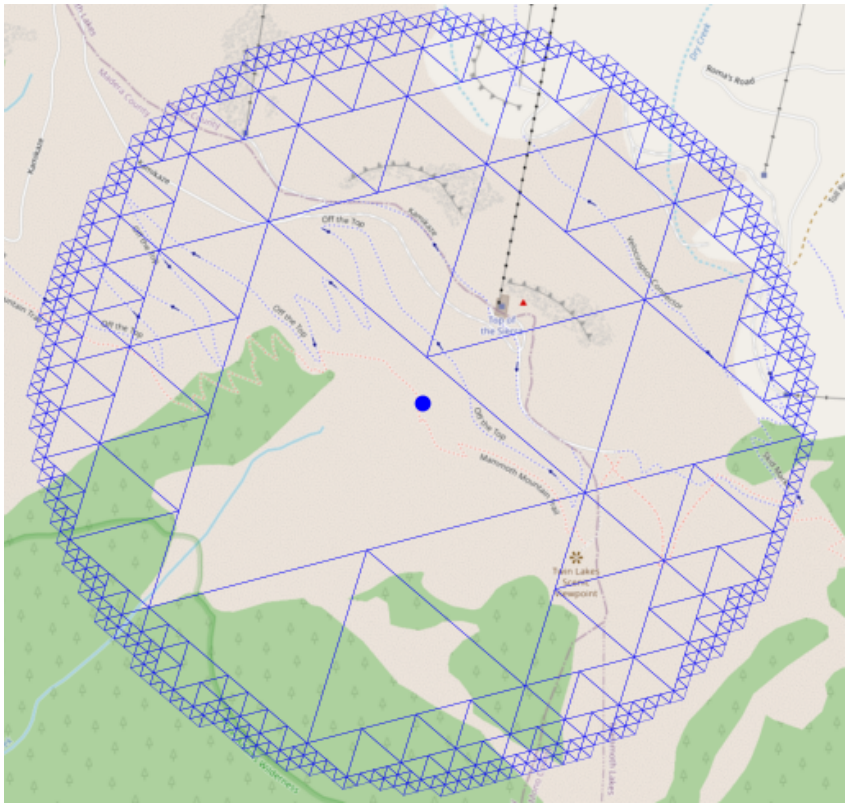


Figure 3.10: A circular cover around a center point created with STARE.

denoted by a SID covers. The library further provides methods to look up the latitudes and longitudes of the vertices and center points of a trixel denoted by a SID. Those methods have proven helpful in creating visualizations of STARE-based geospatial analysis. Finally, STARE can evaluate spatial coincidence/overlap of two individual SIDs and between two sets of SIDs.

3.4.2 PySTARE

While the C++ base library's API contains a minimum set of methods to perform STARE based geospatial analysis, we recognize that geospatial analysis is often performed in an exploratory and ad-hoc manner in higher-level programming languages. Using Simplified Wrapper and Interface Generator (SWIG)[14], we, therefore, created Python bindings to a subset of the base

[14] `https://swig.org/`

libraries' API and exposed them in a Pythonic API in the PySTARE project[15]

Python became a convenient choice since it is commonly used and popular in geospatial analysis. Various Python libraries exist to read data formats commonly used in geospatial analyses (e.g., Hierarchical Data Format (HDF)4/5, Network Common Data Form (NetCDF), Structured Query Language (SQL), comma-separated values (CSV), shapefiles, geopackages) and to visualize geospatial data (Matplotlib, GeoPandas). PySTARE is intended to be the primary user interface to STARE. The development, therefore, focuses on complete documentation (published on Read the Docs (RTD)[16]), tight adherence to Python Enhancement Proposals (PEPs)[17] style guides, test-driven continuous integration, and a simple path for installation: We provide pre-compiled wheels distributed through Python Package Index (PyPI)[18], allowing PySTARE and all its dependencies to be installed with a single command.

The following code snippet demonstrates how points and rings are converted to SIDs and how the spatial relations between sets of SIDs can be evaluated with PySTARE.

[15] Github: `https://github.com/SpatioTemporal/pystare`;
Readthedocs: https://pystare.readthedocs.io.

[16] `https://pystare.readthedocs.io/en/latest/`

[17] `https://peps.python.org/`

[18] `https://pypi.org/`

```
import numpy
import pystare

points_lats = [52.52063377345684, 48.86507804019062]
points_lons = [13.40137845762151, 2.3357209301448676]
points_sids = from_lonlat(lons, lats, level)



ring_lats = [53.75702912049104, 54.98310415304803, 53.69393219666267,
             50.128051662794235, 49.01778351500333, 47.62058197691181,
             47.467645575544, 50.266337795607285, 53.75702912049104]

ring_lons = [14.119686313542559, 9.921906365609118, 7.100424838905269,
             6.043073357781111, 8.099278598674744, 7.466759067422231,
             12.932626987365948, 12.240111118222558, 14.119686313542559]

sids_ring =  pystare.cover_from_ring(lat, lon, 5)

pystare.intersects(cover, sids, method='binsearch')
array[True, False]
```

3.4.3 STAREMaster and STARE sidecar files

Conventionally, the locations of remote sensing observations are represented either as geolocated IFOV features, where each observation is associated with an Earth location (usually WGS 84 longitude and latitude) or as gridded and projected fields of observations, in which the grid indices can be converted to geolocations. In order to perform STARE-based geospatial analysis

on these data, the SID for each observation or grid cell has to be calculated. Since these calculations involve expensive transcendental functions, the index values are calculated once and then stored.

STAREMaster_py[19] is a library and set of command line tools that allow looking up the STARE representation of commonly used remote sensing products and storing those STARE representations into companion ("sidecar") files (Gallagher, Hartnett, et al., 2021), which are intended to be read together with the data files during the analysis. STAREMaster_py is written to be easily extendable to allow ingestion of other products. It handles a subset of MODIS, VIIRS, and various microwave products. STAREMaster_py further includes convenience utilities to verify the integrity of local granule+sidecar collections (e.g., to detect missing sidecars.)

During the SID lookup for each observation, the STARE quadfurcation level of each SID is adapted so that the corresponding trixel area matches the extent of the observation's IFOV as closely as possible.

STAREMaster_py can also create sidecar files for gridded products, such as the MODIS level-3 surface reflectance product MOD09GA[20]. MOD09GA uses a sinusoidal projection, and the MODIS sinusoidal grid divides the Earth into 460 nonoverlapping "tiles" of approximate 10°x10°. MODIS observations are spatially binned into those tiles. This makes accessing data for a given region simple: A user only once has to evaluate which tiles cover the ROI and then request data for those tiles. Each tile contains 2400x2400 cells (at 500 m resolution), into which individual observations are further binned. Even though the individual observations are irregularly spaced, the cells remain fixed. This means that all granules of the same tile can share a single sidecar file. In other words: Only one sidecar for each of the 460 tiles has to be created once.

Besides looking up the SIDs for each observation of a granule, STAREMaster_py also computes the set of SIDs that cover the footprint of the granule (the "STARE cover") from the granule's extent information. For granules that do not contain extent information, the STARE cover is

[19] Github: `https://github.com/SpatioTemporal/STAREmaster_py`

[20] (Eric Vermote and Robert Wolfe, 2021). [DOI 10.5067/MODIS/MOD09GA.006

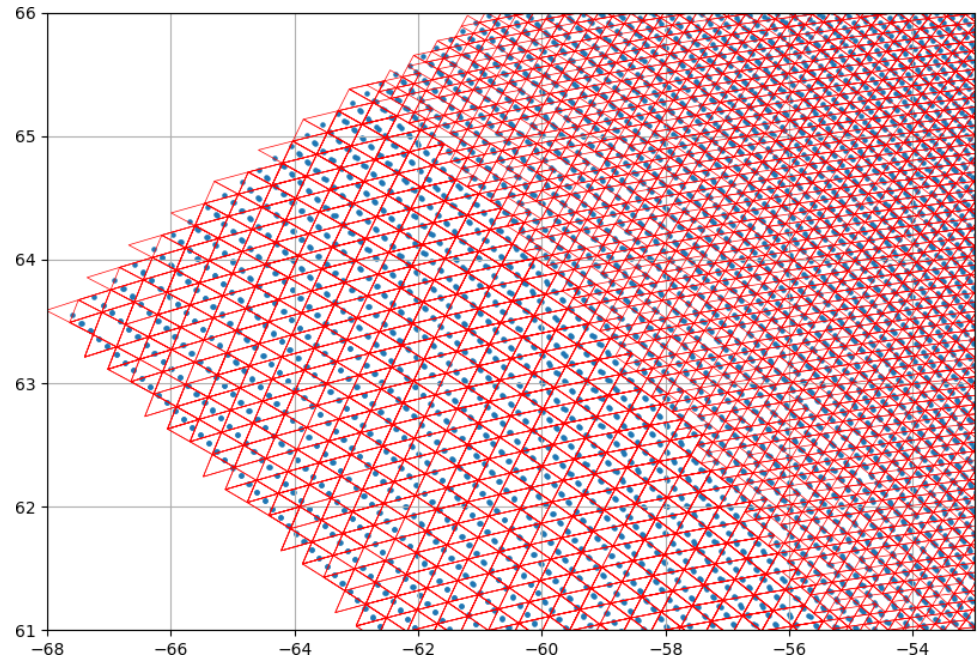


Figure 3.11: The geolocations of a MOD05 granule in blue and their corresponding resolution adapted trixel representations. Note the resolution change towards the center of the swath in the northeast direction.

computed by dissolving the set of the SIDs of the observations. "Dissolving" means that if a set contains four child nodes of the same parent, the four child nodes get replaced by the parent node.

The following code snippet shows the command line interface of STAREMasters_py's `create_sidecar_files.py` program. A user will specify individual granules or a folder containing a collection of granules for which sidecars should be created. Alternatively, the user can specify a grid, such as a MODIS tile, for which a sidecar should be created. The user can further optionally specify where the sidecars should be created, provide a hint for what product the granules are, how many workers should be used in parallel, and if a sidecar archive should be created. The sidecar archive registers all granules for which a sidecar has successfully been created. This can be helpful to avoid (accidental) redundant creation of sidecar files, e.g., when dealing with growing collections or when the sidecar creation of an extensive collection gets interrupted.

```
usage: create_sidecar_files.py [-h] [--folder folder]


[--files file1 [file2 ...]] [--grid files] [--out_path OUT_PATH]
[--product product] [--cover_res cover_res]
[--workers n_workers] [--archive archive] [--parallel_files]


Creates Sidecar Files
options:
 -h, --help              show this help message and exit
 --folder folder         the folder to create sidecars for
 --files file1 [file2 ...] the files to create a sidecar for
 --grid files            the grid to create a sidecar for (e.g. IMERG)
 --out_path OUT_PATH     the folder to create sidecars in;
                         default: next to granule
 --product product       product (e.g. cldmsk_l2_viirs, hdfeos,
                         l2_viirs, mod05, mod09, vj102dnb,
                         vj103dnb, vnp02dnb, vnp03dnb, ssmi)
 --cover_res cover_res   max STARE level of the cover.
                         Default: min resolution of IFOVs
 --workers n_workers    use n_workers (local) dask workers
 --archive archive      Create sidecars only for granules
                        not listed in the archive file.
                        Record all created sidecars and their
                        corresponding granules in it.
 --parallel_files       Process files in parallel rather than
                        looking up SIDs in parallel
```

STARE sidecar files are written out in NetCDF format, containing the following variables:

- The SID for each observation or grid cell. If a granule contains more than one resolution (such as the MODIS surface reflectance product MOD09[21], containing 250 m, 500 m, and 1000 m resolution), one SID variable is created for each resolution.
- The set of spatial index values that cover the footprint of the granule (STARE Cover).
- Optionally, the latitudes and longitudes of each IFOV; one variable for each resolution.

Ultimately, we expect that STARE sidecar files will be distributed, along with the corresponding data, by the data producers. Until then, the generation of sidecar files will remain the first step a user must perform in the STARE-based data harmonization process. Since we recognize that the generation of sidecar files is a compute-intensive but parallelizable process, we implemented STAREMaster_py to natively support multiprocessing, either for calculating multiple SIDs in parallel observations in a single granule or for processing multiple granules in parallel. The multiprocessing is handled by the dask[22] library. A local dask cluster is started by default, but STAREMaster_py can be adapted to use any (remote) dask cluster.

3.4.4 STAREPandas

STAREPandas[23] is a Python library that provides a high-level interface to STARE and a unified data representation. It allows users to perform STARE-based spatial operations and related tests on sets of features that would otherwise require more extensive tooling, e.g., by using a STARE-extended spatial database or GIS.

[21] (M. L. S. Team, 2017) DOI: 10.5067/MODIS/MOD09.006

[22] `https://www.dask.org/`

[23] `https://github.com/SpatioTemporal/STAREPandas`; `https://starepandas.readthedocs.io`

Introduction

The GIS literature (Bolstad, 2002; Worboys and Duckham, 2004; Mitchell and Environmental Systems Research Institute (Redlands, 1999; Green and Tukman, 2017; Wise, 2003; Law and Collins, 2015; Rigaux, Scholl, and Voisard, 2002) conventionally distinguishes between two types of spatial data: raster and vector. Vector data describes locations and shapes using coordinates to represent points or the vertices of lines, rings, and polygons. On the other hand, raster data represent locations as cells of a rectangular array ("grid"). The geolocation of a grid cell is implicitly given by the array indices (the row and column coordinates) of the cell and a transformation (matrix) that may be used to translate the array indices into a geospatial (geographic or projected) coordinate system. Vector datasets are often conceptually represented as tables where each row corresponds to a feature and each column to an attribute. For raster datasets, identically shaped arrays are stacked as bands to associate multiple attributes with a single cell. Both vector and raster data may be used to describe discrete features, such as roads, houses, or the outline of countries, and to represent spatially continuous phenomena such as land use, elevation, or surface temperature. However, vector data are typically used to represent feature data, while raster data is used to represent continuous phenomena. Since raster data discretizes locations, the determination of spatial coincidence between two raster datasets is trivial: two cells are coincident if they share the same indices. On the other hand, determining spatial coincidence between two vector datasets requires potentially expensive point-in-ring calculations.

Geolocated swath data[24] fall in neither the feature data category nor the raster data category. Even though swath data is typically stored in arrays, which may make them seem raster-like. However, rather than using a transformation matrix, swath data use *maps*: two ancillary arrays of the same shape as the data (one containing latitudes, the other containing longitudes) to

[24]NASA Earth-observing System Data and Information System EOSDIS distinguishes between 5 processing levels (0 through 5). Levels 1 and 2 are data "at full resolution, time-referenced, and annotated with[..] georeferencing parameters", which is what we reference with swath data. `https://www.earthdata.nasa.gov/engage/open-data-services-and-software/data-information-policy/data-levels`

provide individual (center) coordinates for each IFOV. There are arguments to be made for interpreting swath data as raster data; after all, swath observations are usually continuously recorded, and neighboring array cells represent neighboring observations. The EarthDB project (Planthaber, Stonebraker, and J. Frew, 2012; Planthaber, 2012), as well as (Tan, Yue, and Gong, 2017) and (Krčál and Ho, 2015), demonstrate the challenges with processing geolocated remote sensing swath data as rasters. In their approaches, data is indexed through integerized latitude-longitudes grids, requiring re-indexing and compromises between array sparseness and data fidelity. On the other hand, one may argue that it is merely an artifact that swath data is represented in 2-D arrays. Since the geolocation of a cell cannot be implicitly calculated from the indices (row and column numbers) but must be retrieved from the map, geolocated swath data may just as well be represented in flattened 1-D arrays or tables, making them appear much more feature-like.

STARE removes the need for distinguishing between vector and raster datasets. Since a single SID encodes both location and extent, STARE can express grid cells, IFOVs, points, and arbitrarily shaped areas. The dilemma of conceptualizing swath data as raster or vector is thus resolved.

Data Structure

STAREPandas provides a uniform data structure to hold any geospatial data type. It represents any location (point, polygon, grid cell, IFOV) as a single feature, making geospatial data genuinely interoperable. STAREPandas extends GeoPandas[25] with a STARE spatial type. It inherits GeoPandas' relational model to tie together locations and attributes of collections of features. STAREPandas exposes STARE functionality using extensions to the GeoPandas API, lowering technical hurdles for Python programmers to perform STARE - based geospatial analysis.

[25] geopandas.org; `https://github.com/geopandas/geopandas`

In contrast to GeoPandas' `GeoDataFrames,` where geometries are represented as simple features (ISO 19125-1:2004), STAREPandas' `STAREDataFrames` represent geometries as SIDs, corresponding to trixels of variable sizes and resolutions. Polygons are represented as sets of SIDs whose corresponding trixels cover the polygon, while points are represented as individual trixels at the HTM tree's leaf resolution. Single SIDs at varying quadfurcation levels represent grid cells and features such as sensor IFOVs at a quadfurcation level corresponding to the s/IFOV's spatial extent/area.

A `STAREDataFrame` has one row per feature and one column per attribute. The `STAREDataFrame` has the particular SID column, which holds the STARE representation of the location and on which all STARE - based geospatial operations are executed.

Conversions and I/O

Read feature data Using GeoPandas' I/O capabilities, STAREPandas can read most vector-based data formats. STAREPandas can then convert GeoPandas' internal simple feature representation of geometries such as points, polygons, and multipolygons to STARE's spatial SID representations using `make_sids()`[26]. While PySTARE has methods to convert points, convex hulls, and rings to SIDs, it cannot handle more complex geometries such as "swiss cheese" polygons (polygons with holes) or multipart geometries (which may be discontiguous). STAREPandas adds those capabilities.

Since GeoPandas can read and write most geospatial feature data formats, STAREPandas is a convenient way of converting conventional geospatial feature data (e.g., shapefiles) to STARE representations.

[26] `make_sids()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.read_geotiff.html#starepandas.make_sids`

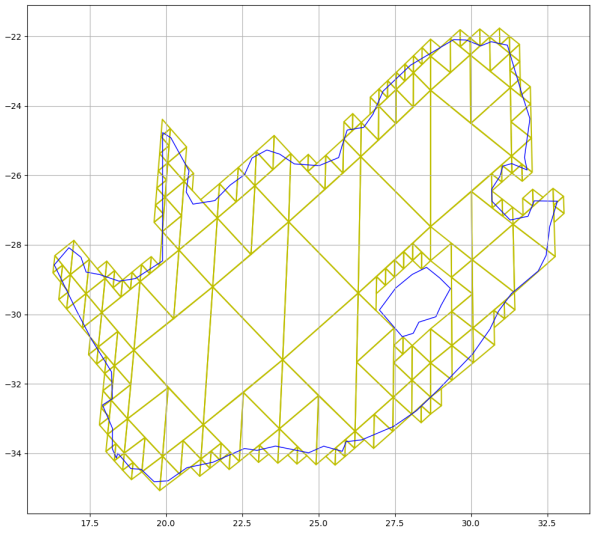


Figure 3.12: STAREPandas handles polygons with inner rings: The figure displays outline of the Republic of South Africa (RSA) with Lesotho as a hole in blue and the trixel representation of RSA in yellow.

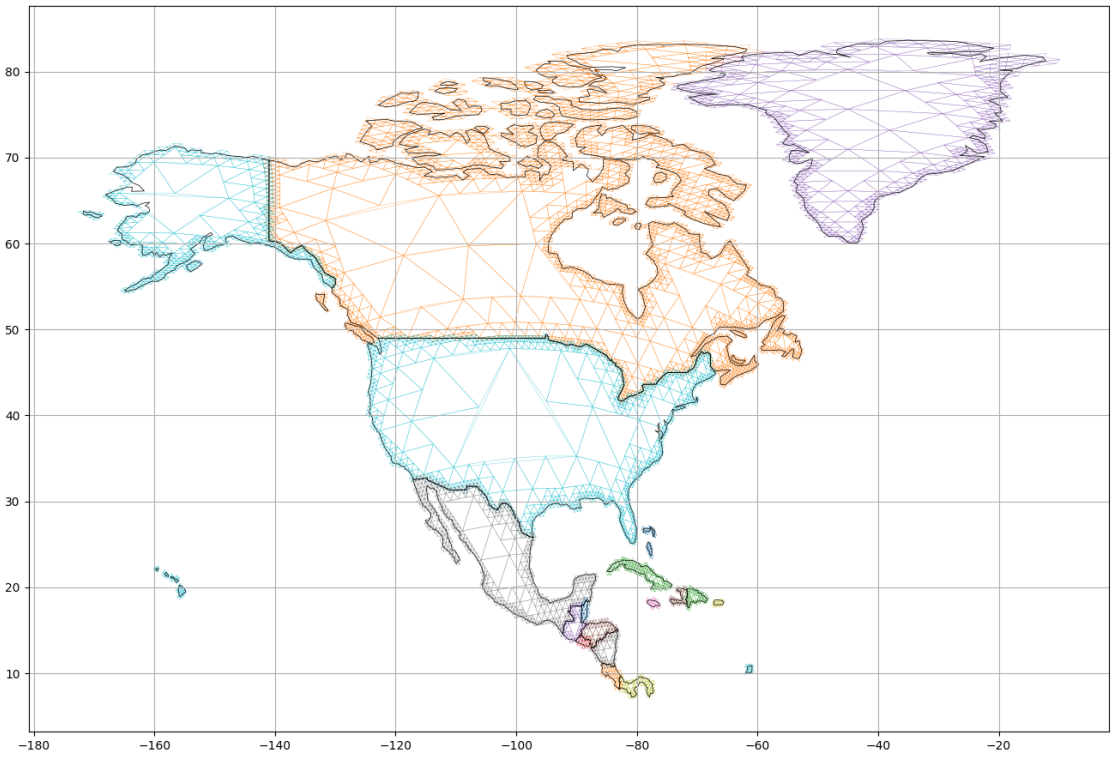
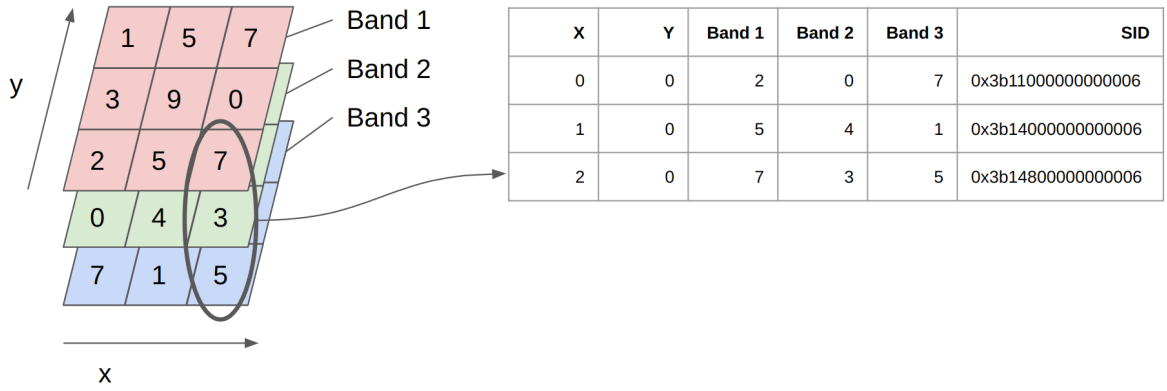


Figure 3.13: STAREPandas can handle discontinuous ("multi") polygons. The figure displays the outlines of north and central American countries and Greenland in black and their trixel representations in colors. Note how e.g., discontinuous USA is represented.



X	Y	Band 1	Band 2	Band 3	SID
0	0	2	0	7	0x3b11000000000006
1	0	5	4	1	0x3b14000000000006
2	0	7	3	5	0x3b14800000000006

Figure 3.14: Conversion of a multi-band GeoTIFF to a `STAREDataframe`. The different bands get converted into columns of the `STAREDataFrame`

Read Raster Data STAREPandas extends GeoPandas' capabilities to load raster data from GeoTIFFs using `read_geotiff()`[27]. STAREPandas will extract the transformation matrix from the GeoTIFF's metadata and compute the WGS 84 coordinates of each grid cell center. It then computes the SID for each grid cell center and adapts the STARE quadfurcation level so that the trixel area will match the cell area as closely as possible. Each cell is represented as a feature with all band values as attributes. A user may additionally choose to add the latitudes and longitudes, the array coordinates, or the projected coordinates as attributes. STAREPandas' ability to read raster data may be compared to a vectorization operation of a conventional GIS. However, while vectorization of a grid is expensive and results in increased storage requirement, STARE naturally collapses the two spatial dimensions into one.

Read Granules

While fundamentally, the same data structures are used in disseminating remotely sensed data (points, arrays), data products often idiosyncratically represent differences in orbits, viewing strategies, spatial and temporal resolutions, etc. (K.-S. Kuo and Michael Lee Rilee, 2017). As

[27] `read_geotiff()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.read_geotiff.html#starepandas.read_geotiff`

a result, each sensor's product requires specific domain knowledge and tailor-made tools to load and interpret the data, especially the location information. STAREPandas adds methods to load selected gridded remote sensing data and geolocated swath data into a `STAREDataFrame` to facilitate working with remote sensing data. The STARE representations can either be generated on-the-fly during loading or read from a pre-generated STARE "sidecar" file. The `read_granules()`[28] facilities are designed to be easily extensible to support additional products.

A user can choose to add the WGS 84 latitudes and longitudes, as well as the array coordinates, as attributes. Keeping the array indices as attributes can be of interest to maintain the neighborhood of each observation or (for scanning sensors) to calculate viewing geometries. Finally, the array indices can be used to reconstruct the original array from the dataframe using `to_array()`[29].

Similarly to reading raster data, all variables (in HDF 4 terms: scientific datasets) of the granule will be read and added as attributes for each feature. For granules containing multiple resolutions, a user chooses a single resolution to be read at a time. This means that each resolution is read separately into a separate `STAREDataFrame`, which subsequently may or may not be concatenated.

Read Folders / Create Catalogs Maintaining, searching, and subsetting extensive local collections of granules can become challenging. We thus implemented a method `folder2catalog()`[30] to catalog local collections to perform spatial searches and subsetting using STAREPandas and STARE sidecar files. The catalogs contain one row per granule. Each row contains the path of the granule and the sidecar, the granule acquisition timestamp, and the STARE representation of the granule's cover (as read from the sidecar file). Creating such

[28] `read_granule()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.read_granule.html`

[29] `to_array()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.to_array.html`.

[30] `folder2catalog()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.read_granule.html`.

a catalog dataframe for extensive local collections of granules is helpful in quickly identifying granules that intersect a potentially complex ROI or finding multiple granules that intersect each other.

Persisting STAREDataFrames Like any other Python object, `STAREDataFrames` can be serialized and stored as pickles or HDF 5 files using the inherited Pandas methods. STAREPandas extends Pandas' database functionality to read and write to/from SQLite, Postgresql, and SciDB. The STARE software collection includes STARE extensions to all three databases (STARELite[31], PostSTARE[32], SciDB-STARE[33]). STAREPandas can function as a convenient pivot format for loading data into STARE-enabled databases.

STAREPandas additionally implements its own storage mechanism based on STARE - Parallel Optimized Data Stores (PODS). A dataframe is written into a PODS using the `write_pods()`[34] method. A PODS spatially shards data at a user-defined HTM quadfurcation level. Each shard is a trixel at this user-defined quadfurcation level and only contains data within this trixel. To write a `STAREDataFrame` into a PODS, it is first split into chunks. Each row of a single chunk shares the same SID prefix. E.g., if a PODS is created at quadfurcation level 4, all data in a single chunk are contained with the same level 4 trixel. Thus all SIDs within the same chunk share the same first 9 bits (8 bits plus the initial south/north bit). The individual chunks are then written into the according PODS shards. A PODS can, in turn, be read back into a `STAREDataFrame` using STAREPandas' `read_pods()`[35] method. STARE - PODS are further described in section 3.4.6

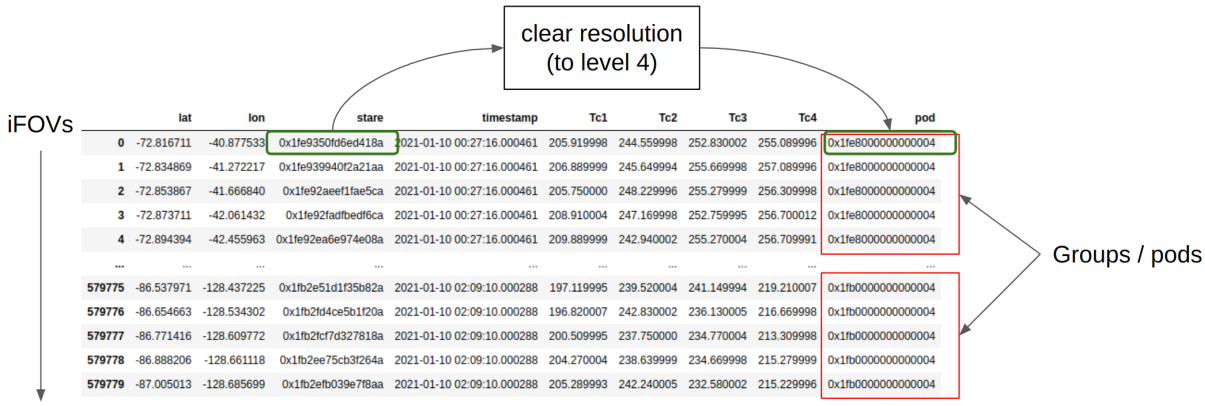
[31] `https://github.com/SpatioTemporal/STARELite`

[32] `https://github.com/SpatioTemporal/StarePostgresql`

[33] `https://github.com/NiklasPhabian/SciDB-STARE`

[34] `write_pods()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.write_pods.html`.

[35] `read_pods()` om STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.read_pods.html`



	lat	lon	stare	timestamp	Tc1	Tc2	Tc3	Tc4	pod
0	-72.816711	-40.877533	0x1fe9350fd6ed418a	2021-01-10 00:27:16.000461	205.919998	244.559998	252.830002	255.089996	0x1fe8000000000004
1	-72.834869	-41.272217	0x1fe939940f2a21aa	2021-01-10 00:27:16.000461	206.889999	245.649994	255.669998	257.089996	0x1fe8000000000004
2	-72.853867	-41.666840	0x1fe92aeef1fae5ca	2021-01-10 00:27:16.000461	205.750000	248.229996	255.279999	256.309998	0x1fe8000000000004
3	-72.873711	-42.061432	0x1fe92fadfbedf6ca	2021-01-10 00:27:16.000461	208.910004	247.169998	252.759995	256.700012	0x1fe8000000000004
4	-72.894394	-42.455963	0x1fe92ea6e974e08a	2021-01-10 00:27:16.000461	209.889999	242.940002	255.270004	256.709991	0x1fe8000000000004
...	...	...	...	...	...	...	...	...	...
579775	-86.537971	-128.437225	0x1fb2e51d1f35b82a	2021-01-10 02:09:10.000288	197.119995	239.520004	241.149994	219.210007	0x1fb0000000000004
579776	-86.654663	-128.534302	0x1fb2fd4ce5b1f20a	2021-01-10 02:09:10.000288	196.820007	242.830002	236.130005	216.669998	0x1fb0000000000004
579777	-86.771416	-128.609772	0x1fb2fcf7d327818a	2021-01-10 02:09:10.000288	200.509995	237.750000	234.770004	213.309998	0x1fb0000000000004
579778	-86.888206	-128.661118	0x1fb2ee75cb3f264a	2021-01-10 02:09:10.000288	204.270004	238.639999	234.669998	215.279999	0x1fb0000000000004
579779	-87.005013	-128.685699	0x1fb2efb039e7f8aa	2021-01-10 02:09:10.000288	205.289993	242.240005	232.580002	215.229996	0x1fb0000000000004

Figure 3.15: Illustration of how STAREDataframes are split geospatial bins, simply by grouping by the SID prefix.

STARE-based spatial operations

STAREPandas allows us to perform the STARE-based geospatial relation tests `stare_intersects()`[36] and `stare_disjoint()`[37]. Those functionalities enable STARE-based spatial subsetting and spatial joins, which are common bottlenecks when working with large collections of irregularly spaced data. They function analogously to GeoPandas `intersects()`[38] and `disjoint()`[39] methods. The relation tests are wrapped in the STARE-based join method `stare_join()`[40], which can spatially join two dataframes.

The following code snippets demonstrate how a `STAREDataFrame` can be bootstrapped from a `GeoDataFrame` and how STARE-based intersects tests can be performed. The data is visualized in figure 15.

[36] `stare_intersects()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.stare_intersects.html`

[37] `stare_disjoint()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.stare_disjoint.html`

[38] `intersects()` on GeoPandas' RTD: `https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoSeries.intersects.html`

[39] `disjoint()` on GeoPandas' RTD: `https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoSeries.disjoint.html`

[40] `stare_join()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.stare_join.html?highlight=join#starepandas.stare_join`

```
import geopandas
import starepandas
import pystare

path = geopandas.datasets.get_path("naturalearth_lowres")
world = geopandas.read_file(path)
n_america = world[world.continent=='North America']
n_america.reset_index(inplace=True)
n_america = starepandas.STAREDataFrame(n_america)
n_america = n_america.set_sids(n_america.make_sids(level=9))

santa_barbara_sid = pystare.from_lonlat([-119.81100397568609],
                                        [34.44687326105255],
                                        level=5)

n_america[n_america.stare_intersects(santa_barbara_sid)].name
United States of America
```

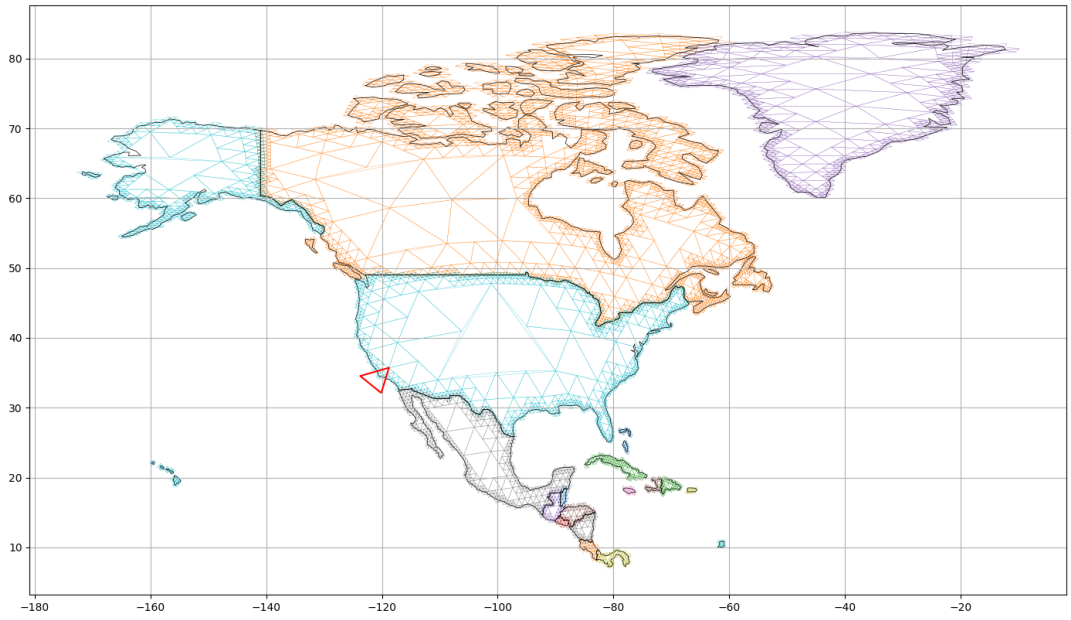


Figure 3.16: North and Central America (colored) and Santa Barbara (red trixel) their trixel representation.

The indexed data structure of STAREPandas and STARE's nesting properties offers a convenient speedup for intersection tests, implemented in STAREPandas' `speedy_subset()`[41] method: Consider an extensive collection of IFOVs, each represented by a single SID at varying levels (i.e., resolutions) and a complex geographic region represented by a set of SID, and the objective to find all IFOVs that intersect the ROI. To speed up the intersect test, we implemented the following algorithm:

1. All IFOVs with a SID smaller than the smallest SID of the ROI and all IFOVs with a SID larger than the largest SID of the ROI are not intersecting the ROI. This can significantly reduce our search space.
2. We determine the highest STARE level of all the SIDs representing the ROI and the highest STARE level of all the SIDs representing the IFOVs. The lower one of the two is

[41] `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.tools.spatial_conversions.speedy_subset.html#starepandas.tools.spatial_conversions.speedy_subset`

the *intersects level*. I.e., the STARE level at which we perform the intersects tests. We can ignore all location bits beyond the intersects level.

3. Since we can ignore the location bits beyond the intersects level, we can take the *set* of all IFOV SIDs at this level. This set will likely be orders of magnitude smaller than all SIDs representing the IFOVs.
4. We now need to perform the STARE spatial intersect tests on this (potentially much) smaller set of (ˆSID).

STAREPandas also provides APIs to query and manipulate the SID level:

- `hex()`[42] returns the SIDs in hexadecimal representation for each feature.
- `spatial_resolution()`[43] returns the STARE level of each feature.
- `trixel_area()`[44] returns the approximate area of the trixel represented by the SID for each feature.
- `to_stare_resolution()`[45] lets the user set the STARE resolution. The user can choose whether the location bits beyond the resolution should be cleared (set to 0).
- `clear_to_resolution()`[46] clears the location bits higher than the selected resolution.

In Pandas dataframes, map-reduce operations are carried out by using the `grouby()`[47] method to group ("map") rows and then applying an aggregate ("reduce") function to the groups. Since conventional aggregate functions (mean, max, sum) cannot be applied to simple feature geome-

[42] `hex()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.hex.html`.

[43] `spatial_resolution()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.spatial_resolution.html`

[44] `trixel_area()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.trixel_area.html`

[45] `to_stare_resolution()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.to_stare_resolution.html`

[46] `clear_to_resolution()` on STAREPandas' RTD`https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.clear_to_resolution.html`

[47] `groupby()` on Pandas' RTD: `https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.groupby.html`

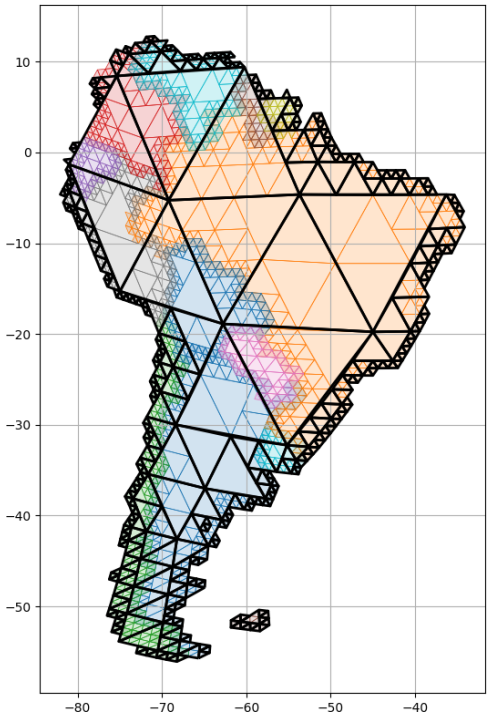


Figure 3.17: Visualization of a `STAREDataFrame` with one feature per country (colored) and the dissolved representation (black).

tries, GeoPandas implements the abstracted map-reduce function `dissolve()`[48], which will yield a `GeoDataframe` in which geometries are aggregated using a spatial `unary_union`[49]. Similarly, STAREPandas implements the STARE-based map-reduce function `stare_dissolve()`[50] that will aggregate SIDs using a `stare_union`[51]. The `stare_union` on a collection of SIDs is created by taking the set of those SIDs and then recursively replacing any four SIDs in that set that share the same parent node with the parent node. Note: Both GeoPandas' `dissolve()` and STAREPandas' `stare_dissolve()` can dissolve by `None`, yielding a dataframe dissolved into a single feature, unionizing all geometries/SIDs.

[48] `dissolve()` on GeoPandas' RTD: `https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoDataFrame.dissolve.html`

[49] `unary_union` on GeoPandas' RTD: `https://geopandas.org/en/stable/docs/reference/api/geopandas.GeoSeries.unary_union.html`

[50] `stare_dissolve()` on GeoPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.stare_dissolve.html`

[51] `stare_union` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.tools.spatial_conversions.compress_sids.html#starepandas.tools.spatial_conversions.compress_sids`

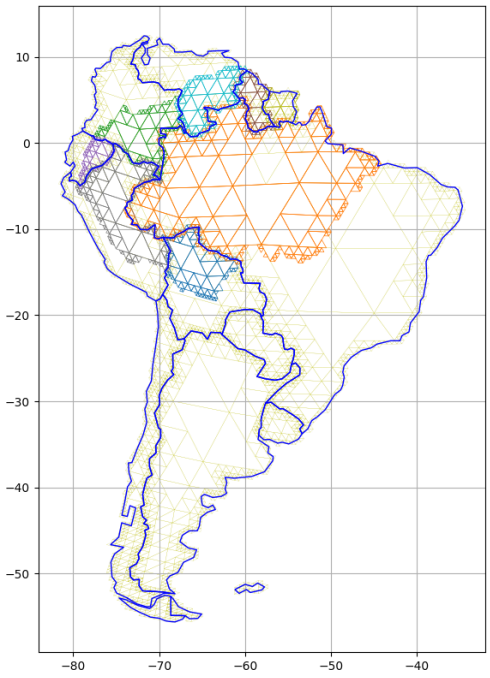


Figure 3.18: Results of an intersection of South American countries with the Amazon region

Finally, STAREPandas implements a stare_intersection() method, which yields the spatial intersection of two STAREDataFrames.

Plotting

Using STARE 's ability to look up the trixel vertices for SIDs, STAREPandas can convert SIDs and collections of SIDs to simple feature geometries. A feature with a single SID, e.g., representing an IFOV or a point, will be converted into a single triangular simple feature polygon. Features with a set of SIDs, e.g., representing a cover, are converted into a simple feature multipolygon of trixels. The `STAREDataFrame` method `make_trixels()`[52] generates a GeoPandas `GeoSeries` with each row containing the simple feature geometry of the trixels. This `GeoSeries` may be attached back to the DataFrame. STAREPandas can then use GeoPandas' rich plotting library to plot the trixels.

[52] `make_trixels()` on STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.make_trixels.html`

The ability to plot features in trixel representation has proven immensely helpful in exploring and communicating STARE-based geospatial analysis.

Parallelized /dask

Akin to dask-geopandas[53], STAREPandas uses Dask[54] to parallelize performance bottleneck functions. We implemented parallelized methods for spatial coincidence tests (intersects, disjoint), for the lookup of SIDs from latitudes and longitudes and shapely objects, and for the generation of trixels from SIDs.

3.4.5 STARELite and PostSTARE

The SDSS SkyServer (Alexander S. Szalay, Gray, et al., 2002; A. R. Thakar et al., 2004; A. Thakar et al., 2003; Budavári, Alexander S. Szalay, and Fekete, 2010) is based on a Microsoft SQL Server extended to use HTM - based spatial indices. (Kondor et al., 2014) adapt the concept to perform the indexing of geospatial (rather than celestial) objects. Similarly, we created extensions for PostgreSQL and SQLite to utilize STARE.

STARELite[55] and PostSTARE[56] are SQLite and PostgreSQL STARE extensions allowing us to perform a subset of STARE-based geospatial operations within the relational databases SQLite and PostgreSQL. STARELite/PostSTARE allow converting conventional representations of locations specified as latitude and longitude table columns or as Well-known binary (WKB) `blobs` (Ryden and Specification, 2005) (used by SpatiaLite[57], GeoPackages[58], and PostGIS[59]) of points or polygons to their STARE representation. They can further perform spatial relation tests, allowing STARE-based spatial joins of tables.

[53] `https://github.com/geopandas/dask-geopandas`
[54] `https://www.dask.org/`
[55] `https://github.com/SpatioTemporal/STARELite`
[56] `https://github.com/SpatioTemporal/StarePostgresql`
[57] `https://www.gaia-gis.it/fossil/libspatialite/index`
[58] http://www.geopackage.org
[59] `https://postgis.net/`

A useful application of STARELite is cataloging volumes of remote sensing granules that researchers often accumulate locally. This application uses STARELite to determine subsets of granules intersecting arbitrary ROIs. Further, STARELite catalogs can be used for the inverse search problem: Determining all spatially coincident granules of an individual granule. A STARELite catalog may leverage other components of the STARE ecosystem, namely STARE sidecars, which hold the trixel index values of each IFOV and a set of trixels representing the cover of each granule; STAREMaster, which is used to generate STARE sidecar files; and STAREPandas, used to load data into the databases.

3.4.6 STARE-Pods (yet another Discrete Global Grid (DGG))

STARE - PODS (Griessbaum, K.-S. Kuo, et al., 2021; M. Rilee, K.-S. Kuo, Griessbaum, et al., 2022) is an approach for geospatial sharding, i.e., partitioning/chunking geospatial data into geospatial bins. For STARE - PODS, each bin is a geographical region defined by a trixel at a relatively low level. On traditional filesystems, a bin (a shard) may be a filesystem directory named after the SID it represents and contains all data chunks that are within the trixel. For cloud object stores, which lack a hierarchical directory structure, each chunk will be stored in an object whose name or key is prepended with the SID of the bin it falls into.

To implement STARE - PODS, we first generate STARE indices for all observations of all granules (using STAREMaster). If we decide, e.g., to use quadfurcation level 4 for partitioning the data, we then repackage data elements within each quadfurcation 4 trixel into a file as a STARE chunk. If we use a directory structure to implement the STARE hierarchy, all spatially close chunks and, thus, observations will be in the same respective level 4 directory. In cloud object stores, all chunks would share the same shard-name prefix. Finding the overlaps between two or more datasets thus becomes trivial and scalable. Since our STARE API can convert arbitrary (spherical) polygons into STARE covers at any given quadfurcation level, finding overlaps between an ROI and any STARE chunk is also trivial.

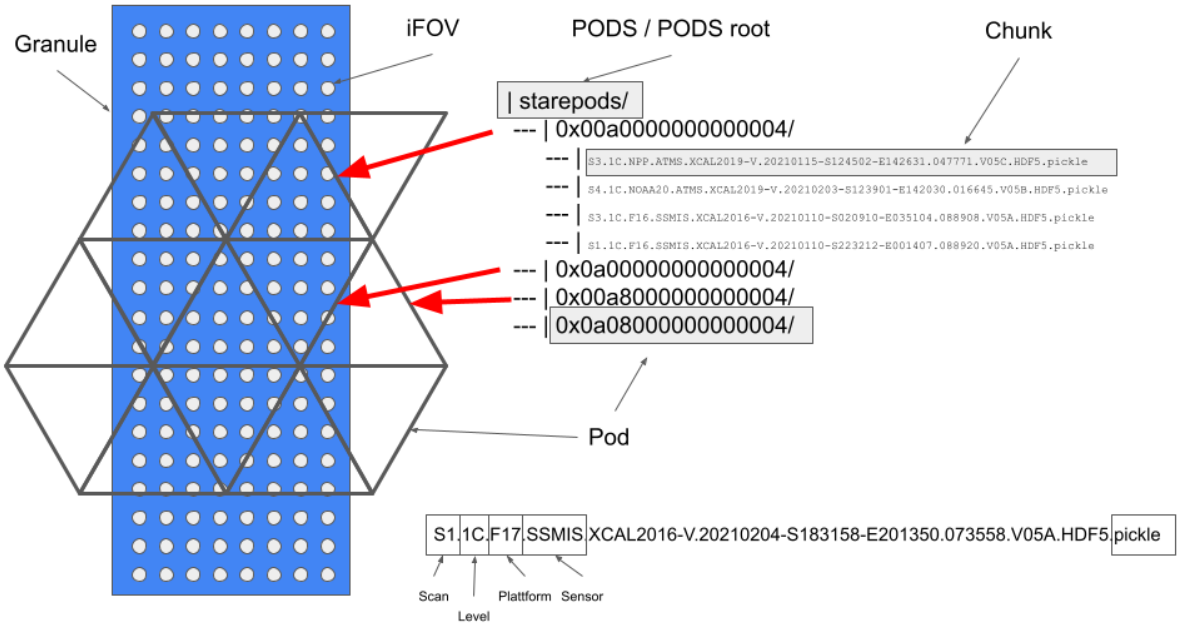


Figure 3.19: Schematic concept of STAREPods. Each observation gets binned into a shard. A shard is a trixel of a predefined STARE level, which may be resembled, e.g., a folder.

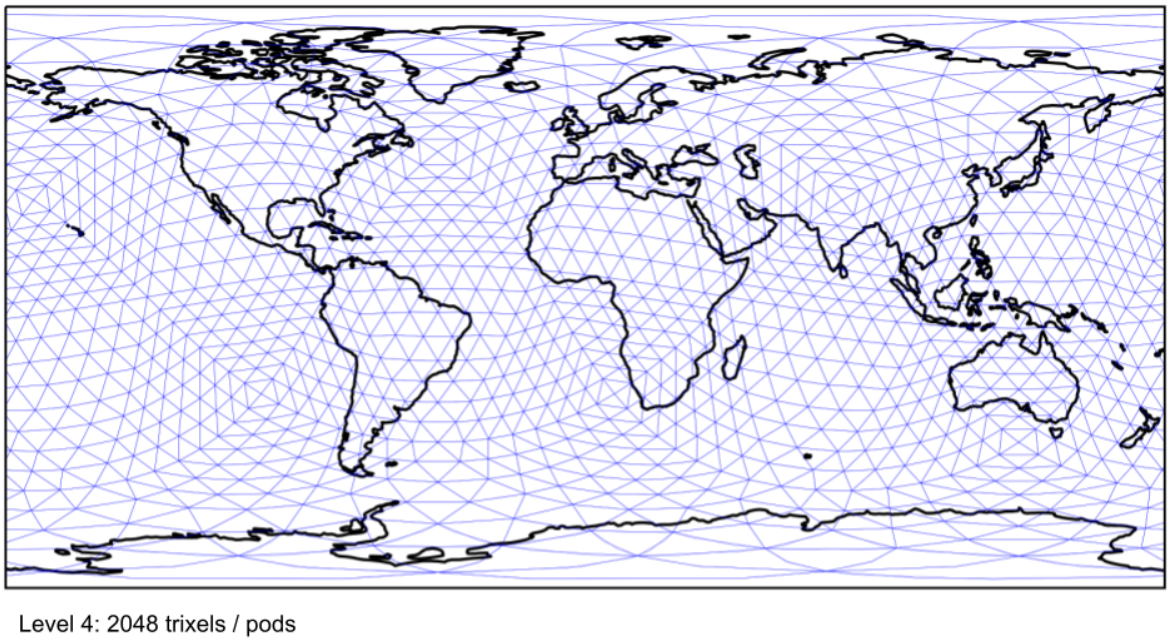


Figure 3.20: At level 4, there are 2048 shards, each with a size of about 640 km

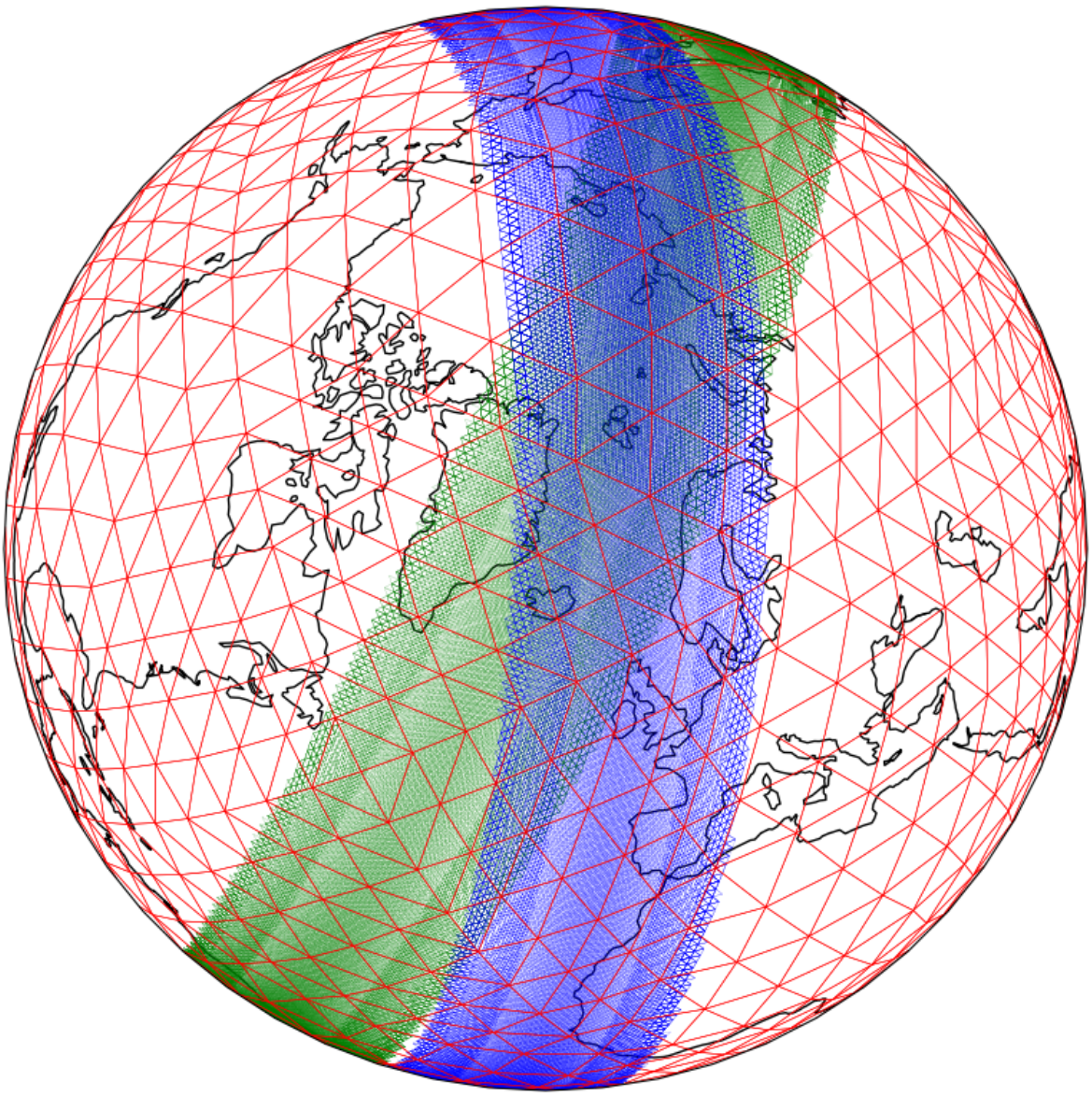


Figure 3.21: The advantage of storing data in PODS: The trixels in green and blue correspond to the IFOV of two swaths stored in two granules. In STAREPods, data from each granule get chunked into the bins/shards corresponding to the red trixels, resulting in geospatially coincident data being stored in the same location.

pod	chunk	chunk_name	ts_min	ts_max	scan	satellite	instrument
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
0x0058000000000004	/tablespace/xcal/starepods/...	S3.1C.F18.SSMIS.XCAL2016-V.20210120-S015907-E034103.058061.V05A.HDF5.pickle	2021-01-20 03:19:54.000361	2021-01-20 03:20:07.000654	S3	F18	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S4.1C.F17.SSMIS.XCAL2016-V.20210207-S160935-E175127.073599.V05A.HDF5.pickle	2021-02-07 16:27:22.000825	2021-02-07 16:28:08.000392	S4	F17	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S2.1C.F16.SSMIS.XCAL2016-V.20210122-S011346-E025540.089077.V05A.HDF5.pickle	2021-01-22 02:33:23.000094	2021-01-22 02:35:34.000105	S2	F16	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S1.1C.F17.SSMIS.XCAL2016-V.20210129-S162909-E181102.073472.V05A.HDF5.pickle	2021-01-29 16:45:45.000314	2021-01-29 16:47:37.000336	S1	F17	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S3.1C.F18.SSMIS.XCAL2016-V.20210116-S144350-E162546.058012.V05A.HDF5.pickle	2021-01-16 14:59:43.000436	2021-01-16 15:01:39.000256	S3	F18	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S2.1C.F16.SSMIS.XCAL2016-V.20210125-S003427-E021621.089119.V05A.HDF5.pickle	2021-01-25 01:55:31.000847	2021-01-25 01:57:21.000972	S2	F16	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S3.1C.F18.SSMIS.XCAL2016-V.20210123-S012048-E030243.058103.V05A.HDF5.pickle	2021-01-23 02:40:30.000924	2021-01-23 02:42:34.000340	S3	F18	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S1.1C.F17.SSMIS.XCAL2016-V.20210130-S024031-E042224.073478.V05A.HDF5.pickle	2021-01-30 04:02:17.000406	2021-01-30 04:04:17.000024	S1	F17	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S4.1C.F18.SSMIS.XCAL2016-V.20210129-S014604-E032800.058188.V05A.HDF5.pickle	2021-01-29 03:05:50.000733	2021-01-29 03:08:32.000123	S4	F18	SSMIS
0x0058000000000004	/tablespace/xcal/starepods/...	S3.1C.F17.SSMIS.XCAL2016-V.20210121-S030005-E044158.073351.V05A.HDF5.pickle	2021-01-21 04:20:47.000097	2021-01-21 04:22:35.000321	S3	F17	SSMIS

```
SELECT
      pod,
      strftime('%Y-%m-%d', ts_min) AS timestamp,
      satellite,
      instrument,
      count(chunk) AS n_chunks
FROM catalog
GROUP BY pod,satellite, instrument, timestamp
ORDER BY pod,timestamp
```

→

	pod	timestamp	satellite	instrument	n_chunks
1	0x0000000000000004	2021-01-10	F16	SSMIS	8
2	0x0000000000000004	2021-01-10	F17	SSMIS	8
3	0x0000000000000004	2021-01-10	F18	SSMIS	4
4	0x0000000000000004	2021-01-10	NOAA20	ATMS	12
5	0x0000000000000004	2021-01-10	NPP	ATMS	4
6	0x0000000000000004	2021-01-11	F16	SSMIS	8
7	0x0000000000000004	2021-01-11	F17	SSMIS	8
8	0x0000000000000004	2021-01-11	F18	SSMIS	8

Figure 3.22: A STARELite Databases used as a catalog for chunks within a PODS .

We have prototyped STARE-PODS with STARE-based data partitioning and organization on a traditional file system using cross-calibrated microwave radiometer/imager datasets produced by the NASA Precipitation Processing System (PPS), which include (cross-)calibrated brightness temperatures from 18 satellite-borne microwave radiometer/imager instruments of NASA 's Global Precipitation Measurement (GPM), such as Advanced Microwave Scanning Radiometer (AMSR) - 2, Advanced Technology Microwave Sounder (ATMS), Global Precipitation Measurement (GMI), Microwave Humidity Sounder (MHS), or Special Sensor Microwave Imager/Sounder (SSMIS). This set of data products exhibits not only data varieties across the products of different instruments but, due to different IFOV resolutions for different microwave frequencies, also within the same product of the same instrument.

For convenience, we built a catalog on top of the PODS (c.f. figure 3.22). The catalog is a STARELite database containing one row per chunk. This makes it trivial to, e.g., find all chunks that spatiotemporally intersect a ROI.

In a testing environment, we used one month's worth of XCAL SSMIS data from F16, F17, and F18 for 2021-01-10 to 2021-02-09, a total of 1314 granules. We created STARE sidecar files for

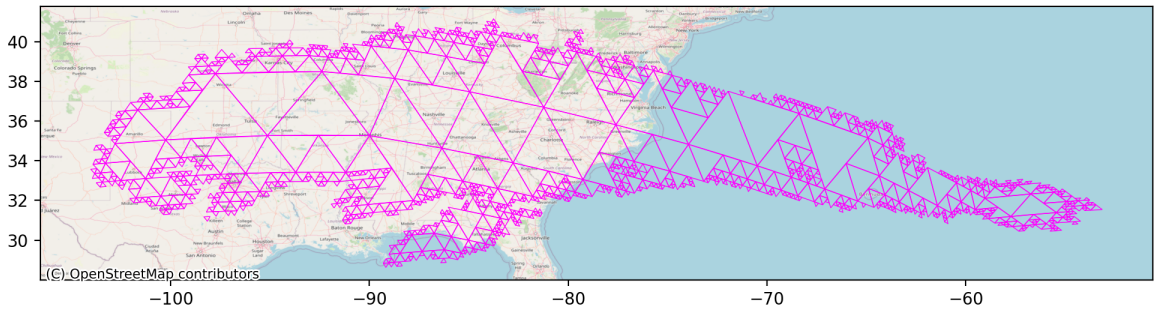


Figure 3.23: The spatial extent of a precipitation event lasting from 2021-01-24 to 2021-01-27. Note the complexity of the shape around its southern edges.

each of those granules. Each sidecar contains the SID of each observation/IFOV/pixel of the granule and a set of SIDs representing the spatial coverage of the entire granule.

We then loaded the collection of granule/sidecar pairs into a level 4 PODS using `STAREPods_py`[60]. A level 4 PODS shards data at the 4th STARE quadfurcation level. Considering the initial solid with 8 faces, a level 4 PODS has 2048 shards. The 1314 original granules are split into 3 675 403 chunks distributed over the 2048 shards. On average, each shard contains about 1800 chunks, though their distribution varies. The size of the chunks varies between 10 kB to 100 kB. Each chunk is a pickled `STAREDataFrame` containing the latitude, longitude, timestamp, SID, and the measurements for each observation.

As a benchmarking test, we evaluated the performance of loading all SSMIS data that intersects a complex spatiotemporal bounding box. As the complex spatiotemporal bounding box, we used the extent of a precipitation event over the southwestern US lasting from 2021-01-24 until 2021-01-27. Our testing environment was a m5.x4large Amazon Web Services (AWS) instance with both the granules and sidecars as well as the PODS residing on a flexFS[61] volume.

In a conventional approach, we utilize the granule naming convention (c.f. figure 3.24) to temporarily subset the granules to the temporal bounding box. This reduces the number of

[60] `https://github.com/SpatioTemporal/STAREPods_py`
[61] `https://www.paradigm4.com/technology/flexfs/`

```
S1.1C.F18.SSMIS.XCAL2016-V.20210201-S144317-E162513.058238.V05A.HDF5
```

Figure 3.24: The filename of an SSMIS XCAL granule. Highlighted is the portion of the filename indicating the timestamp of the granule.

granules from 1314 to 170 candidate granules. We then iteratively read the STARE cover from each sidecar of the candidate granules and verify if the granule spatially intersects our ROI. If not, we discard the candidate. If yes, we load the entire granule into a `STAREDataFrame` and spatially subset it to the ROI (using STARE). Out of the 170 candidate granules, 70 granules spatially intersect the ROI. Finally, we concatenate all subsetted granule `STAREDataFrames`. The whole process took a total time of 46.2 s ± 102 ms (mean ± std. dev. of 10 runs) on our testing server.

We extended the conventional approach by making use of a granule catalog. A granule catalog is a database containing the paths of granules and their sidecars, the start and beginning times of the granules as well as the spatial coverages of the granules. Using the catalog, we can immediately query for the granules that intersect the spatiotemporal ROI. We can then only load those granules and spatially subset them to our ROI. Using the catalog, we slightly improve the runtime to 44.1 s ± 103 ms (mean ± std. dev. of 10 runs)

In the STAREPods approach, we first find all the shards that may contain chunks intersecting the ROI. We can simply achieve this by converting the SIDs of STARE cover of our ROI to level 4 (the level of our PODS) and then taking the set of those SIDs. Only 29 of the 2048 shards do intersect our ROI. We then utilize STAREPandas' `read_pods()`[62] function to load the chunks into a single `STAREDataFrame`. The function `read_pods()` iterates through the candidate shards and loads all chunks whose name contains a specified pattern. Again, we utilize the granule naming convention and specify a pattern corresponding to our temporal bounding box. We finally subset the loaded `STAREDataFrame` to our ROI since it contains

[62] `read_pods()` om STAREPandas' RTD: `https://starepandas.readthedocs.io/en/latest/docs/reference/api/starepandas.STAREDataFrame.read_pods.html`

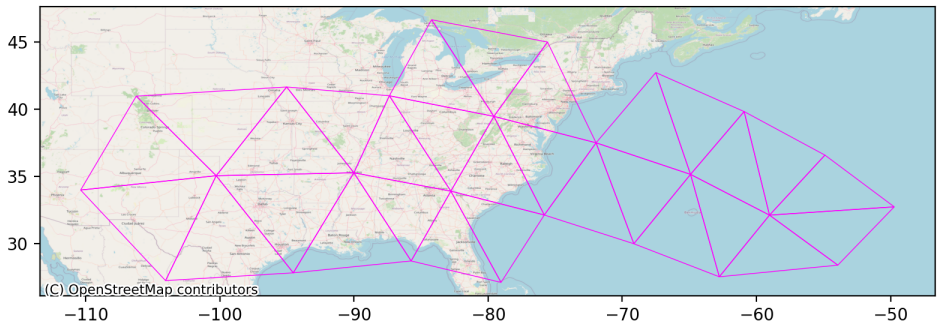


Figure 3.25: The spatial extent of the precipitation event at STARE quadfuraction level 4. There are 29 trixels (and thus shards) covering the event.

observations that coincide with the ROI at level 4, but not at the ROI at its original resolution (level 9). The whole process takes a total time of 2 s ± 15.8 ms (mean ± std. dev. of 10 runs) on our testing server, bringing us a speedup of over 20x compared to the conventional approach.

Each of the three approaches resulted in loading 225,539 individual SSMIS observations.

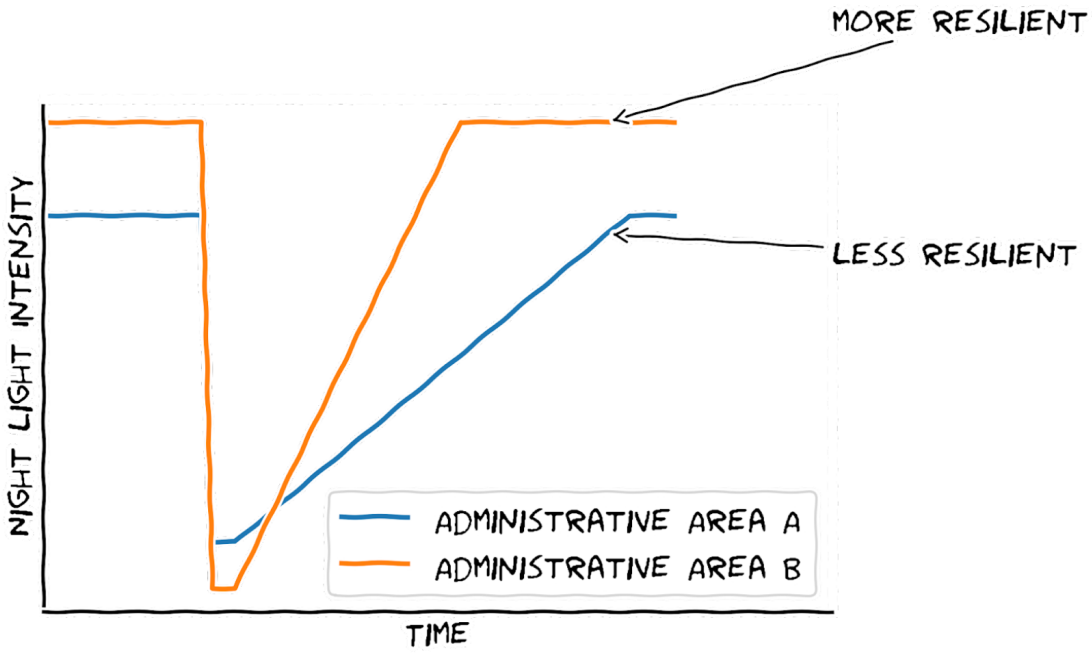
3.5 Application examples

3.5.1 Night lights

> "For more than an hour, Leigh-Cheri stared into the mandala of the sky. 'Does the moon have a purpose?' she inquired of Prince Charming. Prince Charming pretended that she had asked a silly question. Perhaps she had."
>
> From Still Life with Woodpecker (1980), Tom Robbins

(Kondor et al., 2014) demonstrated how an HTM - enabled relational database made it possible spatially to classify billions of geolocated tweets into complex political boundaries. The problem can be generalized as a challenge to associate extensive collections of irregularly spaced observations/events (the tweets) with complex (i.e., discontinuous and/or containing holes) geographic regions.

Similarly, we demonstrate STARE's capabilities by spatially associating an even more extensive collection of geolocated observations to a set of even more complex geographic regions. The challenge was to determine the characteristics of night light intensity drop and recovery during and after natural disasters in the Caribbean. The hypothesis is that communities' disaster resilience can be measured by how quickly their night lights recover (Links et al., 2018). By comparing the measured resilience of a set of administrative areas (e.g., counties), we intended to get insight into the effectiveness of different resilience-improving policies.



A similar undertaking was performed by (Wang et al., 2018), using the gridded global Bidirectional Reflectance Distribution Function (BRDF) adjusted nighttime lights data VNP46A2[63] from the VIIRS, which only became publicly available at the end of 2021. However, at the time of our undertaking, VIIRS nightlight data were only available as stray light corrected monthly composites on Google Earth Engine[64], produced according to (Mills, Weiss, and Liang, 2013)[65], and as the VNP02DNB[66] geolocated (L1B) top-of-the atmosphere (at-sensor) night light product. The temporally aggregated data was too coarse in the temporal dimension for the intended study. We reckoned that with a relatively crude correction for clouds and moonlight and making use of STARE 's ability to handle swath data gracefully, we would be able to create nightlight time series by spatially joining the VNP02DNB product with local political boundaries (in other words: classify each VNP02DNB observation into an administrative area).

We had previously demonstrated the ability to use VNP02DNB night light intensity data to

[63](System, 2019). DOI: 10.5067/VIIRS/VNP46A2.001

[64]Google Earth Engine: VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1; VIIRS Nighttime Day/Night Band Composites Version 1

[65]https://eogdata.mines.edu/products/vnl/

[66]((VCST), 2021a). DOI: 10.5067/VIIRS/VNP02DNB.002

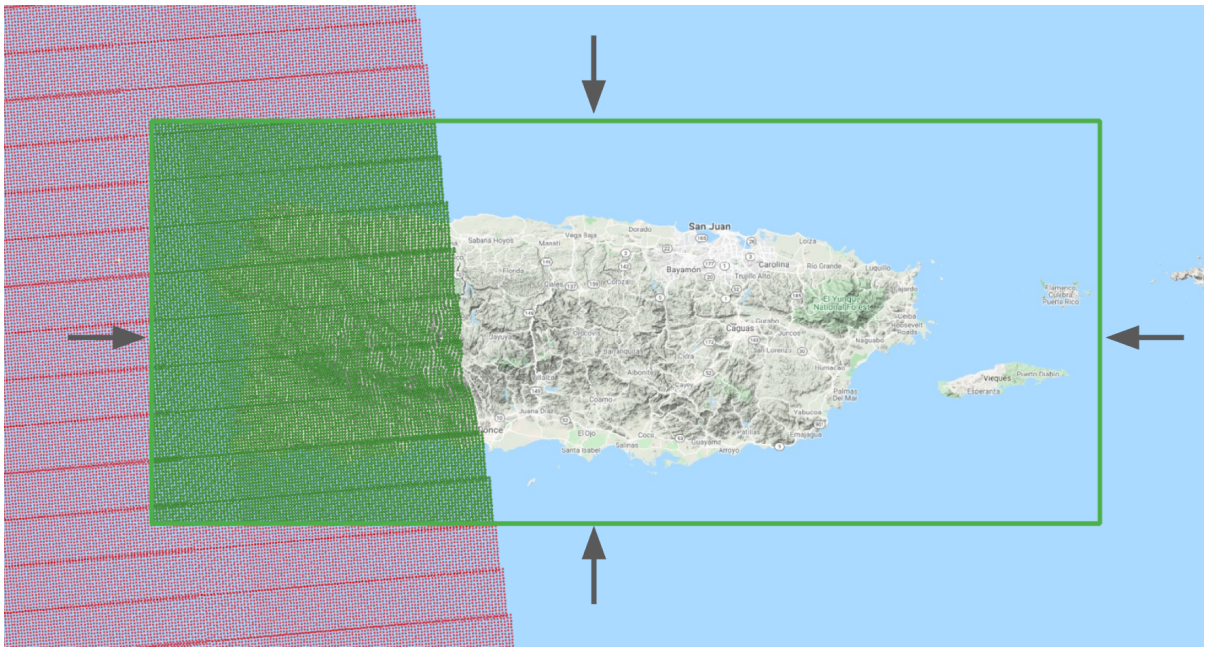


Figure 3.26: Subsetting IFOVs by thresholding latitudes and longitudes using a rectangular bounding box approximating our ROI of Puerto Rico

produce nightlight intensity time series for a relatively small region of interest (Puerto Rico) using conventional methods. By approximating the ROI with a *rectangular* bounding box, we were able to quickly subset each VNP02DNB[67] granule simply by thresholding latitudes and longitudes. We loaded the remaining data into a PostGIS database, where we geospatially mapped each observation to an administrative area (*barrio*). We then temporally aggregated the observations (per day) and averaged the individual observations to create timelines of nightlight intensities per administrative area. This approach was possible since we had a small enough ROI and one whose shape could be closely approximated by a rectangular bounding box, vastly reducing the search space and resulting in a data volume manageable in PostGIS. However, it was obvious that this could not be repeated for larger, discontinuous, more complex ROIs such as the entire Caribbean region.

[67] And the VJ102DNB. We additionally needed to retrieve the geolocation products VNP03DNB/VJ103DNB products to extract the moon illumination intensity and the cloud mask products CLDMSK_L2_VIIRS_SNPP/CLDMSK_L2_VIIRS_NOAA20.

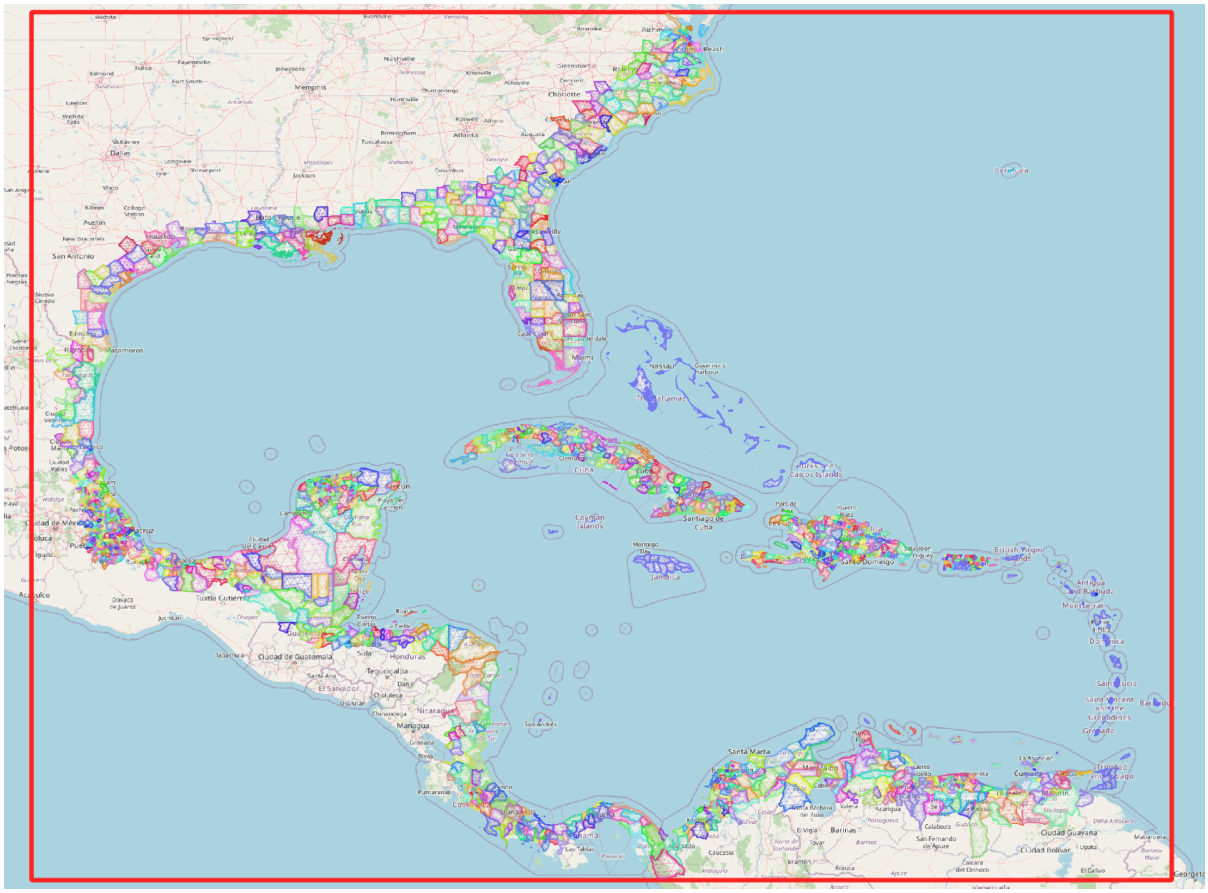


Figure 3.27: The rectangular ROI in red used to query granules from LADSWEB and the STARE representation of all Caribbean coastal second-level administrative subdivisions.

In order to produce night light intensity timelines for each administrative area of all coastal communities of the Caribbean, we utilized STAREPandas to spatially subset and join close to a trillion IFOVs of VIIRS Day Night Band (DNB) with second-level administrative subdivisions.

The undertaking was a success: With the help of STARE and STAREPandas, we were able to classify the irregularly spaced observations of the VNP02DNB (and VJ102DNB from NOAA20) as well the VIIRS cloud masks to the complex geographic regions (administrative boundaries at level 2) and were able to derive nightlight intensity time series with a temporal resolution of two observations per night (one from Suomi, one from NOAA20).

We were able to generate the STARE spatial indices of the 628 076 309 504 IFOVs from a total

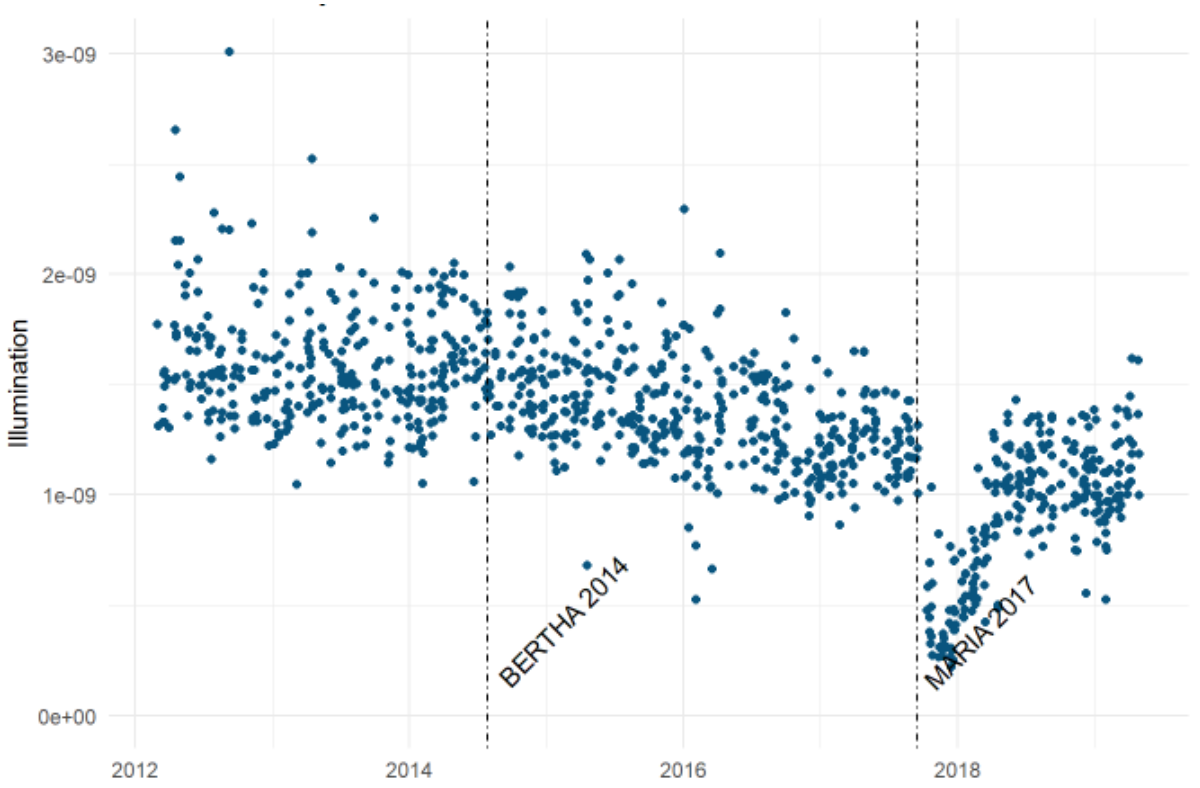


Figure 3.28: Average Night Light intensity time-series for St Vincente, clearly displaying the drop and recovery in night light intensity during and after Hurricane Maria

of 31 905 VIIRS/Suomii and 15 677 granules (each of which containing 13 199 872 observations) within approximately 42 hours (parallelized on 64 processes). Depending on the extent and the degree of the complexity of the geographic regions, the spatial joins of the indexed nightlight data with the geographic regions terminate within multiple seconds to multiple minutes.

We demonstrated the capability of STAREPandas to convert conventionally represented (i.e., ESRI shapefiles) complex feature data (the political boundaries) to their STARE representation. We further demonstrated the usability of STARECatalogs to subset the complete collection of granules that intersect the entire ROI (a rectangular bounding box in the Caribbean) to granules that intersect the political boundaries of an individual country. We finally used STAREPandas' STARE-based intersect tests to classify each observation of each granule with a political boundary at the county level.

However, we experienced issues with the final time series: Since we did not correct for moonlight but filtered observations exceeding a moonlight intensity threshold, we lost a large portion of the observations (approximately a week per month). Further, we were particularly interested in nightlight intensity during Hurricane season, a time of the year that naturally coincides with a frequent cloud presence. We had to filter out a large portion of observations due to cloud presence which reduced our temporal resolution further. In a use-case investigating the impacts of natural disasters such as blizzards or earthquakes, the cloud present may have added less interference.

At the end of 2021, the VNP46A1/2 product (Román et al., 2018) became available. A major feature of this product is that it corrects, rather than filters for moonlight, allowing for fewer data to be filtered out. In its production, it takes the following parameters into account:

- Daytime VIIRS DNB surface reflectance
- BRDF,
- Surface Albedo,
- Nadir BRDF adjusted Reflectance (NBAR),
- Lunar irradiance values

With the presence of this product, extracting nightlight dynamics as we did has become significantly simpler: Given its gridded nature and daily temporal resolution, VNP46A2 is much easier to handle for any user since it does not require processing irregularly spaced observations. It is only a matter of time until it will be available on the google earth engine for users to achieve far more sophisticated analysis. However, it remains to be stated that VNP46A2 is a model output rather than sensor observation, and further use of the product is tied to the grid in which the product is distributed. For the development of new or alternative products, working with ungridded sensor values, and thus VNP02, will continue to be necessary.

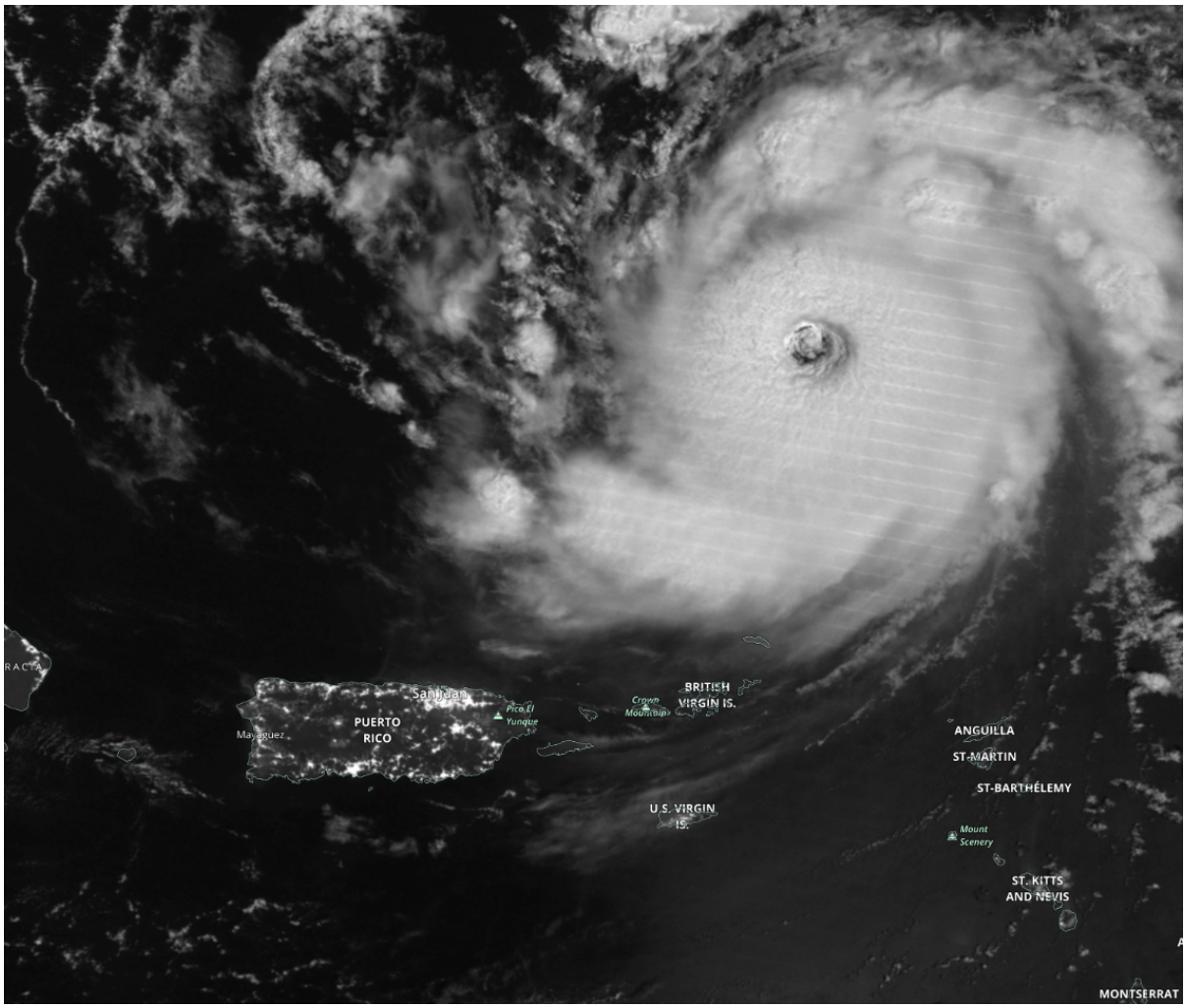


Figure 3.29: Cloud presence visible in the VNP02DNB product during Hurricane Maria approaching Puerto Rico on 2017-09-10. Image retrieved from https://worldview.earthdata.nasa.gov/ on 2022-08-29

3.5.2 Moving objects

Researchers are often interested in analyzing data that spatiotemporally intersects phenomena that evolve over time. Examples of such phenomena may be tropical cyclones, atmospheric rivers, or precipitation events. Rather than fixed locations, ROIs follow events and thus are dynamic. These regions of interest are three-dimensional spatiotemporal volumes, not just spatial extents.

Extracting spatiotemporal events from data and representing spatiotemporal ROIs are challenging. For gridded data, time slices can be stacked into a cube. By postulating that the spatiotemporal extents of moving events are contiguous volumes (i.e., an ordered set of spatial covers that are connected in time, having a spatial overlap between each subsequent time step), discrete events can then be identified and labeled by 3D Connected-component labeling (CCL). The resulting data structure is a three-dimensional cube of labels. Neither the extraction of statistical information from this data structure nor spatiotemporally joining this structure with other data is trivial.

We implemented a STARE-based moving object database approach to this issue, as described by (K.-s. Kuo et al., 2021; M. Rilee, K.-S. Kuo, Griessbaum, et al., 2022). The moving object database represents each time step of each event as an individual feature of a `STAREDataFrame`. Extracting event statistics thus becomes a mere matter of querying the `STAREDataFrame`. The spatial extent of each feature is represented as a STARE cover (i.e., a set of SIDs). This allows us to easily associate events with other spatial data, such as observations from other instruments, or static spatial objects, such as political boundaries or watersheds.

In our demonstration, we identified discrete (i.e., spatiotemporally contiguous) precipitation events from Integrated Multi-satellitE Retrievals for GPM (IMGERG)[68] data. IMGERG is a half-hourly global precipitation-rate product with 0.1° grid resolution. We then stored the

[68] https://doi.org/10.5067/GPM/IMERG/3B-HH-L/06 ; `https://disc.gsfc.nasa.gov/datasets/GPM_3IMERGHHL_06/summary?keywords=%22IMERG%20late%22`

events in our moving object database and spatially joined the database with political boundaries to calculate statistical information about the precipitation.

To extract the discrete events, we temporally stacked one month of IMGERG calibrated precipitation data. Considering the spatial resolution of 0.1° and temporal resolution of 30 minutes, this yielded us an array with dimensions 1800x3600x1440. We then applied a threshold of $1\,\mathrm{mm\,h^{-1}}$, producing a 3D binary mask of the data[69]. We then used an adaptation of the Python package, cc3d[70] (Silversmith, 2021), to check for space and time connection (subject to adjacency). Since cc3d natively only handles basic boundary conditions associated with raster arrays, our adaptation handles Earth's idiosyncratic boundary conditions: an event covering a Pole or straddling +/-180° longitude is recognized as a single event. Connected array elements are said to belong to the same episode (or event) and are assigned a unique label (e.g., an integer). The resulting array of labels is called a labeled mask, which contains the spatiotemporal coordinate information (as its array indices) for each event (c.f. figure 3.30).

The array indices can trivially be converted back into geographic and temporal coordinates. We then converted each discretized location to its STARE representation. (One also could have imagined polygonizing the set of discrete locations of every time slice of every event to produce, e.g., a vector feature for each timeslice).

Having the events represented in our STAREPandas moving object databases, we can easily overlay events with other spatial objects and calculate statistical information. For example, we may obtain attributes such as the total precipitation volume, the timings of the maximum spatial coverage, and the maximum precipitation intensity of each episode. We may, in turn, obtain distributions, and hence statistics derived from them, of these attributes for a collection of episodes satisfying some condition(s), e.g., those in a given season or those with a duration exceeding a criterion.

[69] Various other methods for identifying the presence of an event could have been applied (e.g., a compound criterion composed of multiple simple criteria or a criterion based on the threshold of a new variable derived from existing variables).

[70] `https://github.com/seung-lab/connected-components-3d/`

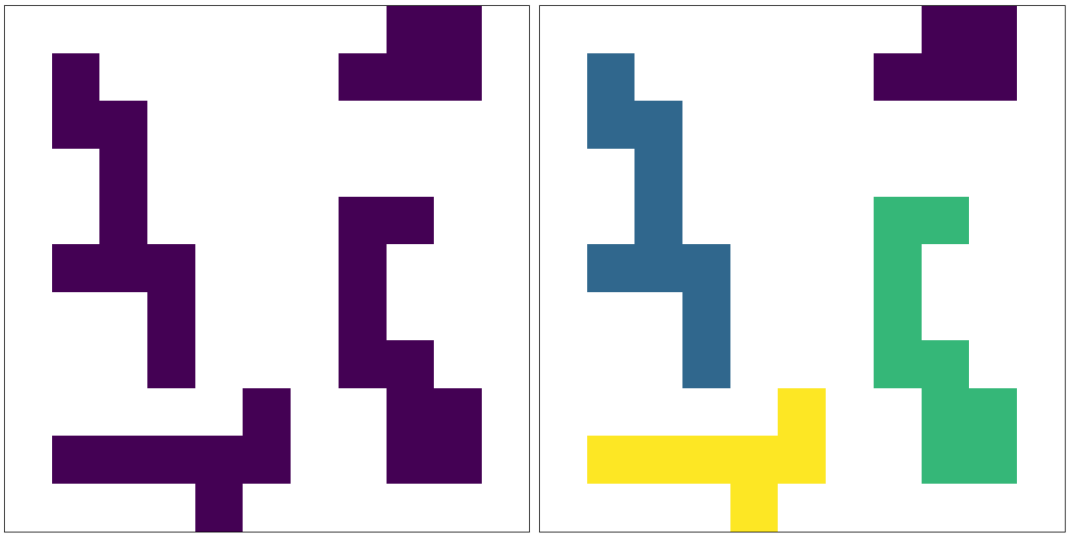


Figure 3.30: Left: Cells with binary data. This could, e.g., be cells of a lat/lon grid in which a threshold of a physical quantity has been exceeded or not. Right: The labeled mask. Cells have been spatially connected and assigned a common label; here represented as different colors.

A simple example: A particular large precipitation event over the pacific lasting from UTC 2022-05-30 15:30:00 to UTC 2022-06-08 18:30:00 had total precipitation of about 130 billion cubic meters. It intersected the San Joaquin watershed in California from UTC 2022-06-04 05:30:00 to UTC 2022-06-04 13:00:00. The total precipitation of this event over Alabama was 14 million cubic meters, about 0.0104 % of the total event's precipitation. The event is visualized in figure 3.32.

Using the STARE cover of a single event, we can also load observations from data stored, e.g., in STAREPods that are spatiotemporally coincident with an event. For example, we may subset and compare quantitative precipitation estimates derived from in-situ rain gauge networks and/or {+NOAA) NEXRAD radars. In addition, one may use wildfire episodes to select/filter data related to water management or air quality, and vice versa, to solicit relation, correlation, or even causation between phenomena. In conclusion, identifying and tracking episodes opens the door to a vast range of event-based statistics for deeper analyses that are impossible with conventional non-event-based analysis.

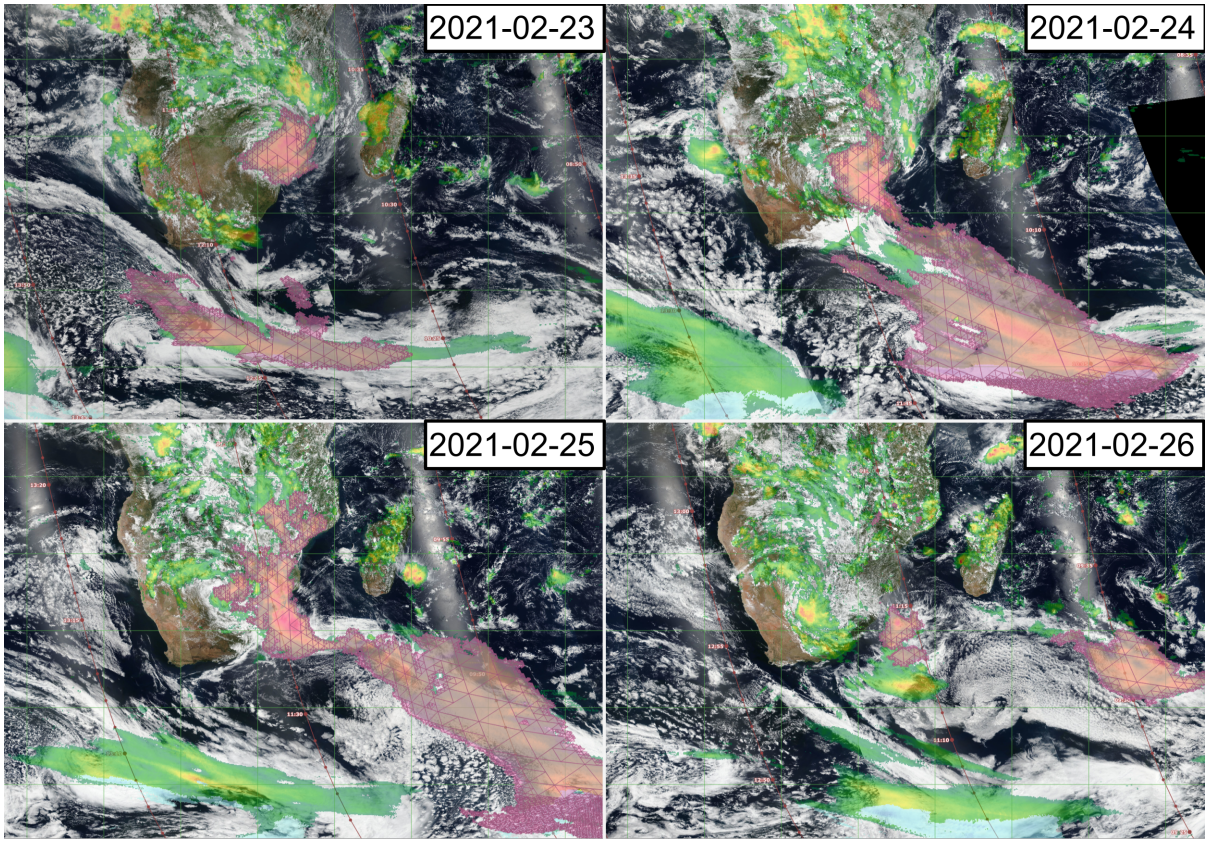


Figure 3.31: The daily-accumulated spatial extents of a four-day (23-26 January 2021) precipitation episode (as defined by a 1-mm/hour threshold of IMERG precipitation rate) are depicted with purple trixel meshes of STARE covers, which overlay the background composed of the daily VIIRS RGB composite image and IMERG precipitation rate (in mostly green and yellow shades) obtained from NASA EarthData Global Imagery Browse Services. The partial daily evolution of a precipitation episode (event) over four days in purple STARE covers based on IMERG data with a 1 mm/hour threshold demonstrating merging and splitting of the event.

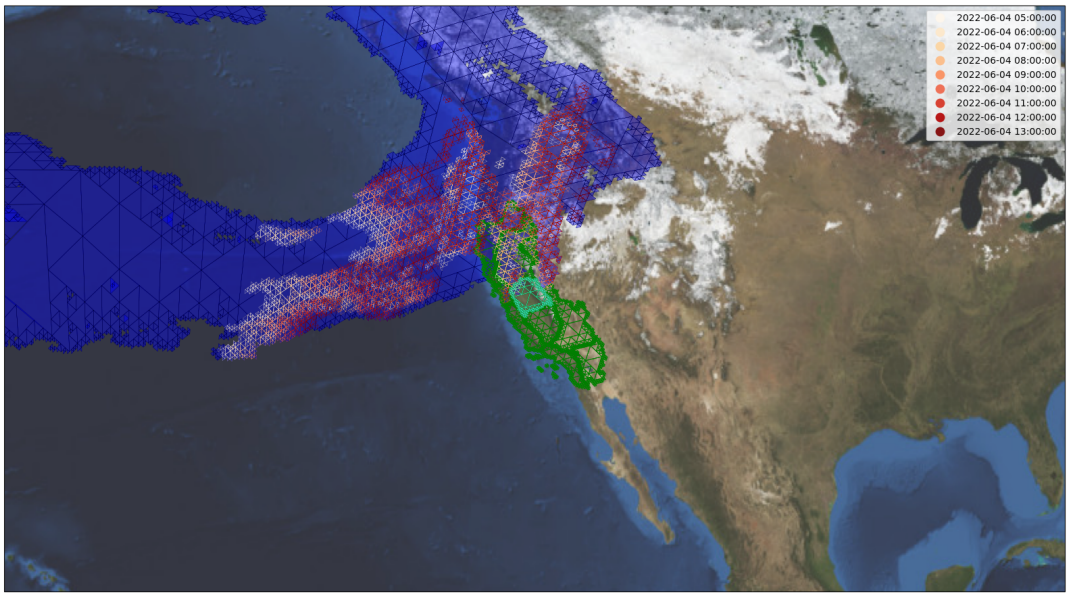


Figure 3.32: An example of using STARE covers for a precipitation event extracted from the IMGERG data. California watersheds are depicted using STARE covers (green), with San Joaquin watershed highlighted in in turquoise. The blue mesh overlaid is the cumulative area of a precipitation event, lasting from from UTC 2022-05-30 15:30:00 to UTC 2022-06-08 18:30:00. The red shaded covers are individual timesteps of the precipitation event.

3.5.3 Improving fractional snow cover estimates through increased spatial fidelity.

Chapter 4 (Griessbaum, 2022) demonstrates how STARE's ability to process remote sensing data at the sensor's resolution allows for increasing the accuracy of fractional snow cover area estimates from surface reflectance data of moderate resolution imaging spectrometers.

3.6 Outlook

3.6.1 STARECache

Calculating STARE representations for complex geographic regions remains computationally intense. We envisage a database with an extensive collection of STARE encoded geographic boundaries called STARECache. The geographic boundaries might include political boundaries, 1°, 0.5°, and 0.25° global grid cells, a Digital Elevation Model (DEM), roads, and a gazetteer. Ideally, this database would be openly accessible for download and exposed through a web API. We expect such a database to drop the usage barrier for STARE, help users understand STARE better, and provide more extensive exposure to STARE technology.

We additionally envisage a STARECache for the grids of MODIS tiles grids and the tiles themselves. Those caches may contain the x/y cell index, the ISIN (Yang and R.E. Wolfe, 2001) grid cell location, the latitudes, and longitudes of the centroids of the cells, a representative SID at an appropriate level for each cell (e.g., level 14/15 for 500 m resolution), a set of SIDs representing the cell cover at various levels (anywhere from the representative resolution up to the products geolocation precision (e.g., 50 m for MODIS: level 17/18))

Finally, we intend to expose a cache for all SIDs up to a certain level containing:

- The SID in integer and hex notation
- The area of each trixel in steradians and m^2
- The location of each trixel center in WKB (Ryden and Specification, 2005) notation, as latitude and longitude and as ECEF vector.
- The trixel in WKB notation, as well as the locations of each trixel corner as latitudes and longitudes and as ECEF vectors
- The edges of each trixel as norm vectors of their great circles.

To a certain level, such a cache could be shipped with STARE, but we would also want to expose it in a web service allowing users to do lookups. e.g.

```
/api.stare.world/getCover?name={name}&level={level}
/api.stare.world/getCellCover?name={name}&x={x}&y={y}&level={level}
/api.stare.world/getTrixel?sid={sid}
/api.stare.world/getArea?sid={sid}
/api.stare.world/getNodes?sid={sid}
/api.stare.world/getEdges?sid={sid}
```

3.6.2 STARESearch

The Level-1 and Atmosphere Archive & Distribution System (LAADS) DAAC's search and distribution platform ladsweb[71] allows users to search for data by time and location. However, it limits users to specify locations as either a latitude-longitude point, a rectangular bounding box, predefined validation sites, ISIN tiles, or countries. However, often a user will be interested in data that intersect other arbitrary shapes (a watershed, a county, a plot of land). Currently, a user's only choice is to approximate these arbitrary shapes with a rectangular bounding box (the resolution is limited to 0.1 degrees, and it is not evident if the edges are treated as great circles or as rhumb lines). This leads to more data being extracted than needed and, thus, an overhead data transfer. We envisage STARESearch, an alternative to ladsweb, which allows users to search granules for arbitrarily shaped locations. Rather than specifying a location as a bounding box, a user will specify the location as STARE cover (or as Well-known text (WKT)[72] (Ryden and Specification, 2005), which STARESearch will internally convert to the STARE cover, e.g., using STARECache). Internally, STARESearch will use a catalog that holds all granule footprints in STARE representation to find the granules that intersect the specified STARE cover.

This approach may be pushed one step further: The OPeNDAP server hyrax was "STARE-

[71] https://ladsweb.modaps.eosdis.nasa.gov/

[72] https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry

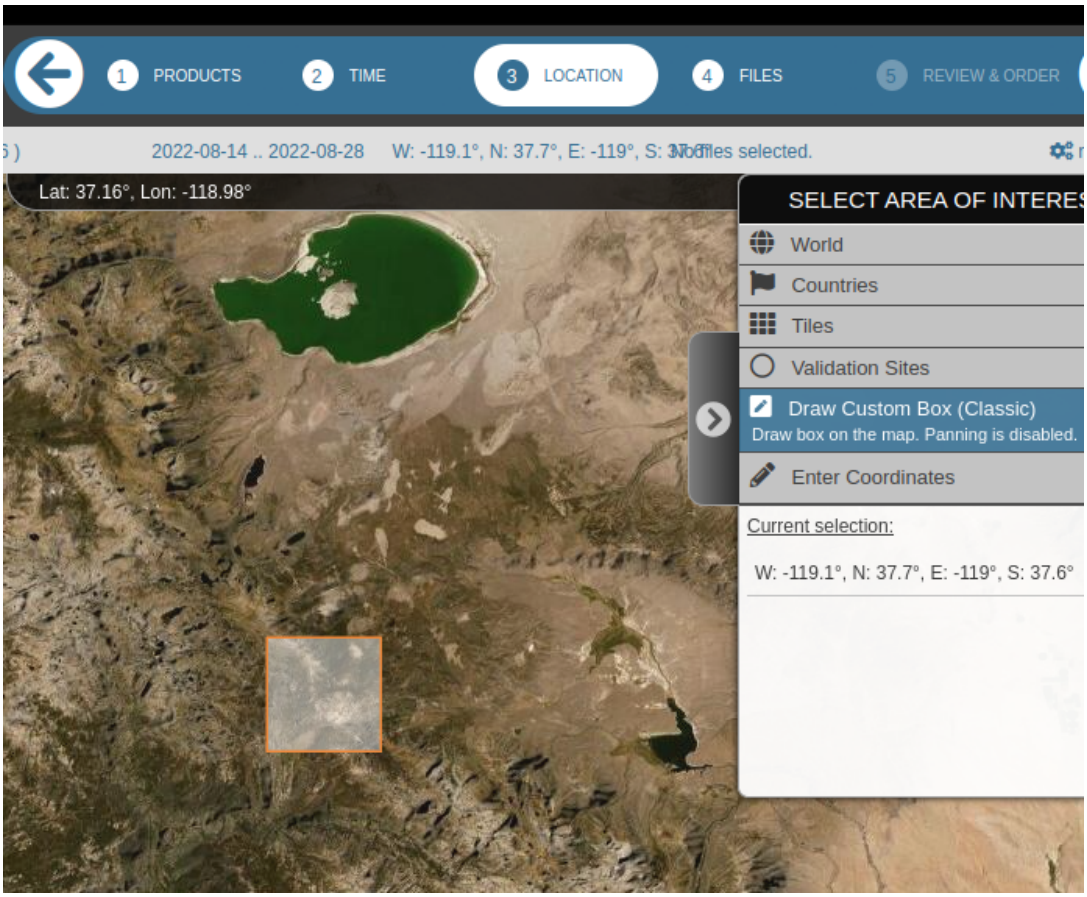


Figure 3.33: The ladsweb search interface allows users to search by locations specified either as a latitude-longitude point, a rectangular bounding box, predefined validation sites, the ISIN tiles, or countries.

enabled" during NASA's Advancing Collaborative Connections for Earth System Science (ACCESS) 2017 project "STARE: SpatioTemporal Adaptive-Resolution Encoding to Unify Diverse Earth Science Data for Integrative Analysis."[73] A hyrax server now accepts STARE covers in the constrain expressions, allowing users to subset, e.g., NetCDF or HDF files, not only by specifying array index slices but also by specifying a STARE cover. Since most publicly available LAADS DAAC product collections are available via OPeNDAP[74], a system in which a user does not merely search for granules intersecting a specified region but remotely subsets the data to the specified region could be conceived.

We could imagine a workflow as follows:

- User has an ROI that they need data for
- User looks up SID representation of the ROI (e.g., from the STARECache service or through STAREPandas)
- User sends a query to STARESearch to find all granules that intersect the ROI (specified as a collection of SIDs). STARESearch sends back all the granule file paths that intersect the ROI.
- Since a STARE enabled hyrax serves the granules, the user can request hyrax to subset each granule to the ROI (again, specified as a collection of SIDs).

As a result, we would have a system that ensures that no data transfer overhead appears.

[73] `https://www.earthdata.nasa.gov/esds/competitive-programs/access/stare`

[74] https://ladsweb.modaps.eosdis.nasa.gov/learn/using-laads-apis/

3.7 Seminal work

3.7.1 Google Earth Engine

At its core, Earth Engine is a tile database built on infrastructures such as Google Bigtable (Chang et al., 2008), Google Spanner (Corbett et al., 2012), and Google Borg (Verma et al., 2015) and a virtually unlimited amount of storage and compute nodes. Earth Engine does not re-grid during data import as a data cube would. Instead, it preserves the resolution and reference system of individual granules and breaks the original granules into tiles, which by default span 256x256 pixels. This package size is a tradeoff between reducing the number of individual tile lookups/scans/reads versus reducing the transfer of unneeded pixels. Individual tiles then are presumably inserted as rows into a Spanner Database with an appropriate index of these tiles[75]. Appropriate spatial indexes are presumably used to shard the datasets across spanservers, which in turn likely shuffle data around for load balancing. Earth Engine creates a cache of exponential lower-resolution images (pyramids) to decrease latency and reduce computing cycles. On top of this Spanner database, Earth Engine implements server-side functions that allow for per-pixel arithmetic and map and reduce operators. Keeping in mind that no re-gridding but rather tiling appears within Earth Engine, it begs the question of how swath data and data variety are handled. The simple answer is, respectively, not at all and arguably not gracefully.

Without re-griding or the ability to express the size of each pixel, Earth Engine has no way of representing and thus visualizing and operating on wide swath data for which pixel sizes vary in respect to the position along scan lines. The fact that Earth Engine advertises itself as no re-gridding, therefore, is only part of the truth: While it is true that it does not do re-gridding itself, it handles exclusively data that already is gridded by the provider (e.g., Land Processes Distributed Active Archive Center (LPDAAC), United States Geological Survey (USGS)).

[75]For all we know, this index might be a bbox btree. Free and Open Source Software (FOSS) implementations of a tile database, including their index, can be found in QuadTiles (`https://wiki.openstreetmap.org/wiki/QuadTiles` and mapnik.)

The sheer availability of computing power combined with the high performance of the underlying stack and the lazy evaluation paradigm employed by Earth Engine makes it feasible to combine data of different reference systems and resolutions ad-hoc. Notably, the range of resolutions of datasets in the Earth Engine data library is within a few orders of magnitude. It appears that this is achieved through re-gridding, contradicting earth engine's claim not to require regridding.

Due to the employment of lazy computing, the re-gridding is sufficiently performant and mainly opaque to the user while browsing through the combined dataset. Lazy computing means that if a user wants to evaluate, e.g., the difference of NDVI calculated from MODIS and NDVI calculated from Landsat 8, the actual difference values are always evaluated under the constraint of an extent and a resolution. The extent is either implicitly provided through the extent, and zoom level of the user interface or, more explicitly, during data export (users implicitly define new grids to export combined data by specifying the export projection and resolution). The cost for the re-gridding, however, materializes during exports, where lazy computing is not an option. High latencies are observable, and a high spatial limitation is in place: The API will block requests stretching over areas in the order of $10\,\mathrm{km}^2$ to $100\,\mathrm{km}^2$. It remains up for speculation what techniques and indexes Earth Engine employ to tune the co-registration performance.

Earth Engine opens up the ability to work on remote sensing for a broad set of users. Considering the virtually infinite horizontal scalability and the sophisticated stack it is built on, it is most likely an ideal tool for processing (or at least pre-processing) single-sensor analysis. However, users are limited to gridded (L2B and L3) products. The underlying implicit re-gridding processes required to handle data variety are not necessarily evident to the user and likely will not gracefully handle the combination of extensive datasets at high spatial resolution.

3.7.2 S2geometry / Hilbert curves

Like STARE, S2geometry[76] is a project attempting to increase the harmony of geospatial data. S2geometry provides spatial data types represented as ECEF vectors. Like STARE, s2geometry promises efficient evaluation of spatial predicates, allowing spatial association of diverse data. Contrary to STARE (which uses an HTM), S2geometry uses Hilbert Space-Filling curves (Lawder and P. J. H. King, 2001) as a spatial index. The S2geometrie's developer guide notes that conversions between points and s2cells are orders of magnitude faster than conversions between HTM nodes and points[77]. We did not verify this claim, nor do we consider this as a relevant drawback of STARE. In the remote sensing applications, we intend STARE to be used for, conversions between conventionally specified locations (e.g., as lat/lon) will be carried out only once (currently at the beginning of an analysis by the users; in the future, centrally by the repository), amortizing the increased lookup costs. The critical question for the application in remote sensing data analysis is how efficient spatial coincidence of data with intra- and inter-variable resolution can be evaluated and how the index can be used for data placement in shared-nothing infrastructures.

Like STARE, S2geometry provides python bindings to the API. However, a significant boilerplate has to be created to perform geospatial analysis using those bindings. On the contrary, in our development of the STARE software stack, we focused on minimizing the effort to perform remote sensing data analysis by adding support to ingest legacy data and mimicking well-known APIs.

[76] `https://s2geometry.io/`

[77] `https://s2geometry.io/devguide/s2cell_hierarchy`

3.8 Discussion and Conclusion

We identified that the inability of conventional methods to performantly evaluate geospatial coincidence between large collections of irregularly spaced geographic objects forces scientists to analyze spatially discretized and sampled remote sensing data rather than sensor data directly. We demonstrated how the STARE software collection empowers users to circumvent this roadblock and enables them to work directly with (calibrated) sensor data. Bringing the scientists closer to the actual observations allows for geospatial analysis with higher spatiotemporal fidelity, leading to insights that would otherwise have required high cost and effort in ETL pipeline development. In our use, the STARE software collection allowed us to utilize the full spatiotemporal resolution of irregularly spaced observations from MODIS and VIIRS, helping us to derive results that would have been inaccessible using spatiotemporally discretized data. With the development focus on low entry hurdles and support of legacy data, we hope that the STARE software collection will experience growth in its user group. The development of the STARE software collection stands only at its beginning. Significant efforts had to be put into enabling the conversion of conventional spatial representations and container (file) formats into STARE representations. In a future with a higher proliferation of STARE, we hope that observation and model data will be distributed alongside STARE representations of the geolocation. The main focus on the future development of the STARE software collection will lay on performance improvements and the integration of further STARE-based geoprocessing methods.

Acknowledgements

We appreciate the founding through NASA's ACCESS 17 program[78] (Award number: 80NSSC18M0118), which allowed us to develop the core functionality of the STARE software collection. We thank

[78] `https://www.earthdata.nasa.gov/esds/competitive-programs/access/stare`

our project collaborators at Bayesics LLC and OPeNDAP and Robert Wolfe. We thank our colleagues Gabriela Alberola and Mark Buntaine at the Bren School for providing exciting use cases.

Chapter 4

Improving fractional snow-coverd area estimations through increased spatial fidelity

Abstract

Gridding of remote sensing products discretizes space and thus makes the evaluation of geospatial coincidence trivial. This dramatically simplifies the development of algorithms that require multiple observations of a single location as their input and further allows for easy algorithm accuracy evaluation against ground truth data. However, the loss in location precision can lead to unnecessary noise in algorithm outputs. The Snow Property Inversion from Remote Sensing (SPIReS) algorithm estimates fractional snow-covered area (fSCA) from surface reflectance observations using a snow-free observation of the same location as a reference. We demonstrate how the discretization of Moderate Resolution Imaging Spectroradiometer (MODIS) surface reflectance data in gridded products leads to spatial mismatching that propagates errors into the estimation of fSCA. We employ an approach forgoing gridded products and instead use the full spatial accuracy of MODIS and Visible Infrared Imaging Radiometer Suite (VIIRS). Our approach uses a Hierarchical Triangular Mesh (HTM) to represent the locations of individual ungridded observations, allowing us to spatially match fractionally snow-covered observations accurately with snow-free reference observations. This reduces the mean absolute error (MAE) of fSCA estimates from 0.064 to 0.037.

4.1 Introduction

Due to its high reflectance and spatial extent, snow is an important factor in Earth's radiation balance and hence the climate (Durand et al., 2017; Hansen and Nazarenko, 2004). Further, significant portions of Earth's population rely on water originating from snowmelt (Barnett, Adam, and Lettenmaier, 2005; Durand et al., 2017).

It is, therefore, crucial to understand, estimate, and predict the spatial distribution and properties of snow, requiring spatially resolved measurements of the snowpack in terms of extent (cover), depth, density, water content (summarizable in the Snow Water Equivalent (SWE)), temperature profile, and albedo (Dozier and Painter, 2004).

Traditional ways of measuring the snowpack are snow pillows, snow courses, and metrological surveys. While these measurements allow for detailed insights into the snowpack's properties, they are sparse, infrequent, and not necessarily representative in inhomogeneous terrain.

Conversely, remote sensing can provide spatiotemporally continuous data on the global extent of snow (Dozier and Painter, 2004; Nolin, 2010): Snow extent can e.g. be retrieved from multispectral surface reflectance data. (Lettenmaier et al., 2015) suggests spatial resolutions of snow extent not coarser than $\approx 100\,\mathrm{m}$ and temporal resolution of not more than one week. The required spatiotemporal resolution exceeds the spatiotemporal resolution of individual multispectral surface reflectance data of spaceborne remote sensing instruments. Since the launch of Landsat 9 in 2021, the combination of Landsat 8/9 and Sentinel-2A/B provides 20 m to 30 m resolution imagery at 3-day average intervals (Each Landsat has a repeat interval of 16 days and each Sentinel repeats at 20-day intervals). However, to this date, the time ranges of those datasets are relatively short. The moderate resolution sensors Moderate Resolution Imaging Spectroradiometer (MODIS) and Visible Infrared Imaging Radiometer Suite (VIIRS) on the other hand have been operational for multiple years (in the case of MODIS, decades). Their repeat interval is about one day each. However, they lack spatial resolution. Since their pixels (aka Instantaneous Field of Views (IFOVs)) are too large to observe pure constituents,

it, therefore, is necessary to map snow cover at sub-pixel accuracy (Dozier and Painter, 2004).

Several algorithms to classify pixels into 'snow' or 'non-snow' (i.e., binary snowmaps) as well as algorithms to estimate fractional snow-covered area (fSCA) (i.e., sub-pixel) from multispectral surface reflectance data exist (Nolin, 2010). Both snow and clouds are highly reflective in the visible part of the spectrum. However, contrary to clouds, snow is highly absorptive in the Short Wave infrared (SWIR) part of the spectrum, allowing us to distinguish snow from clouds by using the ratio of visible and SWIR (Crane and Anderson, 1984) surface reflectances. With the launch of Landsat 4 Thematic Mapper (TM), which included sensors for SWIR, it became possible to discriminate snow from clouds on a global scale for the first time. In this context, (Dozier, 1989) introduced the normalized differences of a visible band and a SWIR band (later termed Normalized Difference Snow Index (NDSI) by (Hall, Riggs, and Salomonson, 1995)) to identify snow. The appeal of NDSI lies in its simplicity: An observation/pixel is identified as snow if its NDSI exceeds a threshold, typically 0.4 (Dozier, 1989; Hall, Riggs, and Salomonson, 1995).

$$NDSI = \frac{R\lambda(VIS) - R\lambda(SWIR)}{R\lambda(VIS) + R\lambda(SWIR)}$$

A challenge in snow-cover mapping is trees obscuring the snow beneath the canopy. (Klein, Hall, and Riggs, 1998) introduced a combination of Normalized Difference Vegetation Index (NDVI) and NDSI to reduce the error of snow cover detection in dense vegetation. The approach was adapted by (Hall, Riggs, Salomonson, et al., 2002; Hall, Riggs, and Salomonson, 2001) to introduce the operational global level-3 snow mapping products for MODIS and VIIRS (MOD10A1[1]/VNP10A1[2]). The approach also includes thermal masks to identify "spurious snow": A pixel is determined not to be snow if its temperature is greater than 277 K.

While the NDSI itself should not be interpreted as fSCA (Stillinger, Rittger, et al., 2022),

[1](Hall and Riggs, 2016). DOI: 10.5067/MODIS/MOD10A1.006

[2](Riggs et al., 2019). DOI: 10.5067/VIIRS/VNP10A1.001

(Salomonson and Appel, 2004; Salomonson and Appel, 2006) developed a regression-based approach to infer fSCA from NDSI: The model is fitted with binary snowmap data from Landsat Enhanced Thematic Mapper (ETM) and results in the following relationships for MODIS/Terra:

$$fSCA = -0.01 + (1.45 * NDSI)$$

This regression approach fails in the transitional periods during accumulation and melt, overestimates fSCA in some areas of the world while underestimating it in others, and has a high median error (Rittger, Painter, and Dozier, 2013).

Higher accuracy of fSCA estimations can be achieved through mixture analysis of multispectral measurements (Stillinger, Rittger, et al., 2022). Spectral mixing assumes that a measured spectrum is a combination of multiple constituent ("endmember") spectra. The measured spectrum is decomposed into the spectra of the constituents, allowing the determination of the proportionate contributions of each constituent to the mixed spectrum (Dozier, 1981; Dozier and Painter, 2004). Hence, spectral unmixing provides a method to retrieve sub-pixel detail (Keshava, 2003).

Specifically, we may assume that the reflectance spectrum R that a sensor observes is a linear spectral mixture of the reflectance spectra R_k of the constituent endmembers k within a pixel.

$$R_\lambda = \epsilon_\lambda + \sum_{k=1}^{N} f_k * R_{\lambda,k}$$

R_λ is the observed reflectance at wavelength λ that is modeled as the weighted sum of the constrained weights/fractions f_k of the endmember k with a reflectance of $R_{\lambda,k}$ and the residual error ϵ_λ. Using a library of endmember reflectance spectra, we then may find the endmember fractional combination (i.e., all f_k) that minimizes the square error of the linear combination.

$$minimize \sqrt{\sum_{k_\lambda=0}^{n} \epsilon_\lambda^2}$$

(Painter, Dozier, et al., 2003) describe Multiple Endmember Snow-Covered Area and Grain Size (MEMSCAG), a method derived from Multiple Endmember Spectral Mixture Models (MESMA) (Roberts et al., 1998), to obtain subpixel snow cover, grain size, and albedo for Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) pixels using spectral unmixing. (Painter, Rittger, et al., 2009) describes MODIS Snow-Covered Area and Grain size (MODSCAG), an progression of MEMSCAG to work on multispectral MODIS pixels rather than hyperspectral AVIRIS pixels. MEMSCAG and MODSCAG use endmembers libraries of snow, different rock and soil types, vegetation, and shade. The reflectances $R_{\lambda,k}$ of the rock/soil vegetation endmembers are measured in the field and the laboratory, while the snow endmembers are modeled for varying grain sizes and solar zenith angles.

The Snow Property Inversion from Remote Sensing (SPIReS) algorithm follows a similar approach. However, rather than solving for the non-snow endmembers, SPIReS exploits the fact that for any given location (in the following referred to as a grid cell), the (mixed) non-snow endmember spectrum R_0 can be measured during the summer[3]. SPIReS uses a snow-free endmember reflectance library containing a single snow-free spectrum for each grid cell. This single snow-free reflectance spectrum is selected from all measured spectra for a given grid cell subject to a set of criteria: The snow-free spectrum must not be quality flagged, be cloud and cloud shadow-free, and have an NDSI of less than zero. From the spectra that pass those criteria, the spectrum with the highest NDVI is selected as the snow-free reference spectrum[4]. Additional advancements of SPIReS include a correction for canopy cover, persistence filters to eliminate false-positive caused by cloud presence, temporal smoothing, and cell clustering in which similar cells are grouped prior to computing to improve performance.

[3]This, of course, excludes regions of permanent snow cover, such as the arctic regions or glaciers

[4]If no spectrum with an NDSI of less than zero exists for a given cell, the spectrum with the lowest band-3 reflectance is selected.

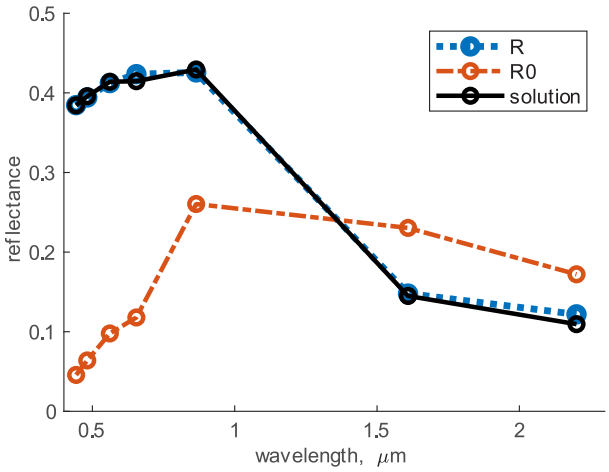


Figure 4.1: The measured spectrum R which was observed under fractionally snow-covered conditions. The snow-free reference spectrum $R0$ was observed under snow-free conditions. SPIReS solved for the solution and thereby was able to estimate the fSCA, the fractional shade, the snow grain size and the Light absorbing particle (LAP) concentration. Figure 2 from (E. H. Bair, Stillinger, and Dozier, 2021), which was licensed under Creative Commons Attribution 4.0 License.

4.2 SPIReS uncertainty and possible causes

> "location, location, location."
>
> Lord Harold Samuel

Both MODSCAG and SPIReS use *gridded* (i.e., level 3) surface reflectance products as their inputs (e.g., MOD09GA[5]). Gridded products bin irregularly spaced observations into a discretized space. Using gridded products greatly simplifies verification efforts in which snow cover estimates derived from other instruments (e.g., high-resolution binary snowmaps) have to be spatially associated with the snow cover estimates from SPIReS. In the case of SPIReS, the (fractionally) snow-covered observations also have to be spatially associated with snow-free observations of the same location, which is trivial using a gridded product and challenging for an ungridded product.

While the gridded products bring the above-stated simplifications, they also introduce a source of uncertainty. The "level 3" MODIS MOD09GA granules are produced by gridding and compositing the irregularly spaced "level 2" MOD09[6] observations. The methodology first bins each MOD09 observation into a grid cell and then selects one of those binned observations as the best (subject to clouds, viewing geometry, and others)[7] (R.E. Wolfe, Roy, and E. Vermote, 1998; Yang and R.E. Wolfe, 2001).

The spatial binning of MOD09GA reduces the spatial resolution by one order of magnitude: While the geolocations of individual IFOVs are precise to approximately 50 m (R. E. Wolfe et al., 2002), they get binned into grid cells of approximately 500 m. Figure 4.2 shows the wide spread of MODIS IFOVs center locations associated with a single cell.

[5](Eric Vermote and Robert Wolfe, 2021). DOI: 10.5067/MODIS/MOD09GA.006

[6](M. L. S. Team, 2017). DOI: 10.5067/MODIS/MOD09.006

[7]MOD09GA actually is an "Gridded Level-2 (L2G)" product, a hybrid between level 2 and level 3. In addition to the level 3 data, L2G products contain "additional observations" for each cell. Those additional observations are values binned to a cell but not selected as the best observation in the composition step.

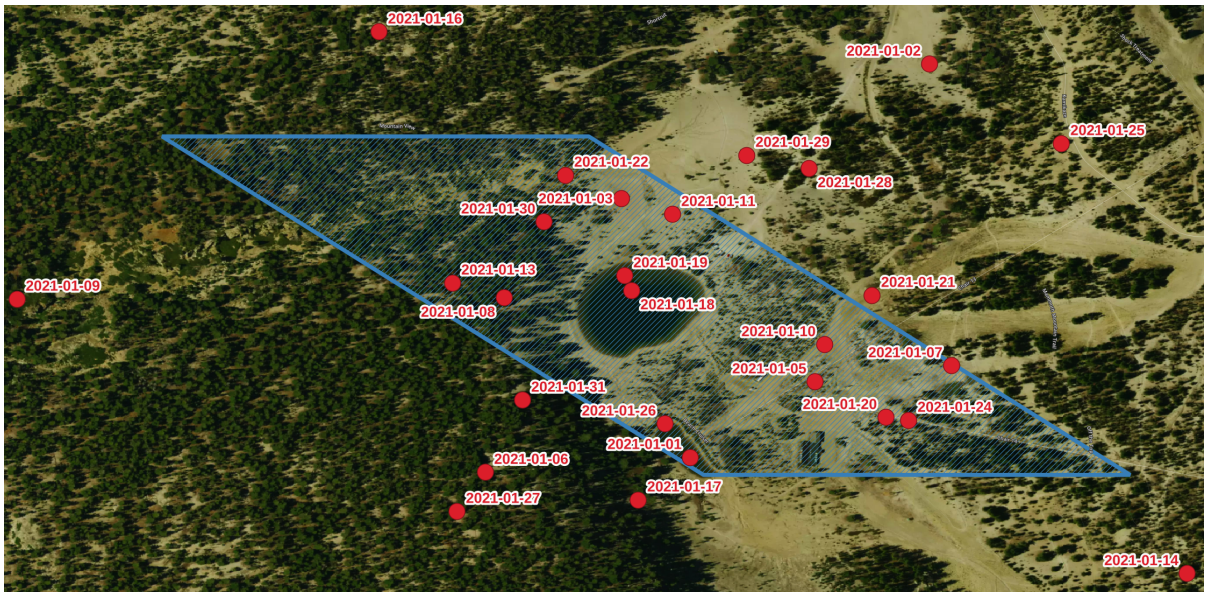


Figure 4.2: Geolocation of MOD09 observations at 500 m resolution associated with a single MODIS grid cell in the Region of Mammoth Lakes, California for January 2021.

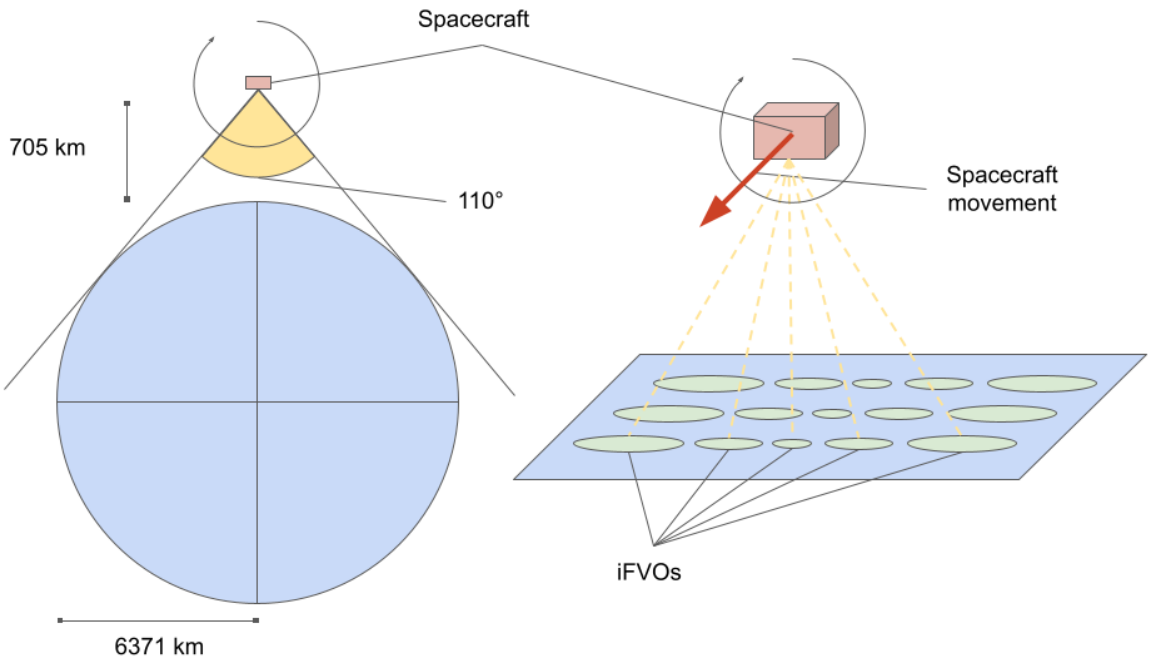


Figure 4.3: Schematic visualization of MODIS' viewing geometry and the effects of the wide scan angle on the size and shape of (ˆIFOVs).

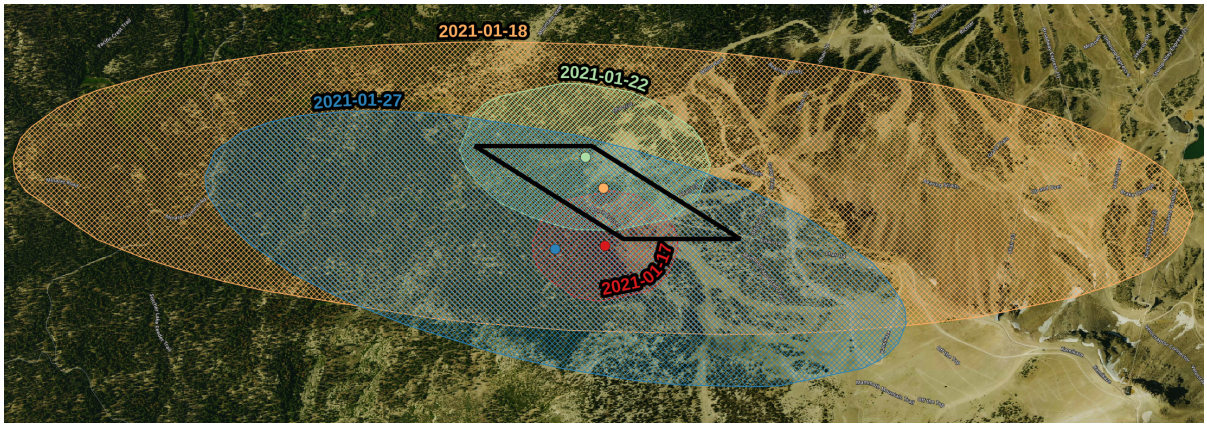


Figure 4.4: Center locations (colored dots) and estimated footprints (colored polygons) of 4 IFOVs in January 2021 associated with the same MODIS cell (black outline).

However, it is not only the center locations that differ for each IFOV. MODIS is a passive imaging spectroradiometer using a continuously rotating double-sided scan mirror. The mirror's rotation axis is in the same plane as the spacecraft's ground track. At a spacecraft altitude of 705 km, the Earth is in sight for about 110° of the mirror's rotation/scan (Barbieri et al., 1997). Consequently, observations are retrieved under viewing/sensor nadir angles ranging from $-55°$ to $55°$[8] (c.f. figure 4.3). When the sensor is at a viewing angle of $\pm 55°$ (at the beginning/end of a scan), the observed footprint is actually about $10\times$ larger via a combination of the telescopic effect in both along- and cross-track directions and an additional cross-track increment (Dozier, Painter, et al., 2008) than when the sensor is pointing straight down at nadir/0°.

The actual extent/footprint of each IFOV thus varies significantly in size and location, which means that two IFOVs binned into the same grid cell may have been for two very different areas. Figure 4.4 shows approximations of the extent of four IFOVs associated with the same cell. It visualizes the significant differences in the approximated extents/footprints that the IFOVs cover depending on the sensor's viewing angle.

Specifically, in the mountains, characterized by high topographic variability, the irregularity of

[8] This refers to the nadir viewing angle at the satellite. The sensor viewing angle (sensor zenith), from Earth surface, is about 65° at the edge of the scan. For a plane-parallel geometry, sensor zenith and nadir would be the same, but not for a sphere or an ellipsoid.

the actual IFOV footprints leads to significant noise in derived estimates. Figure 4.5 displays the time series of NDSI, NDVI, and fSCA estimates from SPIReS[9] for a *seemingly* fixed location: a MODIS grid cell at Reds Lake in Mammoth. We observe strong NDVI, NDSI, and fSCA fluctuations. We also note that the estimated fSCA stays well above 0 during the late summer months, for which we know the area was snow-free. Uncertainties in the atmospheric corrections, cloud cover and shadow, smoke presence, and variations in the solar and sensor zenith may explain some of the noise. In the following, we will demonstrate that the gridding introduces a large portion of the noise: The estimands in figure 4.5 are only seemingly for a fixed location. In reality, the underlying observations' footprints dramatically vary in size and center location. It is thus not that the fSCA estimates from SPIReS necessarily are inherently noisy; the estimates simply were made for varying footprints, some of which having larger fSCA than others.

The question consequently arises of how the accuracy of any derived product, in general, and the fSCA estimations, in particular, should be evaluated. Gridded products may tempt the assumption that each fSCA estimate is for the footprint of a grid cell. Under this assumption, we may find observations of a (higher resolution binary) ground-truth dataset intersecting the grid cell and compare the fSCA estimation with the ground-truth data. The assumption, however, is false and causes the accuracy of the fSCA estimates to appear much worse than they are. Instead, the actual footprint of each observation has to be considered, and the ground truth observations covering this footprint have to be found for the evaluation. Figure 4.6 visualizes the issue: While the fSCA is computed for the actual IFOVfootprint (red), it will be evaluated against data covering the cell if we naively use gridded data.

The location uncertainty introduces an additional issue specifically for SPIReS: SPIReS does not solve for the snow-free endmembers of a fractionally snow-covered observation but instead uses a snow-free observation of the same location as the snow-free endmember. If gridded products are used, the snow-free observations for the "same" location is a (snow-free) observation associated with the same grid cell. Since the gridding blurred the location of the footprints of

[9]The fractional snow cover estimates are for visible snow only.

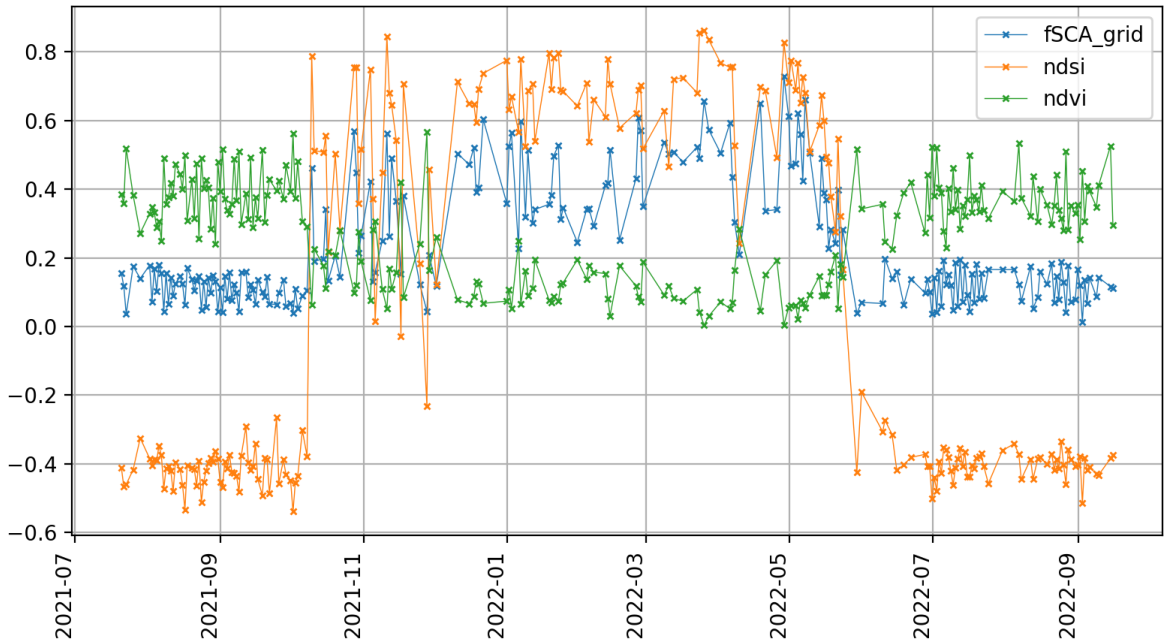


Figure 4.5: NDVI, NDSI, and (visible) fSCA estimate from SPIReS over a 'fixed' location in Mammoth lakes at Reds Lake (tile H08V05, cell x=1373; y=566) over the snow season 2021/2022. Note the strong fluctuations of all measures and a $fSCA > 0$ even during the late summer months. None of the high-frequency fluctuations of any of the measures are plausible. Note: The displayed fSCA are uncorrected visible fSCA estimates.

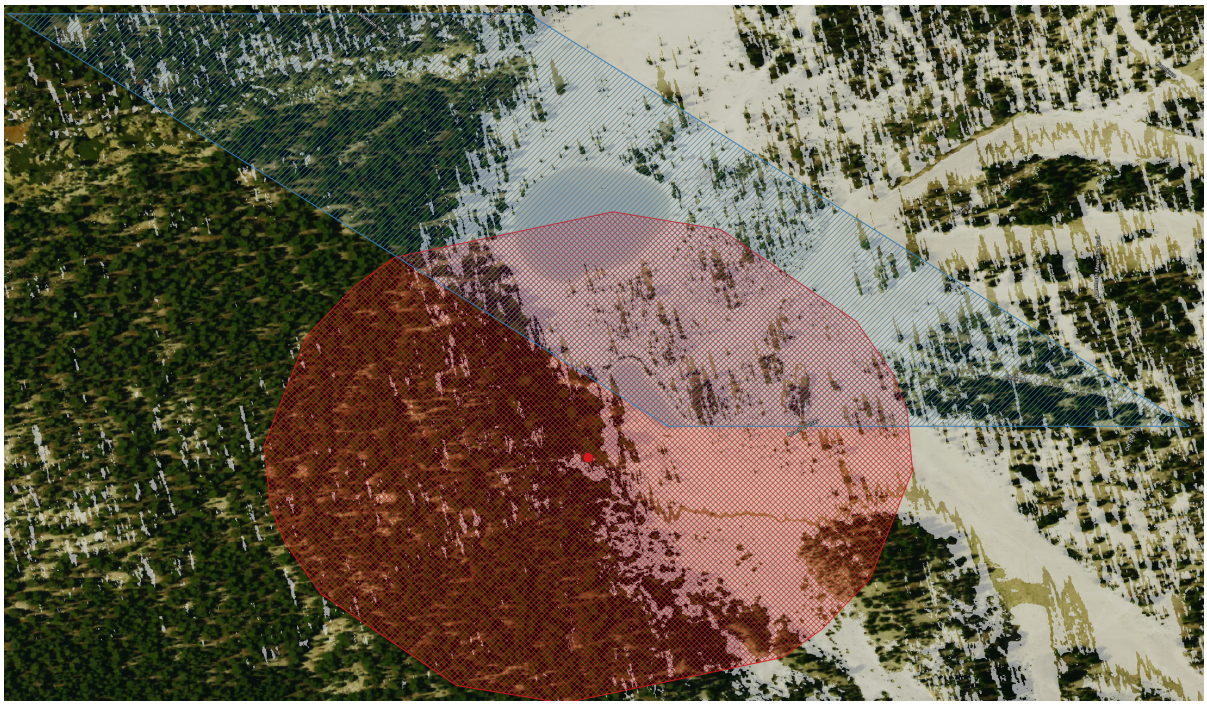


Figure 4.6: A cell (blue) and the estimated footprint of an IFOV(red) overlayed over a binary snowmap (white) derived from Worldview legion data at 0.5 m resolution (Stillinger and N. Bair, 2020). Note how significantly more than half of the cell is snow-covered while the IFOV's footprint is less than half snow-covered. Also note that the cell appears to be about 1/3 forest covered, while the IFOV's footprint appears to be more than half forest.

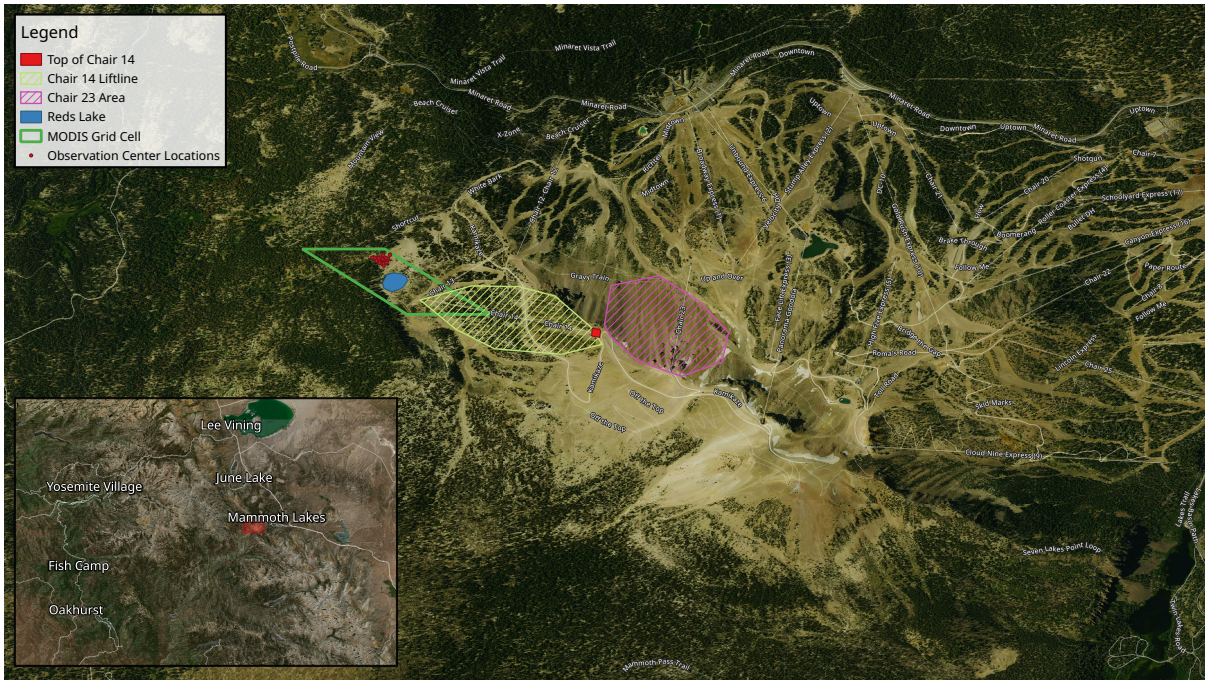


Figure 4.7: Center location of observations used for figure 4.8. All observations are within 50 m.

the observations, this snow-free observation may be for a different area than the fractionally snow-covered observation (even though they have been binned to the same grid cell). It is, e.g., conceivable that a fractionally snow-covered observation covers a region consisting of snow and forest while the associated snow-free observation covers a region consisting of soil and rock.

Finally, a concern is that MOD09GA contains merely the sensors' zenith angles but no trivially accessible information about the actual viewing geometry of the observations[10]. The sensor zenith angle alone does not provide information about whether an observation was acquired while the sensor was pointed left or right. However, knowing the actual viewing geometry may be crucial in mountainous terrain where opposing faces of the same mountain may significantly vary regarding the composition of rocks/soil and vegetation.

Figure 4.8 shows NDVI values for observations at a fixed location at Reds Lake in Mammoth

[10]Theoretically, the pointer files could be backtracked to identify the viewing geometry of each MOD09GA value. However, these pointer files are not archived and would have to be regenerated.

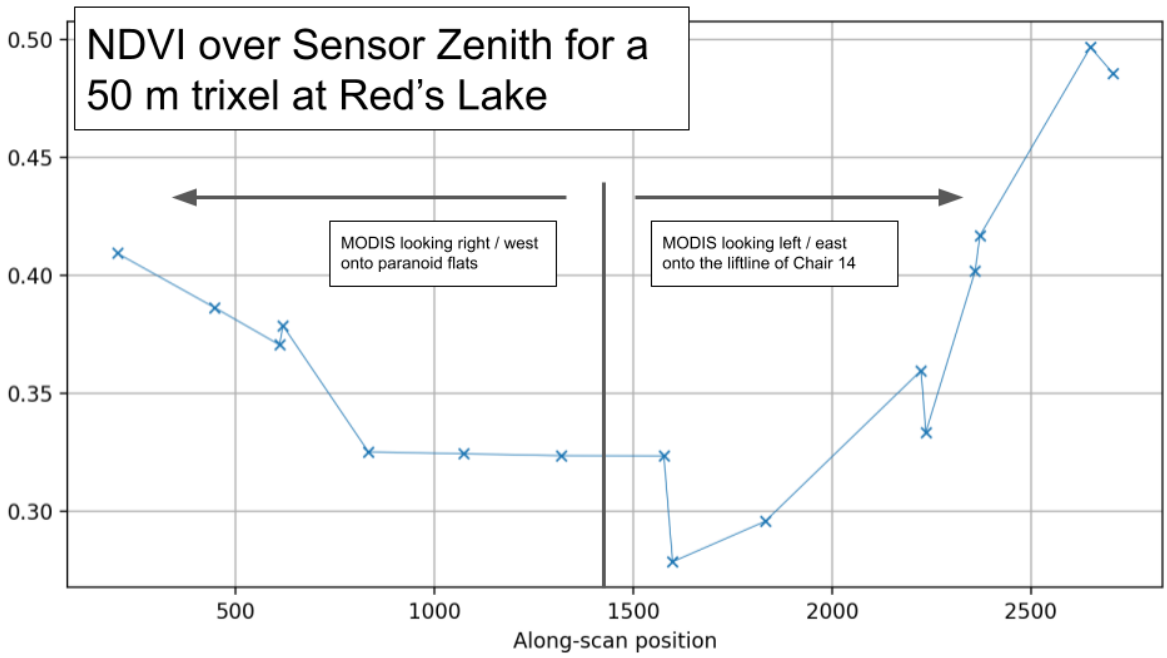


Figure 4.8: NDVI for a fixed location at Reds lake in Mammoth (to the precision of 50 m) for varying along-scan positions. Note that the off-nadir observations have higher NDVI for observations in which the sensor looks east onto the location.

(within a radius of 50 m) (C.f. figure 4.7) under varying viewing geometries (here represented as the along-scan position). The asymmetry demonstrates the influence of the viewing geometry on the reflectance spectrum: When the sensor passes east of the observed location and thus is looking west (i.e., low along-scan position)[11], it observes the eastern face of *Top of Chair 14* (*Chair 23 Area*), which is mainly exposed rock (c.f. figure top 4.9). However, when the sensor passes west of the observed location and thus is looking east (i.e., high along-scan position), it observes the western face below the *Chair 14 lift line*, which is covered with sparse trees (c.f. figure bottom 4.9).

Part of the asymmetry in figure 4.8 may be explained by the forward scattering behavior of vegetation (the Terra overpass at our Region of Interest (ROI) is before noon). However, we observe similar asymmetric behavior for locations where the eastern face of a ridge has more vegetation than the western face. We display the NDVI over the along-scan position for four additional exemplary locations in figure 4.10. An apparent asymmetry is observable for all locations, which cannot be explained by the forward scattering alone.

Therefore, any fractionally snow-covered observation should be associated with a snow-free observation that both is spatially as close as possible, and has a similar viewing geometry.

[11] terrapath

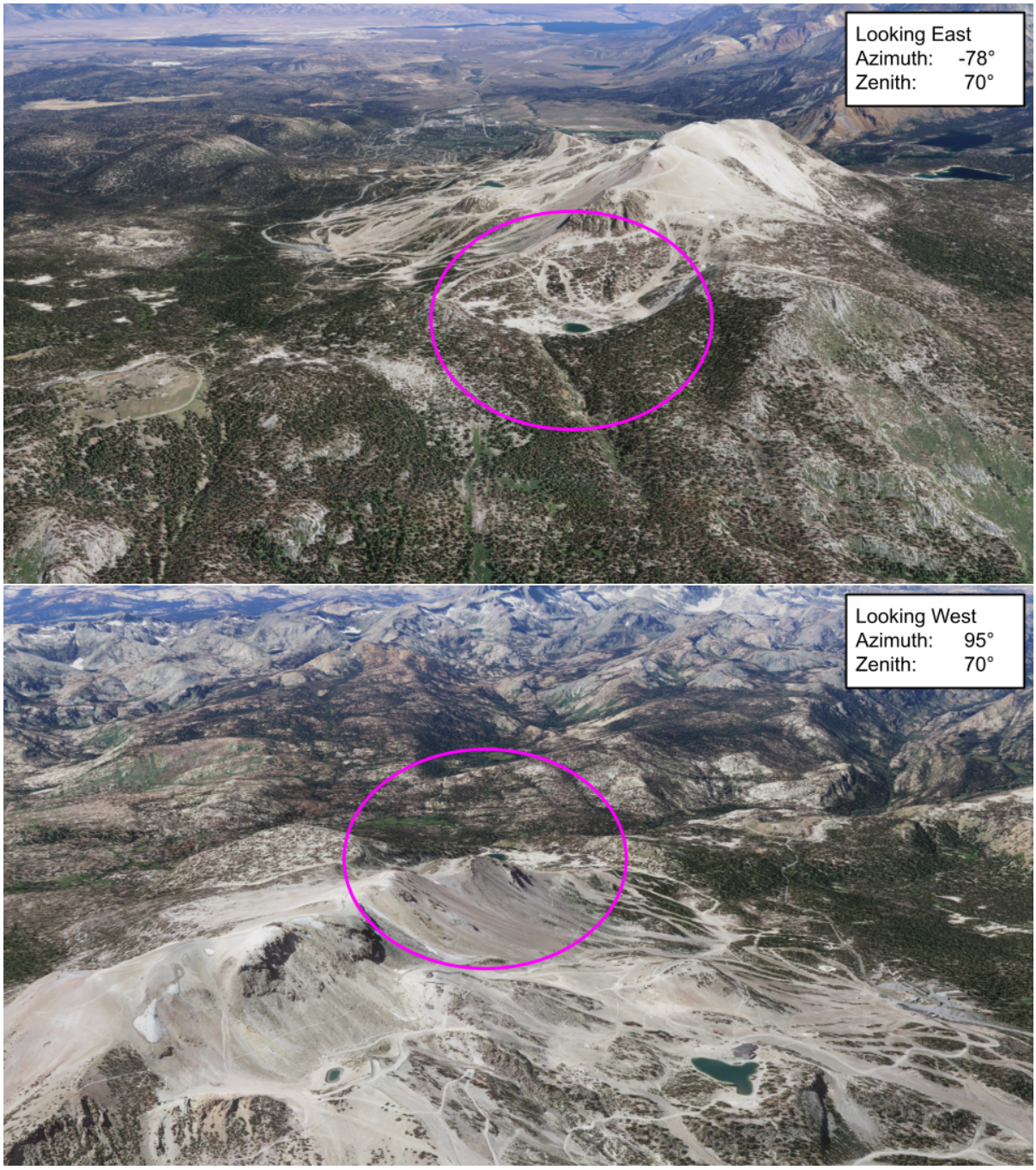


Figure 4.9: View onto the same location at Reds Lake from the east (top) and the west (bottom). Note that looking west, a face with little vegetation will be co-registered while looking east, a face with sparse trees will be co-registered.

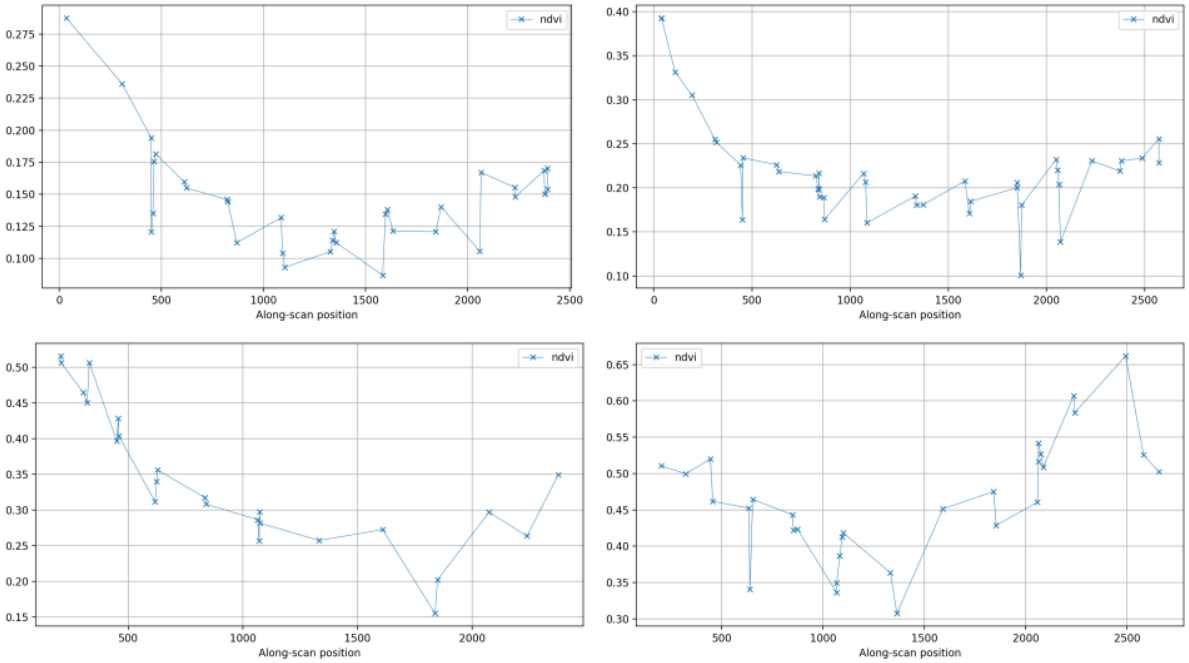


Figure 4.10: NDVI over along-scan position for four additional exemplary locations.
Top left: A location at Reds creek (37.62600691422575, -119.03900649128829).
Top right: A location at dragon's back (37.623934838741974, -119.02511013281583).
Bottom left: A location at Mammoth Rock (37.612298635885914, -118.99655376170831).
Bottom right: A location at twin lake (37.62255106589664, -119.00785780041357).

4.3 STARE Approach

In order to circumvent the introduction of uncertainties by gridding, we need to forego gridded data and work with ungridded level 2 data (i.e., MOD09/VNP09MOD). Working with ungridded data is cumbersome with conventional technologies. We, therefore, use the alternative geolocation representation Spatio-Temporal Adaptive-Resolution Encoding (STARE). STARE firstly enables us to associate each observation with a snow-free observation at approximately the same location and recorded under similar viewing geometries. Secondly, STARE allows us to evaluate the accuracy of the fSCA estimates by comparing them to "ground-truth" data for the approximate footprints of each IFOV.

The remainder of the chapter is structured as follows: We first give a quick overview of STARE. We then describe the data preparation and the creation of our snow-free reflectance library. Finally, we describe our evaluation methods and display metrics on the accuracy improvements achieved.

4.3.1 STARE primer

STARE, described in chapter 2 and in (Michael L. Rilee et al., 2021; Michael L Rilee, K.-S. Kuo, J. Frew, et al., 2020; M. Rilee, K.-S. Kuo, Gallagher, et al., 2019; Michael Lee Rilee, K.-S. Kuo, et al., 2016; K.-S. Kuo and Michael Lee Rilee, 2017), is an alternative spatial geolocation representation based on a Hierarchical Triangular Mesh (HTM). Rather than expressing points as, e.g., latitudes and longitudes, STARE express locations/points as nodes (aka trixels) in the HTM, and arbitrarily shaped regions (polygons) as sets of trixels (c.f. figure. 4.11).

STARE might appear to be yet another gridding approach, using triangular instead of rectangular cells. However, contrary to conventional gridding, the grid resolution (the size of the bins into which individual observations are binned) is not fixed but adaptive. This is achieved by using trixels at varying "quadfurcation" levels in the HTM tree. The adaptive resolution brings immense advantages when working with variable data resolutions, both between datasets, and

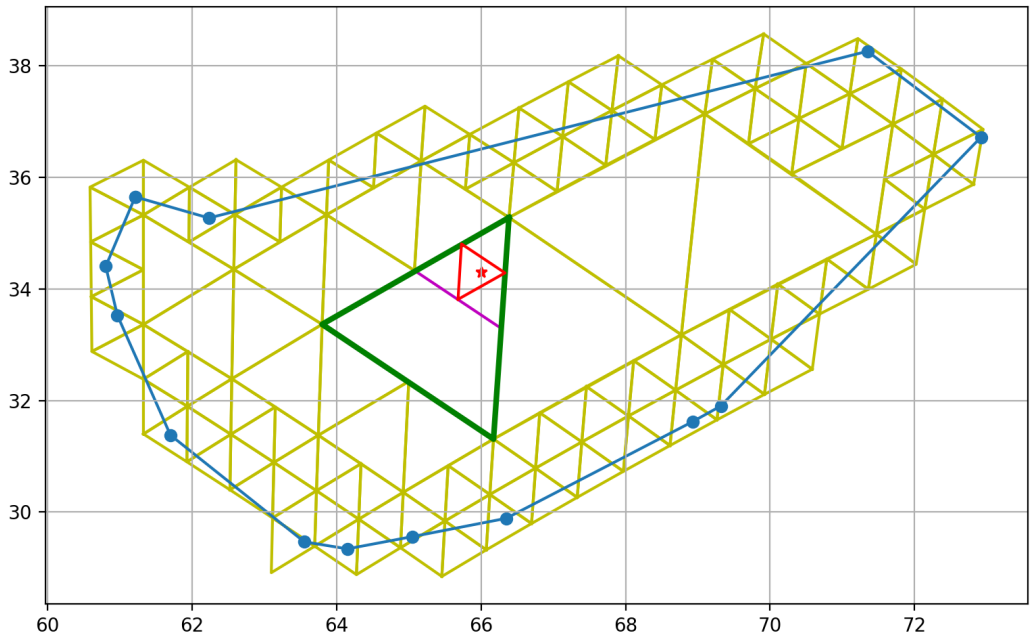


Figure 4.11: A polygon (blue) and its representation as a set of trixels that cover it (yellow). A point (red star) is represented by a single trixel (red triangle) at a chosen STARE level.

within the same datasets. With STARE, identifying the intersection of irregularly shaped and spaced data at varying resolutions is trivial. We thus do not have to decide apriori on a grid resolution into which we have to bin data; instead, we can preserve the original resolution of the data.

The STARE concept is implemented in a collection of software, described in chapter 2. The collection contains software to convert conventional location representations into STARE representations, various storage backends, and the ability to perform STARE-based geoprocessing (such as unions, intersections, and dissolves) and geospatial analysis.

4.3.2 Data Preparation

Our study area ROI is around Mammoth Lakes in the eastern Sierra Nevada, spanning from just north of Lake Thomas Edison to June Lake (c.f. figure. 4.12).

We acquired all Level 2 MODIS/Terra atmospherically corrected surface reflectance granules

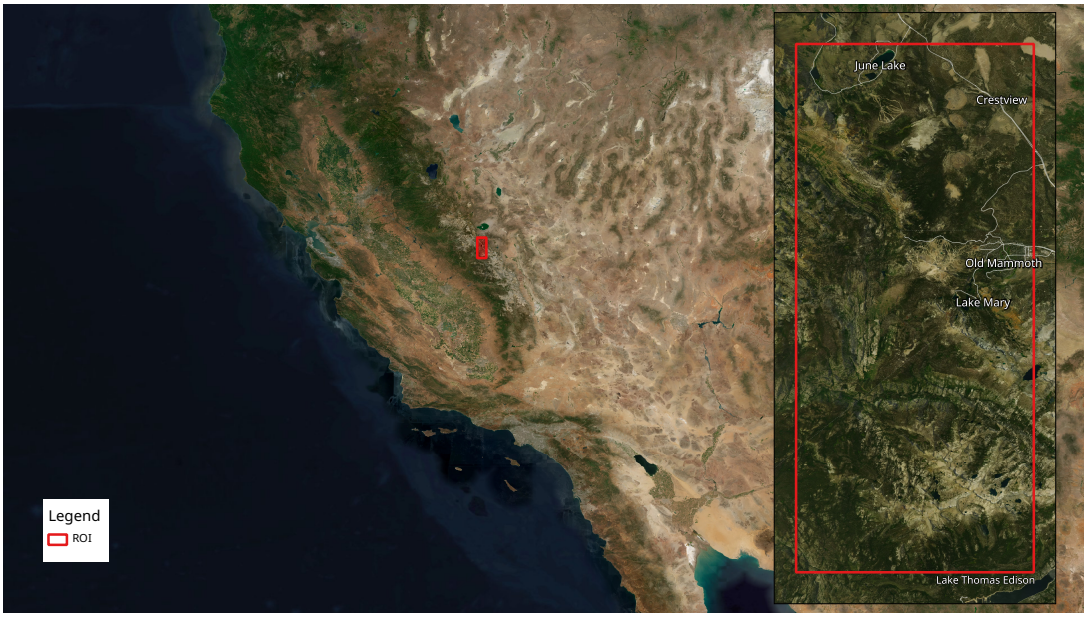


Figure 4.12: Region of interest of our study area (red) in the region of Mammoth lakes spanning from just north of Lake Thomas Edison to June Lake. (BBOX: -119.14, 37.4: -118.96, 37.8).

(MOD09[12]) and their corresponding geolocation companion granules (MOD03[13]) for the entire sensor lifetime from 2000-02-24 until 2022-09-15 (a total of 26 466 granules, containing a total of 145×10^9 IFOVs, 243×10^6 of which intersecting our ROI). We additionally acquired moderate-resolution VIIRS/Suomi surface reflectance granules (VNP09[14]) and their corresponding geolocation companion granules (VNP03MOD[15]) for the entire sensor lifetime from 2012-01-19 to 2022-09-15. We also acquired all gridded MOD09GA[16] granules for tile H08V05[17] for verification purposes.

We then created STARE sidecar companion files for each granule. STARE sidecar companion files contain the geolocation for each IFOV in STARE representation. They are thus analogous to the MOD03/VNP03* companion files, which contain the geolocation of each IFOV in WGS84

[12](M. L. S. Team, 2017). DOI: 10.5067/MODIS/MOD09.006
[13](M. S. D. S. Team, 2017). DOI: 10.5067/MODIS/MOD03.061
[14](N. V. L. S. Team, 2020). DOI: 10.5067/VIIRS/VNP09.001
[15]((VCST), 2021b). DOI: 10.5067/VIIRS/VNP03MOD.002
[16](Eric Vermote and Robert Wolfe, 2021). DOI: 10.5067/MODIS/MOD09GA.006
[17]we used the ladsweb_downloader to download all granules. Ladsweb_downloader is available at `https://github.com/NiklasPhabian/ladsweb_downloader`.

coordinates. The MOD09 granules contain surface reflectances at 1000 m, 500 m, and 250 m resolution. However, MODIS geolocations are only distributed at 1000 m resolution. Since we intend to use the 500 m surface reflectances, we implemented a geolocation interpolation algorithm, described in section 4.3.3.

Finally, we created estimates for each IFOV footprint subject to the viewing geometry. Those footprint approximations are used to evaluate the fSCA estimates against higher-resolution binary snowmaps. The creation of the footprint estimates is described in section 4.3.4.

4.3.3 Resolution interpolation

MODIS collects data at nominal resolutions of 1000 m, 500 m, and 250 m. The calibrated surface reflectance product MOD09 contains reflectance spectra for all three resolutions, but only the geolocation of the 1 km resolution is distributed. In order to use the 500 m resolution data, we therefore need to interpolate the 500 m resolution geolocations. The MODIS Level 1A Earth Location Algorithm Theoretical Basis Document (ATBD) (Nishihama et al., 1997) and the MODIS Level 1B user manual (Toller et al., 2009) give us hints for how to do this:

The MODIS Level 1A Earth Location ATBD (Nishihama et al., 1997) states[18]:

> "MODIS is built so that the start of the weighting function for the 500 m pixel is the same as the start of the weighting function for the 1 km pixel. This means that four 500 m pixels are not contained within a 1 km pixel."

This is visible in figure 4.13. (R. E. Wolfe et al., 2002) further explains:

> "To the first order, the MODIS point-spread function is triangular in the scan direction. The centers of the integration areas of the first observation in each scan

[18] A private communication with the MODIS support team contradicts this statement and suggests the following (from github) "Since the MODIS geolocation is estimated at 1km only, [..] it is provided for 1km datasets only. The best way to use the 1m geolocation will be to co-locate each 500m/250m observations within the corresponding 1km pixel and then use the corresponding 1km geolocation."

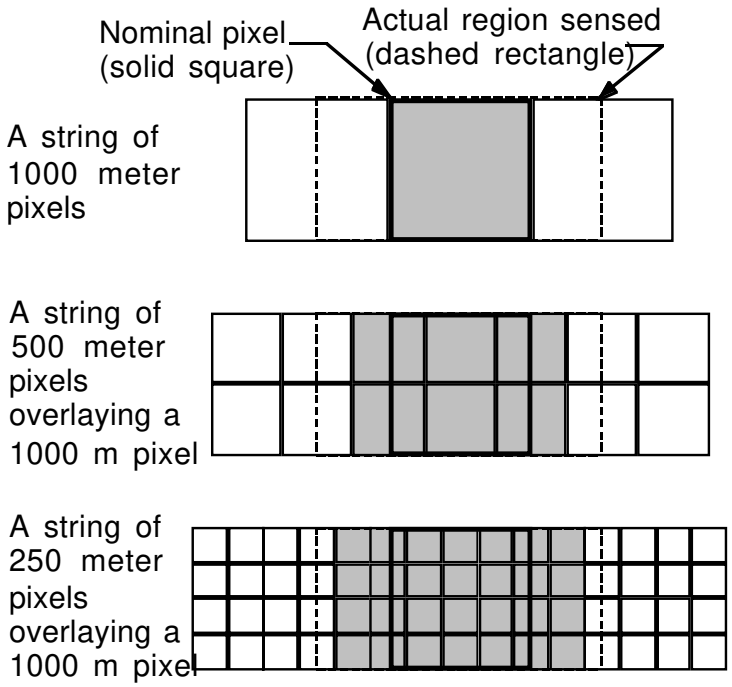


Figure 4.13: Pixel nesting of the 1 km, 500 m, and 250 meter resolutions of MODIS. Source: Figure 2-8 of MODIS Level 1A Earth Location (Nishihama et al., 1997).

> are aligned, in a 'peak-to-peak' alignment.". And: "In the track direction, the point-spread function is rectangular and the observations at the different resolutions are nested, allowing four rows of 250 m observations and two rows of 500 m observations to cover the same area as one row of 1 km observations."

This is visualized in figure 4.14. The MODIS Level 1B User Guide (Toller et al., 2009) further suggests:

> "Interpolation may be used to recover latitude and longitude of all pixels [..]. Note that, due to the overlap between consecutive scans, interpolation across scan boundaries, such as is done by these HDF-EOS swath functions, can produce inaccurate results. Near scan boundaries, it is better to extrapolate from within the same scan, than to interpolate with values from the adjacent scan."

Finally, (Nishihama et al., 1997) states that:

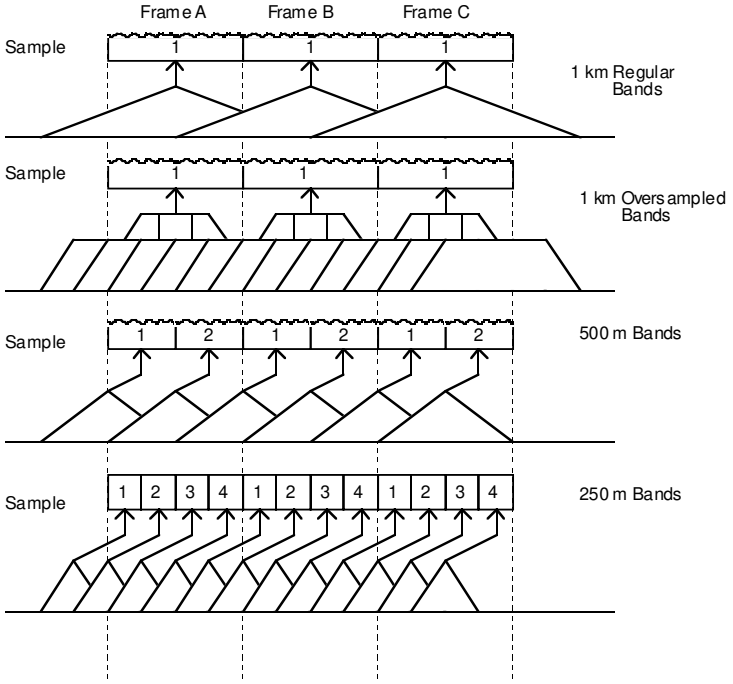


Figure 4.14: MODIS triangular point-spread function for all resolutions. Figure 3-13 of the MODIS Level 1A Earth Location: (Nishihama et al., 1997).

> "The samples for each band are delayed by an appropriate amount so that all samples in a frame of data start at the same point on the ground."

We combined these pieces of information in our geolocation interpolation algorithm[19]: We process one scan group at a time. For the 1 km resolution, a scan group contains 10 IFOVs along-track and 1354 IFOVs along-scan. The first observations of all resolutions are aligned in scan direction. The first 500 m resolution geolocations in scan direction, therefore, sit between the 1000 m resolution in track direction offset by 1/4 of the distance to the following observation in track direction. The same is true for all other odd-numbered observations in scan direction. The even-numbered observations in scan direction sit halfway between the odd-numbered observations. The last observation in scan direction has to be extrapolated.

Figures 4.15, 4.16, and 4.17 display the original 1000 m resolution geolocations (red) and the interpolated 500 m geolocations (blue) for one and a half 1000 m scan groups at the beginning

[19]STAREMaster's 500 m interpolation function on GitHub

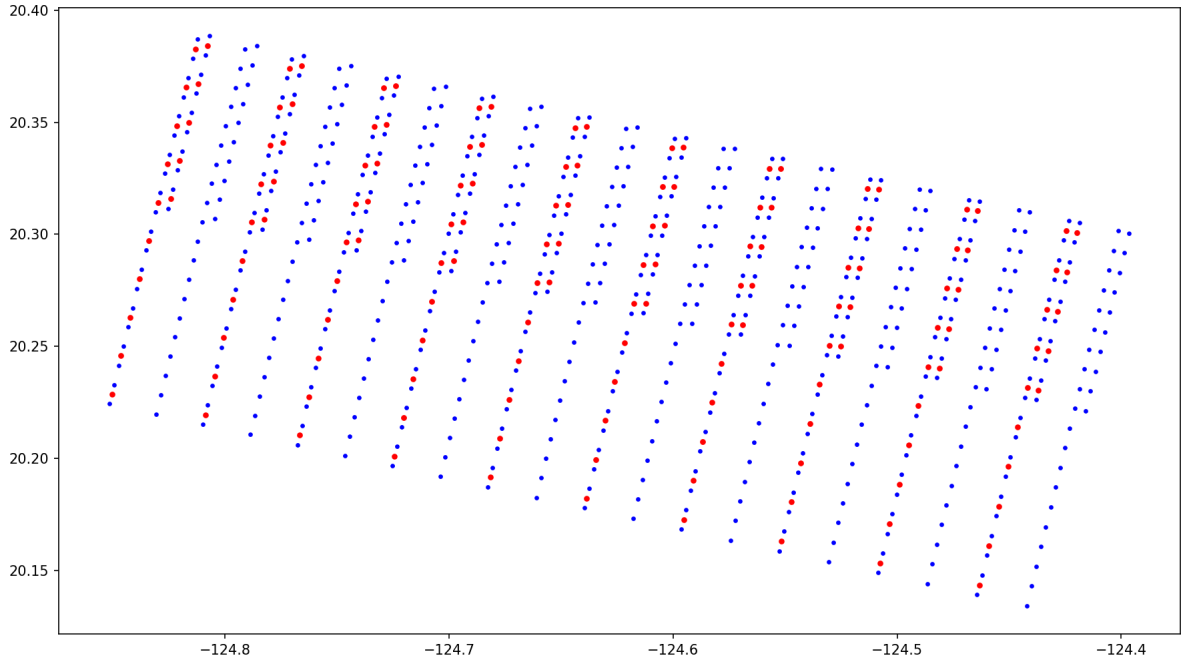


Figure 4.15: MODIS Geolocations of the 1 km resolution (red) and their 500 m interpolations (blue) for the first 1.5 scan groups and the first ten observations in scan direction at the beginning of a scan.

(4.15), the center (4.16), and the end (4.17) of a scan. The geolocations are for MODIS Terra for daytime observations. The beginning of a scan is therefore in the north-west, while the end of a scan is in the south-east. Note how six 500 m resolution observations nest into one 1000 m resolution observation.

4.3.4 IFOV approximation

In order to evaluate the accuracy of a sub-pixel estimands in general and the fSCA from SPIReS in particular, we compare the estimated values against ground truth data. In practice, the ground truth data will often be binary data derived from sensors with higher spatial resolutions.

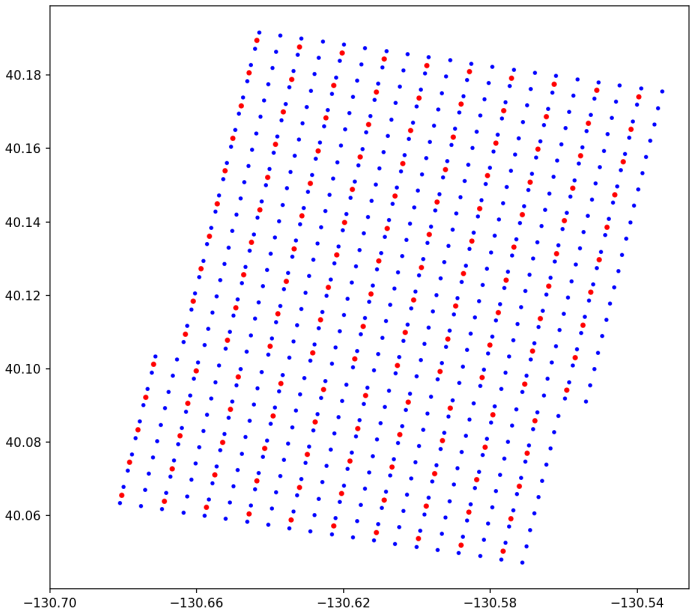


Figure 4.16: MODIS Geolocations of the 1 km resolution (red) and their 500 m interpolations (blue) for 1.5 scan groups at the center of a scan (at nadir).

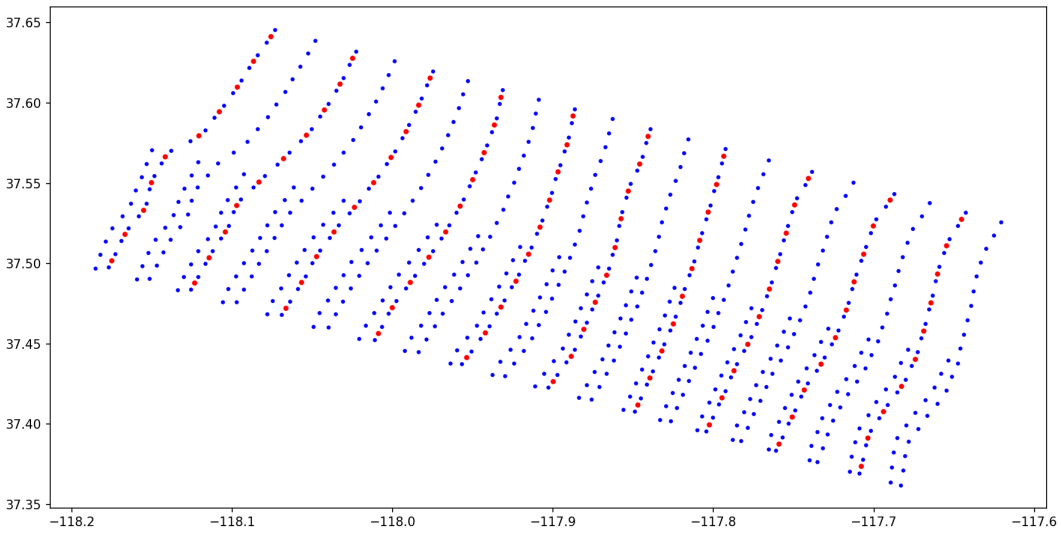


Figure 4.17: MODIS Geolocations of the 1 km resolution (red) and their 500 m interpolations (blue) for the first 1.5 scan groups and the last ten observations in scan direction at the end of a scan.

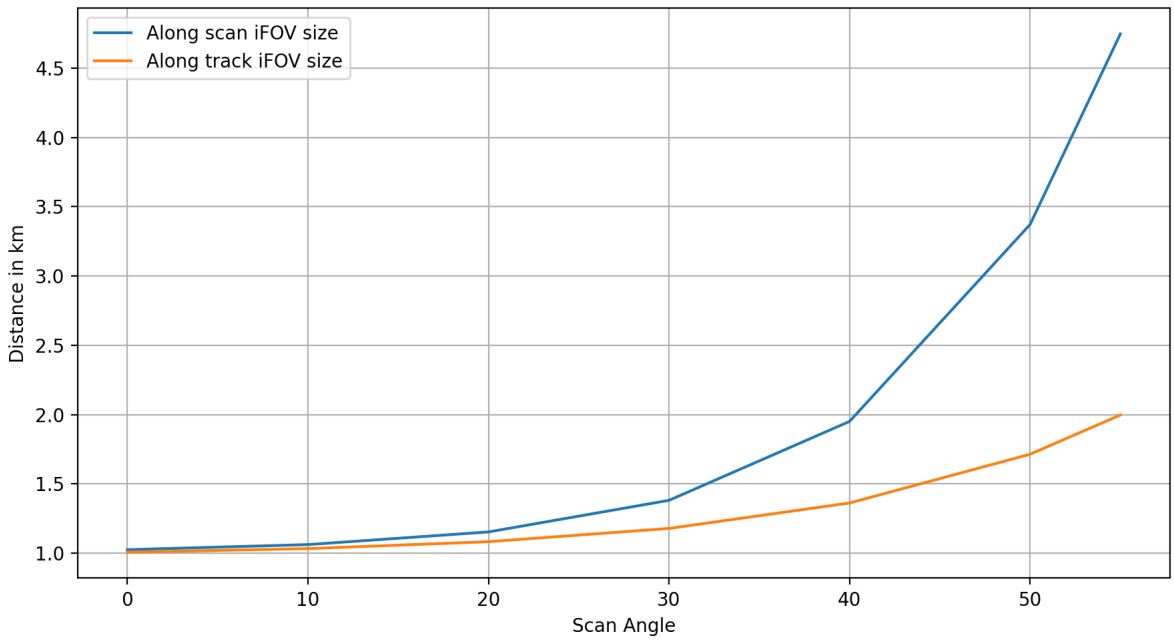


Figure 4.18: Growth of IFOV along-track and along-scan as a function of the nadir scan angle.

E.g., a MODIS IFOV at nadir may have an approximate footprint of 500 m x 500 m[20], covering about 250 Landsat pixels.

Comparing subpixel estimands of an IFOV against verification data requires us to evaluate spatial coincidence between the IFOV footprints and the verification data. We thus need an approximate spatial representation for the IFOV footprints to achieve this. As previously discussed, we consider a grid cell at a constant resolution of 500 m x 500 m as an inapt footprint approximation. We instead created two alternative approximations:

The first approximation assumes that the IFOV's footprint is a circle around the geolocation with a constant diameter equaling the nominal resolution, i.e., 500 m for MODIS and 750 m for VIIRS.

The second approximation assumes that the IFOV's footprint has the shape of an ellipse. We

[20]The actual cell size is $w * w$. With $w = T/n = 463.312\,716\,53$ m. $T = 1\,111\,950$ m is the height and width of each MODIS tile in the projection plane, and $n = 2400$ the number of cells in a MODIS tile in width and height direction (Meister, Zong, and McClain, 2008).

use the fact that we have some idea of the distortion introduced by the sensor's nadir scan angle. From the MODIS Level 1A Earth Location ATBD (Nishihama et al., 1997), we know that the resolution of a 1 km spatial element at a 55° scan angle has ground dimensions of approximately 4800 m along-scan and 2000 m along-track. (Dozier, Painter, et al., 2008) provides an analytic approach for the relation between sensor zenith and along-scan and along-track IFOV pixel expansions. For simplicity of our purposes, we instead fit an exponential function to data provide by (Nishihama et al., 1997) to establish a relation between nadir scan angle and along-scan and along-track IFOV length (c.f. figure 4.18.).

$$length = \frac{e^{a*x}}{b} + c$$

With:

	a	b	c
along-scan	0.09	$40.9\,\mathrm{km}^{-1}$	1.00 km
along-track	0.07	$35.2\,\mathrm{km}^{-1}$	0.98 km

Using the derived along-track and the along-scan lengths, we create ellipses around each IFOV geolocation, using the along-track length as the length of the semi-minor axis and the along-scan length as the length semi-major axis. We finally orient each ellipse according to the sensor's instantaneous azimuth angle. A set of resulting ellipse footprint approximations are visualized in figure 4.4. Figure 4.19 displays IFOVs approximated as circles with constant diameter and as ellipses with semi-minor and semi-major axes lengths adjusted to the sensor nadir scan angle and orientation adjusted to the sensor azimuth angle.

4.3.5 Viewing geometry discretization

Our thesis is that we can improve the accuracy of SPIReS' fSCA estimates by using snow-free reference reflectances R_0 that have been observed under a similar viewing geometry as the

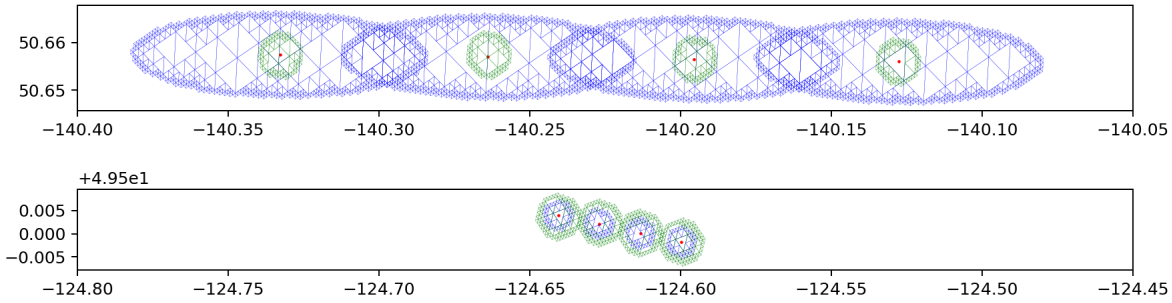


Figure 4.19: IFOVs with their geolocation points (red), approximated as circles with a constant diameter (green) and as ellipses (blue) at the beginning of a scan (top) and the center of a scan (bottom).

reflectances R of the observation for which the fSCA is to be estimated.

MODIS has an orbital repeat cycle of 16 days (M. D. King et al., 2003; R. E. Wolfe et al., 2002). This means that every 16 days, the ground track repeats. Consequently, there are 16 distinct viewing geometries under which MODIS observations are made for any given location outside of latitudes of approximately $\pm 30°$. Due to orbital dynamics and orbit drift, the overpasses in each orbital track vary slightly. Figure 4.20 displays along-scan positions under which MODIS/Terra observations were taken for a fixed location over time. The 16 ground track groups of MODIS's 16-day orbital repeat cycle are visible. It is noteworthy to draw attention to the 'bumps,' presumably caused by orbital anomalies and consequent corrective satellite maneuvers. Towards the end of the time series, the uncorrected orbital drift[21] is observable. MODIS is drifting towards earlier equator crossing times (conceptually: drifting westwards), causing observations for a fixed location to be made increasingly earlier in the day and thus at lower along scan positions[22]. If MODIS were to remain in this drifting orbit long enough, the gaps between the orbit track groups would eventually fill.

The along-scan position is a proxy for the viewing geometry. It gives us information about the sensor zenith (and thus the extent of the IFOV) and whether the sensor looked left, right,

[21] https://terra.nasa.gov/about/terra-orbital-drift-information

[22] Terra flies south and MODIS scans west to east for daytime observations.

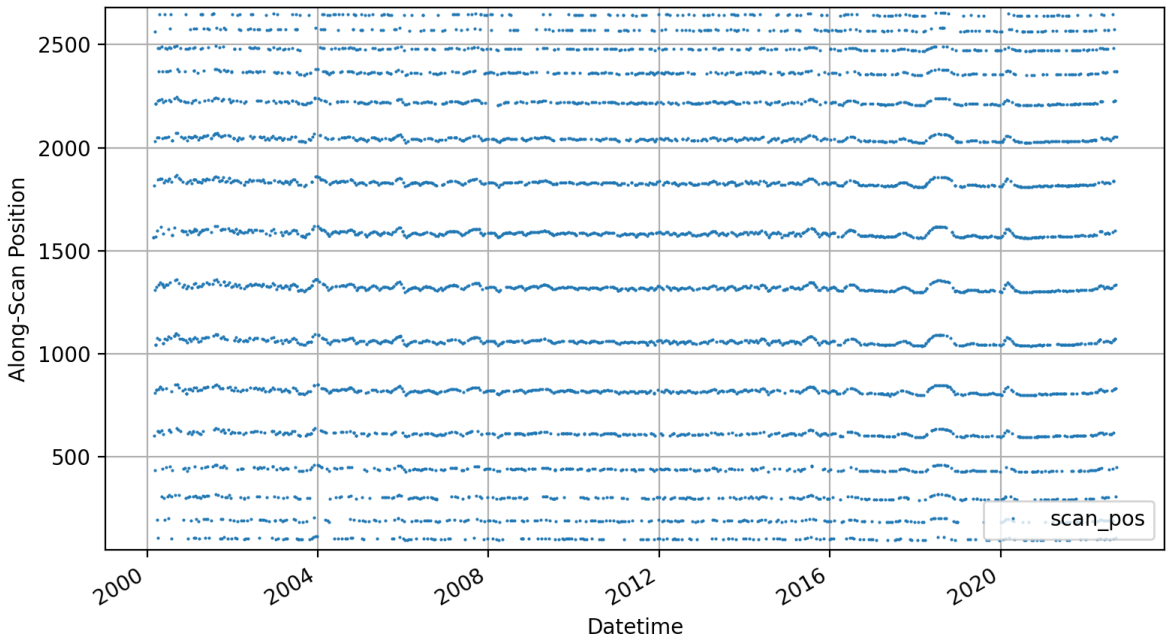


Figure 4.20: Along-scan position vs. time of MODIS observations of a fixed location. The 16 distinct groups' orbital tracks are visible. Also note the bumps in, e.g., 2004, 2018, and 2020 presumably caused by orbital anomalies and consequent corrective satellite maneuvers.

or (approximately) straight down. Since we want to find a R_0 for every R observed under a similar viewing geometry, we discretize the along-scan position at different resolutions. At a discretization of 1, we do not distinguish between different viewing geometries. At a discretization of 2, we only distinguish whether the sensor looked left or right. At a discretization of 3, we distinguish between the sensor looking left, right, or approximately straight down. At higher discretizations, we add further fidelity to the viewing geometry.

4.3.6 Snow-Free Reflectance (R0) Library

To compute the fSCA for a reflectance spectrum R observed for a given location, SPIReS requires a snow-free reflectance spectrum R_0 for the same location. The question of what "same location" means immediately arises, considering that each observation has a unique geolocation and footprint[ˆunique_footprint]. One could find the spatially closest observation by evaluating geodesic distances. However, evaluating possibly trillions of distances ad-hoc is computationally infeasible. The correct answer is that some degree of location discretization is required.

[ˆunique footprint]: [+MODIS] has a repeat cycle of 16 days. Therefore, every 16 days observations under *similar* viewing conditions are taken. However, subject to the nonlinearity in the dynamics of the spacecraft trajectory, the optical properties of the atmosphere, and the topography of the Earth's surface, the viewing conditions do not exactly repeat. Thus every single observation has to be considered to have a unique geolocation and footprint.

Discretization enables us to bin observations sufficiently close to each other and consider them spatially coincidental. When using the standard gridded surface reflectance products, we assume every observation binned to the same grid cell is spatially coincident. As previously stated, our thesis is that this is a too strong a coarsening specifically since the geolocation is known to have at least an order of magnitude higher precision. We thus create a R_0 library at a higher spatial resolution.

The algorithm for finding a R_0 for each grid cell works as follows[23]:

```
For each cell:
    1. Consider all observed spectra for the cell
    2. Discard observations with a SensorZenith > 30 degree
    3. Find the observation with the lowest NDSI
    4. If the lowest NDSI < 0:
        1. Discard observations that
            1. Have internal cloud mask set, or
            2. Have internal snow mask set, or
            3. Have MOD35 cloudmask set, but not saltpan
        2. The observation with maximum NDVI is R0
    5. Else: (e.g., permanent snow)
        1. Discard observations with Band 3 < 0.1
        2. The observation with the lowest Band 3 value is R0
```

Note that the cutoff at 30° sensor zenith angle eliminates approximately 65 % of all observations.

We adapt this algorithm to create R_0 for STARE trixels rather than for grid cells and create R_0 libraries at different STARE "quadfurcation" levels. We additionally take the viewing geometry, represented by the along-scan position, into account. The choices of the quadfurcation level and the along-scan position discretization are a tradeoff. While conceptually, we want a high resolution in geolocation and viewing geometry, we decrease the number of candidate snow-free reflectances per bin with increasing resolution. If the count of the number of candidates per bin is too low, we may not find good (i.e., cloud-free, snow-free, and not quality-flagged) snow-free spectra.

[23] createR0.m on github

Approaching an appropriate discretization level: At the latitude of our ROI, a 500 m x 500 m cell is roughly observed once per day (i.e., the revisit period is one day). A data duration of 8239 days (from 2000-02-24 to 2022-09-15) thus gives us 8239 candidate spectra to choose from when discretizing to the MODIS grid cells. A STARE trixel at quadfurcation level 14 has roughly the same area as a MODIS grid cell. We thus expect roughly one observation per day falling into a level 14 trixel. At level 15, we expect the revisit period to be 1/4 of that since a level 15 trixel is four times smaller than a level 14 trixel.

Table 4.2: Approximate trixel areas, edge lengths, and MODIS revisit periods and total visits for the data duration of 8239 days

Level	Area (A_{trixel})	Edge (l_{trixel})	Revisit Period	n Visits
12	$5.11 \times 10^6\,\mathrm{m}^2$	$3.20 \times 10^3\,\mathrm{m}$	0.05 days	167 544.0
13	$1.28 \times 10^6\,\mathrm{m}^2$	$1.60 \times 10^3\,\mathrm{m}$	0.20 days	41 885.9
14	$3.19 \times 10^5\,\mathrm{m}^2$	$7.99 \times 10^2\,\mathrm{m}$	0.78 days	10 471.5
15	$7.99 \times 10^4\,\mathrm{m}^2$	$4.00 \times 10^2\,\mathrm{m}$	3.13 days	2 617.9
16	$2.00 \times 10^4\,\mathrm{m}^2$	$2.00 \times 10^2\,\mathrm{m}$	12.52 days	654.5
17	$4.99 \times 10^3\,\mathrm{m}^2$	$9.99 \times 10^1\,\mathrm{m}$	50.08 days	163.6
18	$1.25 \times 10^3\,\mathrm{m}^2$	$5.00 \times 10^1\,\mathrm{m}$	200.32 days	40.9

Table 4.2 displays the areas and edge lengths of trixels at varying quadfurcation levels. It also contains the estimated *revisit periods* which is the nominal resolution divided by the area of the trixels, assuming one overpass per day. It is to be understood as the maximum revisit period. Revisit periods of less than 1 indicate that multiple observations per day fall into the trixel. The column *n Visits* is the revisit period multiplied by the data duration (8239 days). It gives us a rough approximation of how many candidate spectra we will have per spatial bin.

Note that the average trixel area A_{trixel} and edge length l_{trixel} may be naively computed as

indicated below. However, depending on the position of the trixels in the initial solid's faces, the trixel areas vary across the globe. Table 4.2 shows the actual areas and lengths computed for trixels in our ROI.

$$r_{earth} = 6\,371\,007\,\mathrm{m}$$
$$A_{earth} = 4 * \pi * r_{earth}^2$$
$$A_{trixel} = A_{earth}/8/(4^{level})$$
$$l_{trixel} = (A_{trixel} * 2)^{0.5}$$

The MODIS geolocation precision is approximately 50 m (R. E. Wolfe et al., 2002) ($2.5\times10^3\,\mathrm{m}^2$), matching trixel areas at quadfurcation level between 16 and 17. This gives us the upper bound for the quadfurcation level. A cell with an edge length of 463.3 m has an area of $2.14 \times 10^5\,\mathrm{m}^2$, matching quadfurcation level between 14 and 15, giving us the lower bound for the quadfurcation level.

We created R_0 libraries for quadfurcation levels 14 to 17 using the same logic as stated above. We then further binned the observations for each trixel by their viewing geometry, discretized in 1 to 7 groups. This would lead to a total of 28 R_0 libraries. We recognize we have too few candidate spectra at high spatial resolution and viewing geometry discretization. We, therefore, discarded the R_0 libraries with quadfurcation levels above 16 and[24] viewing geometry discretizations above 4. This leaves us with a total of 22 R_0 libraries.

Table 4.3 gives the actual average number of observations falling into trixels at different quadfurcation levels, approximately matching our theoretically expected number of revisits in table 4.2. Figure 4.21 visualizes the number of observations that fell into each trixel of a level 15 STARE grid in our ROI.

[24] logical and

Table 4.3: Number of observations that fell into trixels at varying quadfurcation levels

Level	n Visits (mean)
14	6 696
15	1 767
16	486
17	145
18	49

4.3.7 Computation of fSCA

To compute the fSCA in our STARE approach for set of reflectance observations O, we proceed as follows: First, we choose the discretization level of our R_0 library. This means we a) select a spatial resolution (STARE quadfurcation level) at which we evaluate spatial coincidence, and b) select a viewing geometry discretization.

Secondly, we iterate over each observation O for which we want to calculate the fSCA. Using STARE, we find all snow-free observations O_0 that spatially coincide with each O. From those, we find the observation with the most similar along-scan position to O. We then feed the reflectance spectra of O and O_0 (R and R_0, respectively) to SpiPy[25], which returns the fSCA, the fractional shade cover (fShade), the snow's grain size, and LAP concentration for O.

[25]https://github.com/edwardbair/SpiPy

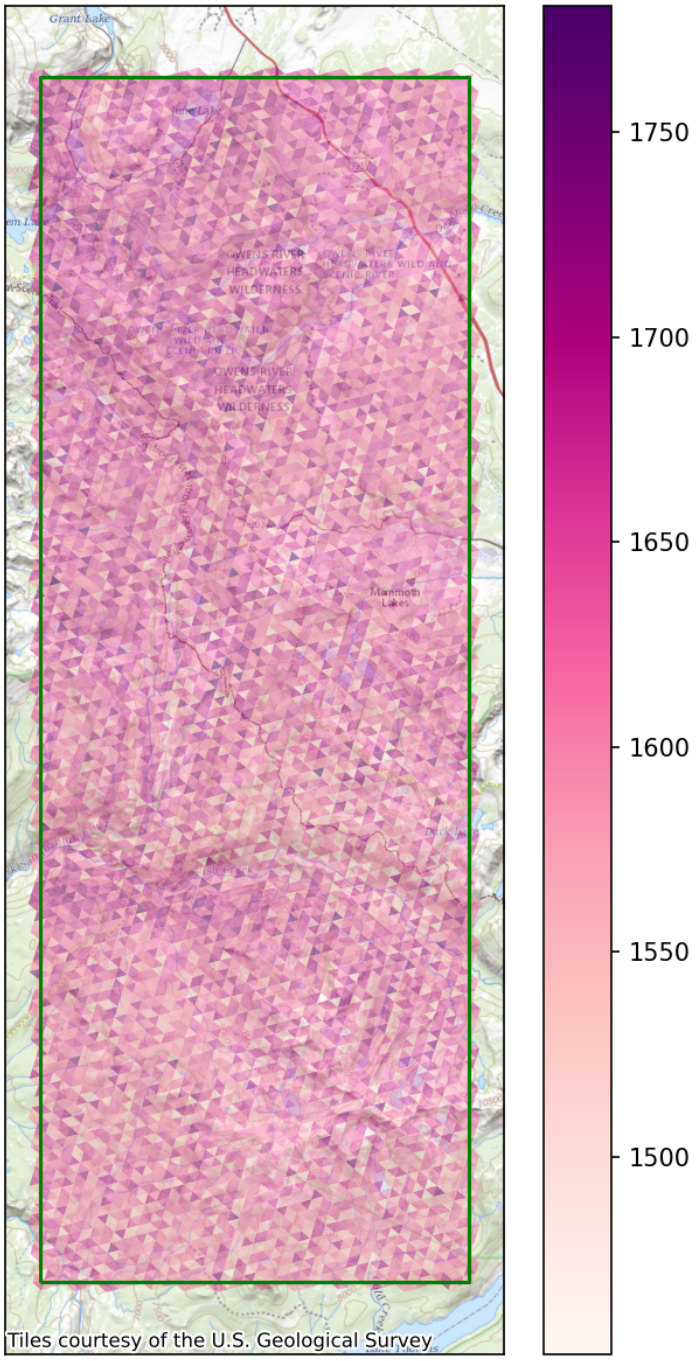


Figure 4.21: Number of observations that fell into each trixel of a level 15 STARE grid over our ROI. Each cell had more than 1400 observations.

4.4 Evaluation and Results

We carry out two evaluation efforts. First, we evaluate the accuracy of fSCA retrievals for a short period over a large ROI. We here evaluate the influence of the spatial resolution and viewing angle discretization of the R_0 library. We evaluate the accuracy by comparing the fSCA values to high-resolution binary snowmaps of the approximate IFOV footprints. In the second effort, we evaluate the plausibility of seasonal fSCA time series over small selected locations. We further attempt to join fSCA retrievals from MODIS/Terra and VIIRS/Suomi.

The efforts differ from conventional SPIReS processing and validation as follows:

- We ensure that each fractionally snow-covered observation is associated with a snow-free observation that is as close as possible regarding location and viewing geometry.
- We use the approximate footprint of each IFOV to allow a more accurate comparison to ground-truth data to better evaluate the fSCA accuracy.

4.4.1 fSCA accuracy

To evaluate the fSCA accuracy, we use a Binary Viewable Snow Covered Area Validation Mask (Stillinger and N. Bair, 2020) derived from cloud-free WorldView-2/3 data for 2017-12-11 over our ROI.

In figure 4.22, we display snow depth measured at the Cold Regions Research and Engineering Laboratory (CRREL)) and University of California, Santa Barbara (UCSB) Energy Site (CUES) (Colee, 2016) in the vicinity of McCoy Station in Mammoth, for the days before and after 2017-12-11[26]. Since we observe only a slight variation in snow depth during this period, we conclude that neither significant melting nor a snow accumulation event occurred. We consequently assume the Binary Snow Covered Area Mask to be representative of the period between 2017-12-07 and 2017-12-13.

[26]Data retrieved from `https://snow.ucsb.edu/`

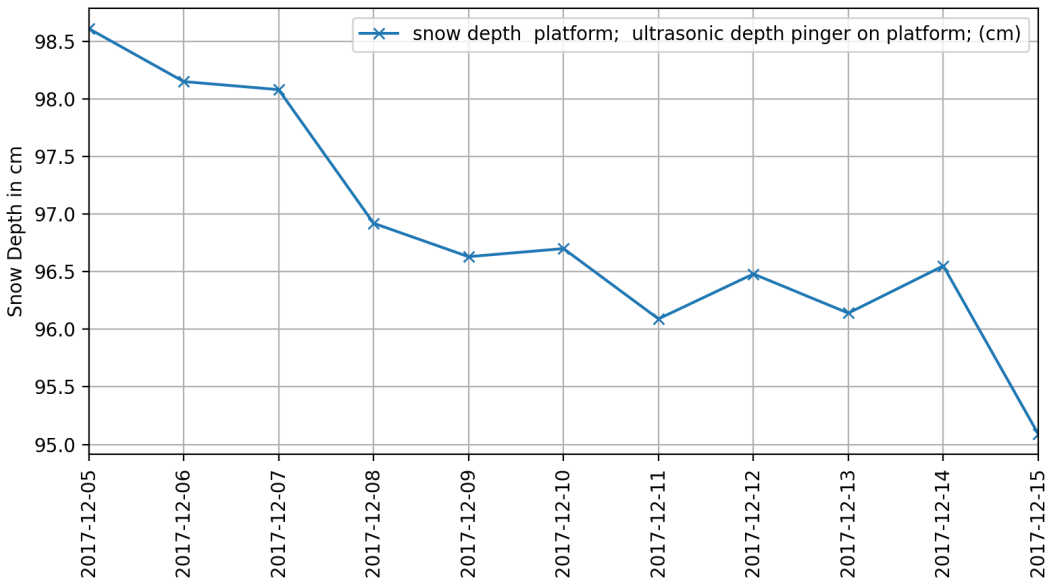


Figure 4.22: Average daily snow depth as measured at CUES between 2017-12-05 and 2017-12-19. Data retrieved from snow.ucsb.edu. No significant snowmelt nor snow precipitation event appeared in the period.

We compute the fSCA for each MODIS and VIIRS IFOV falling into our ROI using snow-free endmembers for all of our R_0 libraries for six days between 2017-12-07 and 2017-12-13, skipping 2017-12-09 since it had a far-off nadir viewing angle for MODIS/Terra over our ROI. A total of 13 709 MODIS and 8912 VIIRS IFOVs fell into our spatiotemporal bounding box. We additionally compute the fSCA using a standard MODIS-grid R_0 library as a reference.

MODIS

The six days contained six distinctly different MODIS overpasses, as displayed in figure 4.23: Two overpasses appeared close to the nadir, two overpasses far off-nadir, and two overpasses at an intermediate distance, one each with MODIS passing east and west of our ROI. We can notice a slight influence of smoke from the Thomas Fire in Santa Barbara and Ventura county (c.f. figure 4.24) over our ROI.

We then use our IFOV footprint approximations to retrieve the pixels of the high-resolution Binary Viewable Snow Covered Area Validation Mask that intersect each footprint. For each

Figure 4.23: MODIS overpasses (magenta) over our ROI (green) for our temporal extent between 2017-12-07 and 2017-12-13. Two overpasses appeared close to the nadir, two overpasses far-off nadir, and two overpasses at an intermediate distance. Basemap: NASA GIBBS

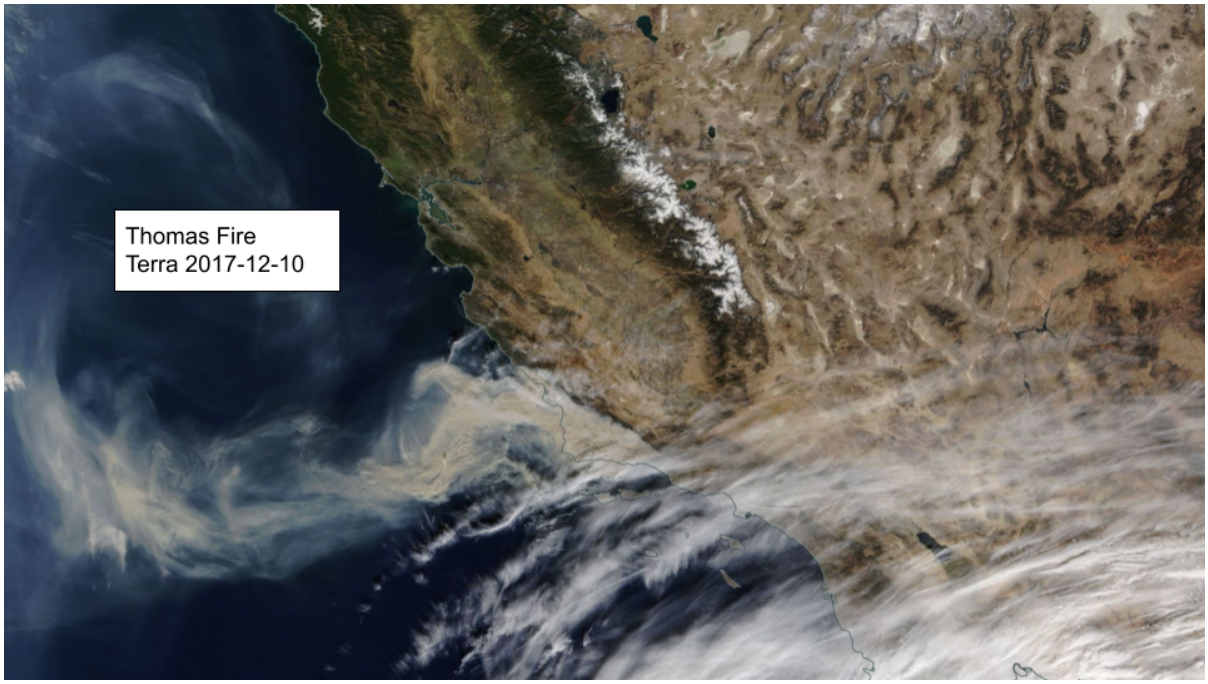


Figure 4.24: Smoke over Santa Barbara county from the Thomas fire. Source: NASA worldview.

footprint, we compute the "ground truth" fractional snow cover $fSCA_{gt}$ as the ratio of pixels marked as snow-covered to the total number of pixels intersecting the footprint. We then declare the difference between the fSCA retrieved for the footprint from SPIReS $fSCA_{spires}$ and the $fSCA_{gt}$ as the estimation error and calculate the mean absolute error (MAE), root-mean-square error (RMSE), and the variance over all IFOVs.

$$fSCA_{gt} = \frac{n_{snow}}{n_{total}}$$

$$MAE = \sum_{i=0}^{n} \frac{|fSCA_{spires} - fSCA_{gt}|}{n}$$

$$RMSE = \sqrt{\sum_{i=0}^{n} \frac{(fSCA_{spires} - fSCA_{gt})^2}{n}}$$

The fSCA errors are displayed in figure 4.25. The three leftmost boxes-and-whiskers are the errors of fSCA computed using MODIS-grid R_0 library values. In the leftmost box-and-whisker, we assumed the IFOV footprints to be the MODIS grid cells. This scenario is thus equivalent to the conventional SPIReS approach. In the second box-and-whisker, we assume the IFOV footprints to be circles with a constant radius. In the third box-and-whisker, we assumed the IFOV footprints to be ellipses. The following boxes-and-whiskers represent the errors of the fSCA computed using STARE-based R_0 values at increasing spatial resolution and viewing angle discretization. We can immediately observe the following results:

1) The fSCA estimates accuracy improves when assuming a circular IFOV footprint centered over the IFOV's geolocation and improves marginally further when assuming an ellipse as the IFOV footprint
2) The fSCA estimate accuracy is better for the STARE R_0 libraries than for the MODIS grid R_0 libraries. This is also true for quadfurcation level 14, which has a similar spatial resolution as the MODIS grid.

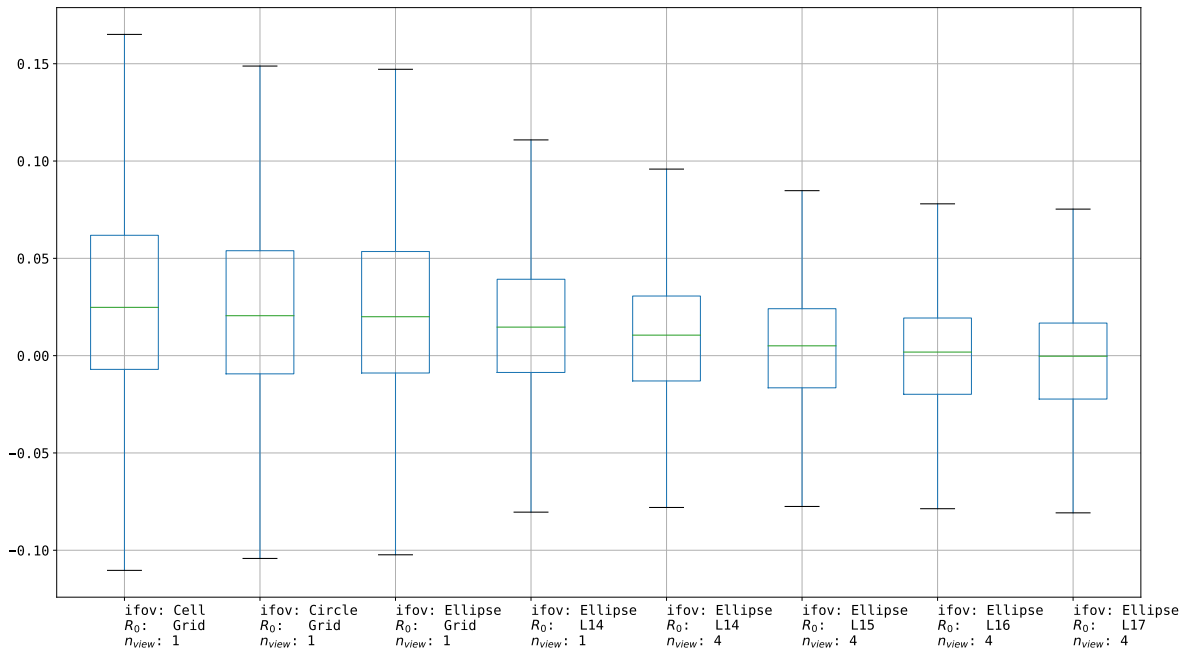


Figure 4.25: Results of the fSCA errors. The three leftmost columns are fSCA computed using the grided R_0 library.

3) The fSCA estimates accuracy improves with higher quadfurcation levels and viewing geometry discretization.

4) The best fSCA estimates are for high spatial resolution (quadfurcation levels 16 and 17) and a medium number of viewing geometry bins (4). The highest accuracy was achieved for STARE quadfurcation levels 16 and 17 and 4 viewing geometry bins.

Table 4.4.1 shows the mean absolute error MAE, the RMSE, and the variance of the fSCA estimates for the scenarios named above. (E. H. Bair, Stillinger, and Dozier, 2021) and (Stillinger, Rittger, et al., 2022) state the RMSE for SPIReS with 0.12 and 0.1, respectively. Those values approximately match the RMSE of our conventional scenario (0.0915). In terms of MAE, the best STARE-based accuracy is achieved at a spatial resolution of 16/17 and 4 viewing geometry discretizations. The MAE is almost halved compared to the conventional grid-based approach.

Table: Mean absolute error, root mean square error, and variance of the fSCA estimates for

different R_0 scenarios. **** | R_0 res |R_0 view bins | IFOV extent | MAE | RMSE | Variance | |:————— |———-: |:———— |——-:|——-:|————:| | grid | 1 | cell | **0.0644** | **0.0915** | 0.0081 | | grid | 1 | circle | 0.0534 | 0.0736 | 0.0051 | | grid | 1 | ellipse | 0.0520 | 0.0718 | 0.0049 | | level 14 | 1 | ellipse | 0.0432 | 0.0632 | 0.0039 | | level 14 | 2 | ellipse | 0.0421 | 0.0621 | 0.0038 | | level 14 | 3 | ellipse | 0.0410 | 0.0620 | 0.0038 | | level 14 | 4 | ellipse | 0.0400 | 0.0611 | 0.0037 | | level 14 | 5 | ellipse | 0.0401 | 0.0616 | 0.0038 | | level 14 | 6 | ellipse | 0.0392 | 0.0609 | 0.0037 | | level 14 | 7 | ellipse | 0.0387 | 0.0606 | 0.0037 | | level 15 | 1 | ellipse | 0.0404 | 0.0616 | 0.0038 | | level 15 | 2 | ellipse | 0.0394 | 0.0608 | 0.0037 | | level 15 | 3 | ellipse | 0.0384 | 0.0608 | 0.0037 | | level 15 | 4 | ellipse | 0.0374 | 0.0600 | 0.0036 | | level 15 | 5 | ellipse | 0.0377 | 0.0605 | 0.0037 | | level 15 | 6 | ellipse | 0.0372 | 0.0600 | 0.0036 | | level 15 | 7 | ellipse | 0.0370 | 0.0602 | 0.0036 | | level 16 | 1 | ellipse | 0.0387 | 0.0607 | 0.0037 | | level 16 | 2 | ellipse | 0.0375 | 0.0598 | 0.0036 | | level 16 | 3 | ellipse | 0.0368 | 0.0597 | 0.0035 | | level 16 | 4 | ellipse | **0.0365** | **0.0596** | 0.0035 | | level 17 | 1 | ellipse | 0.0371 | 0.0591 | 0.0035 | | level 17 | 2 | ellipse | 0.0374 | 0.0603 | 0.0036 | | level 17 | 3 | ellipse | 0.0367 | 0.0603 | 0.0036 | | level 17 | 4 | ellipse | **0.0366** | 0.0602 | 0.0036 |

We interpret the 4 previous findings as follows:

1. Using an IFOV footprint approximation centered around the geolocation is relevant. The circles and ellipses around the relatively precise geolocation are a better approximation of the IFOV than cells of a fixed grid. The triangular point-spread function likely causes the merely marginal improvement of ellipse vs. circle: Most of the information that a sensor collects comes from the area immediately around the center. Therefore, respecting the center location is mainly relevant, while the exact shape of the IFOV approximation may not play a significant role.

2. The MODIS gridding algorithm 'forces' there to be one observation per grid cell per day. On days with far-off nadir overpasses, this leads to oversampling, meaning that a single observation is associated with multiple grid cells (c.f. figure 4.26). Since each grid cell has

its own R_0, each one of those grid cells will have a different fSCA estimate, but not for the right reason. If we compute fSCA for IFOVs rather than for grid cells, we circumvent this error, explaining the immediate accuracy improvement even at STARE quadfurcation level 14.

3. The R_0 library quadfurcation level dictates how closely O and O_0 center locations are matched, while the viewing geometry discretization dictates how closely the footprint shape of the IFOVs of O and O_0 are matched. The closer the footprints match (in terms of location and shape), the less noise from co-registration appears. Higher quadfurcation, therefore, result in higher accuracy. We would not expect further gains at quadfurcation levels higher than the geolocation accuracy.

4. The number of candidates for an ideal snow-free observation during the creation of the R_0 library decreases with increasing resolutions (both in terms of quadfurcation level and viewing geometry discretization). At too-high resolutions, some bins will end up with suboptimal R_0 spectra (e.g., cloud or smoke contamination), driving down the overall accuracy.

VIIRS

We conducted the same analysis for the VIIRS surface reflectance data as we did for the MODIS surface reflectance data. Considering the similar band-passes of MODIS and VIIRS, we also evaluated the accuracy of fSCA estimates when using VIIRS spectra for R and MODIS R_0 spectra. This is interesting as creating an R_0 library at high resolutions requires a long data range, which is not available for new sensors. Thus, the ability to use R_0 from a different sensor than R allows calculating fSCA estimates with SPIReS as soon as a new sensor becomes available.

Table 4.4 summarizes the accuracy metrics of fSCA estimates from VIIRS observations. The overall fSCA accuracy is lower for the VIIRS surface reflectance data than for the MODIS

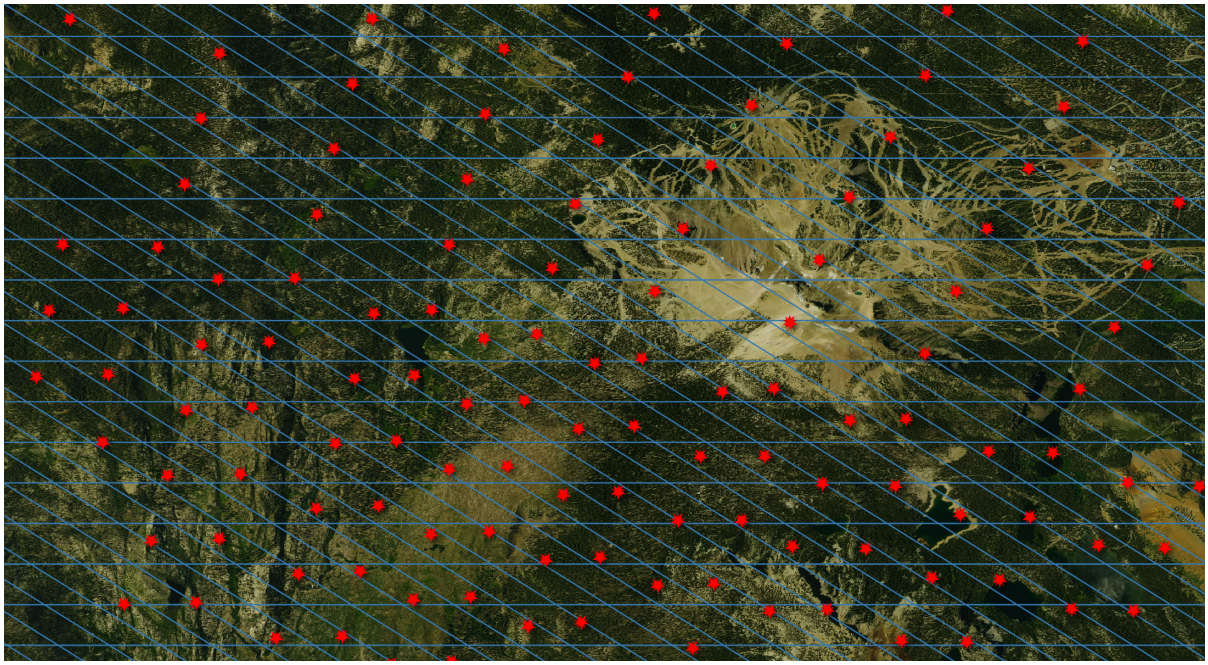


Figure 4.26: Visualization of oversampling for a far off-nadir overpass. Red stars: geolocations of IFOV. Red boxes: MODIS grid cells. There are notably far fewer IFOVsthan grid cells.

surface reflectance data. VIIRS' lower spatial resolution may partially explain this. The viewing geometry discretization does not improve the fSCA accuracy for VIIRS data. This makes sense considering the quasi-constant spatial resolution of VIIRS IFOVs. For VIIRS, the highest accuracy is achieved for lower quadfurcation levels than for the MODIS data. Again, this may be explained by the lower spatial resolution of VIIRS.

Table 4.4: Mean absolute error, root mean square error, and variance of the fSCA estimates for different R_0 scenarios

R_0 res	R_0 view bins	MAE	RMSE	Variance
level 14	1	0.0763	0.1048	0.0107
level 14	2	0.0834	0.1102	0.0116
level 14	3	0.0795	0.1071	0.0111
level 14	4	0.0775	0.1059	0.011

R_0 res	R_0 view bins	MAE	RMSE	Variance
level 14	5	0.0764	0.1048	0.0107
level 14	6	0.074	0.1028	0.0104
level 14	7	0.0742	0.1037	0.0106
level 15	1	0.0742	0.1048	0.0108
level 15	2	0.0798	0.1072	0.0112
level 15	3	0.0765	0.1051	0.0109
level 15	4	0.0745	0.1044	0.0108
level 15	5	0.0738	0.1038	0.0107
level 15	6	**0.0721**	0.1043	0.0108
level 15	7	**0.0721**	0.1052	0.011
level 16	1	0.0743	0.1107	0.0122
level 16	2	0.0779	0.1072	0.0113
level 16	3	0.0757	0.1089	0.0118
level 16	4	0.0742	0.1095	0.012
level 17	1	0.076	0.118	0.0139
level 17	2	0.0775	0.1147	0.0131
level 17	3	0.0801	0.1248	0.0156
level 17	4	0.0812	0.1298	0.0168

We found similar accuracies when calculating fSCA from VIIRS observations using R_0 libraries from MODIS observations. The MAEs range from 0.072 to 0.081.

R_0 res	R_0 view bins	AME	RMSE	Variance
level 14	1	0.0796	0.1064	0.0108
level 14	2	0.0827	0.1090	0.0113
level 14	3	0.0798	0.1063	0.0108

R_0 res	R_0 view bins	AME	RMSE	Variance
level 14	4	0.0780	0.1048	0.0106
level 14	5	0.0768	0.1039	0.0105
level 14	6	0.0763	0.1035	0.0104
level 14	7	0.0754	0.1028	0.0103
level 15	1	0.0771	0.1048	0.0106
level 15	2	0.0808	0.1078	0.0112
level 15	3	0.0774	0.1047	0.0107
level 15	4	0.076	0.1038	0.0105
level 15	5	0.0747	0.1025	0.0103
level 15	6	0.074	0.1021	0.0103
level 15	7	0.073	0.1014	0.0102
level 16	1	0.0746	0.1029	0.0104
level 16	2	0.079	0.1065	0.011
level 16	3	0.0755	0.1032	0.0105
level 16	4	0.0745	0.1027	0.0104
level 17	1	**0.072**	0.1005	0.0099
level 17	2	0.0771	0.1049	0.0108
level 17	3	0.0738	0.1022	0.0103
level 17	4	0.0731	0.1019	0.0103

4.4.2 Time series Plausibility

Figure 4.27 shows a time series of fSCA computed for a grid cell at Reds Lake and CUES using a MODIS-grid R_0 library. Two things are immediately observable:

1) There are substantial fluctuations in the fSCA. These fluctuations are not plausible and

thus are to be interpreted as noise[27].

2) The fSCA values were notably larger than zero for the summer months. However, there was no snow at the two locations for the late summer months - neither in late 2021 nor 2022.

While part of the noise certainly is caused by uncertainties in the atmospheric corrections, cloud cover and shadow, or smoke presence, a significant part of the noise in the time series is induced by the IFOV footprint variations, the spatial mismatching of R and R_0, and oversampling.

In figure 4.28, we display the fSCA computed for a level 15 trixel at Reds Lake, and in figure 4.29 for a level 15 trixel at CUES for MODIS and VIIRS. We computed the fSCA in figure 4.28 and 4.29 with the STARE R_0 library at level 17 and 3 viewing geometry bins. For comparison, we added the snow depth measured at CUES to identify snow precipitation events. Note that we do not expect the snow depth to be proportional to the fSCA at Reds Lake or CUES. However, they are correlated. The timing of the snow accumulation events closely matches the increases in fSCA. We also note that the fSCA during the summer months is closer to zero than for the gridded data in figure 4.27. This is likely caused by a better spatial match of R and R_0. The remaining noise in the summer months may partially be caused by wildfire smoke. The difference between the MODIS fSCA and the VIIRS fSCA can be explained by their differing spatial resolution of 500 m vs. 750 m. Overall, compared to figure 4.27, we traded off temporal resolution for location accuracy, leading to a significantly less noisy signal.

The time series in figure 4.28 and 4.29 include all IFOVs whose geolocation fell into the respective trixels. The blue dots in figure 4.30 are the geolocations of the IFOVs associated with the grid cell around Reds Lake used for figure 4.27. The red and green dots are the geolocations of the MODIS and VIIRS IFOVs that fell into the level 15 trixel used for figure 4.28.

We did not filter out any observations subject to their sensor zenith angle. Figure 4.28 and 4.28

[27]Whether to consider the fluctuations noise or errors depends on the point of view. If we assume that the location precision is irreversibly lost due to gridding, we may call the fluctuations noise. If we on the other side assume that gridding unnecessarily coarsened the resolution, we would conclude the fluctuations to be errors.

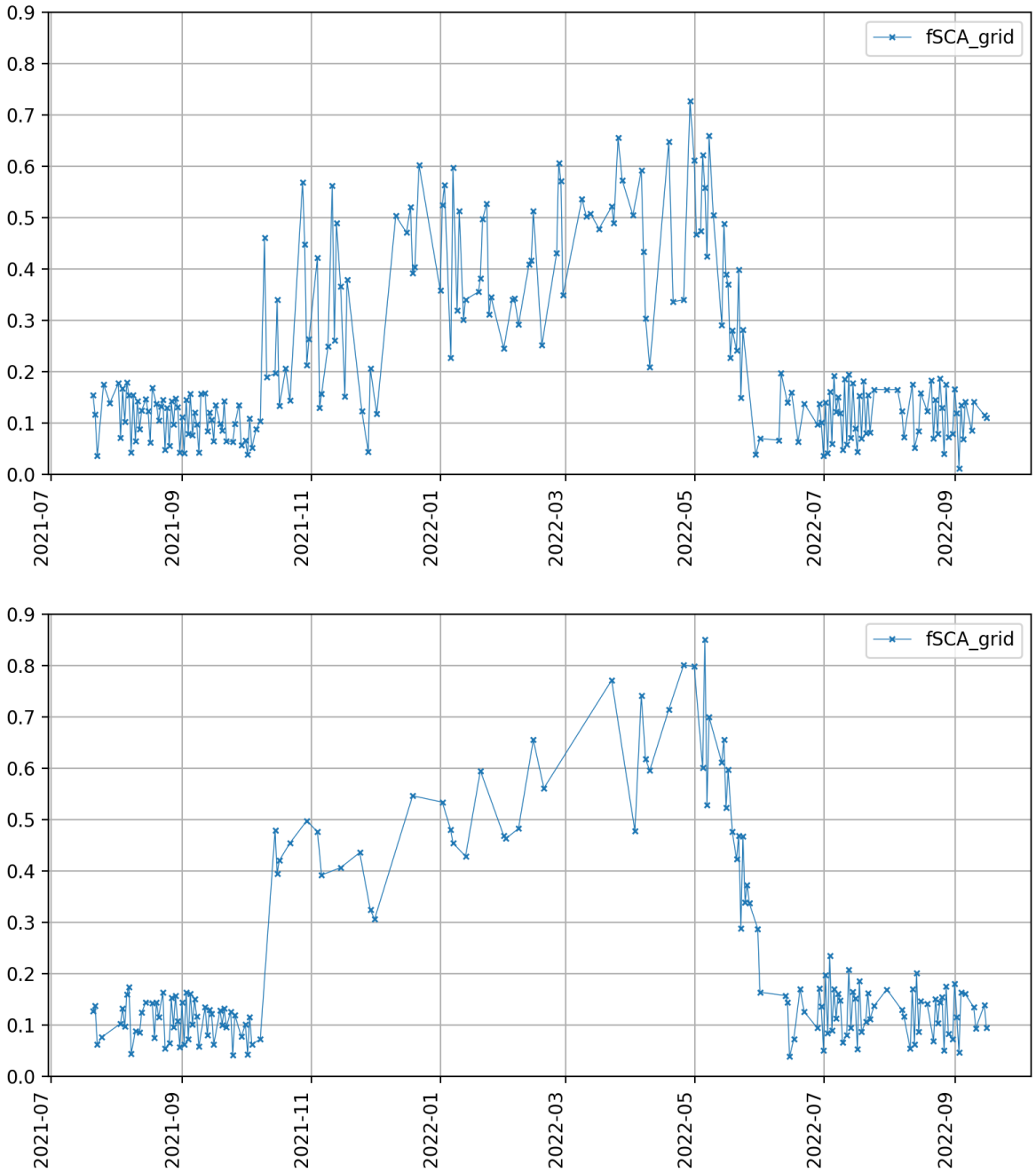


Figure 4.27: SPIReS fSCA for a single grid cell at Reds lake (top) and CUES (bottom). Reds Lake: x = 1373; y = 566. CUES x = 1379; y = 565 of tile H08V05

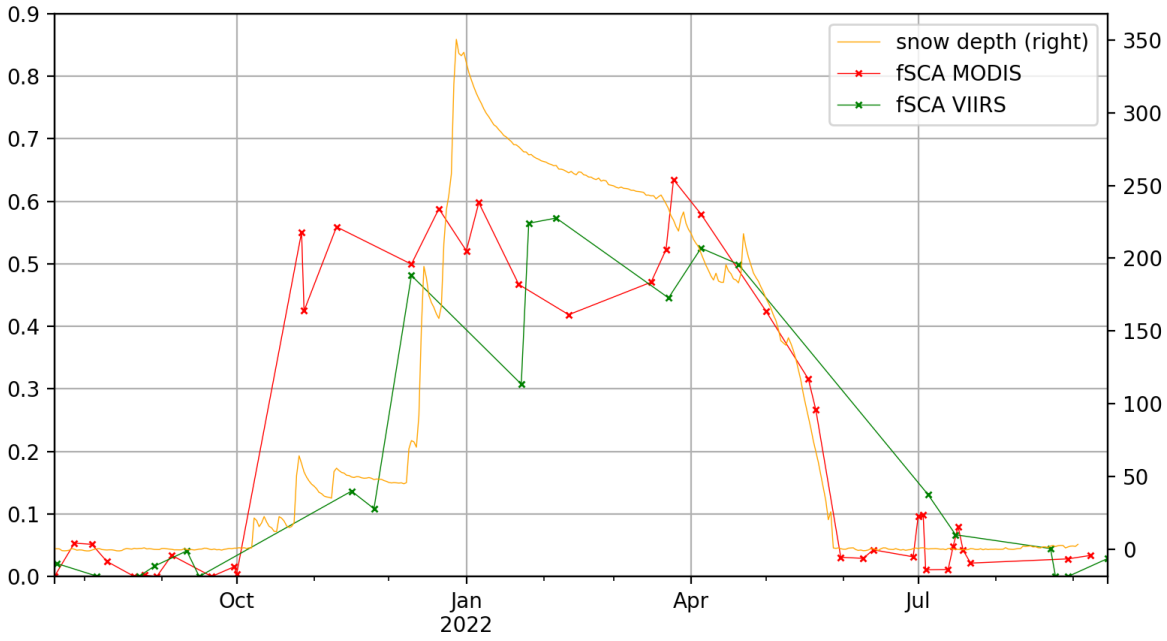


Figure 4.28: SPIReS fSCA for a level 15 trixel at Reds lake.

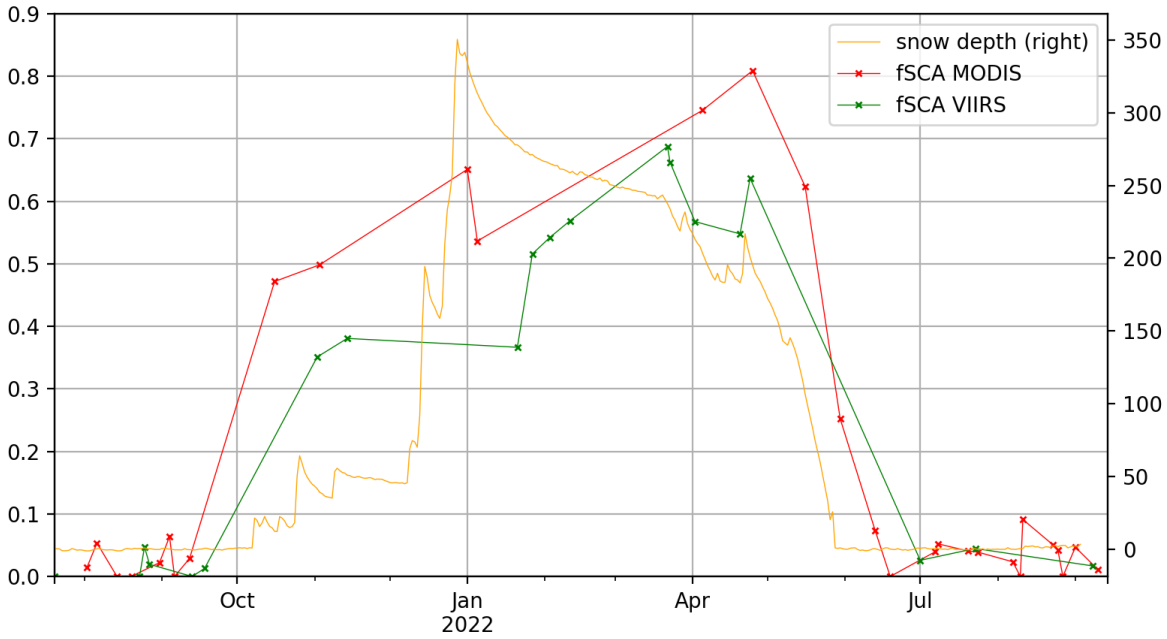


Figure 4.29: SPIReS fSCA for a level 15 trixel at CUES.

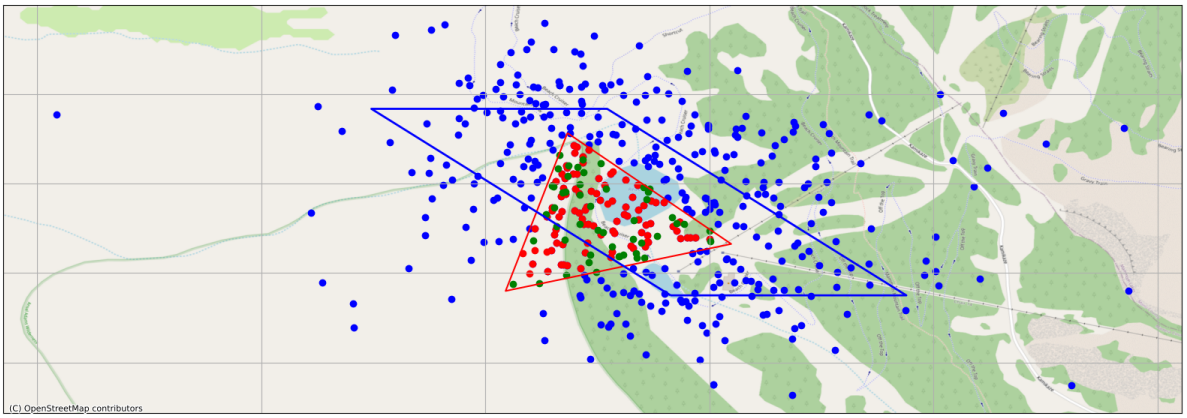


Figure 4.30: A grid cell (blue parallelogram) and all IFOVgeolocations associated with it for the snow season 2021-2022 (blue dots) around Reds lake. The red triangle is a level 15 trixel. The red dots are the MODIS IFOVsthat fell into this trixel, and the green dots are the VIIRS IFOVsthat fell into it for the 2021/2022 snow season.

thus contain IFOVs with significantly larger footprints than others. However, the IFOV centers were all at approximately the same location. Since MODIS and VIIRS have a triangular sensor response function, it may be assumed that the majority of the information of any given pixel does come from the area close to the center location, explaining the smoothness of the curves despite possibly significant differences in the IFOV footprint sizes.

We generated the previous smooth fSCA time series by pegging the location to a sufficiently small extent. However, more often than not, we are interested in the fSCA of an arbitrarily shaped region, such as a control site, a meadow, or a lake. Using STARE, we may define such regions as a set of trixels. STARE makes it easily possible to find all observations that intersect this region, regardless of at what resolution the observations were made.

Figure 4.31 again displays the MODIS cell around Reds lake. Additionally, we added the approximate extent of the meadow around the lake and all MODIS and VIIRS IFOV geolocations intersecting this meadow. We then calculated the fSCA for all those observations and resampled them to weekly values. The resulting signal is displayed in 4.32. For comparison, we also plotted the fSCA of the grid cell resampled to weekly values. The signal for the combined MODIS

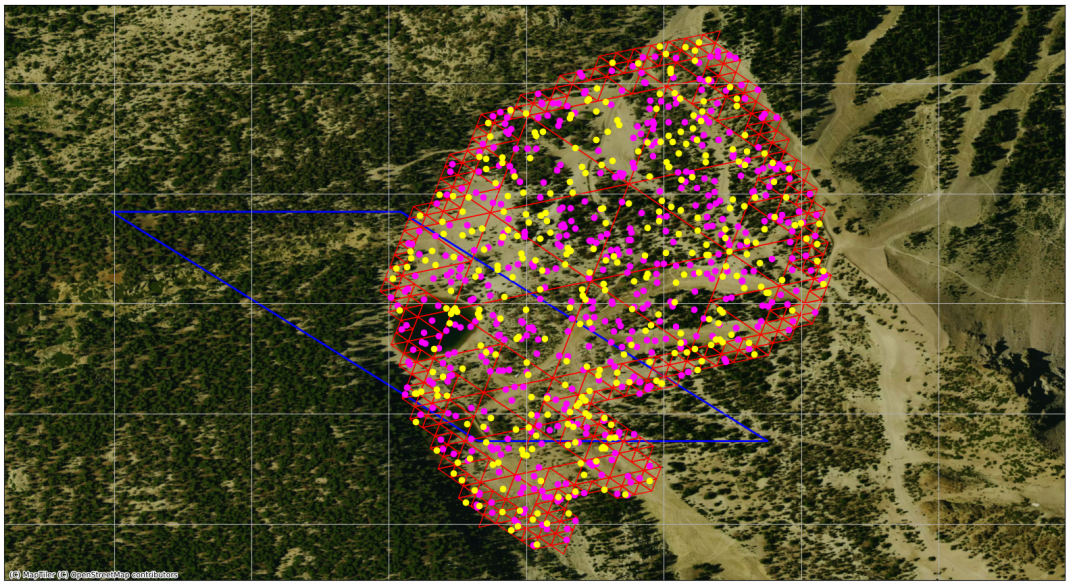


Figure 4.31: Meadow around Reds Lake represented by trixel cover (red triangles). All MODIS and VIIRS observations geolocations that fell into this Region are marked as magenta and yellow dots.

- VIIRS observations is smoother and generally follows the timing of the snowing events. Also, note that the fSCA estimates drop closer to 0 than they do for the gridded fSCA estimates.

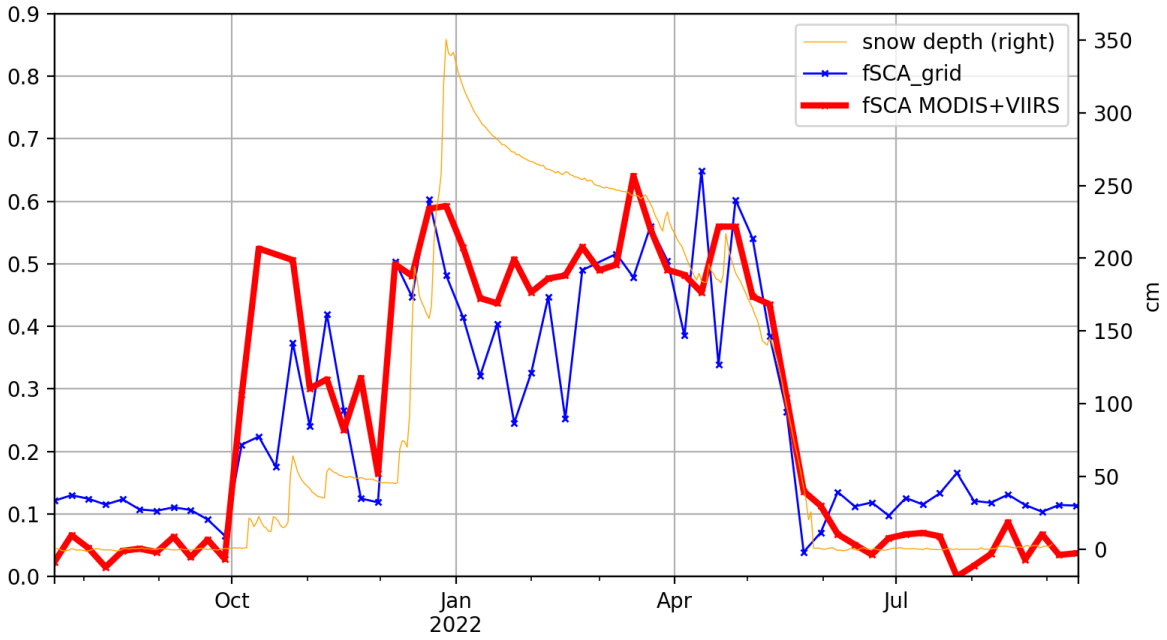


Figure 4.32: fSCA timeline for a complex region for combined MODIS and VIIRS observation (red) and an adjacent grid cell (blue). Both curves have been resampled to 7 days. Note that the fSCA estimates stay well above 0.1 in the summer months for the cell estimates.

4.5 Conclusions and outlook

Since SPIReS requires finding a spatially coinciding snow-free reference spectrum to estimate the fSCA of an observation, it is sensitive to the accuracy of spatial matching of observations. However, the spatial discretization of gridded data disallows for accurate spatial matching. Further, gridded data disallows for a precise evaluation of the estimation accuracy. Gridded data simply does not allow us to determine for what exact area an fSCA estimation was done. Therefore, finding the ground truth data that precisely intersects an observation is impossible. Using STARE, we were able to work directly with ungridded swath data. By forgoing gridded data and working directly with ungridded swath data, we were able to exploit the full spatial fidelity of MODIS surface reflectance data. This allowed us to find snow-free observations that more closely match the area and viewing geometry of observations for which the fSCA are to be estimated. Further, we defined approximate footprints of IFOVs, which enabled us to find the ground truth data that actually intersect the footprints, allowing us to improve the accuracy evaluation. The improvements almost halve the mean absolute error in the fSCA estimations. We further demonstrate that noise in fSCA timeseries can be reduced by pinning down the spatial location and thus trading spatial resolution for temporal resolution. This may be considered a more physic-based smoothing than post-hoc artificial temporal smoothing.

Further efforts will focus on implementing STARE's temporal functionality in the base library and exposing it to pystare, STAREMaster_py, and STAREPandas. Currently, only simple functions to convert between Julian dates and STARE temporal representation (which is based on a hierarchical calendrical partitioning (K.-s. Kuo et al., 2021)) are implemented. Functionality to convert between other temporal representations and methods to perform temporal coincide evaluations need to be implemented to harmonize spatiotemporal data in both spatial and temporal dimensions. Further, efforts have to be devoted to integrating other datasets, such as surface reflectance data from NOAA20 VIIRS, Landsat, MODIS aqua, and GOES. Larger ROIs with different topography should be processed to evaluate the influence of the topography on

the results. Finally, more research needs to be carried out to understand the relation between fSCA estimate accuracy and spatial matching of R_0, viewing geometries, oversampling, and specular effects.

Acknowledgements

This works was partially funded though the National Aeronautics and Space Administration (NASA) Advancing Collaborative Connections for Earth System Science (ACCESS) program (Award number: 80NSSC18M0118), which allowed us to develop the core functionality of the STARE software collection. We thank our project collaborators at Bayesics LLC and Open-source Project for a Network Data Access Protocol (OPeNDAP). We thank Ned Bair, Jeff Dozier, Karl Rittger, and Timbo Stillinger for their invaluable guidance on remote sensing of snow.

Chapter 5

Conclusions

File-centric data analysis is a paradigm in which files are the smallest unit of data. While file-centricity simplifies the task of archiving and distributing data, it pushes the burden of extracting, transforming, and loading (ETL) data before performing any data analysis to the data users. Data-centric data analysis, on the other hand, is a paradigm in which the smallest unit of data are individual observations (e.g., instances/objects in some form of a schema), and data thus are stored in some form of a database and accessed by some form of query.

I addressed two questions that need to be solved to allow data repositories and data users to move toward data-centricity:

In chapter 2, I described Open-source Project for a Network Data Access Protocol (OPeNDAP) Citation Creator (OCCUR); a system providing identity and citations to data in a data-centric world. Identifying data in a data-centric world is different from identifying data in a file-centric world: In a file-centric world, data is accessed through, e.g., file paths or Uniform Resource Locators (URLs), and files themselves can be understood as the identities of the data they contain. In a data-centric world, the notion of files does not exist. Data is instead accessed through queries; therefore, we must be able to identify the result sets of queries. OCCUR allows us to identify data that is queried through OPeNDAP. However, it is intended to be a reference

implementation that can be adapted to any other data distribution system in a data-centric world. That is, any data accessed through some form of a query can be identified by a system similar to OCCUR.

In chapter 3, I provide a solution to harmonize spatial data. Harmonizing data means ensuring that things that are the same are referred to as the same. Only if data is harmonized can we associate data from different datasets and perform data analysis across multiple datasets. Harmonizing data, therefore, is a requirement to void the necessity of ETL and fully leverage the benefits of data-centricity. In the context of spatial data, harmonization means that there needs to be one unified method to express location. The concept of Spatio-Temporal Adaptive-Resolution Encoding (STARE) provides such a unified method, and the STARE software collection implements the STARE concept. Any spatial object can be represented within STARE. Further, evaluating spatial relations between spatial objects represented in STARE representation is cheap, allowing one to associate different datasets by their location. While the world may remain in a transitional phase between file-centricity and data-centricity for some time, the STARE software collection provides a bride toward data-centric spatial data analysis. With its rich capabilities to load and convert conventional spatial representation formats, the STARE software collection provides a method to harmonize any spatial data, e.g., at the beginning of a data processing pipeline, and thus allow for a data-centric workflow.

In chapter 4, I provide an example of how scientific data analysis of remotely sensed data can be performed in a data-centric world and highlight the benefits of performing spatial data analysis on spatially harmonized data with the STARE software collection. Previous methods to harmonize spatial data have been based on location discretization and data sampling using regular grids. Those methods fail us when working with data collected at varying spatial resolutions since they require us to re-grid and re-sample data, both of which likely entail suboptimal compromises. Further, the location discretization reduces data's spatial fidelity, resulting in noise in derived products. By working with data that has been harmonized through STARE instead, I avoided both of these problems. By avoiding spatial discretization, I increased the accuracy

of an algorithm for fractional snow cover estimations and reduced the noise in the time series of those fractional snow cover estimations. By having all input data harmonized through STARE, I could effortlessly spatially associate grid cells, regions of interest, and individual observations from WorldView Legion, Moderate Resolution Imaging Spectroradiometer (MODIS), and Visible Infrared Imaging Radiometer Suite (VIIRS). Other algorithms and data analysis efforts likely will benefit from working with data that has been spatially aligned through STARE, both in terms of simplifying the ETL process and in terms of using the full spatial fidelity of observations.

Bibliography

VIIRS Calibration Support Team (VCST). *VIIRS/NPP Day/Night Band 6-Min L1B Swath SDR- 750m*. 2021. DOI: 10.5067/VIIRS/VNP02DNB.002. URL: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/VNP02DNB/ (cit. on p. 83).

VIIRS Calibration Support Team (VCST). *VIIRS/NPP Moderate Resolution Terrain-Corrected Geolocation L1 6-Min Swath- 750m*. 2021. DOI: 10.5067/VIIRS/VNP03MOD.002. URL: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/VNP03MOD/ (cit. on p. 123).

Abdussalam Alawini et al. "Automating Data Citation: The eagle-i Experience". In: *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. IEEE, June 2017, pp. 1–10. ISBN: 978-1-5386-3861-3. DOI: 10.1109/JCDL.2017.7991571. URL: http://ieeexplore.ieee.org/document/7991571/ (cit. on p. 25).

Micah Altman, Christine Borgman, et al. "An Introduction to the Joint Principles for Data Citation". In: 41.3 (2015) (cit. on pp. 9, 11, 14, 16).

Micah Altman and Gary King. "A Proposed Standard for the Scholarly Citation of Quantitative Data". In: *D-Lib Magazine* 13 (Mar. 2007). DOI: 10.1045/march2007-altman. URL: http://www.dlib.org/dlib/march07/altman/03altman.html (cit. on pp. 9, 12–14, 16).

Massimiliano Assante et al. "Are Scientific Data Repositories Coping with Research Data Publishing?" In: *Data Science Journal* 15.6 (Apr. 2016), pp. 1–24. ISSN: 1683-1470. DOI: 10.5334/dsj-2016-006. URL: http://dx.doi.org/10.5334/dsj-2016-006%20http://datascience.codata.org/articles/10.5334/dsj-2016-006/ (cit. on pp. 10, 11).

Tim Austin. "Towards a digital infrastructure for engineering materials data". In: *Materials Discovery* 3 (Mar. 2016), pp. 1–12. ISSN: 23529245. DOI: 10.1016/j.md.2015.12.003. URL: http://dx.doi.org/10.1016/j.md.2015.12.003%20http://linkinghub.elsevier.com/retrieve/pii/S235292451600003X (cit. on p. 25).

Edward H. Bair, Timbo Stillinger, and Jeff Dozier. "Snow Property Inversion From Remote Sensing (SPIReS): A Generalized Multispectral Unmixing Approach With Examples From MODIS and Landsat 8 OLI". In: *IEEE Transactions on Geoscience and Remote Sensing* 59.9 (Sept. 2021), pp. 7270–7284. ISSN: 0196-2892. DOI: 10.1109/TGRS.2020.3040328. URL: https://ieeexplore.ieee.org/document/9290428/ (cit. on pp. 109, 143).

Alex Ball and Monica Duke. *How to Cite Datasets and Link to Publications*. Tech. rep. Edinburgh: Digital Curation Centre, 2015. URL: http://www.dcc.ac.uk/resources/how-guides (cit. on pp. 14, 15).

Anita Bandrowski et al. "The Resource Identification Initiative: A cultural shift in publishing". In: *Brain and Behavior* 6.1 (2016), pp. 1–14. ISSN: 21623279. DOI: 10.1002/brb3.417 (cit. on pp. 10, 12).

Richard Barbieri et al. *The MODIS Level 1B Algorithm Theoretical Basis Document Version 2.0*. Tech. rep. Greenbelt, MD: NASA Goddard Space Flight Center, 1997, p. 68 (cit. on p. 112).

Tom Barclay, Robert Eberl, et al. "Microsoft TerraServer". In: *CoRR* cs.DB/9809.June (Sept. 1998). arXiv: 9809011 [cs]. URL: https://arxiv.org/abs/cs/9809011%20http://arxiv.org/abs/cs/9809011 (cit. on p. 48).

Tom Barclay, Jim Gray, and Don Slutz. "Microsoft TerraServer: A Spatial Data Warehouse". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00* cs.DB/9907.June (July 1999), pp. 307–318. ISSN: 01635808. DOI: 10.1145/342009.335424. arXiv: 9907016 [cs]. URL: cs/9907016%20http://portal.acm.org/citation.cfm?doid=342009.335424%20http://arxiv.org/abs/cs/9907016 (cit. on p. 48).

T. P. Barnett, J. C. Adam, and Dennis P. Lettenmaier. "Potential impacts of a warming climate on water availability in snow-dominated regions". In: *Nature* 438.7066 (Nov. 2005), pp. 303–309. ISSN: 0028-0836. DOI: 10.1038/nature04141. arXiv: arXiv:1011.1669v3. URL: http://www.nature.com/articles/nature04141 (cit. on p. 105).

P Barret. "Application of the Linear Quadtree to Astronomical Databases". In: *Astronomical Data Analysis Software and Systems IV*. Ed. by R.A. Shaw, H.E. Payne, and J.J.E. Hayes. Vol. 77. 1995, p. 472. URL: http://adsabs.harvard.edu/abs/1995ASPC...77..472B (cit. on p. 48).

Sean Bechhofer et al. "Why linked data is not enough for scientists". In: *Future Generation Computer Systems* 29.2 (Feb. 2013), pp. 599–611. ISSN: 0167739X. DOI: 10.1016/j.future.

2011.08.004. URL: http://linkinghub.elsevier.com/retrieve/pii/S0167739X11001439 (cit. on p. 14).

Christopher W. Belter. "Measuring the Value of Research Data: A Citation Analysis of Oceanographic Data Sets". In: *PLoS ONE* 9.3 (Mar. 2014). Ed. by Howard I. Browman, e92590. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0092590. URL: http://dx.plos.org/10.1371/journal.pone.0092590 (cit. on p. 10).

Michael Bevis and Jean-Luc Chatelain. "Locating a point on a spherical surface relative to a spherical polygon of arbitrary shape". In: *Mathematical Geology* 21.8 (Oct. 1989), pp. 811–828. ISSN: 0882-8121. DOI: 10.1007/BF00894449. URL: http://link.springer.com/10.1007/BF00894449 (cit. on p. 51).

P Bolstad. *GIS Fundamentals: A First Text on Geographic Information Systems*. Eider Press, 2002. ISBN: 9780971764705. URL: https://books.google.com/books?id=c-ksAQAAMAAJ (cit. on p. 61).

Tamás Budavári, Alexander S. Szalay, and György Fekete. "Searchable Sky Coverage of Astronomical Observations: Footprints and Exposures". In: *Publications of the Astronomical Society of the Pacific* 122.897 (Nov. 2010), pp. 1375–1388. ISSN: 0004-6280. DOI: 10.1086/657302. arXiv: 1005.2606. URL: http://www.journals.cambridge.org/abstract%7B%5C_%7DS0140525X00060003%20http://arxiv.org/abs/1005.2606%20http://dx.doi.org/10.1086/657302%20http://iopscience.iop.org/article/10.1086/657302 (cit. on p. 74).

Peter Buneman, Susan B. Davidson, and James Frew. "Why data citation is a computational problem". In: *Communications of the ACM* 59.9 (Aug. 2016), pp. 50–57. ISSN: 00010782. DOI: 10.1145/2893181. URL: http://delivery.acm.org/10.1145/2900000/2893181/p50-buneman.pdf?ip=128.111.110.169%7B%5C&%7Did=2893181%7B%5C&%7Dacc=CHORUS%7B%5C&%7Dkey=CA367851C7E3CE77.022A0CC51A76093F.4D4702B0C3E38B35.6D218144511F3437%7B%5C&%7D%7B%5C_%7D%7B%5C_%7Dacm%7B%5C_%7D%7B%5C_%7D=1529972578%7B%5C_%7Db344595fdf25a6f12b2f1e1ccca9a8d8%20https://dl.acm.org/citatio (cit. on pp. 11, 12, 14, 16).

Peter Buneman and Gianmaria Silvello. "A Rule-Based Citation System for Structured and Evolving Datasets". In: *IEEE Data Eng. Bull.* 33 (2010), pp. 33–41 (cit. on pp. 12–14, 16, 26).

Sarah Callaghan et al. "Making Data a First Class Scientific Output: Data Citation and Publication by NERC's Environmental Data Centres". In: *International Journal of Digital Curation* 7.1 (Mar. 2012), pp. 107–113. ISSN: 1746-8256. DOI: 10.2218/ijdc.v7i1.218. URL: http://www.ijdc.net/article/view/208 (cit. on pp. 9, 11, 14).

Robert G. Chamberlain and William H. Duquette. “Some algorithms for polygons on a sphere”. In: *Jet Propulsion Laboritory* April (2007), p. 30. URL: http://trs-new.jpl.nasa.gov/dspace/handle/2014/41271%7B%5C%%7D5Cnhttp://trs-new.jpl.nasa.gov/dspace/bitstream/2014/40409/1/07-03.pdf (cit. on p. 51).

Fay Chang et al. “Bigtable: A Distributed Storage System for Structured Data”. In: *ACM Transactions on Computer Systems* 26.2 (June 2008), pp. 1–26. ISSN: 07342071. DOI: 10.1145/1365815.1365816. URL: http://portal.acm.org/citation.cfm?doid=1365815.1365816 (cit. on p. 98).

CODATA-ICSTI Task Group on Data Citation Standards and Practices. “Out of Cite, Out of Mind: The Current State of Practice, Policy, and Technology for the Citation of Data”. In: *Data Science Journal* 12.0 (2013), CIDCR1–CIDCR75. ISSN: 1683-1470. DOI: 10.2481/dsj.OSOM13-043. URL: http://datascience.codata.org/articles/abstract/10.2481/dsj.OSOM13-043/ (cit. on pp. 9, 12–14, 16).

Michael Colee. *CRREL/UCSB Energy Site*. 2016. DOI: 10.21424/R4159Q. URL: http://www.snow.ucsb.edu/level-2-fully-fitered-files (cit. on p. 139).

Robert B Cook et al. “Implementation of data citations and persistent identifiers at the ORNL DAAC”. In: *Ecological Informatics* 33 (May 2016), pp. 10–16. ISSN: 15749541. DOI: 10.1016/j.ecoinf.2016.03.003. URL: http://dx.doi.org/10.1016/j.ecoinf.2016.03.003%20http://linkinghub.elsevier.com/retrieve/pii/S1574954116300140 (cit. on p. 25).

James C Corbett et al. “Spanner: Google’s Globally-Distributed Database”. In: *Osdi* 31.3 (2012), pp. 1–14. ISSN: 07342071. DOI: 10.1145/2491245 (cit. on p. 98).

Mark J Costello. “Motivating Online Publication of Data”. In: *BioScience* 59.5 (May 2009), pp. 418–427. ISSN: 0006-3568. DOI: 10.1525/bio.2009.59.5.9. URL: https://academic.oup.com/bioscience/article-lookup/doi/10.1525/bio.2009.59.5.9 (cit. on pp. 9, 14).

R. G. Crane and M. R. Anderson. “Satellite discrimination of snow/cloud surfaces”. In: *International Journal of Remote Sensing* 5.1 (1984), pp. 213–223. ISSN: 13665901. DOI: 10.1080/01431168408948799 (cit. on p. 106).

Mercè Crosas. “The Dataverse Network®: An Open-Source Application for Sharing, Discovering and Preserving Data”. In: *D-Lib Magazine* 17.1/2 (Jan. 2011), pp. 1–8. ISSN: 1082-9873. DOI: 10.1045/january2011-crosas. URL: http://www.dlib.org/dlib/january11/crosas/01crosas.html (cit. on pp. 13, 14, 25).

Data Citation Synthesis Group. *Joint Declaration of Data Citation Principles*. 2014. DOI: 10.25490/a97f-egyk. URL: https://www.force11.org/group/joint-declaration-data-citation-principles-final (cit. on pp. 9, 16).

Susan B. Davidson et al. "Data Citation: a Computational Challenge". In: *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems - PODS '17*. Vol. Part F1277. New York, New York, USA: ACM Press, 2017, pp. 1–4. ISBN: 9781450341981. DOI: 10.1145/3034786.3056123. URL: http://dl.acm.org/citation.cfm?doid=3034786.3056123 (cit. on pp. 13, 14).

Khoa Doan et al. "Evaluating the Impact of Data Placement to Spark and SciDB with an Earth Science Use Case". In: *2016 IEEE International Conference on Big Data (Big Data)*. Washington D.C.,USA, 2016, pp. 341–346. DOI: 10.1109/BigData.2016.7840621. URL: https://ieeexplore.ieee.org/document/7840621/ (cit. on p. 48).

Jeff Dozier. "A method for satellite identification of surface temperature fields of subpixel resolution". In: *Remote Sensing of Environment* (1981). ISSN: 00344257. DOI: 10.1016/0034-4257(81)90021-3 (cit. on p. 107).

Jeff Dozier. "Spectral signature of alpine snow cover from the landsat thematic mapper". In: *Remote Sensing of Environment* 28.1 (Apr. 1989), pp. 9–22. ISSN: 00344257. DOI: 10.1016/0034-4257(89)90101-6. URL: http://linkinghub.elsevier.com/retrieve/pii/0034425789901016 (cit. on p. 106).

Jeff Dozier and Thomas H. Painter. "MULTISPECTRAL AND HYPERSPECTRAL REMOTE SENSING OF ALPINE SNOW PROPERTIES". In: *Annual Review of Earth and Planetary Sciences* 32.1 (May 2004), pp. 465–494. ISSN: 0084-6597. DOI: 10.1146/annurev.earth.32.101802.120404. URL: http://www.annualreviews.org/doi/10.1146/annurev.earth.32.101802.120404 (cit. on pp. 105–107).

Jeff Dozier, Thomas H. Painter, et al. "Time–space continuity of daily maps of fractional snow cover and albedo from MODIS". In: *Advances in Water Resources* 31.11 (Nov. 2008), pp. 1515–1526. ISSN: 03091708. DOI: 10.1016/j.advwatres.2008.08.011. URL: http://linkinghub.elsevier.com/retrieve/pii/S0309170808001437%20https://linkinghub.elsevier.com/retrieve/pii/S0309170808001437 (cit. on pp. 112, 130).

Mike Durand et al. *NASA SnowEx Science Plan: Assessing Approaches for Measuring Water in Earth's Seasonal Snow*. Tech. rep. 2017, pp. 1–68 (cit. on p. 105).

Geoffrey Dutton. “Encoding and Handling Geospatial Data with Hierarchical Triangular Meshes”. In: *Proceedings of the 7th Symposium on Spatial Data Handling*. Zürich, 1996, pp. 505–518. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.461.5603%20http://www.spatial-effects.com/papers/conf/GDutton%7B%5C_%7DSDH96.pdf (cit. on pp. 43, 47).

Geoffrey Dutton. “Geodesic modelling of planetary relief”. In: *Cartographica* 21 (1984), pp. 188–207. ISSN: 03177173 (cit. on p. 47).

Geoffrey Dutton. “Modeling locational uncertainty via hierarchical tesselation”. In: *The Accuracy of Spatial Databases*. Ed. by Michael F. Goodchild and Sucharita Gopal. Santa Barbara: Taylor & Francis, 1989. URL: https://books.google.com/books?id=HL2O6J-XtLAC%7B%5C&%7Dprintsec=frontcover%7B%5C#%7Dv=onepage%7B%5C&%7Dq%7B%5C&%7Df=false (cit. on p. 47).

ESIP Data Preservation and Stewardship Committee. “Data Citation Guidelines for Earth Science Data. Ver. 2”. In: (2019), pp. 1–20. DOI: 10.6084/m9.figshare.8441816 (cit. on p. 14).

Nickolas Faust et al. “Real-time global data model for the digital earth”. In: *International Conference on Discrete Global Grids* (2000). URL: http://www.ncgia.ucsb.edu/globalgrids/papers/faust.pdf (cit. on p. 36).

Federation of Earth Science Information Partners (ESIP). *Data Citation Guidelines for Data Providers and Archives*. 2012. DOI: 10.7269/P34F1NNJ. URL: http://commons.esipfed.org/node/308 (cit. on pp. 12, 14).

György Fekete. “Rendering and managing spherical data with sphere quadtrees”. In: *Proceedings of the First IEEE Conference on Visualization: Visualization ‘90*. San Francisco, California: IEEE Comput. Soc. Press, 1990, pp. 176–186. ISBN: 0-8186-2083-8. DOI: 10.1109/VISUAL.1990.146380. URL: http://files/6580/1990%20-%20Fekete%20-%20VIS%20’90%20Proceedings%20of%20the%201st%20conference%20on%20Visualization%20’90.pdf%20https://dl.acm.org/citation.cfm?id=949560%20https://ieeexplore.ieee.org/document/146380/%20http://ieeexplore.ieee.org/document/146380/ (cit. on p. 47).

György Fekete and Lloyd Treinish. “Sphere quadtrees - A new data structure to support the visualization of spherically distributed data”. In: *SPIE 1259, Extracting Meaning from Complex Data: Processing, Display, Interaction*. Vol. 1259. August 1990. 1990, pp. 242–253. DOI: 10.1117/12.19991. URL: https://www.spiedigitallibrary.org/conference-proceedings-

of-spie/1259/1/Sphere-quadtrees--a-new-data-structure-to-support-the/10.1117/12.19991.full?SSO=1%20https://doi.org/10.1117/12.19991 (cit. on pp. 43, 47).

James Frew, Greg Janée, and Peter Slaughter. "Provenance-Enabled Automatic Data Publishing". In: *Scientific and Statistical Database Management*. Ed. by Judith Bayard Cushing, James French, and Shawn Bowers. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 244–252. ISBN: 978-3-642-22351-8. DOI: 10.1007/978-3-642-22351-8_14. URL: http://eri.ucsb.edu%20http://link.springer.com/10.1007/978-3-642-22351-8%7B%5C_%7D14 (cit. on p. 14).

James Frew, Dominic Metzger, and Peter Slaughter. "Automatic capture and reconstruction of computational provenance". In: *Concurrency and Computation: Practice and Experience* 20.5 (Apr. 2008), pp. 485–496. ISSN: 15320626. DOI: 10.1002/cpe.1247. URL: http://doi.wiley.com/10.1002/cpe.1247 (cit. on p. 14).

James E. Frew and Jeff Dozier. "Environmental informatics". In: *Annual Review of Environment and Resources* 37.1 (2012), pp. 449–472. ISSN: 15435938. DOI: 10.1146/annurev-environ-042711-121244. URL: http://dx.doi.org/10.1146/annurev-environ-042711-121244%20http://www.annualreviews.org/doi/10.1146/annurev-environ-042711-121244%20https://www.annualreviews.org/doi/10.1146/annurev-environ-042711-121244 (cit. on pp. 2, 9).

James Gallagher, Edward J. Hartnett, et al. "STARE Companion Files for NASA Earth Science Data". In: *International Geoscience and Remote Sensing Symposium (IGARSS)* (July 2021), pp. 5668–5671. DOI: 10.1109/IGARSS47720.2021.9553950. URL: https://ieeexplore.ieee.org/document/9553950/ (cit. on p. 57).

James Gallagher, Nathan Potter, et al. *The Data Access Protocol - DAP 2.0*. Tech. rep. OPeNDAP, 2007. URL: https://www.opendap.org/pdf/ESE-RFC-004v1.2.pdf (cit. on pp. 4, 11).

Michael F. Goodchild and Yang Shiren. "A hierarchical spatial data structure for global geographic information systems". In: *CVGIP: Graphical Models and Image Processing* 54.1 (Jan. 1992), pp. 31–44. ISSN: 10499652. DOI: 10.1016/1049-9652(92)90032-S. URL: http://linkinghub.elsevier.com/retrieve/pii/104996529290032S%20http://www.sciencedirect.com/science/article/pii/104996529290032S%20https://linkinghub.elsevier.com/retrieve/pii/104996529290032S (cit. on pp. 43, 47).

Jim Gray et al. "Online Scientific Data Curation, Publication, and Archiving". In: *CoRR* (Aug. 2002). Ed. by Alexander S. Szalay, pp. 103–107. DOI: 10.1117/12.461524. URL: http://arxiv.org/abs/cs/0208012%20http://dx.doi.org/10.1117/12.461524%20http:

//proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=874929 (cit. on pp. 3, 10).

K Green and M Tukman. *Imagery and GIS: Best Practices for Extracting Information from Imagery*. Esri Press, 2017. ISBN: 9781589484542. URL: https://books.google.com/books?id=2VQqvgAACAAJ (cit. on p. 61).

Niklas Griessbaum. "Improving fractional snow cover estimates through increased spatial fidelity ." PhD thesis. 2022, pp. 1–45 (cit. on p. 93).

Niklas Griessbaum, James Gallagher, et al. "Solving science use cases with STARE (Demo Paper)". In: *Proceedings of ACM Sigspatial conference (SIGSPATIAL'20). ACM, New York, NY*. 2020 (cit. on p. vi).

Niklas Griessbaum, Kwo-Sen Kuo, et al. "Harmonizing with STARE-PODS to enable best-resolution Analysis-ready data". In: *AGU Fall Meeting Abstracts*. Vol. 2021. Dec. 2021, IN32A–05 (cit. on p. 75).

Dorothy K. Hall and George A. Riggs. *MODIS/Terra Snow Cover Daily L3 Global 500m Grid, Version 6*. 2016. DOI: 10.5067/MODIS/MOD10A1.006. URL: https://nsidc.org/data/mod10a1 (cit. on p. 106).

Dorothy K. Hall, George A. Riggs, and Vincent V. Salomonson. "Algorithm Theoretical Basis Document (ATBD) for the MODIS Snow and Sea Ice-Mapping Algorithms". In: (2001) (cit. on p. 106).

Dorothy K. Hall, George A. Riggs, and Vincent V. Salomonson. "Development of methods for mapping global snow cover using moderate resolution imaging spectroradiometer data". In: *Remote Sensing of Environment* 54.2 (Nov. 1995), pp. 127–140. ISSN: 00344257. DOI: 10.1016/0034-4257(95)00137-P. URL: http://linkinghub.elsevier.com/retrieve/pii/003442579500137P (cit. on p. 106).

Dorothy K. Hall, George A. Riggs, Vincent V. Salomonson, et al. "MODIS snow-cover products". In: *Remote Sensing of Environment* 83.1-2 (Nov. 2002), pp. 181–194. ISSN: 00344257. DOI: 10.1016/S0034-4257(02)00095-0. URL: http://linkinghub.elsevier.com/retrieve/pii/S0034425702000950 (cit. on p. 106).

James Hansen and Larissa Nazarenko. “Soot climate forcing via snow and ice albedos”. In: *Proceedings of the National Academy of Sciences of the United States of America* 101.2 (2004), pp. 423–428. ISSN: 00278424. DOI: 10.1073/pnas.2237157100 (cit. on p. 105).

Tony Hey, Stewart Tansley, and Kristin Tolle. *The Fourth Paradigm - Data-Intensive Scientific Discovery*. 2009, p. 287. ISBN: 9780982544204. URL: http://www.astro.caltech.edu/%7B~%7Dgeorge/aybi199/4th%7B%5C_%7Dparadigm%7B%5C_%7Dbook%7B%5C_%7Dcomplete%7B%5C_%7Dlr.pdf (cit. on pp. 3, 5, 9).

Leah B Honor et al. “Data Citation in Neuroimaging: Proposed Best Practices for Data Identification and Attribution”. In: *Frontiers in Neuroinformatics* 10 (Aug. 2016), p. 34. ISSN: 1662-5196. DOI: 10.3389/fninf.2016.00034. URL: https://www.frontiersin.org/article/10.3389/fninf.2016.00034%20http://journal.frontiersin.org/Article/10.3389/fninf.2016.00034/abstract (cit. on pp. 10, 26).

Robert Huber et al. “Data citation and digital identifiers for time series data / environmental research infrastructures”. In: (2015). DOI: 10.6084/m9.figshare.1285728. URL: http://figshare.com/articles/Data%7B%5C_%7Dcitation%7B%5C_%7Dand%7B%5C_%7Ddigital%7B%5C_%7Didentifiers%7B%5C_%7Dfor%7B%5C_%7Dtime%7B%5C_%7Dseries%7B%5C_%7Ddata%7B%5C_%7Denvironmental%7B%5C_%7Dresearch%7B%5C_%7Dinfrastructures/1285728%7B%5C%%7D0Ahttps://www.bodc.ac.uk/about/outputs/presentations%7B%5C_%7Dand%7B%5C_%7Dpapers/documents/datacitation%7B%5C_%7Djuck.pdf (cit. on pp. 13, 14).

Nirmal Keshava. “A Survey of Spectral Unmixing Algorithms”. In: *Lincoln Laboratory Journal* 14 (Jan. 2003), pp. 55–78 (cit. on p. 107).

Michael D King et al. “Design and Production EOS Data Products Handbook”. In: (2003) (cit. on p. 131).

Andrew G. Klein, Dorothy K. Hall, and George A. Riggs. “Improving snow cover mapping in forests through the use of a canopy reflectance model”. In: *Hydrological Processes* 12.10-11 (Aug. 1998), pp. 1723–1744. ISSN: 08856087. DOI: 10.1002/(SICI)1099-1085(199808/09)12:10/11<1723::AID-HYP691>3.0.CO;2-2. URL: http://doi.wiley.com/10.1002/%7B%5C%%7D28SICI%7B%5C%%7D291099-1085%7B%5C%%7D28199808/09%7B%5C%%7D2912%7B%5C%%7D3A10/11%7B%5C%%7D3C1723%7B%5C%%7D3A%7B%5C%%7D3AAID-HYP691%7B%5C%%7D3E3.0.CO%7B%5C%%7D3B2-2 (cit. on p. 106).

Jens Klump, Robert Huber, and Michael Diepenbroek. “DOI for geoscience data - how early practices shape present perceptions”. In: *Earth Science Informatics* 9.1 (Mar. 2016), pp. 123–

136. ISSN: 1865-0473. DOI: 10.1007/s12145-015-0231-5. URL: http://link.springer.com/10.1007/s12145-015-0231-5 (cit. on pp. 14, 15).

Tibor Koltay. "Digital Research Data". In: *Digital Information Strategies*. Elsevier, 2016, pp. 71–84. DOI: 10.1016/B978-0-08-100251-3.00005-6. URL: http://dx.doi.org/10.1016/b978-0-08-100251-3.00005-6%20http://linkinghub.elsevier.com/retrieve/pii/B9780081002513000056 (cit. on p. 10).

Dániel Kondor et al. "Efficient classification of billions of points into complex geographic regions using hierarchical triangular mesh". In: *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*. Ed. by Christian S. Jensen et al. Aalborg, Denmark, 2014. DOI: abs/1410.0709. URL: http://www.vo.elte.hu/htmpaper/ssdbm2014%7B%5C_%7Dsubmission%7B%5C_%7D47%7B%5C_%7Dcrc.pdf%20http://arxiv.org/abs/1410.0709 (cit. on pp. 48, 74, 82).

John Kratz and Carly Strasser. "Data publication consensus and controversies". In: *F1000Research* 3 (Oct. 2014), p. 94. ISSN: 2046-1402. DOI: 10.12688/f1000research.3979.3. URL: https://f1000research.com/articles/3-94/v3 (cit. on pp. 9, 11, 14).

Luboš Krčál and Shen-Shyang Ho. "A SciDB-based Framework for Efficient Satellite Data Storage and Query based on Dynamic Atmospheric Event Trajectory". In: *BigSpatial'15 Proceedings of the 4th International ACM SIGSPATIAL Workshop on Analytics for Big Geospatial Data*. Bellevue, WA, USA: ACM, 2015, pp. 7–14. DOI: 10.1145/2835185.2835190. URL: http://doi.acm.org/10.1145/2835185.2835190%20https://dl.acm.org/citation.cfm?id=2835190 (cit. on p. 62).

Peter Z Kunszt, Alexander S Szalay, István Csabai, et al. "The Indexing of the SDSS Science Archive". In: *Astronomical Data Analysis Software and Systems IX*. 2000, pp. 141–144. URL: http://www.aspbooks.org/a/volumes/table%7B%5C_%7Dof%7B%5C_%7Dcontents/?book%7B%5C_%7Did=328 (cit. on p. 48).

Peter Z Kunszt, Alexander S Szalay, and Aniruddha R Thakar. "The Hierarchical Triangular Mesh". In: *Mining the Sky*. Ed. by Anthony J Banday, Saleem Zaroubi, and Matthias Bartelmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 631–637. ISBN: 978-3-540-44665-1. DOI: 10.1007/10849171_83. URL: https://doi.org/10.1007/10849171%7B%5C_%7D83 (cit. on pp. 43, 48).

Kwo-Sen Kuo and Michael Lee Rilee. "STARE - Toward Unprecedented Geo-Data interoperability". In: *2017 Conference on Big Data from Space*. November. Toulouse, France, 2017. URL: https://www.researchgate.net/publication/320908197%7B%5C_%7DSTARE%7B%

5C_%7D-%7B%5C_%7DTOWARD%7B%5C_%7DUNPRECEDENTED%7B%5C_%7DGEO-DATA%7B%5C_%7DINTEROPERABILITY (cit. on pp. 5, 43, 49, 50, 65, 121).

Kwo-sen Kuo et al. "Towards A Moving Object Database for Geophysical Phenomenon Episodes Using STARE". In: February. 2021. ISBN: 0000000205 (cit. on pp. vi, 89, 155).

M Law and A Collins. *Getting to Know ArcGIS*. Getting to know ArcGIS Desktop. ESRI Press, 2015. ISBN: 9781589483828. URL: https://books.google.com/books?id=qCrOoQEACAAJ (cit. on p. 61).

J. K. Lawder and P. J. H. King. "Querying multi-dimensional data indexed using the Hilbert space-filling curve". In: *ACM SIGMOD Record* 30.1 (Mar. 2001), pp. 19–24. ISSN: 0163-5808. DOI: 10.1145/373626.373678. URL: https://dl.acm.org/doi/10.1145/373626.373678 (cit. on p. 100).

Bryan Lawrence et al. "Citation and Peer Review of Data: Moving Towards Formal Data Publication". In: 6 (2011) (cit. on pp. 9, 14).

Dennis P. Lettenmaier et al. "Inroads of remote sensing into hydrologic science during the WRR era". In: *Water Resources Research* 51.9 (Sept. 2015), pp. 7309–7342. ISSN: 00431397. DOI: 10.1002/2015WR017616. URL: http://doi.wiley.com/10.1002/2015WR017616 (cit. on p. 105).

Jonathan M. Links et al. "COPEWELL: A Conceptual Framework and System Dynamics Model for Predicting Community Functioning and Resilience after Disasters". In: *Disaster Medicine and Public Health Preparedness* 12.1 (2018), pp. 127–137. ISSN: 1938744X. DOI: 10.1017/dmp.2017.39 (cit. on p. 82).

Michael Karl Löffler and Niklas Griessbaum. "Storage devices for heat exchangers with phase change". In: *International Journal of Refrigeration* (2014). ISSN: 01407007. DOI: 10.1016/j.ijrefrig.2014.04.016 (cit. on p. vi).

JA Lugo and KC Clarke. "Implementation of triangulated quadtree sequencing for a global relief data structure". In: *Autocarto-Conference-* August (1995), pp. 147–156. URL: http://scholar.google.com/scholar?hl=en%7B%5C&%7DbtnG=Search%7B%5C&%7Dq=intitle:Implementation+of+Triangulated+Quadtree+Sequencing+for+a+Global+Relief+Data+Structure%7B%5C#%7D0 (cit. on p. 47).

Gerhard Meister, Yuqin Zong, and Charles R. McClain. "Derivation of the MODIS Aqua Point-Spread Function ocean color bands". In: ed. by James J. Butler and Jack Xiong. Aug. 2008, 70811F. DOI: 10.1117/12.796980. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.796980 (cit. on p. 129).

Stephen Mills, Stephanie Weiss, and Calvin Liang. "VIIRS day/night band (DNB) stray light characterization and correction". In: *Earth Observing Systems XVIII* 8866.September 2013 (2013), 88661P. ISSN: 0277786X. DOI: 10.1117/12.2023107 (cit. on p. 83).

A Mitchell and Calif.) Environmental Systems Research Institute (Redlands. *The ESRI Guide to GIS Analysis: Geographic patterns \& relationships*. The ESRI Guide to GIS Analysis. ESRI Press, 1999. ISBN: 9781879102064. URL: https://books.google.com/books?id=FOj8L-iDMq0C (cit. on p. 61).

Gordon E. Moore. "Progress in digital integrated electronics [Technical literaiture, Copyright 1975 IEEE. Reprinted, with permission. Technical Digest. International Electron Devices Meeting, IEEE, 1975, pp. 11-13.]" In: *IEEE Solid-State Circuits Society Newsletter* 11.3 (2006), pp. 36–37. ISSN: 1098-4232. DOI: 10.1109/N-SSC.2006.4804410. URL: http://ieeexplore.ieee.org/document/4804410/ (cit. on p. 3).

Jakob Nielsen. *Nielsen's Law of Internet Bandwidth*. 1998. URL: https://www.nngroup.com/articles/law-of-bandwidth/ (cit. on p. 3).

Kyle E. Niemeyer, Arfon M. Smith, and Daniel S. Katz. "The challenge and promise of software citation for credit, identification, discovery, and reuse". In: *Journal of Data and Information Quality* 7.4 (Jan. 2016), pp. 1–5. ISSN: 19361955. DOI: 10.1145/2968452. arXiv: 1601.04734. URL: http://dl.acm.org/citation.cfm?doid=3006343.2968452%20http://arxiv.org/abs/1601.04734%20http://dx.doi.org/10.1145/2968452 (cit. on p. 11).

Mash Nishihama et al. *MODIS Level 1A Earth Location: Algorithm Theoretical Basis Document*. Tech. rep. 1997 (cit. on pp. 124–126, 130).

Anne W. Nolin. "Recent advances in remote sensing of seasonal snow". In: *Journal of Glaciology* 56.200 (Sept. 2010), pp. 1141–1150. ISSN: 0022-1430. DOI: 10.3189/002214311796406077. URL: https://www.cambridge.org/core/product/identifier/S002214300021335X/type/journal%7B%5C_%7Darticle (cit. on pp. 105, 106).

Oxford English Dictionary. *"database, n"*. 2022. URL: https://www.oed.com/view/Entry/47411 (cit. on p. 4).

Thomas H. Painter, Jeff Dozier, et al. “Retrieval of subpixel snow-covered area and grain size from imaging spectrometer data”. In: *Remote Sensing of Environment* 85.1 (Apr. 2003), pp. 64–77. ISSN: 00344257. DOI: 10.1016/S0034-4257(02)00187-6. URL: http://linkinghub.elsevier.com/retrieve/pii/S0034425702001876 (cit. on p. 108).

Thomas H. Painter, Karl Rittger, et al. “Retrieval of subpixel snow covered area, grain size, and albedo from MODIS”. In: *Remote Sensing of Environment* 113.4 (Apr. 2009), pp. 868–879. ISSN: 00344257. DOI: 10.1016/j.rse.2009.01.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0034425709000029 (cit. on p. 108).

M. a. Parsons and P. a. Fox. “Is Data Publication the Right Metaphor?” In: *Data Science Journal* 12.February (2013), WDS32–WDS46. ISSN: 1683-1470. DOI: 10.2481/dsj.WDS-042. URL: http://datascience.codata.org/articles/abstract/10.2481/dsj.WDS-042/ (cit. on p. 12).

Irene Pasquetto. “Labor Out of Place: On the Varieties and Valences of (In)visible Labor in Data-Intensive Science”. In: 6 (2020), pp. 111–132. DOI: 10.17351/ests2020.341 (cit. on p. 14).

Gary Planthaber. “MODBASE: A SciDB-Powered System for Large-Scale Distributed Storage and Analysis of MODIS Earth Remote Sensing Data”. PhD thesis. 2012 (cit. on p. 62).

Gary Planthaber, Michael Stonebraker, and James Frew. “EarthDB: Scalable Analysis of MODIS Data using SciDB”. In: *1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*. Redondo Beach, California: ACM, 2012, pp. 11–19. DOI: 10.1145/2447481.2447483. URL: https://dl.acm.org/citation.cfm?id=2447483%20http://delivery.acm.org/10.1145/2450000/2447483/p11-planthaber.pdf?ip=128.111.110.169%7B%5C&%7Did=2447483%7B%5C&%7Dacc=ACTIVE%20SERVICE%7B%5C&%7Dkey=CA367851C7E3CE77.022A0CC51A76093F.4D4702B0C3E38B35.4D4702B0C3E38B35%7B%5C&%7D%7B%5C_%7D%7B%5C_%7Dacm%7B%5C_%7D%7B%5C_%7D=1529564323%7B%5C_%7D1cbeb (cit. on p. 62).

Giri Prakash et al. “Data Always Getting Bigger—A Scalable DOI Architecture for Big and Expanding Scientific Data”. In: *Data* 1.2 (Aug. 2016), p. 11. ISSN: 2306-5729. DOI: 10.3390/data1020011. URL: http://www.mdpi.com/2306-5729/1/2/11 (cit. on pp. 14, 26).

Stefan Pröll and Andreas Rauber. “Scalable data citation in dynamic, large databases: Model and reference implementation”. In: *2013 IEEE International Conference on Big Data*. IEEE, Oct. 2013, pp. 307–312. ISBN: 978-1-4799-1293-3. DOI: 10.1109/BigData.2013.6691588. URL: http://ieeexplore.ieee.org/document/6691588/ (cit. on pp. 14, 26).

Andreas Rauber, Ari Asmi, Dieter Van Uytvanck, et al. “Data Citation of Evolving Data. Recommendations of the Working Group on Data Citation (WGDC)”. In: (2015) (cit. on pp. 10, 14).

Andreas Rauber, Ari Asmi, Dieter Van Uytvanck, et al. “Identification of Reproducible Subsets for Data Citation, Sharing and Re-Use”. In: *Bulletin of the IEEE-TCDL* (2015), pp. 6–15. URL: https://rd-alliance.org/system/files/documents/RDA-Guidelines%7B%5C_%7DTCDL%7B%5C_%7Ddraft.pdf%20http://www.ibm.com/developerworks/linux/library/l- (cit. on pp. 10, 13–16, 30).

Andreas Rauber, Bernhard Gößwein, et al. “Precisely and Persistently Identifying and Citing Arbitrary Subsets of Dynamic Data”. In: *Harvard Data Science Review* 3.4 (Oct. 2021), pp. 1–29. DOI: 10.1162/99608f92.be565013. URL: https://hdsr.mitpress.mit.edu/pub/si7wzxxa (cit. on pp. 10, 27).

P Rigaux, M Scholl, and A Voisard. *Spatial Databases: With Application to GIS*. Series in Data Management Systems. Elsevier Science, 2002. ISBN: 9781558605886. URL: https://books.google.com/books?id=o8LfhpFOnPwC (cit. on p. 61).

George A. Riggs et al. *VIIRS/NPP Snow Cover Daily L3 Global 375m SIN Grid, Version 1*. 2019. DOI: 10.5067/VIIRS/VNP10A1.001. URL: https://nsidc.org/data/VNP10A1/versions/1 (cit. on p. 106).

Michael Rilee, Niklas Griessbaum, et al. “STARE-based Integrative Analysis of Diverse Data Using Dask Parallel Programming Demo Paper”. In: *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, Nov. 2020, pp. 417–420. ISBN: 9781450380195. DOI: 10.1145/3397536.3422346. URL: https://dl.acm.org/doi/10.1145/3397536.3422346 (cit. on p. vi).

Michael Rilee, Kwo-Sen Kuo, James Gallagher, et al. *STARE for scalable unification of diverse data within Earth, Space, and Planetary Science*. Dec. 2019. DOI: https://doi.org/10.1002/essoar.10501446.1. URL: https://doi.org/10.1002/essoar.10501446.1 (cit. on pp. vi, 43, 50, 121).

Michael Rilee, Kwo-Sen Kuo, Niklas Griessbaum, et al. “SpatioTemporal Adaptive Resolution Encoding STARE-enabled Event-Based Analysis Using STARE-PODS Concepts”. In: (2022). DOI: 10.6084/m9.figshare.19597234.v1. URL: https://esip.figshare.com/articles/presentation/SpatioTemporal%7B%5C_%7DAdaptive%7B%5C_%7DResolution%7B%5C_%7DEncoding%7B%5C_%7DSTARE-enabled%7B%5C_%7DEvent-Based%7B%5C_%7DAnalysis%7B%5C_%7DUsing%7B%5C_%7DSTARE-PODS%7B%5C_%7DConcepts/19597234 (cit. on pp. 75, 89).

Michael L Rilee, Kwo-Sen Kuo, James Frew, et al. “Stare Towards Integrative Analysis with Minimized Data Wrangling Hassle”. In: *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, Sept. 2020, pp. 901–904. ISBN: 978-1-7281-6374-1. DOI: 10.1109/IGARSS39084.2020.9323859. URL: https://ieeexplore.ieee.org/document/9323859/ (cit. on pp. vi, 43, 50, 121).

Michael L Rilee, Kwo-Sen Kuo, Niklas Griessbaum, et al. “A Portable Approach to Integrating Diverse Geo-Science Data Using Stare-Aware Databases and Transitioning to Cloud”. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, July 2021, pp. 2428–2431. ISBN: 978-1-6654-0369-6. DOI: 10.1109/IGARSS47720.2021.9553975. URL: https://ieeexplore.ieee.org/document/9553975/ (cit. on p. vi).

Michael L. Rilee et al. “STARE into the future of GeoData integrative analysis”. In: *Earth Science Informatics* Stanford 2017 (Jan. 2021). ISSN: 1865-0473. DOI: 10.1007/s12145-021-00568-8. URL: http://link.springer.com/10.1007/s12145-021-00568-8 (cit. on pp. vi, 43, 50, 121).

Michael Lee Rilee, Kwo-Sen Kuo, et al. “Addressing the big-earth-data variety challenge with the hierarchical triangular mesh”. In: *2016 IEEE International Conference on Big Data, Big Data 2016*. 2016. ISBN: 9781467390040. DOI: https://doi.org/10.1109/BigData.2016.7840700 (cit. on pp. 2, 48, 50, 121).

Michael Lee Rilee, Kwo-sen Kuo, et al. *SpatioTemporal Adaptive-Resolution Encoding to Unify Diverse Earth Science Data for Integrative Analysis*. 2018 (cit. on p. 43).

Karl Rittger, Thomas H. Painter, and Jeff Dozier. “Assessment of methods for mapping snow cover from MODIS”. In: *Advances in Water Resources* 51 (Jan. 2013), pp. 367–380. ISSN: 03091708. DOI: 10.1016/j.advwatres.2012.03.002. URL: http://dx.doi.org/10.1016/j.advwatres.2012.03.002%20https://linkinghub.elsevier.com/retrieve/pii/S0309170812000516%20http://linkinghub.elsevier.com/retrieve/pii/S0309170812000516 (cit. on p. 107).

Dar A. Roberts et al. “Mapping Chaparral in the Santa Monica Mountains Using Multiple Endmember Spectral Mixture Models”. In: *Remote Sensing of Environment* 65.3 (Sept. 1998), pp. 267–279. ISSN: 00344257. DOI: 10.1016/S0034-4257(98)00037-6. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve%7B%5C&%7Ddb=PubMed%7B%5C&%7Ddopt=Citation%7B%5C&%7Dlist%7B%5C_%7Duids=11766439%20http://linkinghub.elsevier.com/retrieve/pii/S0034425798000376%20https://linkinghub.elsevier.com/retrieve/pii/S0034425798000376 (cit. on p. 108).

Miguel O. Román et al. “NASA’s Black Marble nighttime lights product suite”. In: *Remote Sensing of Environment* 210.April (June 2018), pp. 113–143. ISSN: 00344257. DOI: 10.1016/j.rse.2018.03.017. URL: https://linkinghub.elsevier.com/retrieve/pii/S003442571830110X (cit. on p. 87).

Keith Ryden and Implementation Specification. “Open Geospatial Consortium Inc . OpenGIS ® Implementation Specification for Geographic information - Simple feature access - Part 1 : Common”. In: (2005) (cit. on pp. 74, 94, 95).

Vincent V. Salomonson and Igor Appel. “Development of the aqua MODIS NDSI fractional snow cover algorithm and validation results”. In: *IEEE Transactions on Geoscience and Remote Sensing* 44.7 (2006), pp. 1747–1756. ISSN: 01962892. DOI: 10.1109/TGRS.2006.876029 (cit. on p. 107).

Vincent V. Salomonson and Igor Appel. “Estimating fractional snow cover from MODIS using the normalized difference snow index”. In: *Remote Sensing of Environment* 89.3 (2004), pp. 351–360. ISSN: 00344257. DOI: 10.1016/j.rse.2003.10.016 (cit. on p. 107).

Hanan Samet. “An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures”. In: *Theoretical Foundations of Computer Graphics and CAD* (1988), pp. 51–68. DOI: 10.1007/978-3-642-83539-1_2. arXiv: arXiv:1011.1669v3. URL: http://www.springerlink.com/index/10.1007/978-3-642-83539-1%7B%5C_%7D2 (cit. on p. 47).

Schubert, Seyerl, and Sack. “Dynamic Data Citation Service—Subset Tool for Operational Data Management”. In: *Data* 4.3 (2019), p. 115. DOI: 10.3390/data4030115 (cit. on p. 14).

Gianmaria Silvello. “Theory and Practice of Data Citation”. In: *Journal of the Association for Information Science and Technology* (2017). arXiv: 1706.07976.pdf. URL: http://arxiv.org/abs/1706.07976 (cit. on pp. 10, 14).

William Silversmith. *cc3d: Connected components on multilabel 3D & 2D images.* 2021. DOI: https://zenodo.org/record/5535251 (cit. on p. 90).

Joan Starr et al. “Achieving human and machine accessibility of cited data in scholarly publications”. In: *PeerJ Computer Science* 1 (May 2015), e1. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.1. URL: https://peerj.com/articles/cs-1 (cit. on pp. 10, 11, 14).

N Stephenson. *Snow Crash: A Novel*. Random House Worlds, 1992. ISBN: 9780553898194. URL: https://books.google.com/books?id=RMd3GpIFxcUC (cit. on p. 35).

Timbo Stillinger and Ned Bair. *Viewable Snow Covered Area Validation Masks over Rugged and Forested Terrain*. Sept. 2020. DOI: 10.5281/ZENODO.4031446. URL: https://zenodo.org/record/4031446 (cit. on pp. 115, 139).

Timbo Stillinger, Karl Rittger, et al. "Landsat , MODIS , and VIIRS snow cover mapping algorithm performance as validated by airborne lidar datasets". In: August (2022), pp. 1–37 (cit. on pp. 106, 107, 143).

NASA VIIRS Land Science Investigator-Led Processing System. *VIIRS/NPP Gap-Filled Lunar BRDF-Adjusted Nighttime Lights Daily L3 Global 500m Linear Lat Lon Grid*. 2019. DOI: 10.5067/VIIRS/VNP46A2.001. URL: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/VNP46A2/ (cit. on p. 83).

Alexander Szalay and Jim Gray. "2020 computing: science in an exponential world." In: *Nature* 440.7083 (Mar. 2006), pp. 413–4. ISSN: 1476-4687. DOI: 10.1038/440413a. URL: https://www.nature.com/articles/440413a%20http://www.nature.com/articles/440413a%20http://www.ncbi.nlm.nih.gov/pubmed/16554783 (cit. on p. 3).

Alexander S Szalay et al. *Indexing the Sphere with the Hierarchical Triangular Mesh*. Tech. rep. Redmond: Microsoft Research, 2005. URL: https://arxiv.org/pdf/cs/0701164.pdf%20https://www.researchgate.net/publication/1960619%7B%5C_%7DIndexing%7B%5C_%7Dthe%7B%5C_%7DSphere%7B%5C_%7Dwith%7B%5C_%7Dthe%7B%5C_%7DHierarchical%7B%5C_%7DTriangular%7B%5C_%7DMesh (cit. on p. 48).

Alexander S. Szalay and José A. Blakeley. "Gray's Laws: Database-centric Computing in Science AlExANDER". In: *The Fourth Paradigm - Data-Intensive Scientific Discovery*. Ed. by Stewart Tansley Tony Hey, Kristin Michele Tolle. Microsoft Research, 2009. ISBN: 978-0-9825442-0-4 (cit. on pp. 2, 3, 5).

Alexander S. Szalay, Jim Gray, et al. "The SDSS SkyServer: public access to the Sloan Digital Sky Server data". In: *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin, June 3-6, 2002* 1.1 (2002), pp. 570–581. ISSN: 07308078. DOI: 10.1145/564691.564758. arXiv: 0202013 [cs]. URL: http://doi.acm.org/10.1145/564691.564758 (cit. on pp. 48, 74).

Zhenyu Tan, Peng Yue, and Jianya Gong. "An Array Database Approach for Earth Observation Data Management and Processing". In: *ISPRS International Journal of Geo-Information* 6.7 (July 2017), p. 220. ISSN: 2220-9964. DOI: 10.3390/ijgi6070220. URL: http://www.mdpi.com/2220-9964/6/7/220 (cit. on p. 62).

MODIS Land Science Team. *MODIS/Terra Atmospherically Corrected Surface Reflectance 5-Min L2 Swath 250m, 500m, 1km.* 2017. DOI: 10.5067/MODIS/MOD09.006. URL: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/MOD09 (cit. on pp. 60, 110, 123).

MODIS Science Data Support Team. *MODIS/Terra Geolocation Fields 5-Min L1A Swath 1km.* 2017. DOI: 10.5067/MODIS/MOD03.061. URL: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/MOD03 (cit. on p. 123).

NASA VIIRS Land Science Team. *VIIRS/NPP Atmospherically Corrected Surface Reflectance 6-Min L2 Swath IP 375m, 750m.* 2020. DOI: 10.5067/VIIRS/VNP09.001. URL: https://ladsweb.modaps.eosdis.nasa.gov/missions-and-measurements/products/VNP09/ (cit. on p. 123).

Carol Tenopir et al. "Data Sharing by Scientists: Practices and Perceptions". In: *PLoS ONE* 6.6 (June 2011). Ed. by Cameron Neylon, e21101. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0021101. arXiv: 0803973233. URL: http://dx.plos.org/10.1371/journal.pone.0021101 (cit. on pp. 9, 10).

Ani Thakar et al. "Migrating a multiterabyte archive from object to relational databases". In: *Computing in Science and Engineering* 5.5 (2003), pp. 16–29. ISSN: 15219615. DOI: 10.1109/MCISE.2003.1225857 (cit. on pp. 48, 74).

Aniruddha R Thakar et al. "The Sloan Digital Sky Survey Science Archive: Migrating a Multi-Terabyte Astronomical Archive from Object to Relational DBMS". In: *Cs/0403020V1* (2004). URL: http://arxiv.org/abs/cs/0403020%7B%5C%%7D5Cnhttp://www.arxiv.org/pdf/cs/0403020v1.pdf (cit. on p. 74).

Gary N Toller et al. "MODIS Level 1B Product User's Guide". In: (2009) (cit. on pp. 124, 125).

Abhishek Verma et al. "Large-scale cluster management at Google with Borg". In: *Proceedings of the Tenth European Conference on Computer Systems - EuroSys '15*. New York, New York, USA: ACM Press, 2015, pp. 1–17. ISBN: 9781450332385. DOI: 10.1145/2741948.2741964. URL: http://dl.acm.org/citation.cfm?doid=2741948.2741964 (cit. on p. 98).

Eric Vermote and Robert Wolfe. *MODIS/Terra Surface Reflectance Daily L2G Global 1km and 500m SIN Grid V061.* 2021. DOI: 10.5067/MODIS/MOD09GA.061. URL: https://lpdaac.usgs.gov/products/mod09gav061/ (cit. on pp. 57, 110, 123).

Z. Wang et al. “MONITORING DISASTER-RELATED POWER OUTAGES USING NASA BLACK MARBLE NIGHTTIME LIGHT PRODUCT”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-3.3 (Apr. 2018), pp. 1853–1856. ISSN: 2194-9034. DOI: 10.5194/isprs-archives-XLII-3-1853-2018. URL: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-3/1853/2018/ (cit. on p. 83).

Mark D. Wilkinson et al. “The FAIR Guiding Principles for scientific data management and stewardship”. In: *Scientific Data* 3.1 (Dec. 2016), p. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4792175%7B%5C&%7Dtool=pmcentrez%7B%5C&%7Drendertype=abstract%20http://www.nature.com/articles/sdata201618 (cit. on pp. 4, 9).

S Wise. *GIS Basics*. Taylor \& Francis, 2003. ISBN: 9780203164549. URL: https://books.google.com/books?id=uxRnQEq5qm8C (cit. on p. 61).

R.E. Wolfe, D.P. Roy, and E. Vermote. “MODIS land data storage, gridding, and compositing methodology: Level 2 grid”. In: *IEEE Transactions on Geoscience and Remote Sensing* 36.4 (July 1998), pp. 1324–1338. ISSN: 01962892. DOI: 10.1109/36.701082. URL: http://ieeexplore.ieee.org/document/701082/ (cit. on p. 110).

Robert E. Wolfe et al. “Achieving sub-pixel geolocation accuracy in support of MODIS land science”. In: *Remote Sensing of Environment* 83.1-2 (Nov. 2002), pp. 31–49. ISSN: 00344257. DOI: 10.1016/S0034-4257(02)00085-8. URL: http://linkinghub.elsevier.com/retrieve/pii/S0034425702000858 (cit. on pp. 41, 110, 124, 131, 136).

M F Worboys and M Duckham. *GIS: A Computing Perspective, Second Edition*. Taylor \& Francis, 2004. ISBN: 9780415283755. URL: https://books.google.com/books?id=x4e2IVV0u9gC (cit. on p. 61).

K. Yang and R.E. Wolfe. “MODIS level 2 grid with the ISIN map projection”. In: *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*. Vol. 7. 8. IEEE, Feb. 2001. ISBN: 0-7803-7031-7. DOI: 10.1109/IGARSS.2001.978332. URL: http://ieeexplore.ieee.org/document/978332/ (cit. on pp. 94, 110).