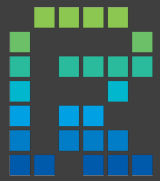
astræa

Bring Satellite & Drone Imagery into Your Data Science Workflows

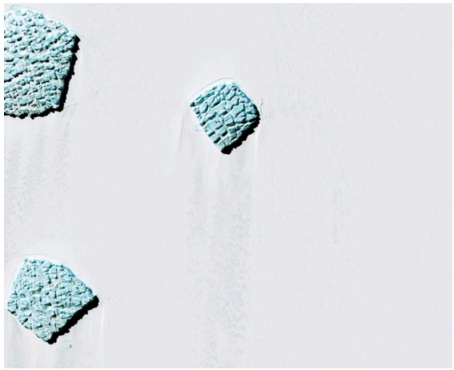
Jason Brown

Senior Data Scientist, Astraea, Inc.

Spark+AI Summit 2020



RasterFrames



Antarctic Icebergs

Antofagasta

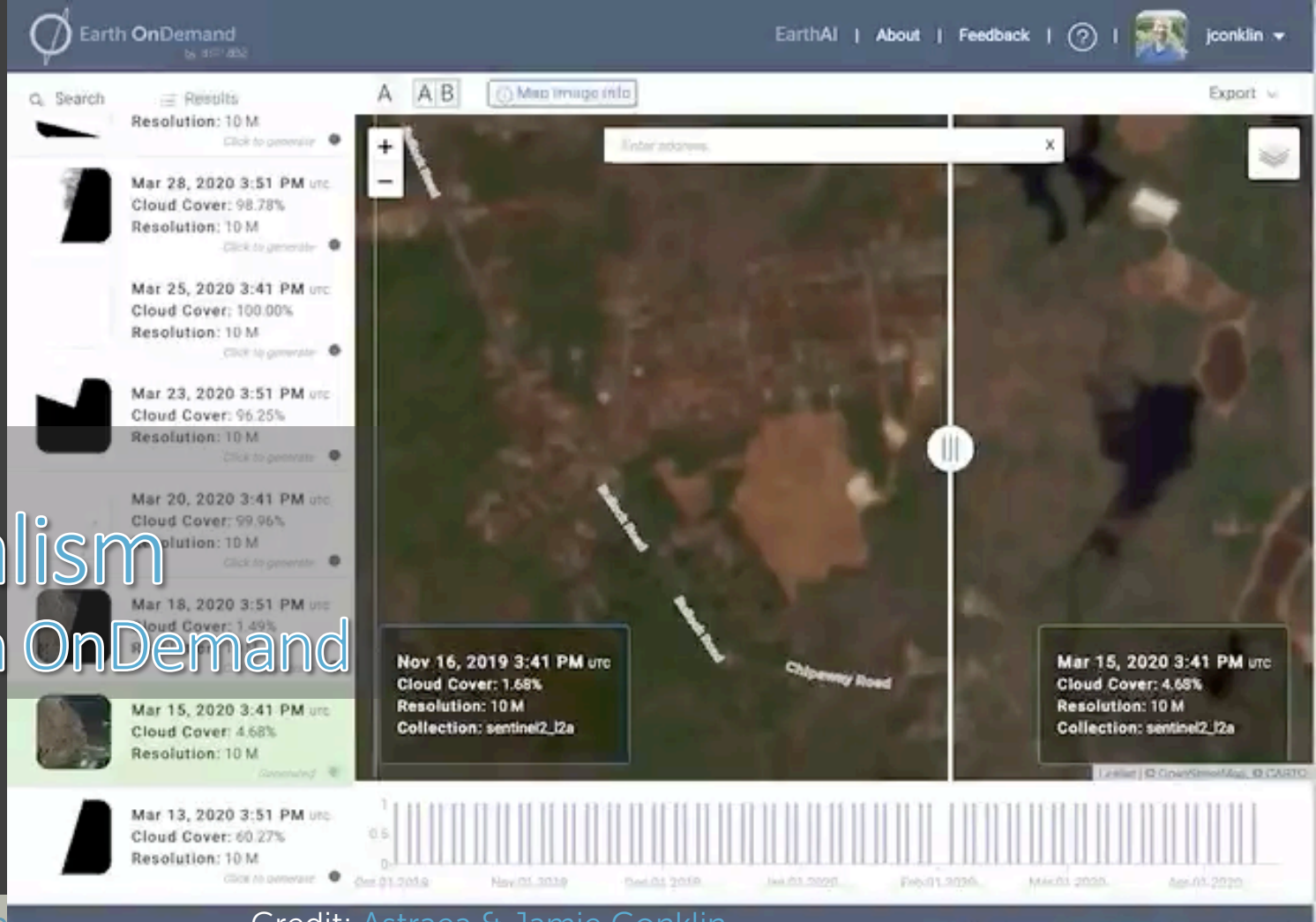
Arc de Triomphe

As art

Daily Overview print shop

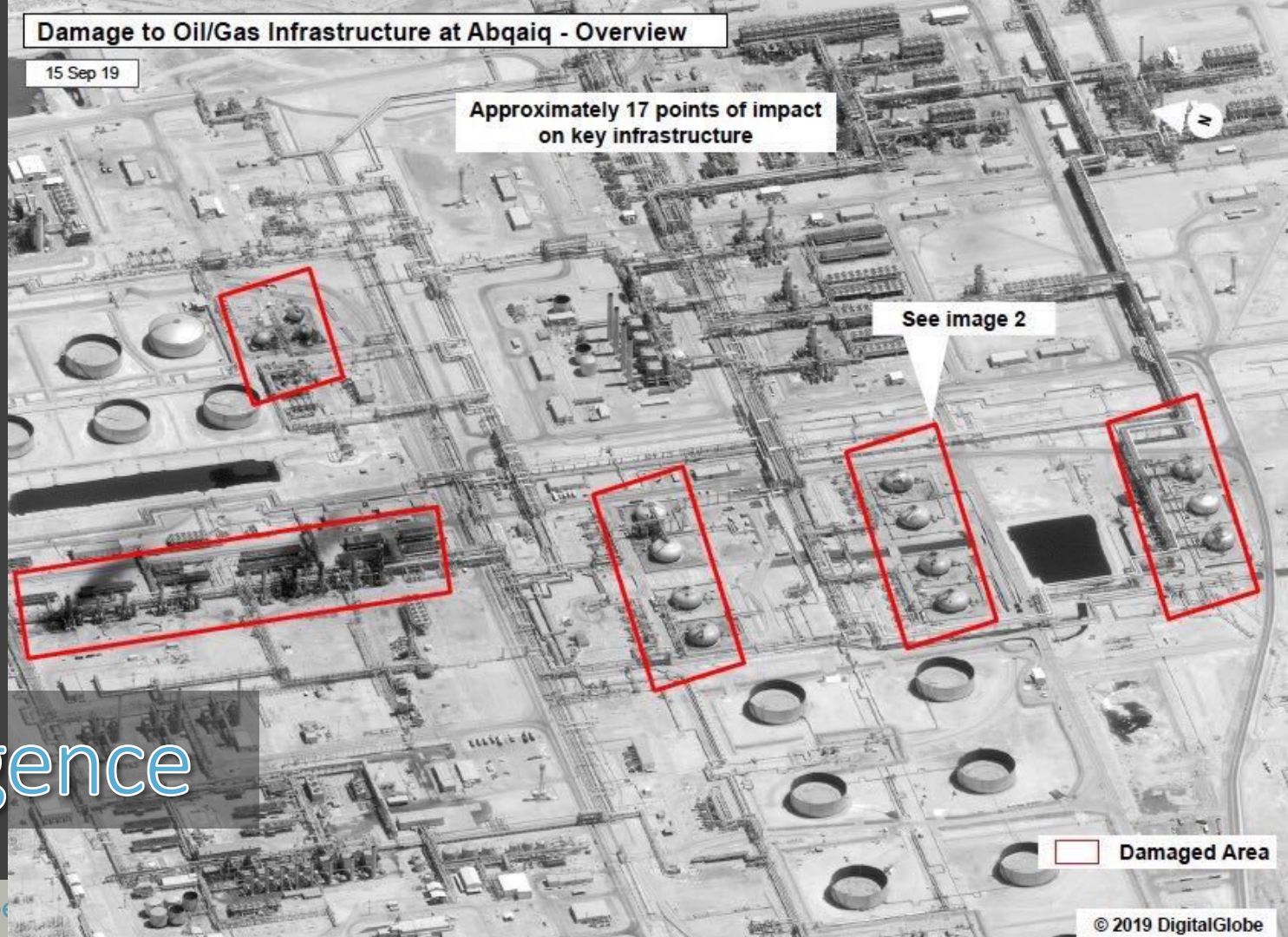
As journalism

Astræa Earth OnDemand



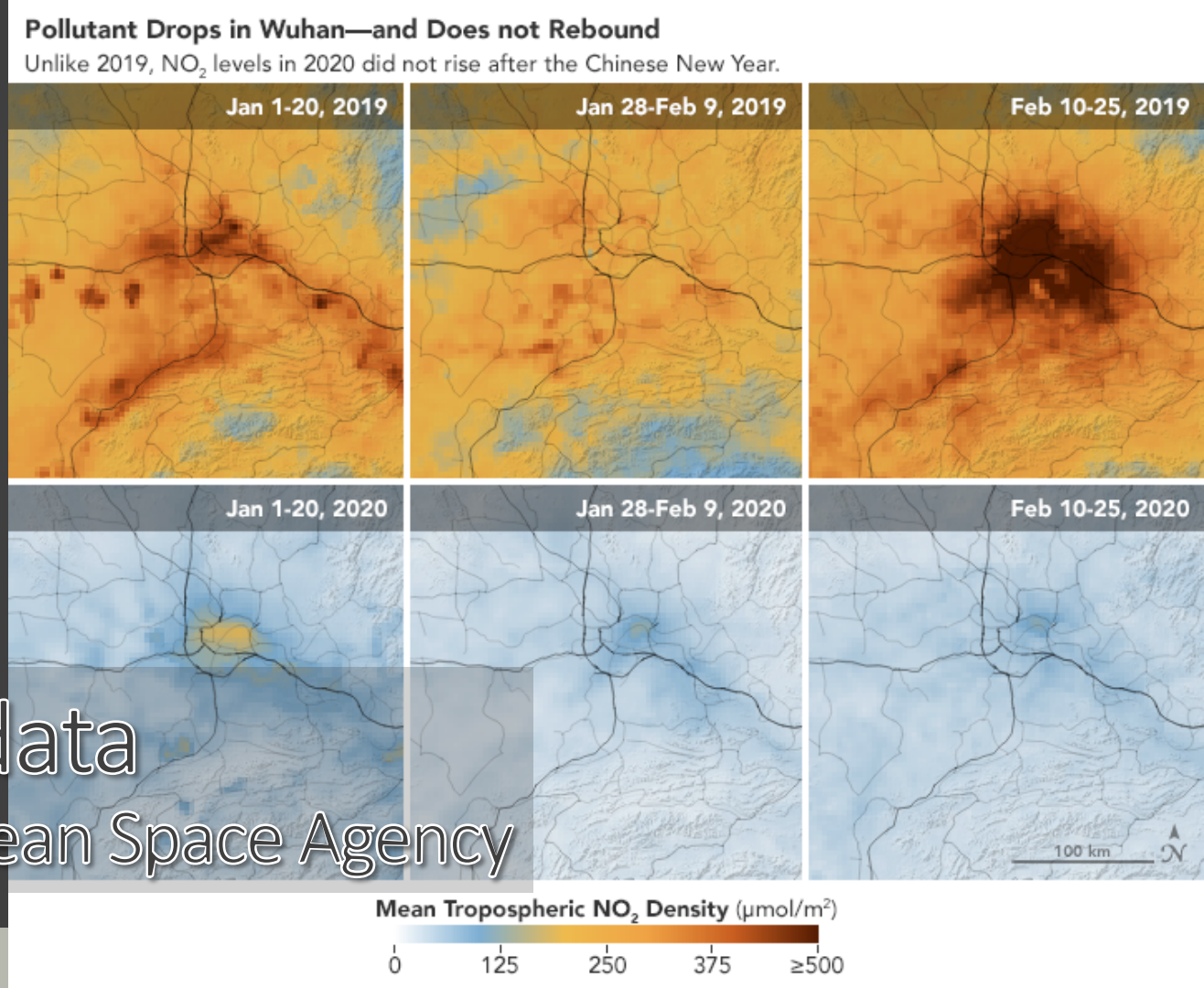
See the Earth as it could b

Credit: Astraea & Jamie Conklin



As intelligence

Credit: Aviation Geek Club



As scientific data

US NASA & European Space Agency

Credit: US NASA

Big data

Variety:

- Naming? Overhead imagery, Earth observation, remote sensing, raster
- > 150 file formats

Velocity:

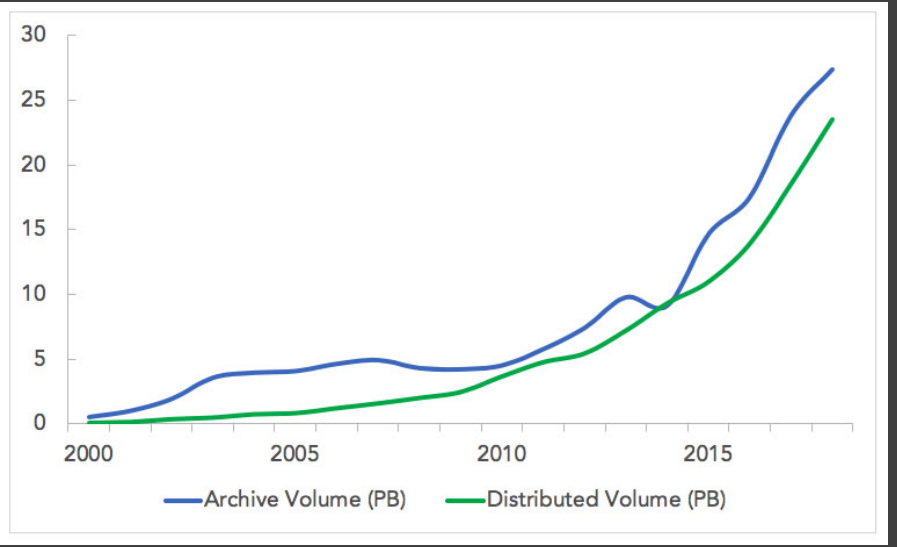
- European Space Agency Sentinel-2 mission generating ~6TB per day
- Government operators in USA, Europe, Japan, South Korea, China, Brazil
- Worldwide private organizations operating hundreds satellites

Volume:

- 2 years ago, US NASA archives at 30 PB and growing
- Plus other public and private operators
- Plus drones

Open data policies (US, EU) and open standards for cloud access

US NASA EOSDIS Data Volumes (PB)



Source: US NASA

What about data science?

Long standing open source software for overhead imagery has produced well developed but narrow tools

- QGIS: visual desktop app for cartography, processing of image to enable human interpretation
- Geospatial Data Abstraction Library (GDAL): mature, high quality open source C and C++ API
- GDAL programs: higher level operations, file I/O orientation

Necessary but not sufficient for data science

RasterFrames origin story:

- What makes overhead imagery “special”?
- How can we work with Volume of imagery in general-purpose tools?

Overhead image data in 5 minutes

Starting from array or tensor representation of image data

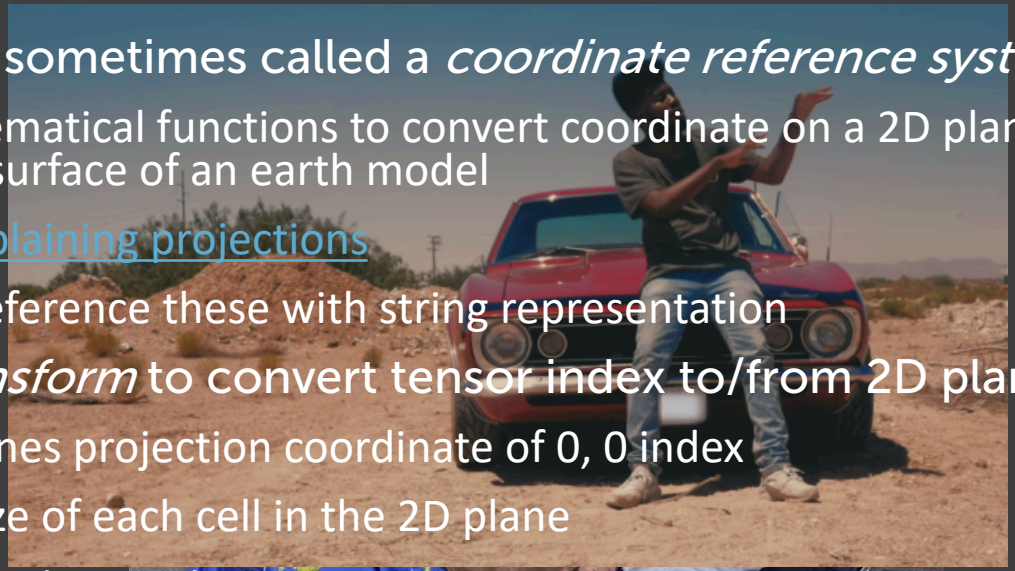
What is different?

- Typical sizes 50-150 megapixel
- Each channel typically 16-bit integers (at rest), often interpreted as float
- Arbitrary number of channels representing more parts of electromagnetic spectrum: e.g. ultraviolet, infrared, etc.

What is additional?

- Location!
- More metadata on each image: sensor, capture time, processing details from publisher
- Encoding NaN in the integer values with a sentinel value

Overhead image data in 5 minutes



Has a Projection, sometimes called a *coordinate reference system (CRS)*

- Invertible mathematical functions to convert coordinate on a 2D plane to/from a location on the surface of an earth model
- Link to video explaining projections
- In practice we reference these with string representation

Has an *affine transform* to convert tensor index to/from 2D plane coordinates

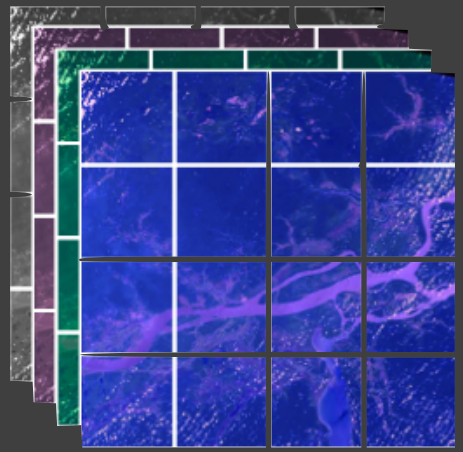
- Translation: defines projection coordinate of 0, 0 index
- Scale: defines size of each cell in the 2D plane
- Shear: rarely seen in practice

Without shear, we can equivalently express the affine transform for a tensor of known size as *extent*, a pair of 2D coordinates

astræa

RasterFrames

Representing large image files in a DataFrame



timestamp	geometry	metadata	B1	B2	B3	B4

RasterFrames

astræa

Making the DataFrame model a practical reality

- Strong Python API - `pip install pyrasterframes`
- Custom datasource - `spark.read.raster`
- `Tile` User Defined Type
  - Single column to carry tensor, projection, and extent
  - In Python, the associated `Tile` class wraps a Numpy `ndarray`
- Column functions to extract or operate on the location and tensor
  - Location examples: reproject, check intersection with other geometry
  - Tensor examples: cell-wise arithmetic, column aggregates
- `spark.ml Transformers` to work with the `Tile` UDT in `Pipelines`
- Visualization and inspection of `Tile` UDT

Case study

Github Gist

https://gist.github.com/vpipkt/09734d56082e7c74b187e9a635b6e764

RasterFrames project

Currently incubating under Eclipse Foundation LocationTech

- Open source under Apache 2.0 license and commercial friendly intellectual property governance

Try it:

- https://rasterframes.io/getting-started.html
- https://pypi.org/project/pyrasterframes/
- Free trial of EarthAI Notebook

Contribute!

- gitter.im for questions, engaging with ideas to contribute https://gitter.im/locationtech/rasterframes
- GitHub for tracking issues & pull requests
- https://github.com/locationtech/rasterframes/blob/develop/CONTRIBUTING.md

astræa

Thank you!

Visit https://rasterframes.io to try it out!

Trends in data publishing

- Another hypothesis: new standards for data publishing are opening up new possibilities for tooling to use satellite and drone data as found data
- Cloud optimized GeoTIFF
  - More efficient parallel processing of large files
- Cloud hosted storage
  - STAC specification: open standard for image metadata and search
  - Cloud providers hosting open datasets

Overhead Imagery Software & Data Science

Hypotheses:

1. Data science innovations often driven from application of “found data”. E.g. 1 million Flickr images lead to facial recognition
2. Specialist tools have hindered the use of this data as “found data”

Observations:

- Many use cases satisfied by visual desktop apps for cartography, processing of image to enable human interpretation
- Typical science use case focuses on a small region, often very sophisticated processing of the data
- Science use cases well met by Geospatial Data Abstraction Library (GDAL) API

- GDAL
  - Advantages: mature, high quality open source C and C++ API
  - Disadvantages: higher level tools are oriented to Unix style read-process-write
- Data science, found data, and overhead imagery
  - Data scientist needs to inspect some images but not a feasible tactic for creating a scalable model / algorithm / analytic
  - Need to do some file reads, but want to use in memory processing
  - In reality, a lot of data science and ML engineering is already done in Apache Spark SQL and ML APIs