Hylatis,
for Hyperspectral Imagery Analysis

Technical

Anne Wilson
Doug Lindholm
Chris Lindholm
Peter Pfister
the LASP Web Team

Subject Matter Experts

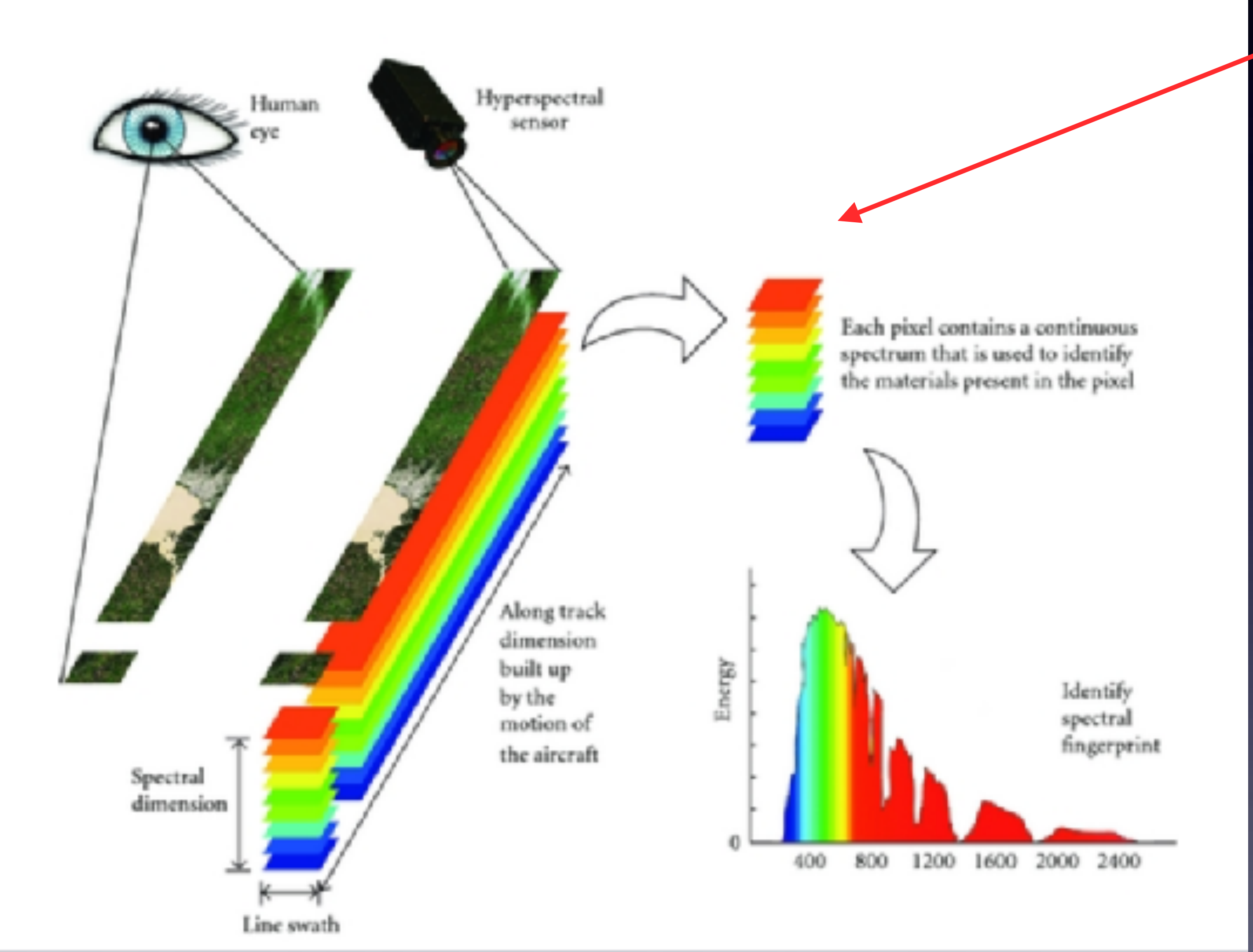
Odele Coddington
Peter Pilewski
Julia Kent

Laboratory for Atmospheric and Space Physics (LASP)
University of Colorado, Boulder

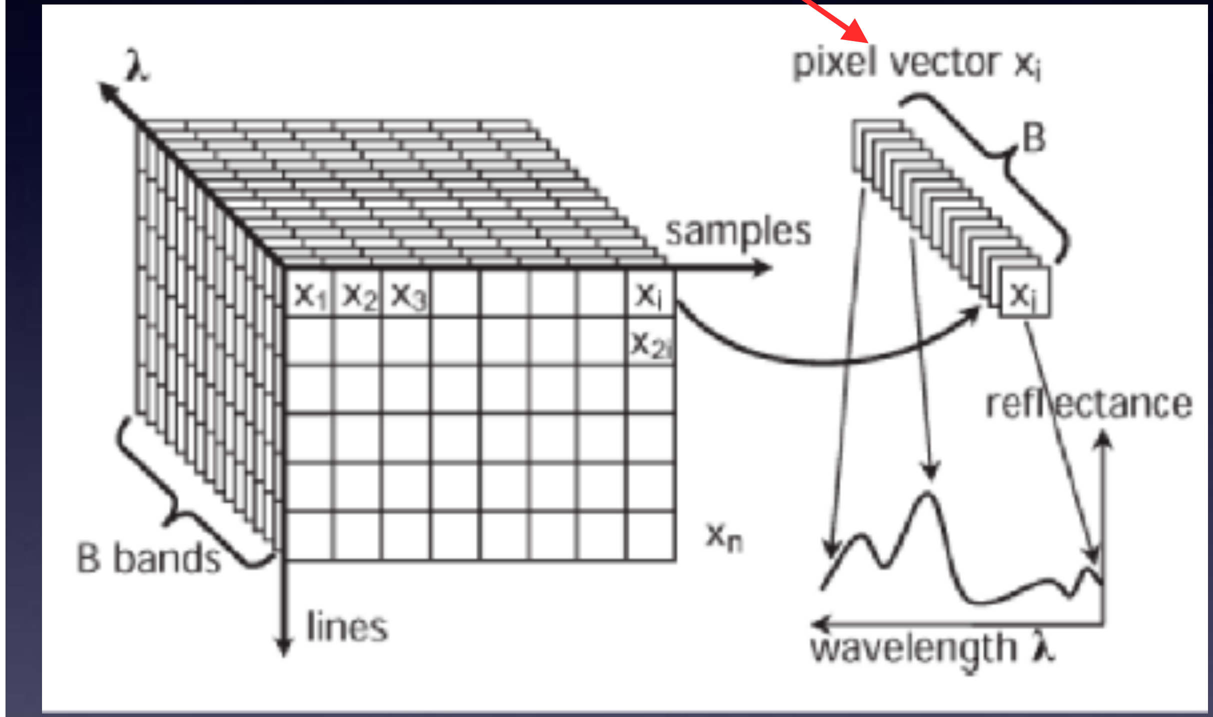
Presentation to ESIP Data Stewardship Cluster, Sept 17, 2018

Hyperspectral cubes

spectral fingerprint


Human eye
Hyperspectral sensor
Each pixel contains a continuous spectrum that is used to identify the materials present in the pixel
Along track dimension built up by the motion of the aircraft
Spectral dimension
Line swath
Energy
0
400
800
1200
1600
2000
2400
Identify spectral fingerprint

Spectral Distortion in Loss Compression of Hyperspectral Data, Aiazzi, et al, Journal of Electrical and Computer Engineering, 2012(5):1817-1819, Aug 2003, DOI: 10.1109/IGARSS.2003.1294260



Bodkin Engineering, http://www.bodkindesign.com/products-page/hyperspectral-imaging/hyperspectral-imaging/

Continuous coverage of an area

Uses, Implications

- Increasing usage and reliance on HI
  - agriculture, environmental monitoring, food safety, forestry, anthropology, medical imaging, reconnaissance, surveillance, …
- Very high data volume is a barrier to use
  - Today, an analysis could take 100TB
- Ever increasing spectral and spatial resolution

Tools are needed.

Terms: ’hyper’ vs ‘multi’ spectral

- Scientists often want to compare, integrate the two
- Same shape, they are treated uniformly from Hylatis perspective, treated uniformly

Hylatis Project Goals

- Toolset for multispectral and hyperspectral datasets in the cloud
- Evolve LaTiS domain agnostic data model and reusable framework to leverage cloud parallelism
- Long term: develop a flexible, domain agnostic platform for science data representation and analysis, via
  - Data model, providing a common representation that any domain can write to it
  - Framework, supporting development of arbitrary operations on arbitrary datasets in a structured, principled way
- A software research project on how to represent and operate on scientific datasets in the cloud

Core project principles

- Modeling data mathematically, for flexibility in representation of disparate datasets
- Commitment to principles of mathematics to operate on datasets
- Use of functional programming style, with benefits especially for science:
  - Based on math, lambda calculus
  - More thoughtful, rigorous, up front development produces code that is more understandable, correct, easily parallelizable
    - Benefits for science
  - Capture of generalizations without domain specificity, fosters code reuse
    - Benefits for systems design, development, and maintenance

Representing data mathematically

- A ‘functional data model’, in the mathematical sense
- Datasets are represented as functions of independent and dependent variables
  - temperature = f(lat,lon)
  - (temperature, pressure) = f(lat,lon,elevation)

Mathematical operations on datasets

- basic math
- unit, time format conversion
- replace (e.g., missing values)
- smooth (reduce noise), stride
- subset, filter, slice
- bin (into partitions)
- statistics: mean, standard deviation
- integration
- …

Build more complex, project specific algorithms from these fundamental operations

Selected Datasets

1.HySICS: LASP hyperspectral instrument to fly on CLARREO

- using calibration data from a balloon flight
- each image is 480 x 640, and cube has 4200 images

2.GOES-R: multispectral, 16 bands

- 1 - 6 measure outgoing radiance at the top of the atmosphere
- 7 - 16 are digital maps of outgoing radiance values at the top of the atmosphere

3.MODIS

- Level 1b, 1km spatial resolution, 5 minute temporal resolution, 36 spectral bands

4.POLDER: 242x548 pixels

- extra dimension: polarization data

User story 1: browsing multiple datasets via RGB images

Load 4 multispectral datasets into cloud

- Demonstrates handling of 4 different datasets via a uniform API that supports operations on multispectral data

Via a web form that invokes that API, user selects: a dataset, a geospatial region, 3 wavelength bands, and gets back an RGB image of that selection

- Demonstrates data sharing by multiple simultaneous users

User story 2: data fusion

- Via a web page, users will
  - select 1 - 4 datasets
  - define a spatial region, select 3 wavelength
  - create a new dataset that is the fusion of the selected datasets, via simple interpolation

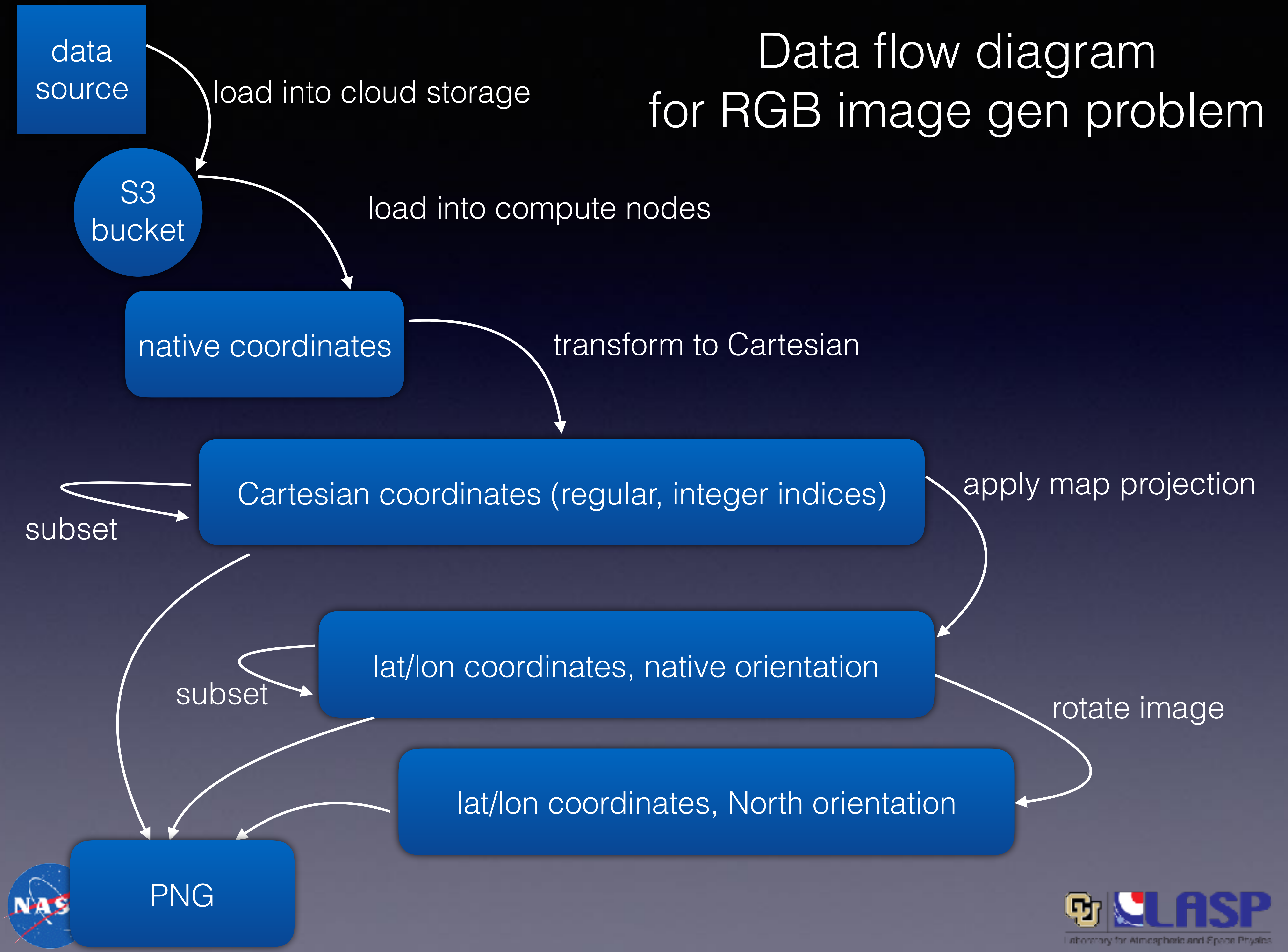
Terms: data ‘fusion’

- To some, any composition of datasets
- To others, implies higher level of thoughtfulness and complexity
- Hylatis is about the framework rather than the algorithms, and supports the implementation of any algorithm
    - Algorithm complexity is a moot issue here
    - We will use the term ‘fusion’

What we learned to love and appreciate

It takes many transformations of a spectral dataset to move from native coordinates to coordinates and orientations that are meaningful to people.

Data flow diagram
for RGB image gen problem



data source

load into cloud storage

S3 bucket

load into compute nodes

native coordinates

transform to Cartesian

Cartesian coordinates (regular, integer indices)

subset

apply map projection

lat/lon coordinates, native orientation

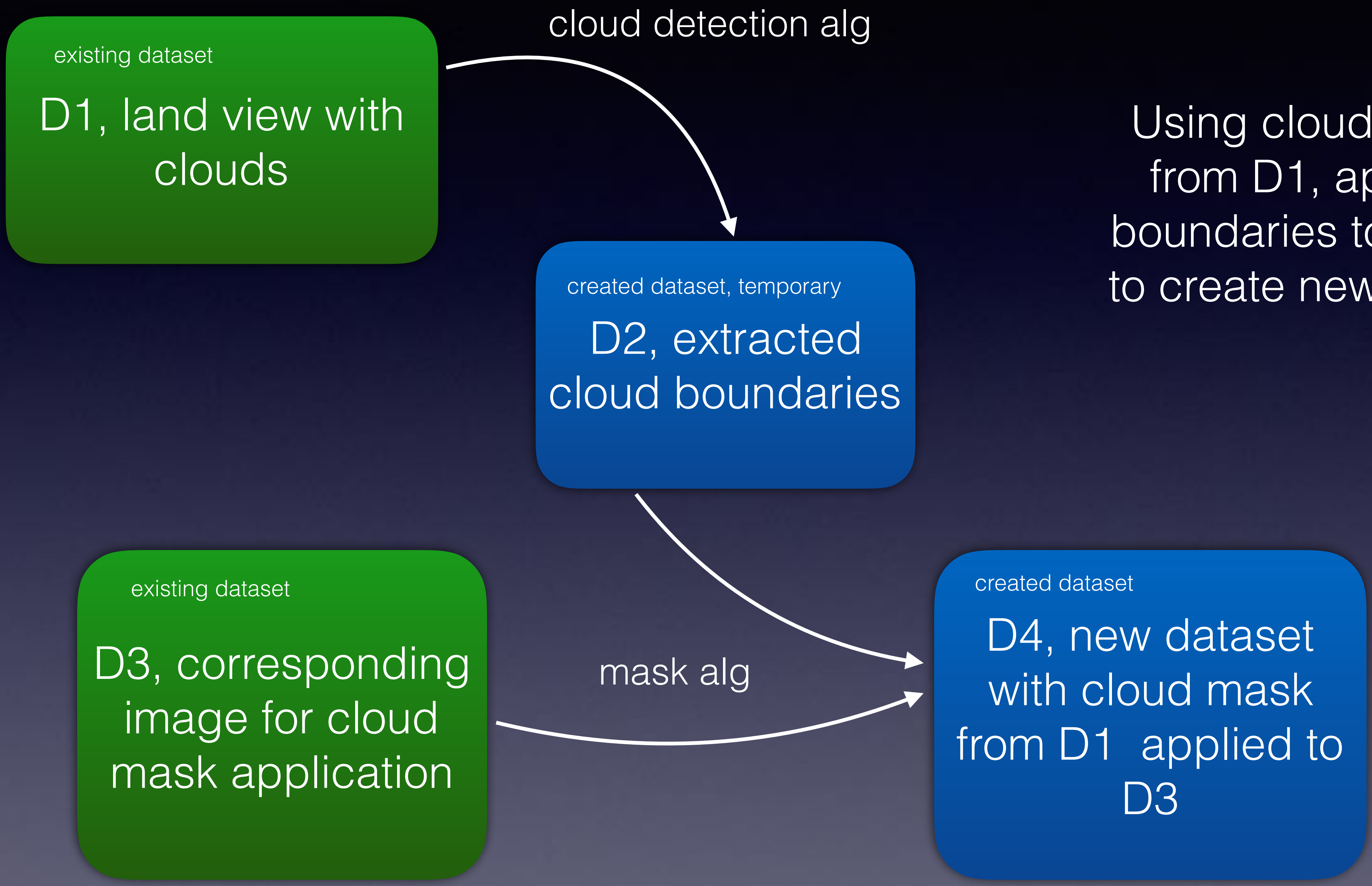
subset

rotate image

lat/lon coordinates, North orientation

PNG

Transformations via function composition



cloud detection alg

existing dataset

D1, land view with clouds

created dataset, temporary

D2, extracted cloud boundaries

existing dataset

D3, corresponding image for cloud mask application

mask alg

created dataset

D4, new dataset with cloud mask from D1 applied to D3

Using clouds detected from D1, apply cloud boundaries to image D3, to create new dataset D4

D4 = mask((cloud_detection(D1)), D3)

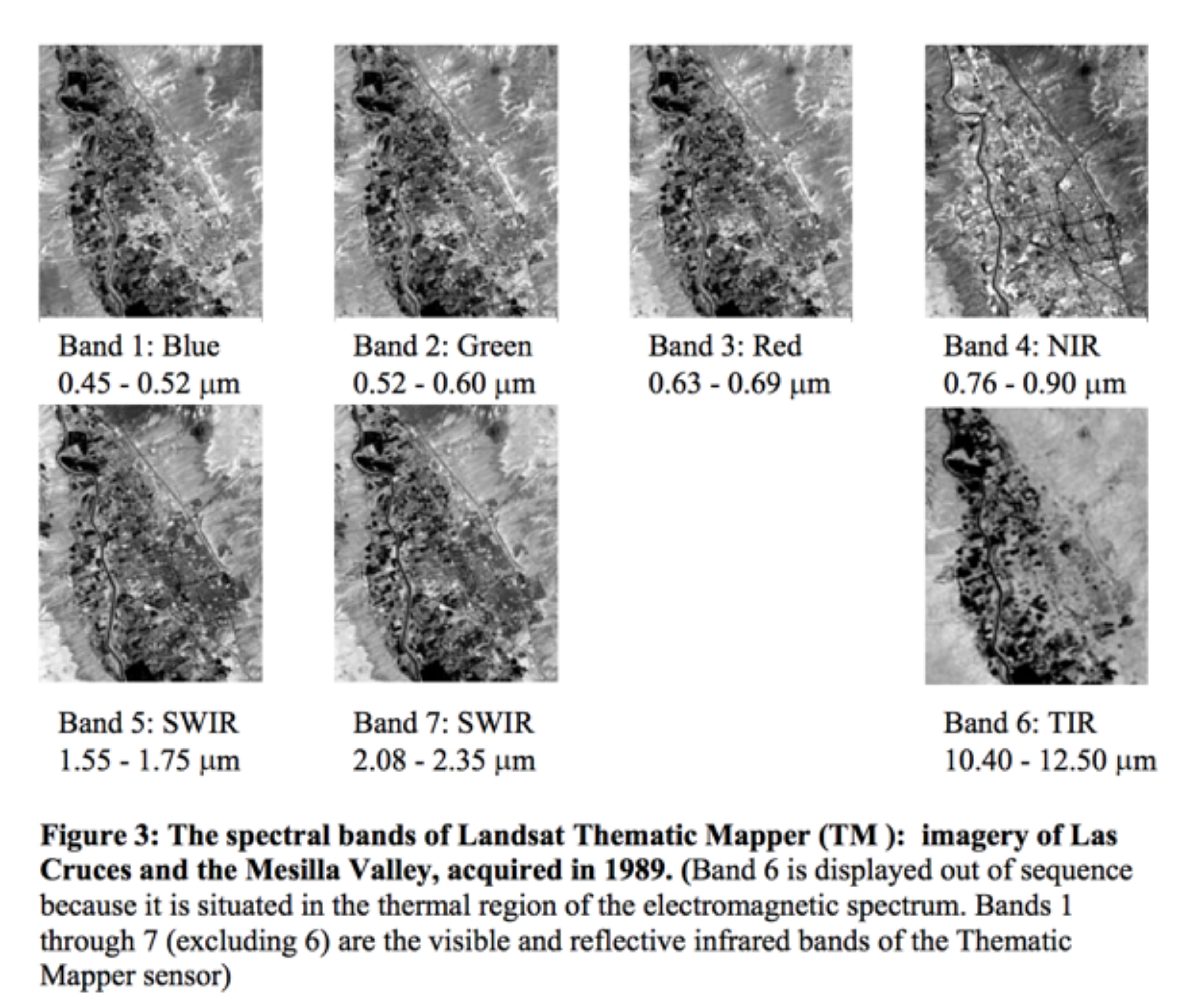
a functional programming technique

Other lessons

- Even more transformations, details than we anticipated
- Focus on level 3 datasets
  - No more level 2
- Jupyter notebook limitation: Spark context bound to single notebook

Thank you!

Example hyperspectral imagery



Band 1: Blue
0.45 - 0.52 µm

Band 2: Green
0.52 - 0.60 µm

Band 3: Red
0.63 - 0.69 µm

Band 4: NIR
0.76 - 0.90 µm

Band 5: SWIR
1.55 - 1.75 µm

Band 7: SWIR
2.08 - 2.35 µm

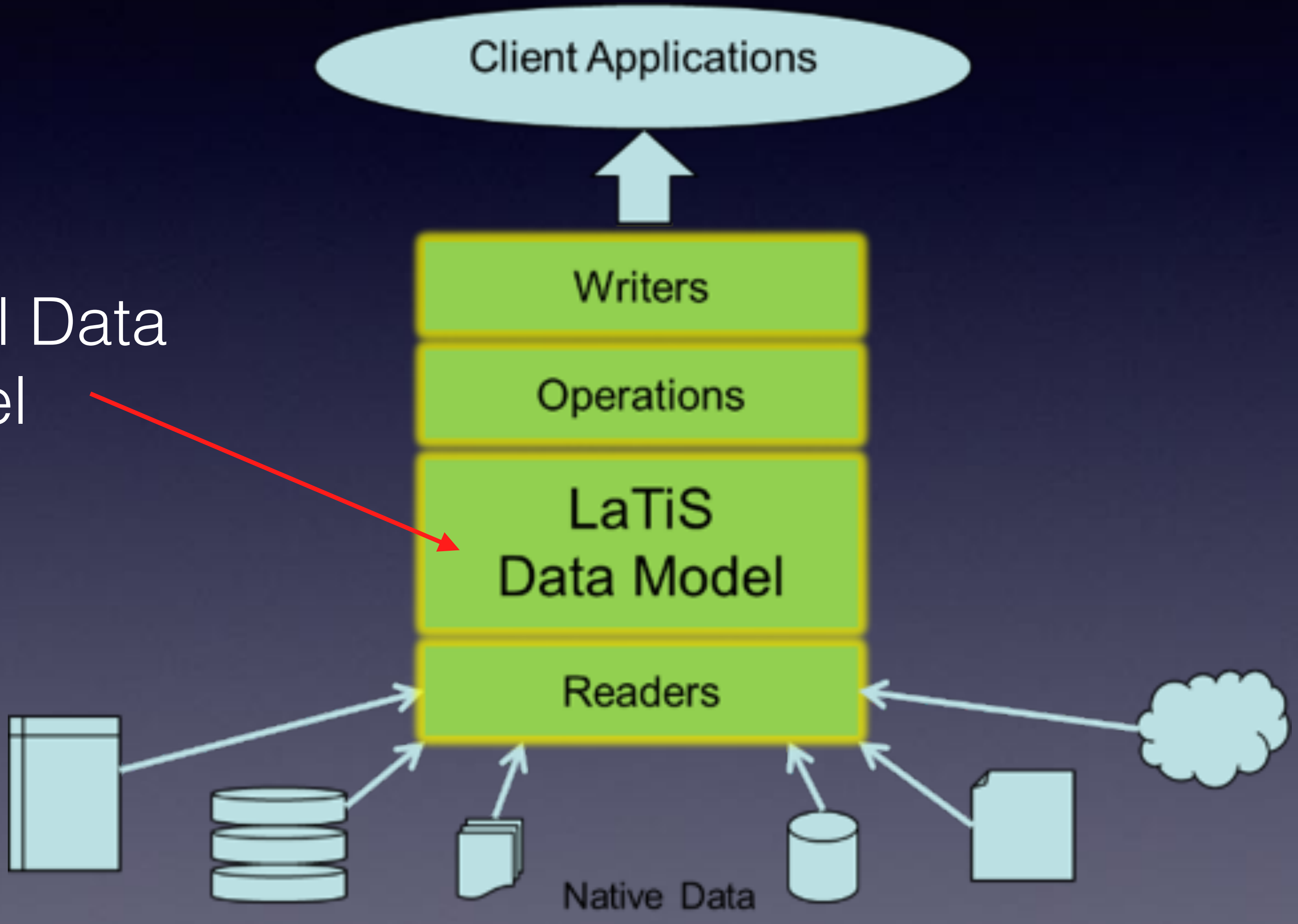
Band 6: TIR
10.40 - 12.50 µm

Figure 3: The spectral bands of Landsat Thematic Mapper (TM): imagery of Las Cruces and the Mesilla Valley, acquired in 1989. (Band 6 is displayed out of sequence because it is situated in the thermal region of the electromagnetic spectrum. Bands 1 through 7 (excluding 6) are the visible and reflective infrared bands of the Thematic Mapper sensor)

http://web.nmsu.edu/~aulery/docs/Lab_9_appendix_e.pdf

LaTiS Architecture

Functional Data Model



HySICS Level 2 data

This is L2 HySICS calibration data, from a short balloon flight, viewed by users as snapshots taken at a single point in time

Users would not subset on time

Native coordinate system is tabular: row, column

Geolocation info is in metadata in separate files, which needed to be applied to make a new geolocated dataset

HySICS geolocation tarpit

To geolocate, first transform from native coordinates (here, table indices) to a spatial grid.

Required simplifying assumptions:

- the line of travel is linear
- the imager lenses are always perpendicular to the line of travel, with no jitter
- the imager is looking straight down (no curvature)
- dX assumed constant, based on detector field of view and altitude, meters
- dY assumed constant, based on lat/lon of detector at start and end of observation, meters
- there is never more than one image associated with any georeferenced point (though jitter might induce that situation)
- and more…

HySICS lessons

Learned

- Need to understand and plan for all dataset specific transforms
  - With HySICS needed 2: new spatial x,y coordinate system and transform, and transform to lat/lon space
  - A distraction - our focus is on the framework, not algorithmic details
- Will use geolocated level 3 products and avoid specific and/or complex algorithm development where possible
- Subsetting is much faster in x,y space due to binary search capability, rather than testing every grid cell, so keep data in x,y space
- To reference wavelengths via geolocation, use a function that is composition of two functions, one to transform from lat/lon to x,y, and another to select the wavelengths in x,y space
- Leverage GeoTools to transform to lon/lat for image generation
- Can leverage existing image generation code PNG Writer for data in x,y space, code reuse

GOES: Yet more georeferencing

With GOES, users would subset on time, unlike HySICS

GOES data, available in AWS, NetCDF

Native coordinate system is Advanced Baseline Imager (ABI) x,y coordinate system

Though level 3, still needs geolocation algorithm applied

Learned

- Some users don’t want to deal issues introduced by applying a map projection?
- May be the norm that a georeferencing transformation must be applied
  - Must budget time
  - Expect development cost to go down over time as more reusable code is developed

There may be geolocated L3 data available, we are investigating that

GOES native ABI coordinate system

GOES data native coordinate system is ABI, a radial coordinate system that is new to us

Learned

- Radial coordinate system makes sense for a geostationary imager
- For each new dataset, consider that handling native coordinate system may take resources
- Have expectation to develop reusable algorithms for all GOES data in ABI coordinates
- Code may be extensible to geostationary imagers in general