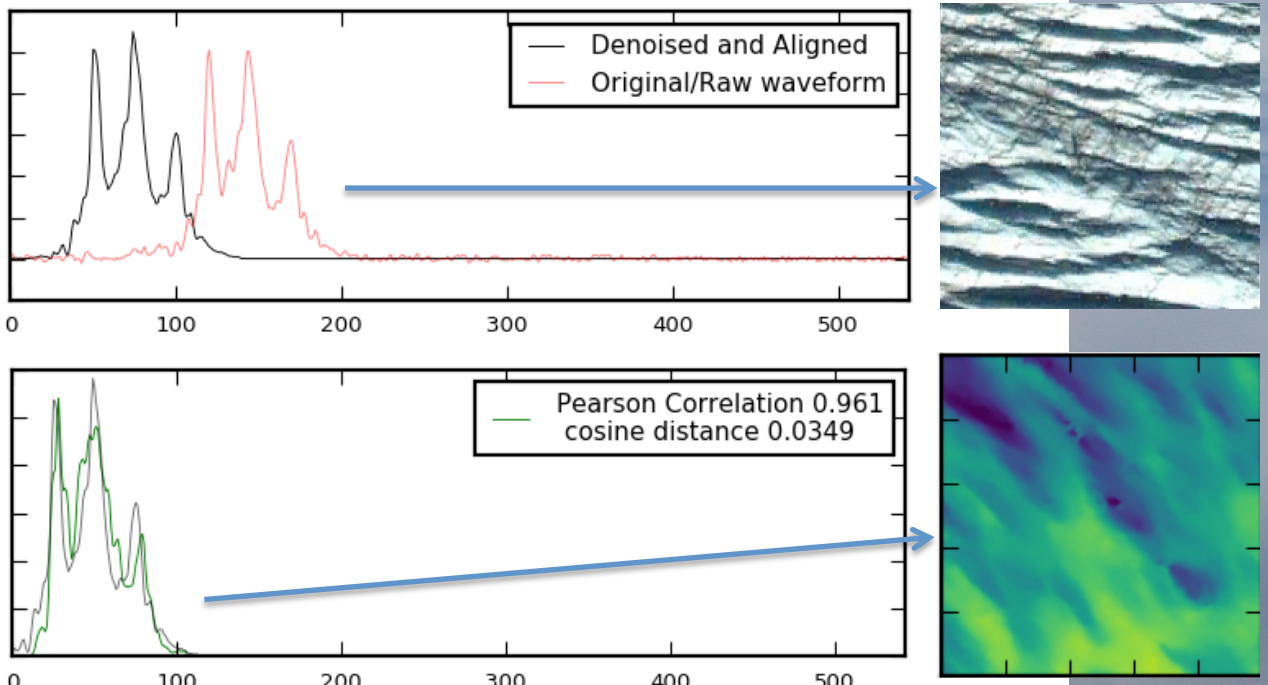
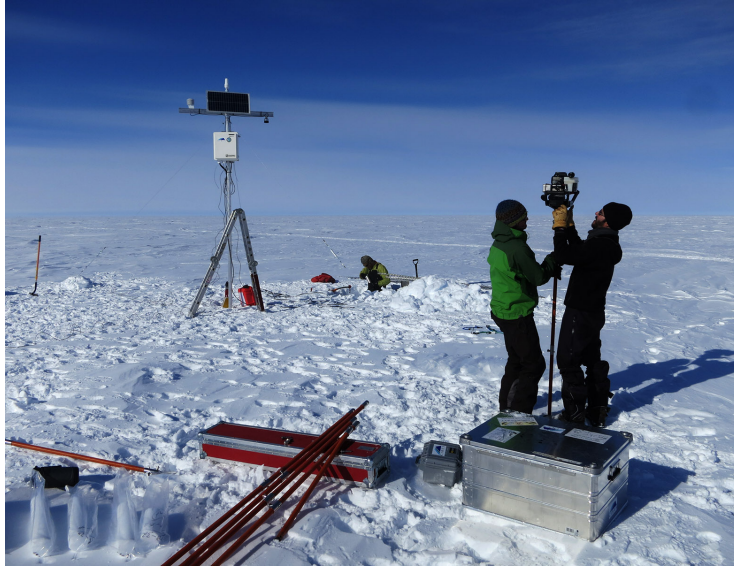
Deep Learning with Geospatial Data

Shane Grigsby

shane.grigsby@colorado.edu



…a quick intro…



- Glaciologist
- Remote Sensing Scientist
  - Anything optical (Active or Passive)
  - Both spaceborne and airborne platforms
- ~70% time algorithm development
- ~15% time manual labor in cold places
- On github at espg

…I mostly watch ice melt from space

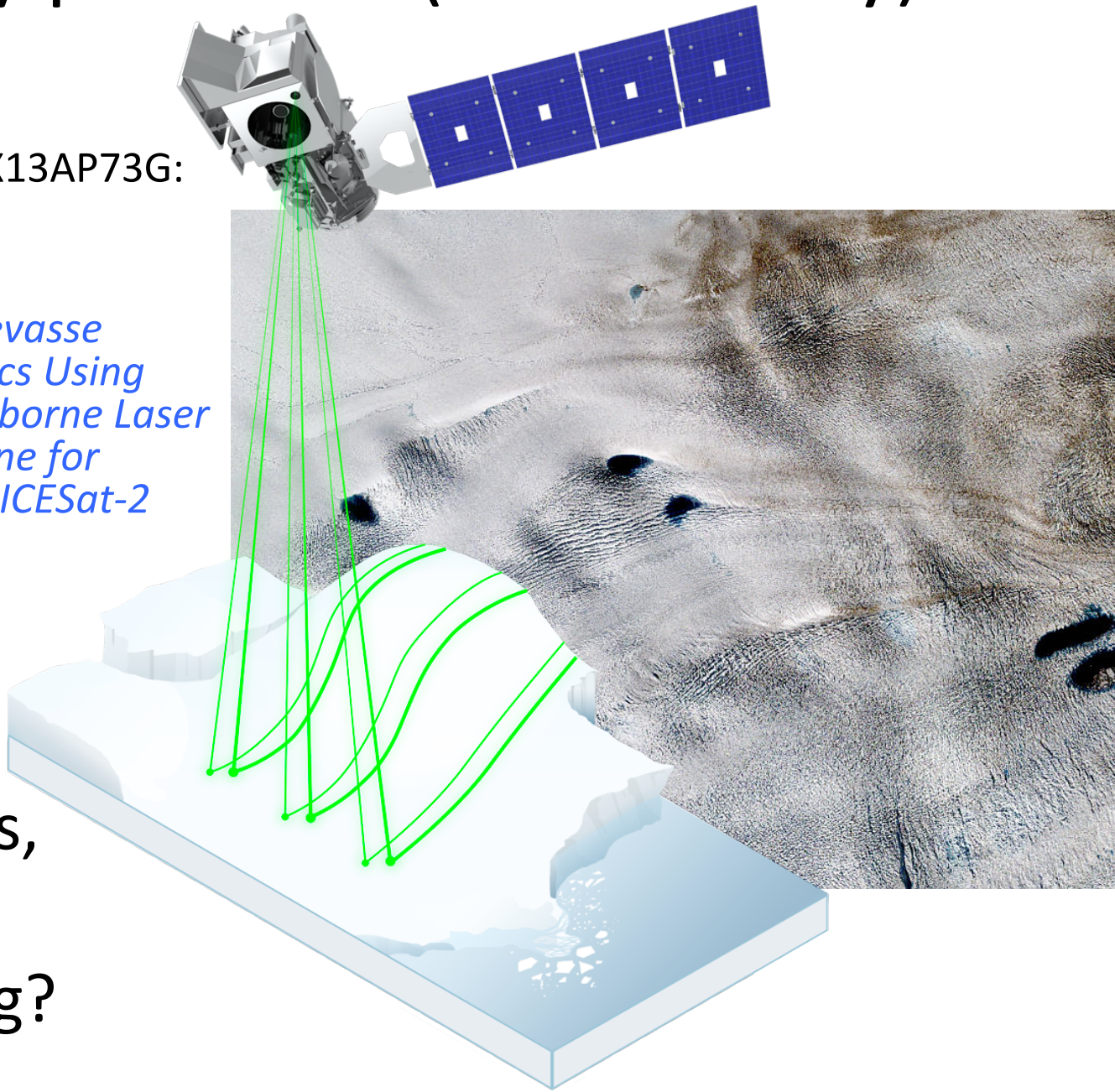
CIRES
Cooperative Institute for Research in Environmental Sciences

ICESat-2
ICE, CLOUD, AND LAND ELEVATION SATELLITE-2

So what's my problem? (Scientifically)

NASA ROSES award NNX13AP73G:

Assessing Greenland Crevasse Extent and Characteristics Using Historical ICESat and Airborne Laser Altimetry Data: A Baseline for Assessing Changes with ICESat-2

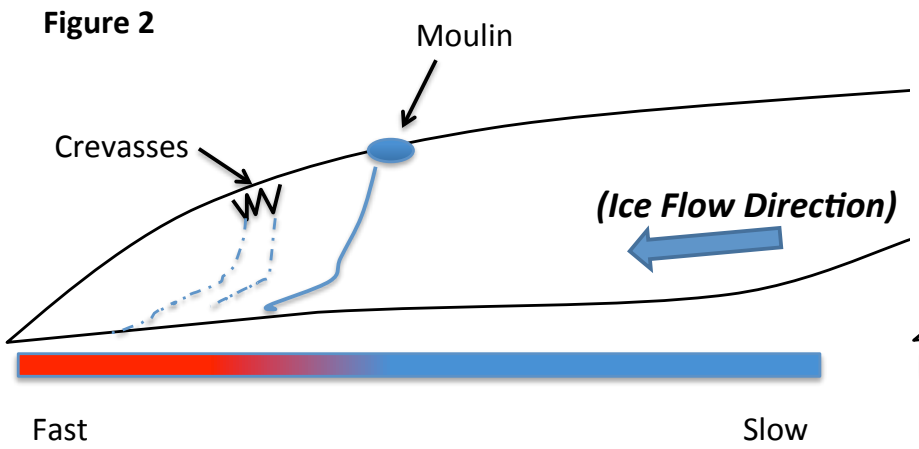


Basically, where are the crevasses, and are they changing/moving?

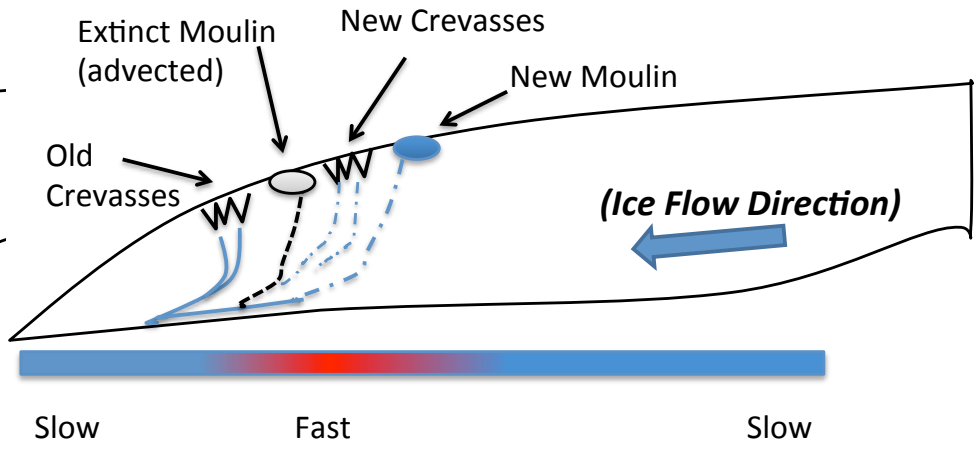
So why do we care about crevasses?

- Crevasses are how we get water ***off*** the ice sheet (and into the ocean to for sea level rise)
- Crevasses may also creep upslope with runoff

Figure 2



Phase 1: Underdeveloped subglacial hydrologic system is overwhelmed by melt water, leading to acceleration. Solid lines denote efficient channelization; dashed lines show Inefficient / vulnerable to being overwhelmed by melt water



Phase 2: Mature subglacial hydrologic system leads to deceleration at the front; however, previous acceleration has resulted in new crevassing, which opens new underdeveloped hydrologic conduits which are vulnerable to acceleration and further crevassing

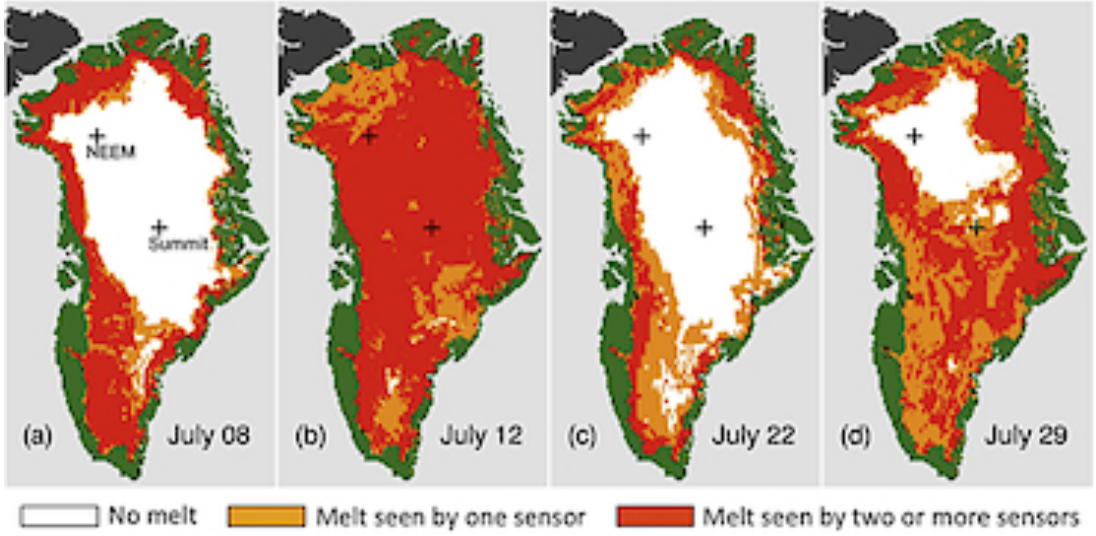
So why do we care about crevasses?

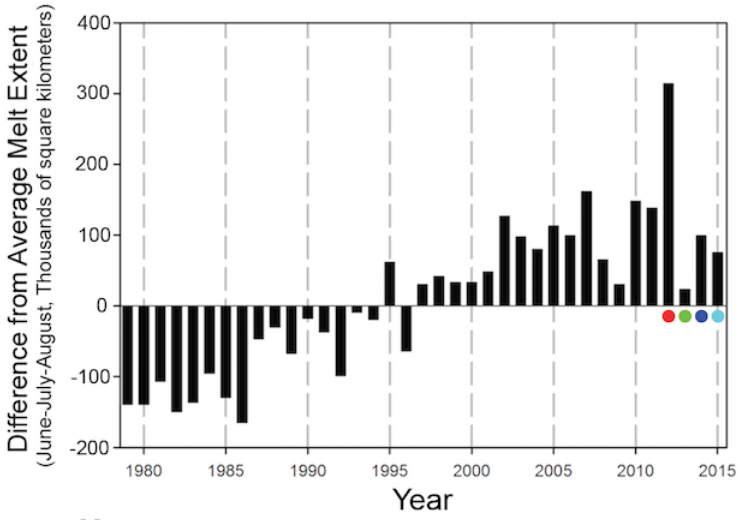
Surface melt is increasing in Greenland:

- Increasing occurrence of extreme melt events (2012)
- Consistent heavy melt years
  - 2007, 2010, 2011

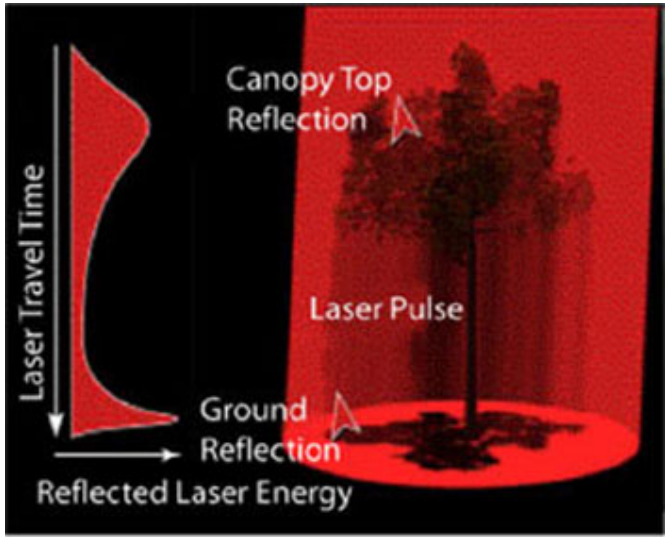
To know if the melt water evacuation system is going upslope, we need to know what it was historically...

Nghiem, et al 2012



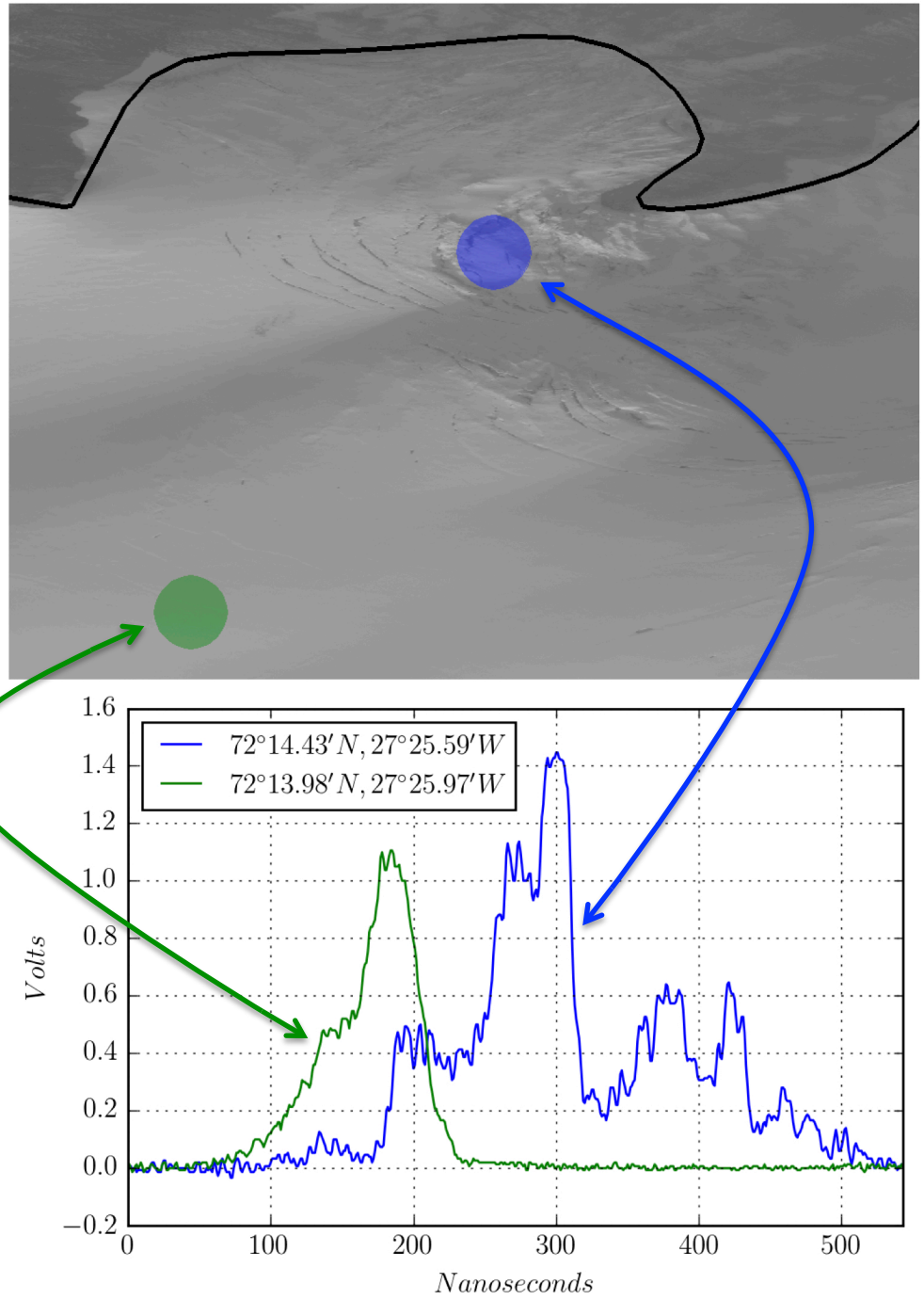


Thomas Mote, University of Georgia

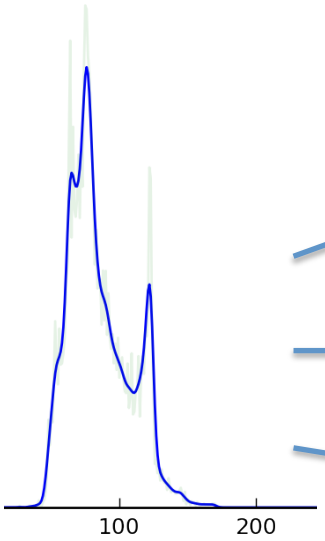


So what's my problem? (Practically)

- Want to know crevassed status from 2004
- There's no optical imagery from before 2009…
- Need to invert from time spectrum to surface type



So, to recap:



Crevasse?

Ridge?

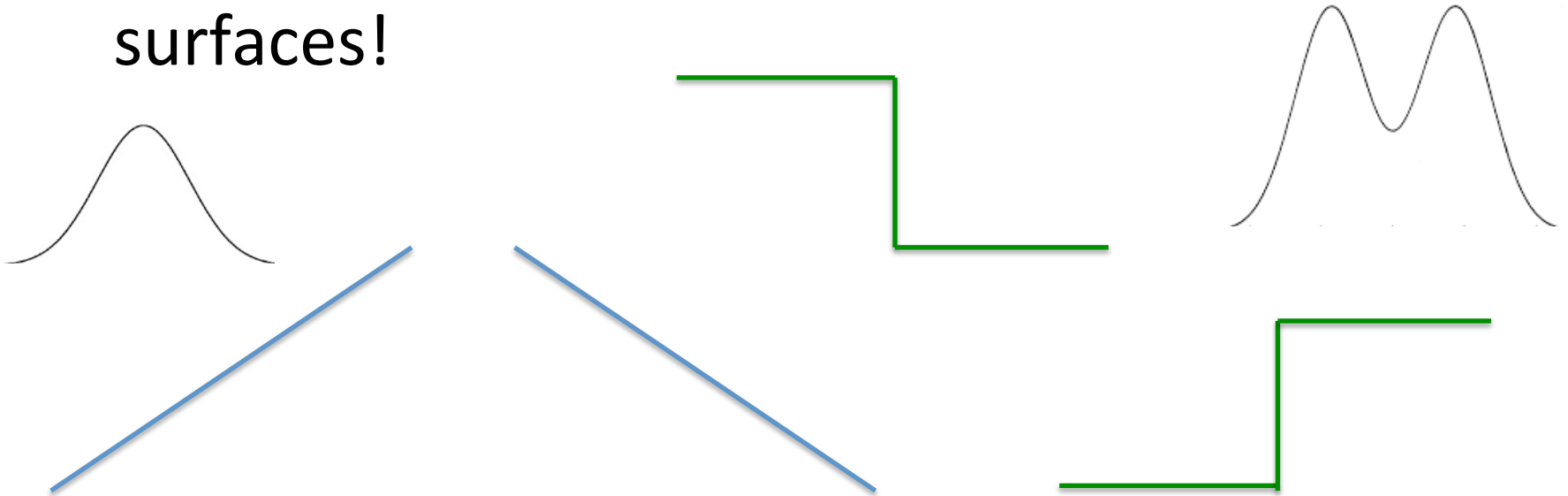
Bowl?

…Easy right?

Inversion of ICESat Waveforms

Inverting to a surface is non-trivial

- Sparse or no validation data
- ICESat Waveforms do not map to unique surfaces!

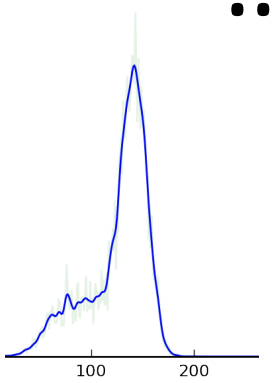


…So how do we solve this?

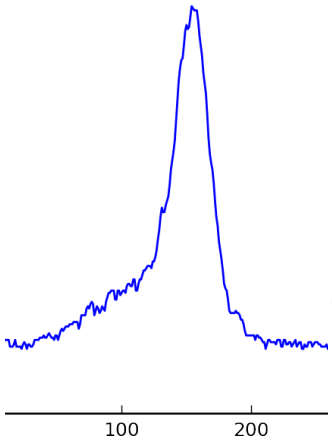
Can we solve it?

- Forward model is impossible…

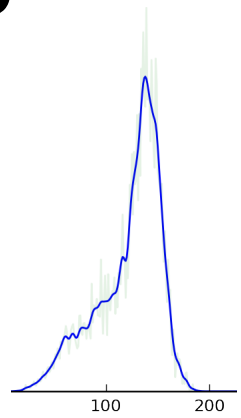
…does uniqueness matter?



Derived Waveform



ICESat Waveform



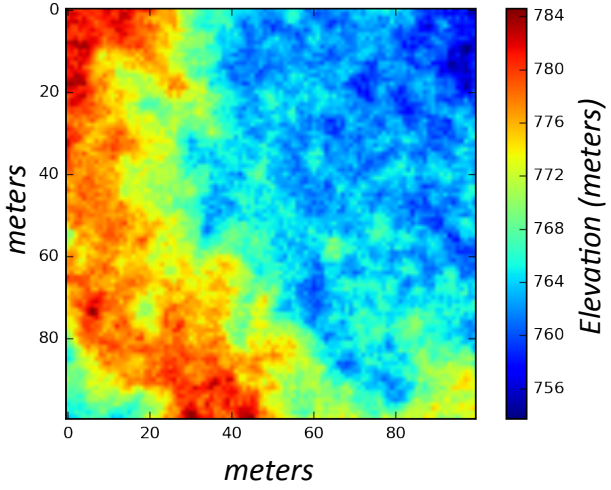
Derived Waveform

?

?

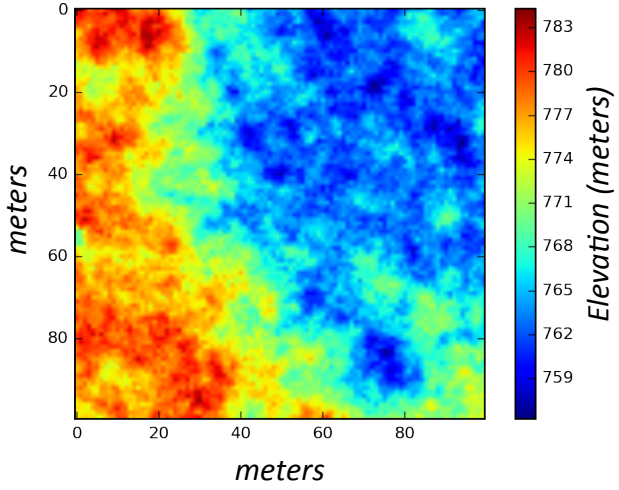
?

?



Derived Surface

Can we find a family of similar waveforms / surfaces, and infer surface characteristics?



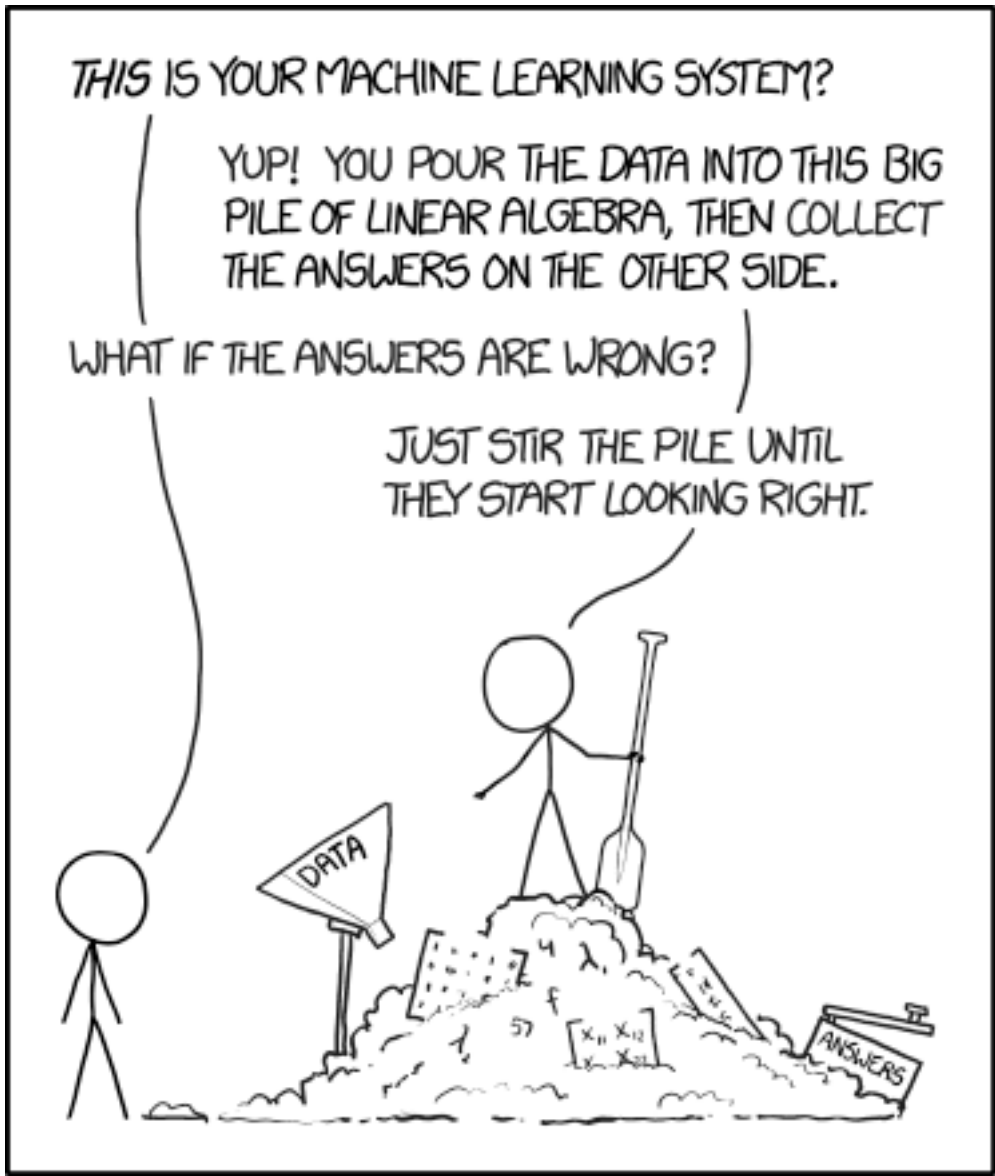
Derived Surface

…So how do we solve this?

Can we solve it?

- Forward model is impossible
- Backward model may be possible… with ***LOTS*** of training data

- Two levers to play with:
  - Models and model parameters
  - **Input training data**



Processing Pipeline Plan

- Simulate waveforms from known surfaces
  - Requires know set of surfaces to simulate from
- Match empirical waveforms (ICESat) to simulated waveforms—and linked surfaces
  - Requires either removing noise from ICESat… or adding noise to simulated waveforms
- Label / catagorize surfaces

Forcing validation by simulation is the major theme!

OIB Flights April 21st, 2012

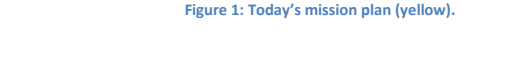


Figure 1: Today's mission plan (yellow).

OIB Flights April 20th, 2012

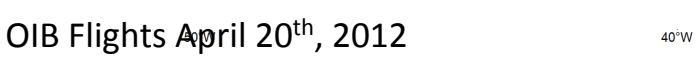


Figure 1: Today's mission plan (yellow).

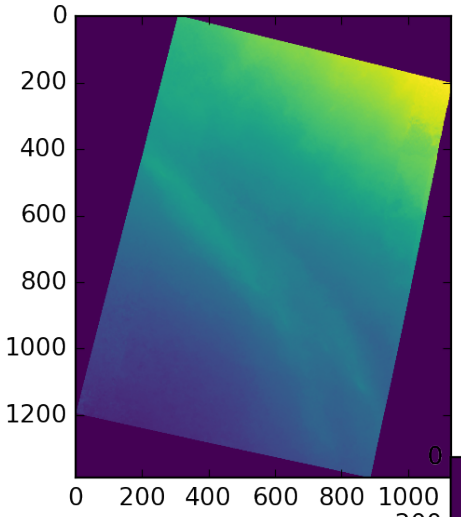
Source data

30 gigabytes per day of Operation IceBridge DMS L3 Photogrammetric DEMs

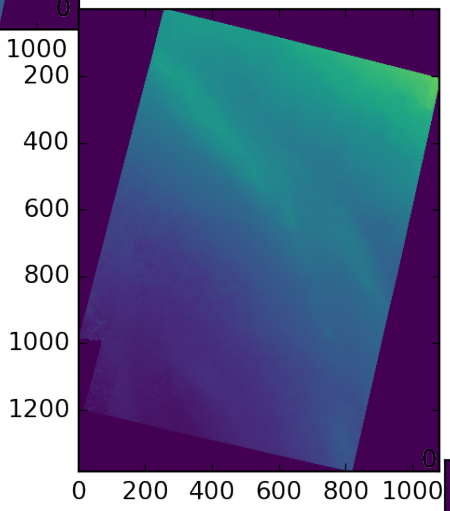
- Combined LiDAR and photogrammetry product
  - Airborne data
- Collected over the same areas and feature types as we'd like to extract from ICESat-1
  - (Southwest Greenland)

…also about 3 million waveforms from ICESat-1

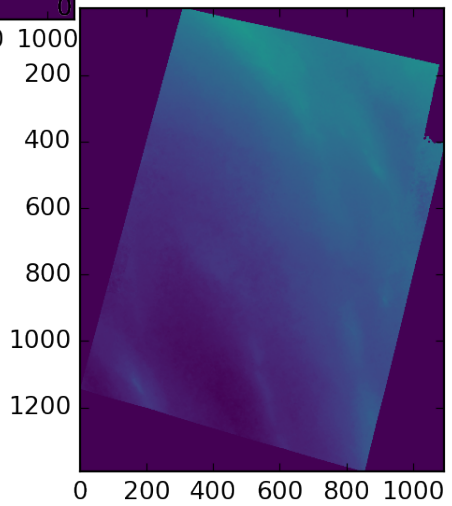
Tiling



44.86 cm pixels



45.24 cm pixels



45.44 cm pixels

Other details:

- Resampling via GDAL
  - bilinear

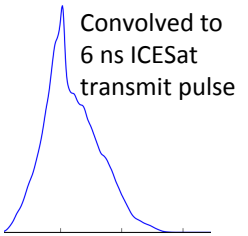
- What do you normalize?
  - Elevation
- What do keep?
  - Location, via georasters
  - Only tiles with valid data (no nans)
- Resolution?
  - Half meter
  - 100 by 100 meter tiles (200 x 200)
- Overlap?
  - Yes, Please!
- Duplicates?
  - Output features files names via hashing
- Dataset shuffling?
  - See above!

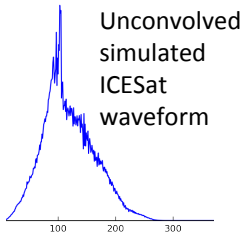
Output with georasters allows slice notation subsetting to geotiffs

Simulating waveforms

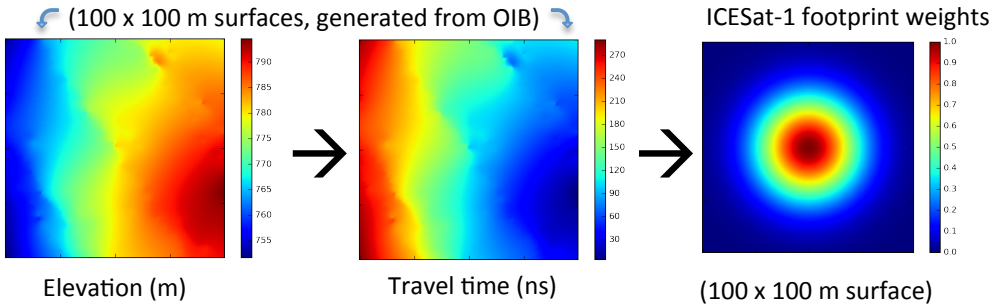
~250,000 image tiles derived from the OIB DMS L3 Photogrammetric DEMs

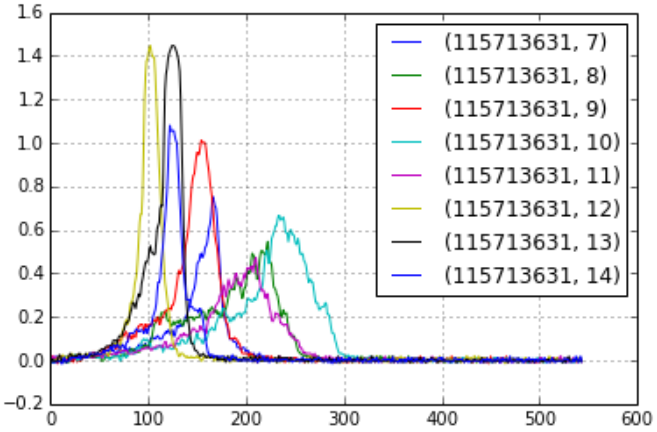
- 100 x 100 meter tiles @ half meter resolution
- Convolved to waveforms using ICESat parameters





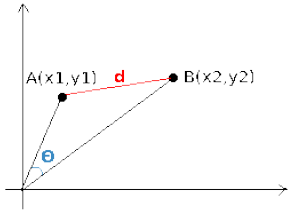
…numba's auto-jit is awesome for convolutions





A few caveats:

- These waveforms are 'perfect'
  - No noise
  - No signal degradation
    - Output energy = observed energy
  - All signals are 'aligned'

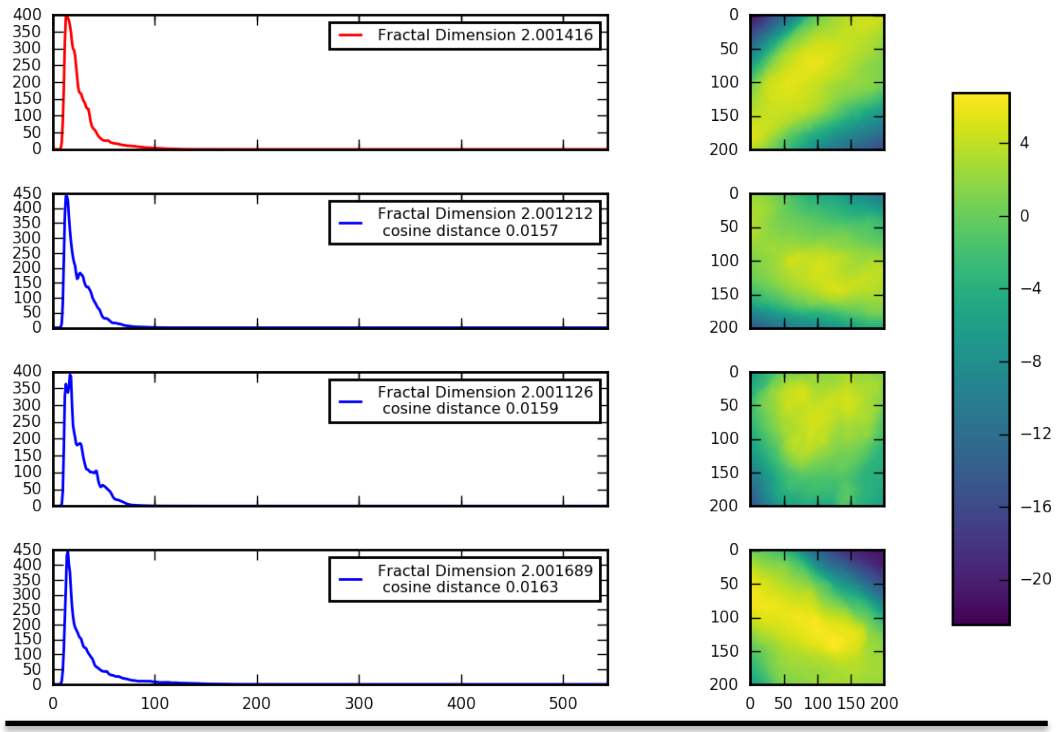


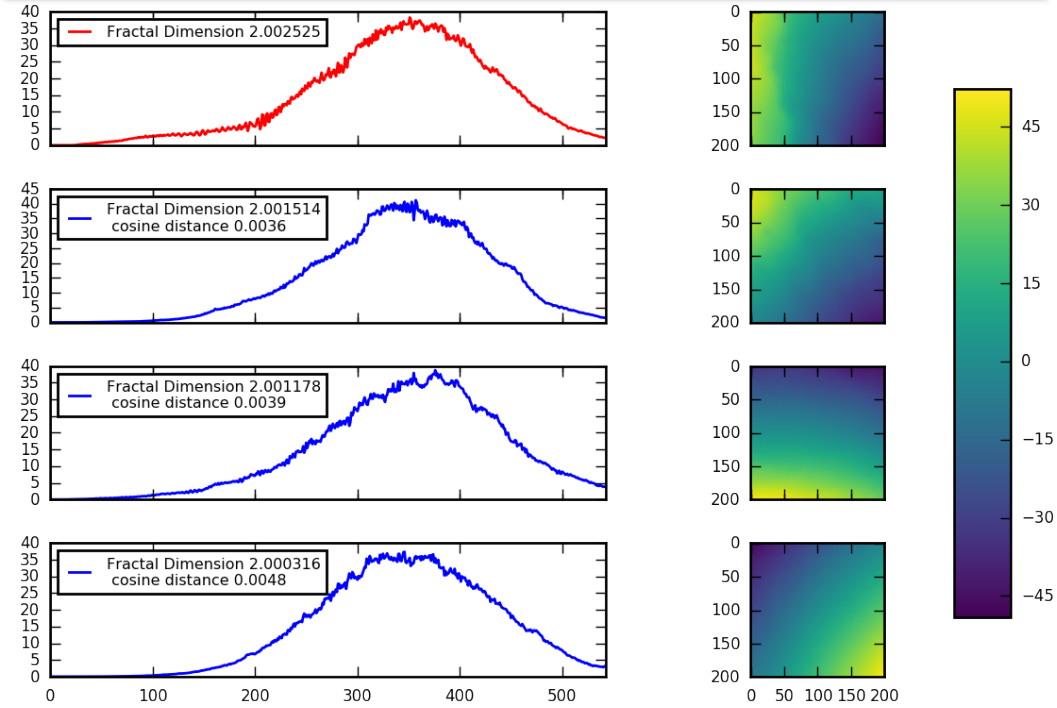
Waveform matching

Locality sensitive hashing database

- Waveforms are cast as high dimensional ‘points’
- Match proximity is measured as the angle between the vectors that the points make from the origin

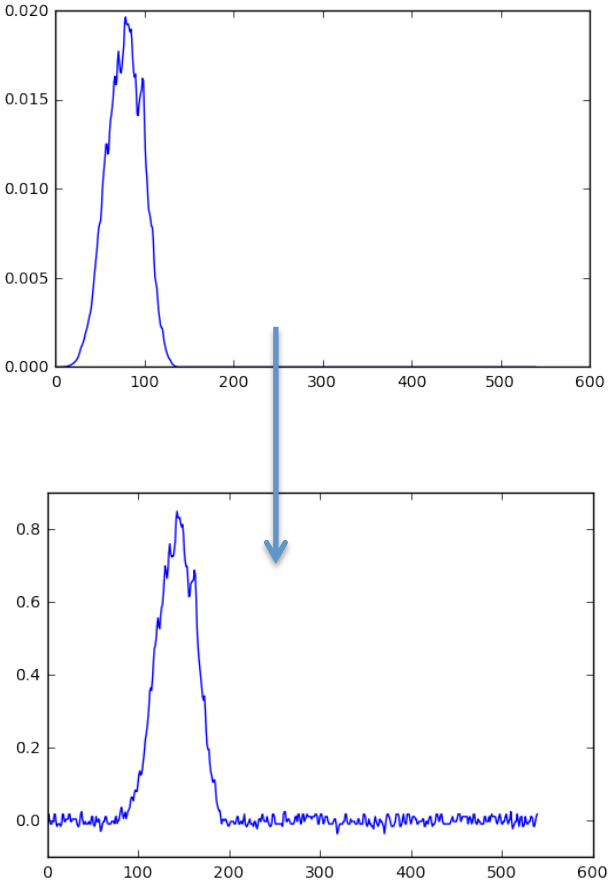
Simulations provide new insights into waveform expressions





Denoising and Centering

- Choose a simulated waveform
  - Scale (from scales distribution)
  - Shift (from shifts distribution)
  - Add noise (from empirical noise database)
- Train in keras
  - 1D convolutional inception net
    - Modified Auto-encoder
  - Batch train
    - Each batch uses a different dataset

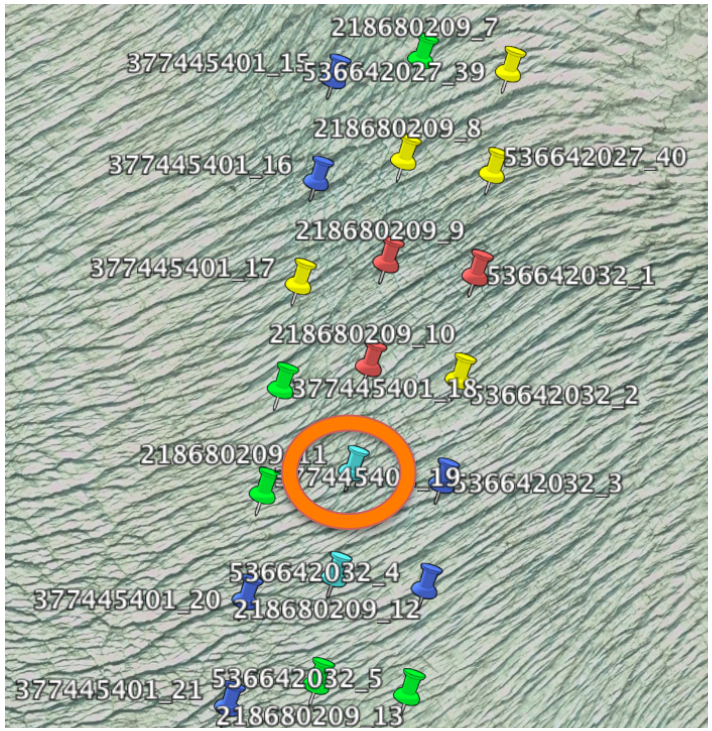


Denoises, scales, and centers

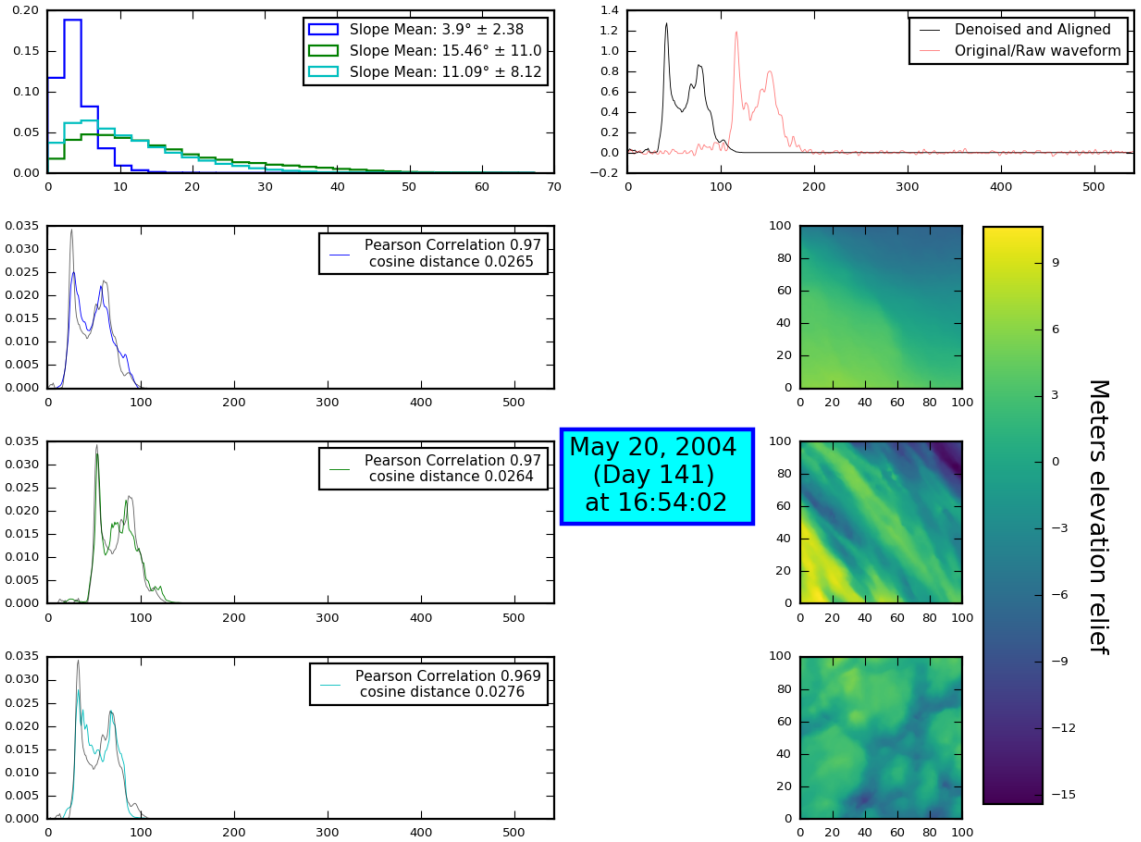
Empirical waveform matching

- Electro-optical noise removed
- Waveforms matched at different offsets
  - Pearson Correlation used to rank candidates

Probabilistic interpretation of surface type and characteristics



Digital Globe image, 2011

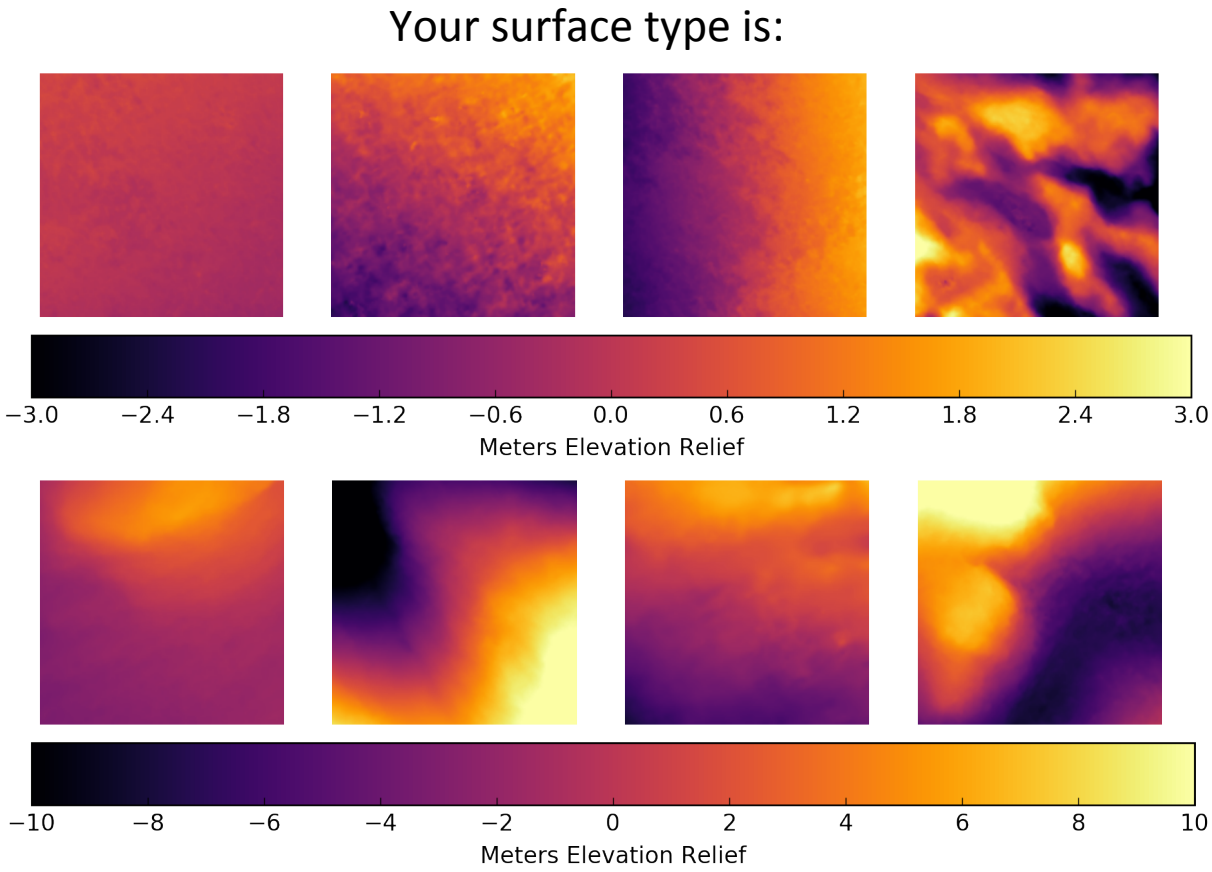


Great, surfaces from waveforms!

…but we still don't know what *type* the surfaces are ☹

Options for labeling:

- Crowd / intern sourcing
  - Not a general solution ☹
- Label a subset
  - Use supervised methods to extend to rest of dataset
  - Imposes our own concept and number of categories ☹
- Unsupervised methods…



Another neural network: SOMs

Self Organizing Maps (also called Kohonen maps)

- Competitive, single layer neural network
- Group like inputs together in output feature space
  - Still requires post processing to cluster output
- Implemented in python via the 'somoclu' package

…it would often be absurd to use primary signal elements, such as temporal samples of speech waveform or pixels of an image, for the components of x directly.

~Teuvo Kohonen, 1990

Self Organizing Map—Take 1

provide a high resolution (< 0.5 meter pixel) digital reference surface of the ice sheet. These reference images were sub-tiled into 212,534 unique 100 by 100 meter surface patches at 0.5m pixel spacing, with an additional 53,133 surface patches produced as a holdout validation data set.

As self organizing maps are computationally intensive, we use a smaller dataset for this exploratory analysis in order to refine our methodology. The data used for this exploratory analysis are 10,000 randomly selected reference tiles.

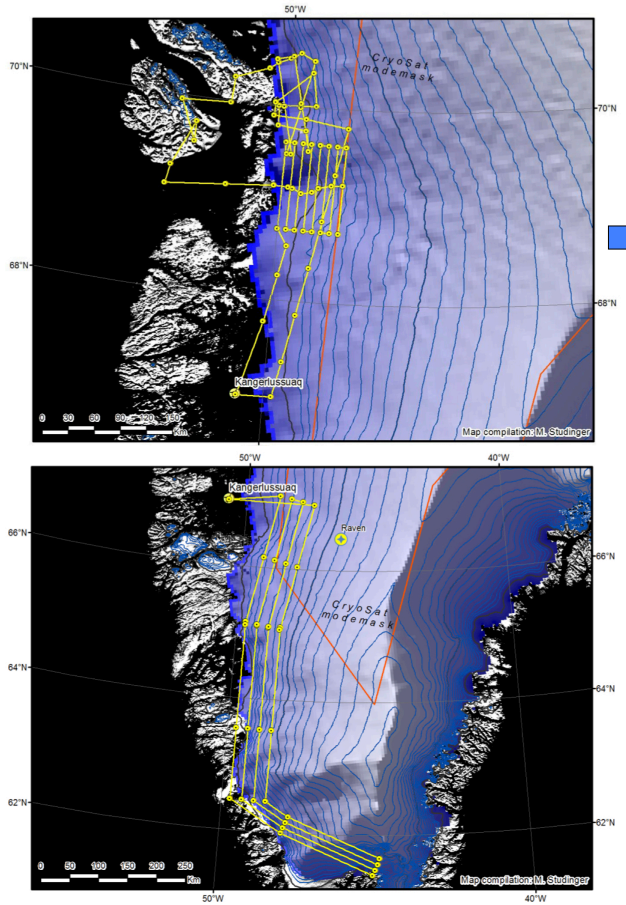
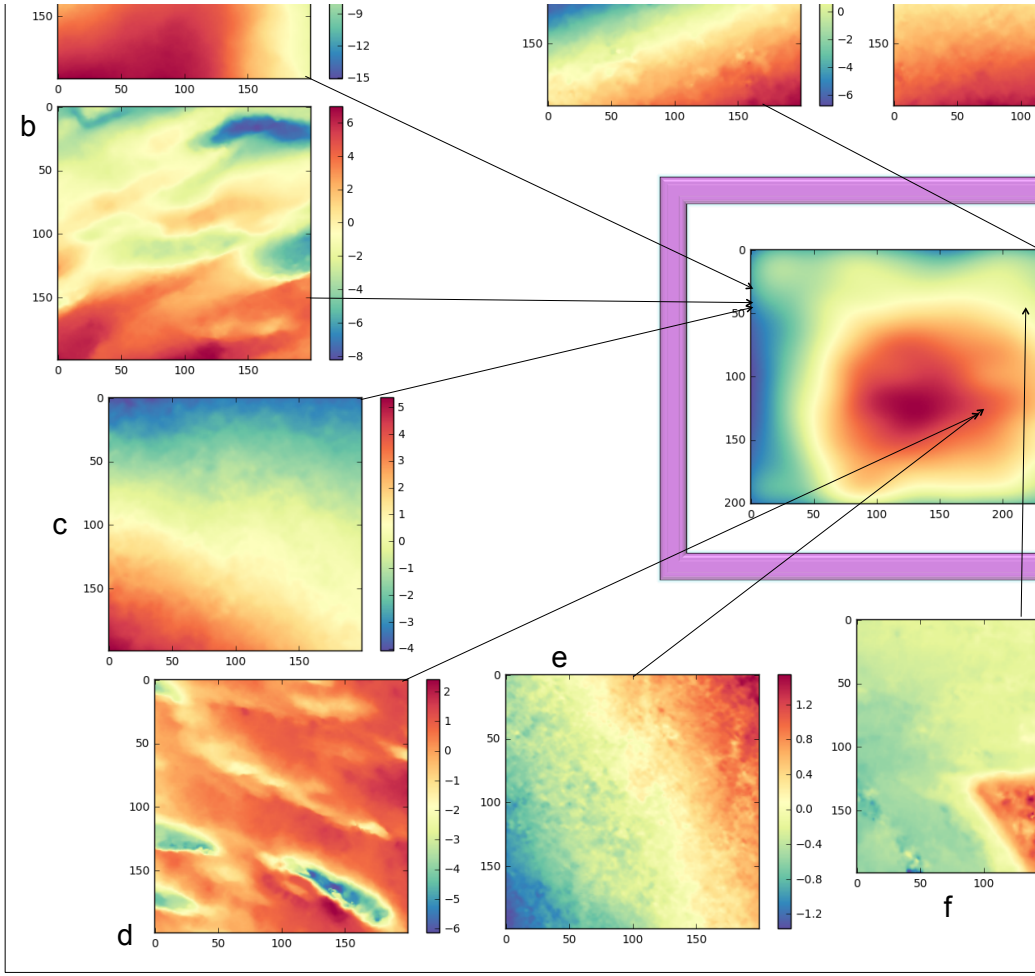


***Figure 1** – (above) Operation Ice Bridge flights that acquired DEMs from which image tiles are derived; April 21st 2012 flight over Disco Bay and Jakobshavn (top); April 20th 2012 flight over the south east coast (bottom).*

***Figure 2** – (right) Kernel Density Estimate (KDE) of the SOM (center, purple box outline), and sample tiles within the SOM (letters a – o). Arrow callouts reference approximate location of tiles within the SOM. Color bar of terrain tiles references meters above or below mean tile height; x and y axis of tiles is pixels (0.5 m posting).*

Tile	X	Y	Type
a	43	0	saddle
b	47	0	crevasses
c	48	2	slope
d	182	123	crevasses
e	184	126	flat
f	211	51	ridge
g	278	198	steep bowl
h	284	196	steep bowl
i	286	56	slope
j	243	0	slope
k	247	0	slope / channel
l	249	0	channel
m	245	0	channel
n	241	1	slope
o	240	0	channel

***Table 1** – (above) Sample terrain tile location and types. Entries refer to subfigure a – o in Figure 2. Type classifications were assessed by author.*



Algorithm and Dataset References

- Wittek, P., Gao, S. C., Lim, I. S., & Zhao, L. (2013). Somoclu: An efficient parallel library for self-organizing maps. *arXiv preprint arXiv:1305.1422*.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464-1480.
- Dotson, Ryan C., and John Arvesen. 2014. *IceBridge DMS L3 Photogrammetric DEM*. Boulder, Colorado USA: NASA DAAC at the National Snow and Ice Data Center.

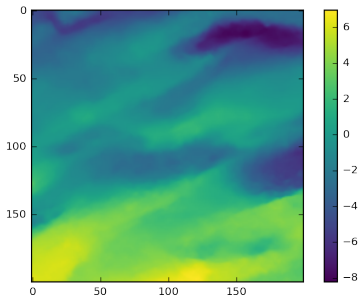
Acknowledg

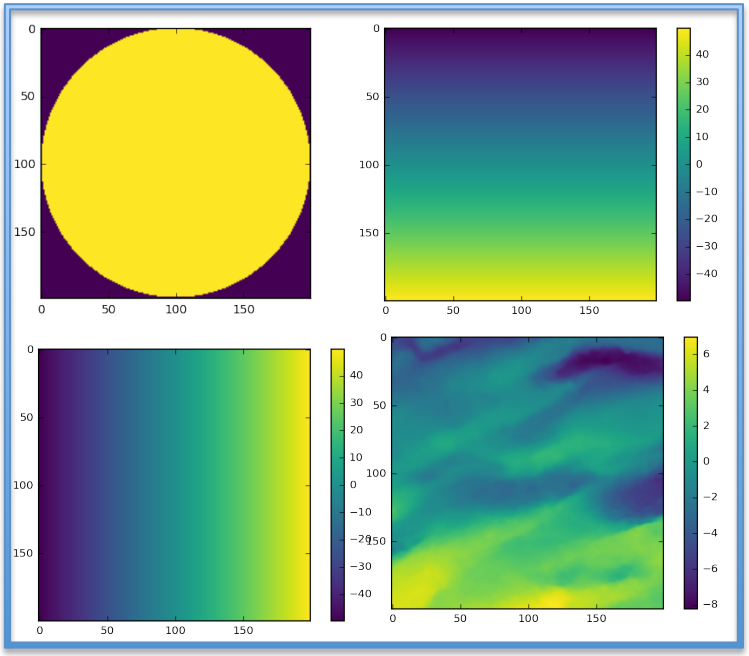
This work was
and Earth Scie
Using Historica

We thank Ryar
updated versio

Tile Normalization

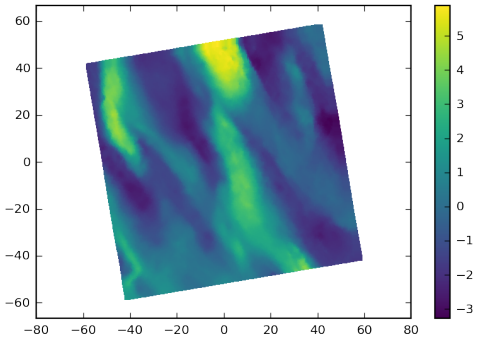
Original Tile



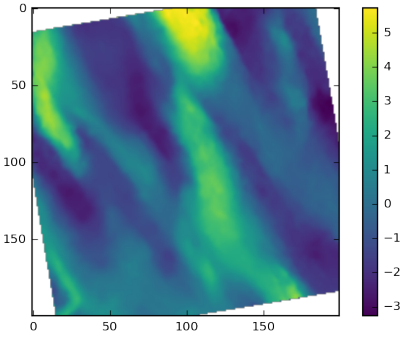


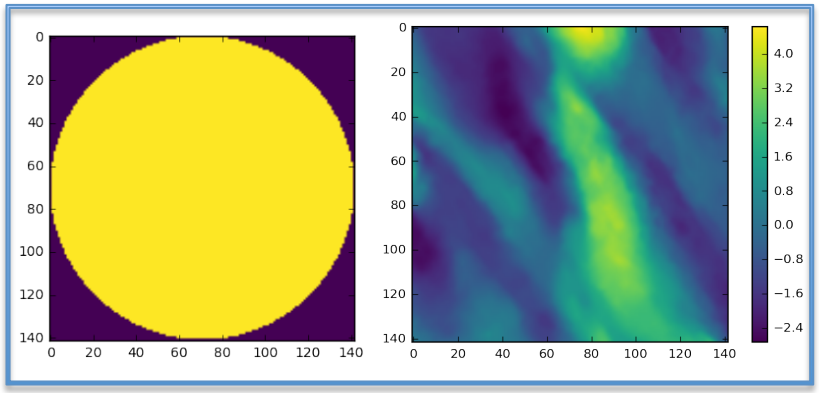
Create XY rasters, circular mask…

Flatten to [X], [Y], [Z]…
…detrend via PCA

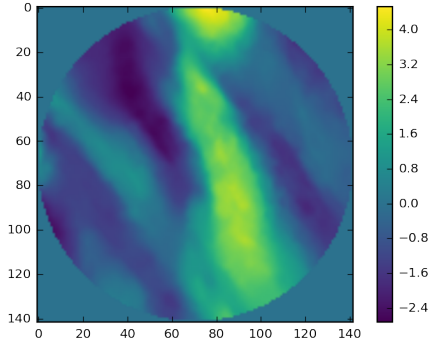


Rasterize output
(scipy.interpolate)

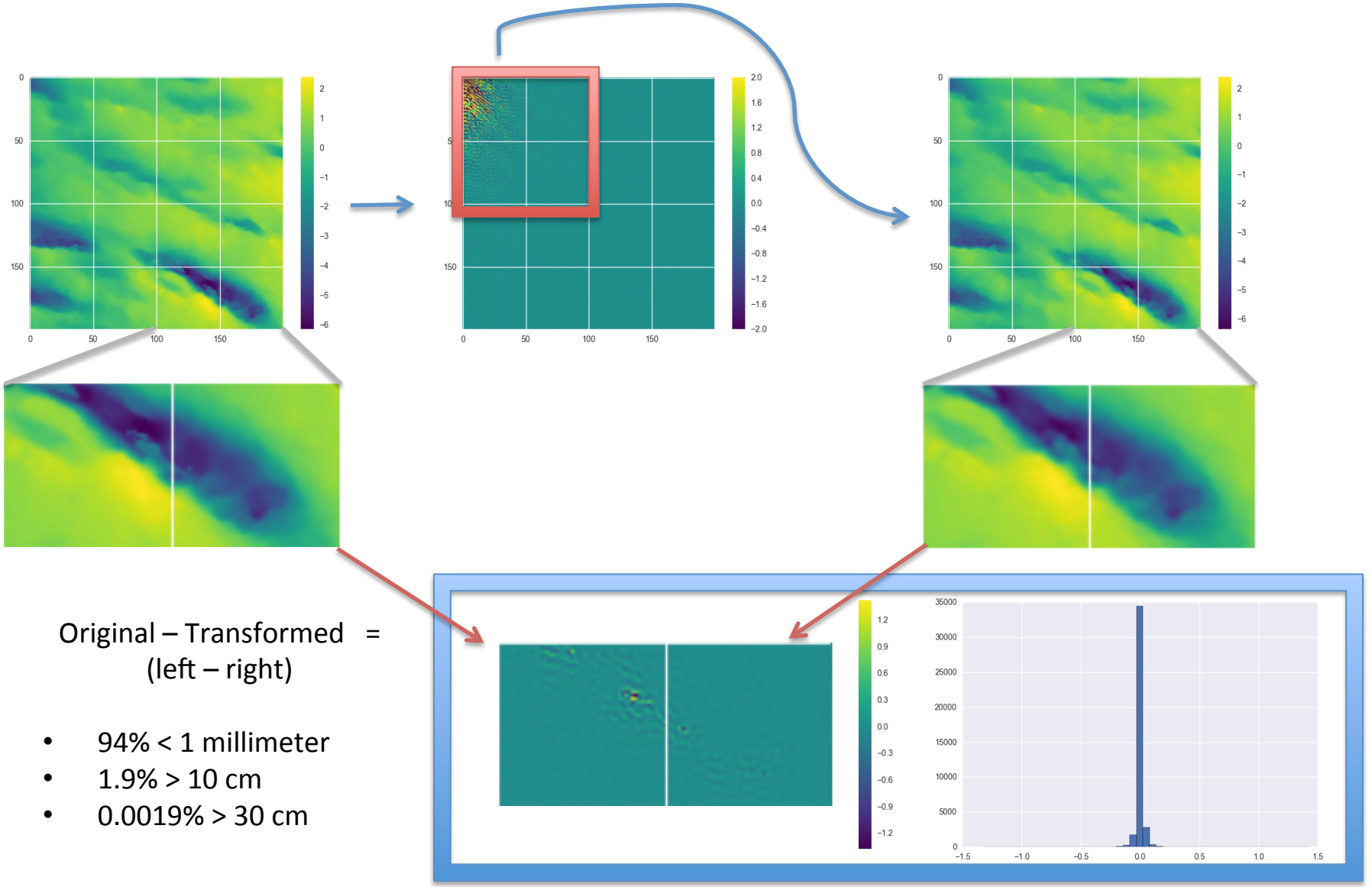




Crop, circular mask

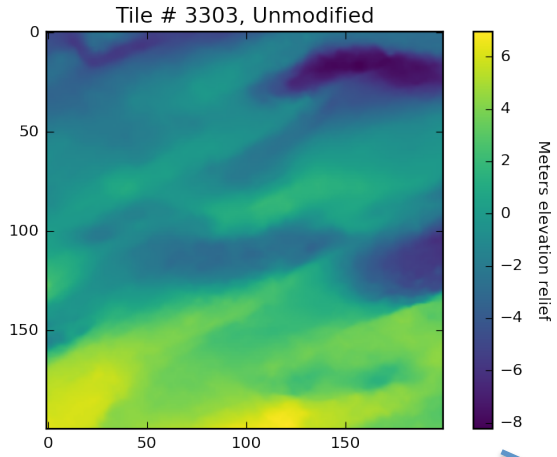


DCT input size reduction



Original – Transformed = (left – right)

- 94% < 1 millimeter
- 1.9% > 10 cm
- 0.0019% > 30 cm



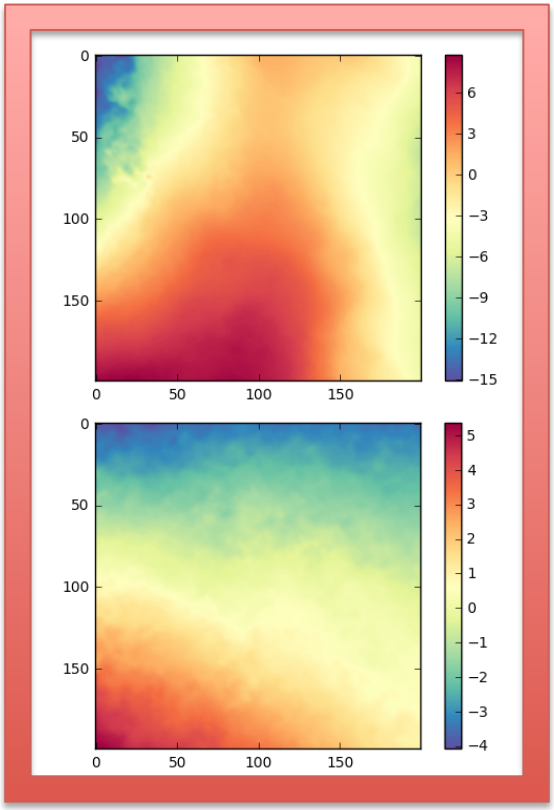
Results:

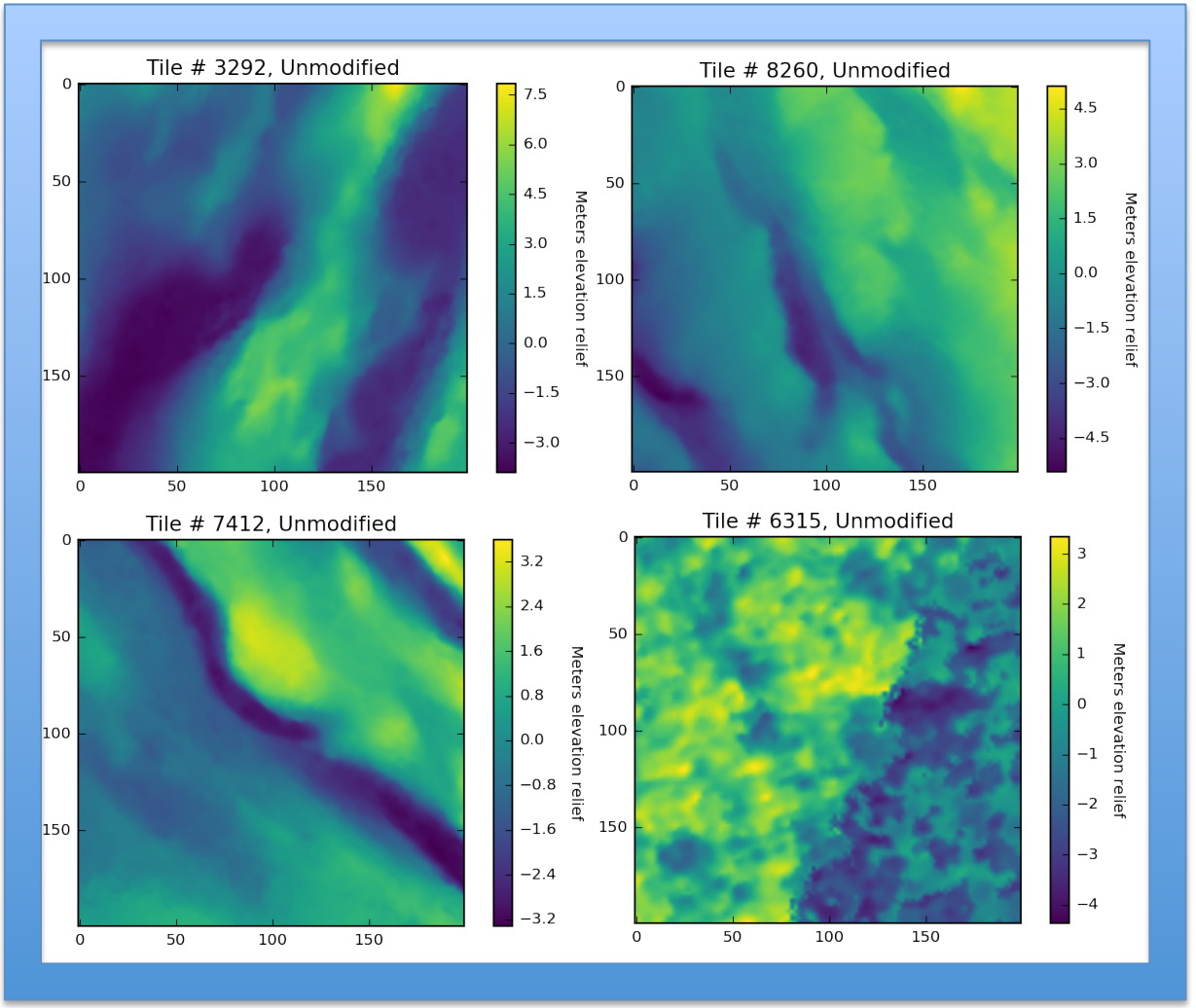
- Reduced input feature space
  - 40,000 → 5,184
- More robust sorting
- Clusters are more seperable

(10K tiles)

~4 hours run time

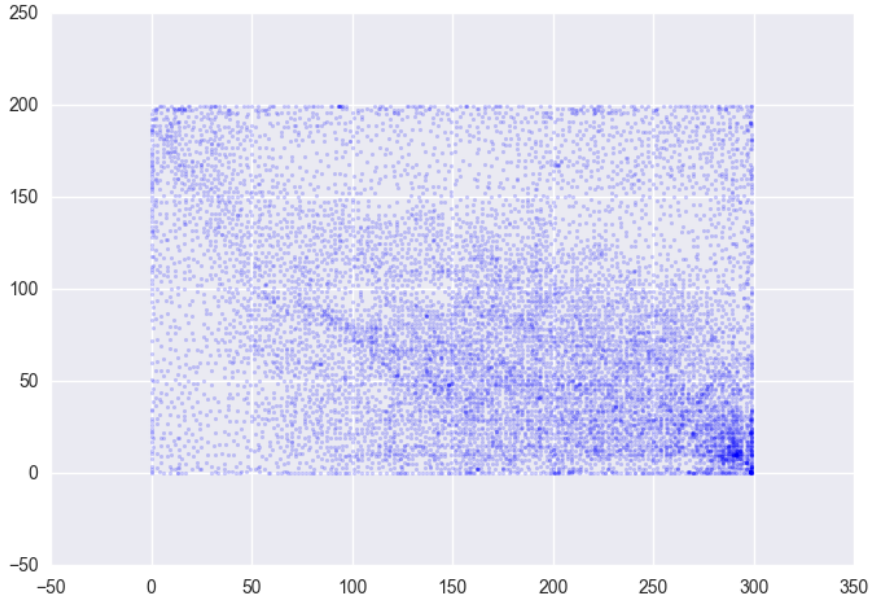
~10 Days run time





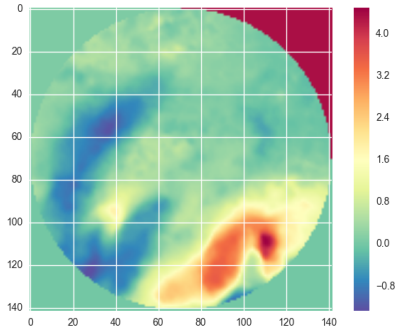
Next steps… & Interesting things

- Ramp up to full dataset
- Cluster output
  - OPTICS
- Label clusters



Things to maybe try:

- Adding non-spatial metadata to SOM
- Projecting output space to more dimensions



Lessons Learned & Best Practices

For tiles:

- Decide on a common data format
  - Same file type, projection and units
    - Hint: probably geotiff, although I do like hdf
- Decide on common resolution and size
  - Bigger than you think you need! Add padding
  - Resolution should be as high as possible
    - I'm biased; you can always resample to lower later
      - Corsen to what is practical as needed
- Enforce Standards
  - No NaNs, Geolocation
- Detrend as much as possible
  - Anything that can be retrieved by metadata (i.e., location)
  - Elevation (zero-mean centered is good for numerics)
  - Slope and rotations (via PCA)

Libraries:

- GDAL for resampling
- Georasters for tiling
  - Subset via slicing and output to geotiff
- Rasterio for reading lots of small tiles
- Scikit learn for…
  - PCA, LSH, clustering...
  - Distance measurements
  - Everything
- Somoclu for SOMs

Questions?