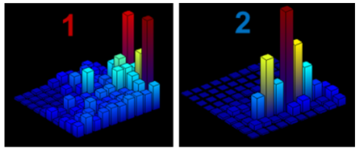
Model Predictive Control Architecture for Optimizing Earth Science Data Collection (PCAES)

ESTF2015

14 -16 June 2016

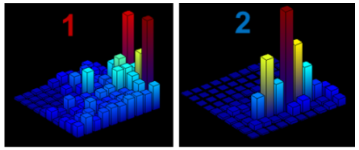
Mike Lieber, Carl Weimer, Reuben Rohrschneider, Lyle Ruppert

Collecting More High Value Earth Science Data



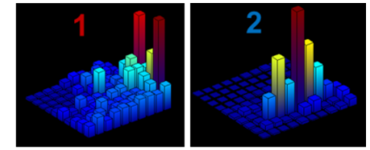
- Challenging budgets demand reconsideration data collection process and data exploitation.
- The remote sensing field benefits from a myriad of sensors and sensor suites of increasing capability and complexity. Meanwhile, on-board systems for real-time control of instruments have been limited in general to a few traditional architectures.
- Focus of this program is to optimize instrument or instruments data collection capability using advanced software architectures.
- Optimized systems many times result in complex systems. Characteristics are:
  - Multiple constraints, nonlinear physics, time-varying systems, interacting, multivariable systems and sometimes sparse data or missing data.
- Multiple Earth Science applications:
  - Trend is for higher capability, scene-directed instruments in the future
  - We focus program on multi-beam lidar systems (for example – electronically steerable flash lidar (ESFL))

Significance of PCAES Work to Future Earth Science Missions

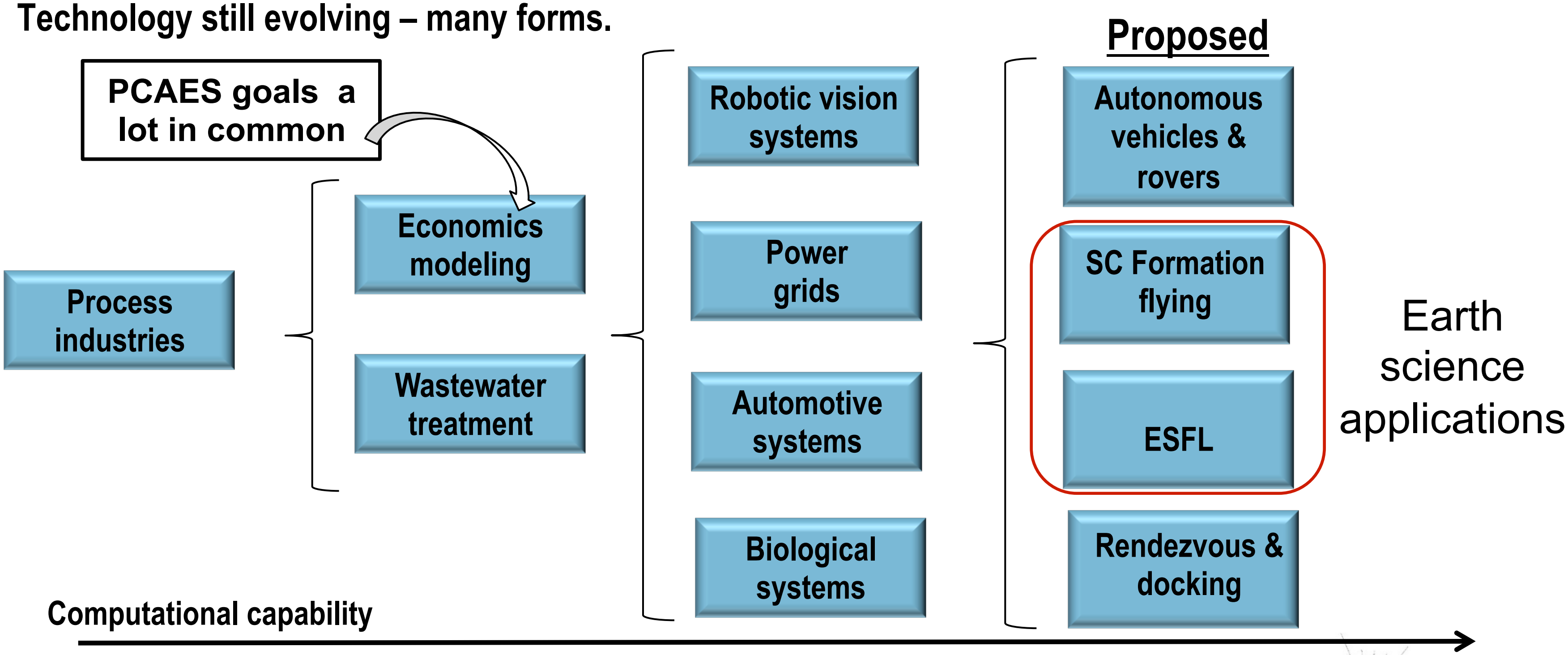


- Tightening NASA budgets require new missions to (1) address the issue of data collection efficiency and (2) consider smaller payloads which can still provide high quality science. The PCAES contract is developing an on-board <u>autonomous</u> software architecture that addresses both these areas.
- PCAES targets a specific application of data collection with adaptive lidar but the architecture is based on ground-based control of complex, hierarchical and sometimes distributed subsystems. It collects much more data of importance for Earth science by optimally targeting areas of interest.
  - The software being developed for PCAES can be used for control of multiple SmallSats /CubeSat's in formation and can optimize overall system performance of the distributed sensors. DeltaSat concept to emphasize strawman mission.
  - PCAES is a new development for space applications (as far as we are aware) which will enable new types of sensor hardware systems and can take advantage of recent developments in sparse signal processing and compressed sensing.
  - PCAES works at the fastest time scales (<1 s), making use of advances in on-board computation speed with FPGA’s.
  - <u>PCAES uses optimization-based control versus classical control</u>. Computationally intensive approach.

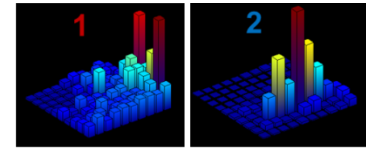
MPC and it's History of Applications



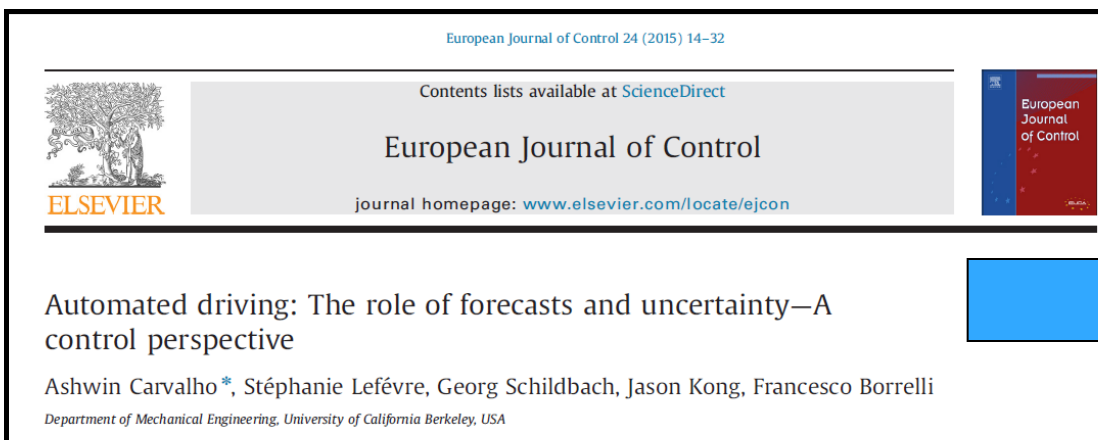
- MPC was invented in the process industry. Large, complex, multivariable systems with constraints (flow rates, volumes, temperatures, etc), multi-level or hierarchical but with very slow time constants. <u>It's an architecture not a set controller</u>.
- Although there has not been problems with systems going unstable, the first stability proofs were not developed until the mid-late 90's.
- Two things have been a deterrent to adoption – computational burden and technology migration.
- Technology still evolving – many forms.



Heuristic Explanation of MPC



- **Works very much how one drives a car.**
  - Continuous adjustment/ optimization of steering and speed using visual feedback of the operating environment while having an embedded model of car operating parameters (acceleration, turning sensitivity, braking) and considering constraints (lanes, other cars, max braking and acceleration). Predict ahead.
  - Multi-layered (the person with smart phone providing directions)



European Journal of Control 24 (2015) 14–32

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Automated driving: The role of forecasts and uncertainty—A control perspective

Ashwin Carvalho*, Stéphanie Lefévre, Georg Schildbach, Jason Kong, Francesco Borrelli

Department of Mechanical Engineering, University of California Berkeley, USA

In fact – many approaches to autonomous cars use a version of MPC

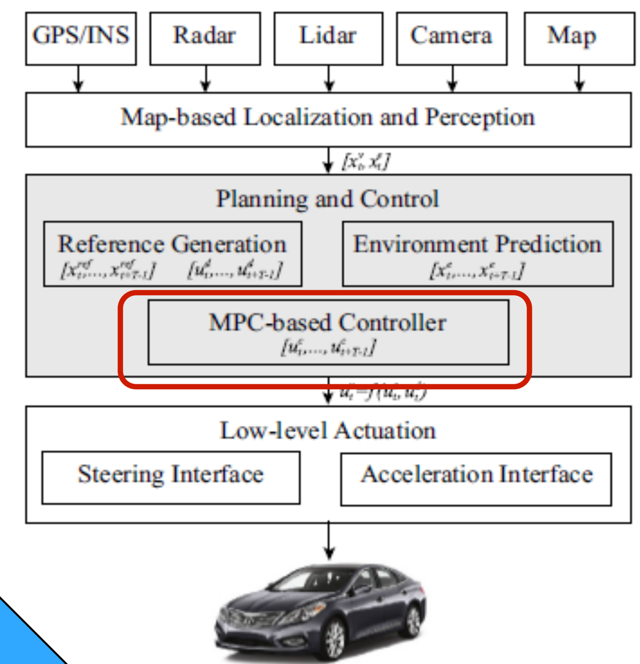
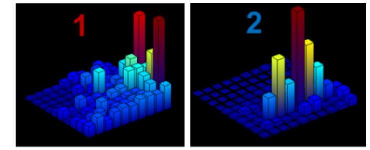
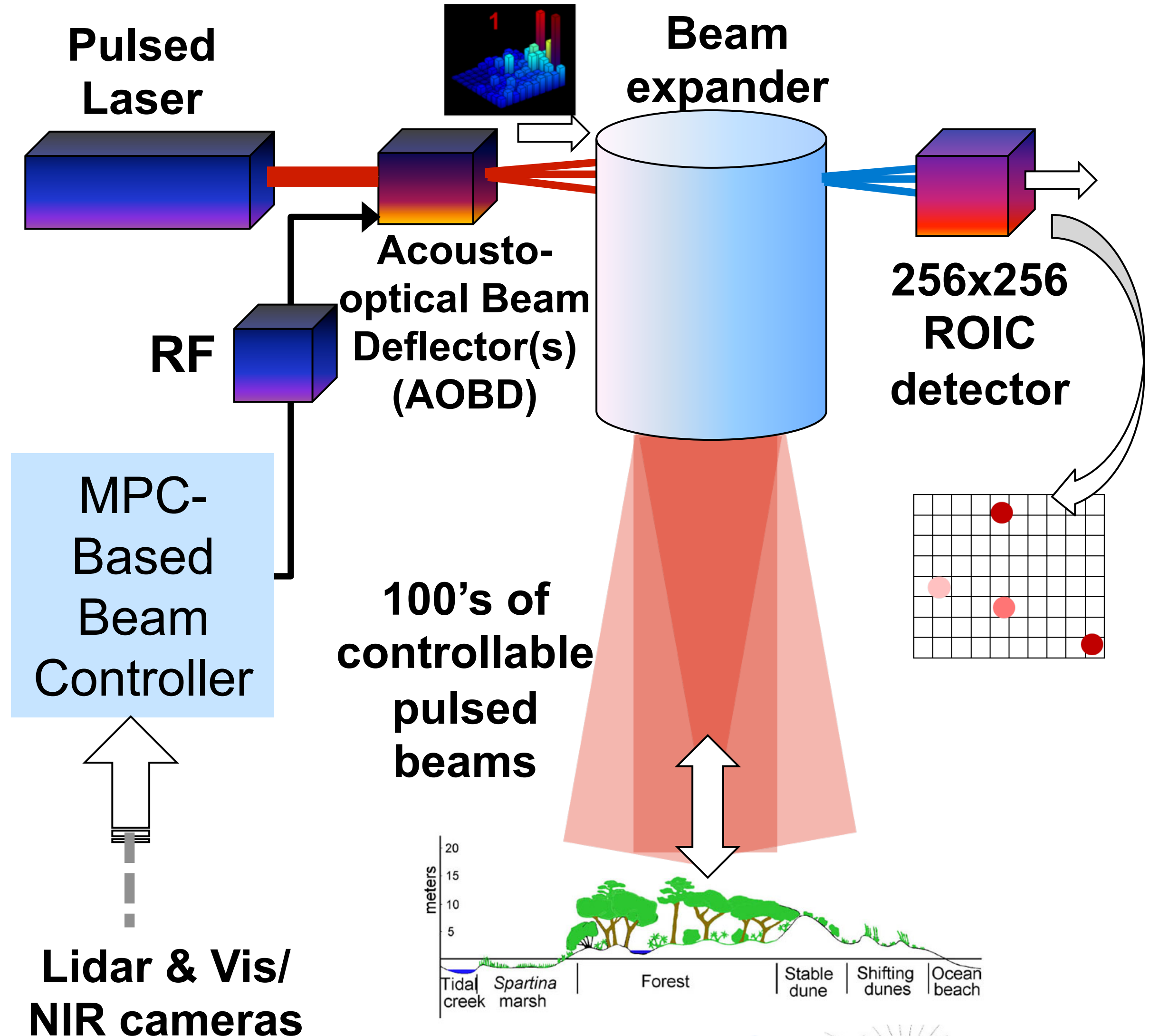


Fig. 1. Intelligent vehicle system overview.

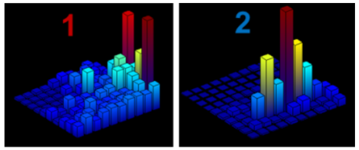
Targeted Application - Adaptive Lidar / ESFL



- We apply MPC architecture to an electronically steerable flash lidar (ESFL)
- AOM controls amplitude and angle (frequency) of each beamlet independently.
- 2D beam steering
- Constraints:
  - Total power
  - SNR
  - Steering angle

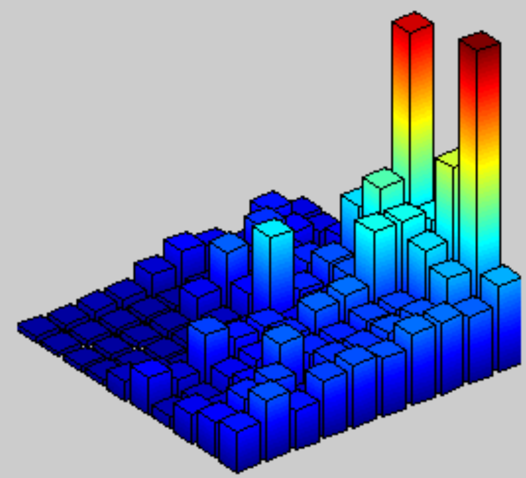


Computational Speed Problem - Time Evolution of Power Map from Landscape and Clouds



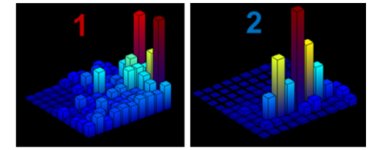
- The example images are 10x10 –how power map changes as we go though a cloudy area. Simple weighting – clouds very penalized, water least penalized. Prior data collection not included
- Estimate center-to-center distance of the 1st to 4th sub-image takes ~14 seconds to traverse. ~20 mi squares or ±1.35 deg (at 700 km)

Ground track and moving FOV over Panama canal

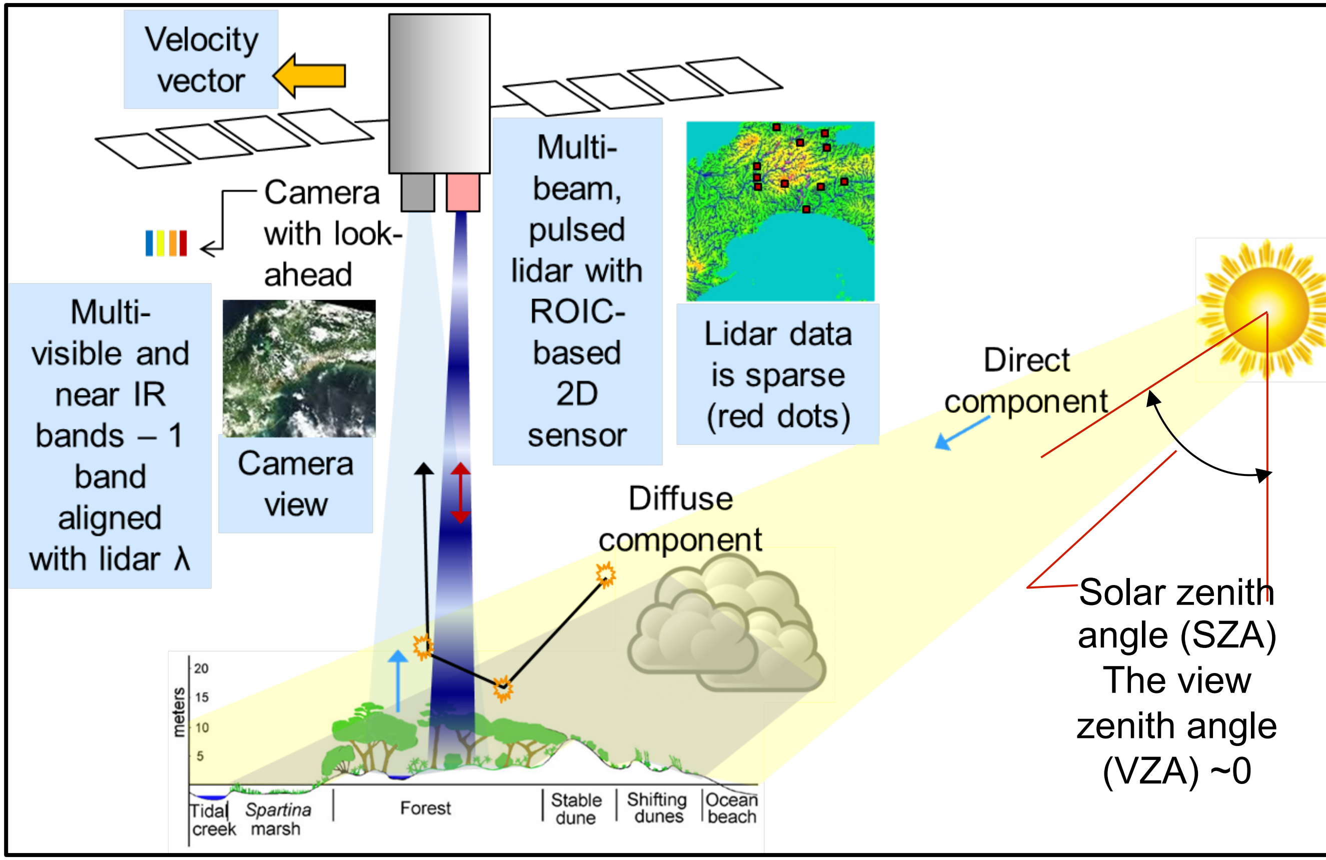


Power map before thresholding

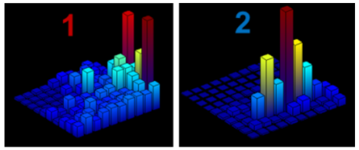
Geometry of the Control Problem



- **Lidar return data sparse – typically 1% of pixels. Need estimated return lidar power in all 256x256 pixels of ROIC receiver.**
- **Light flux measured by the lidar and camera is made up of two components – direct component (non-scattered) and diffuse component (scattered).**
- **Due to solar zenith angle, light seen by camera not same as lidar return signal. <u>Lidar return at hotspot.</u>**

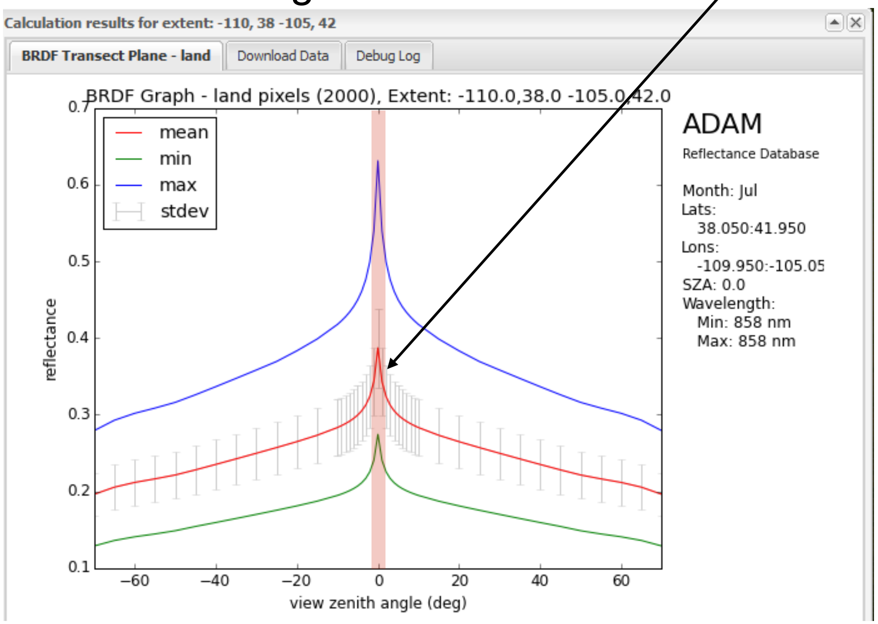


What Camera Sees is Different Than What Lidar Sensor Sees

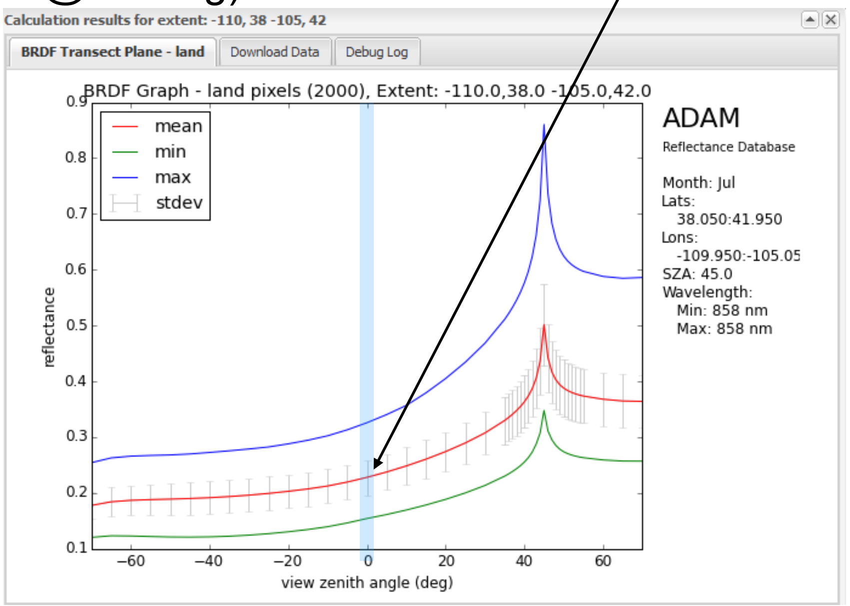


- The difference between the camera measured signal and the lidar return can be significant and gets worse as the solar zenith angle increases.
- Below is shown mean reflectance – every ground pixel will have variability.

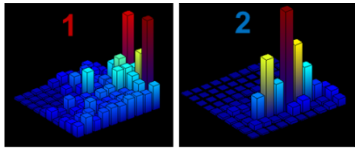
Normalized lidar return signal with noise bars



Camera measured signal (sun @ 45 deg) with noise bars

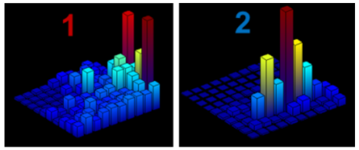


Technical Challenges and Solution Path (1/2)



- **Realized that we need an on-board classification system.**
  - We are starting with simple approaches where possible. We do not need a high end classifier. Experiments by Reuben look promising for simple color ratio classification.
- **Given very large memory storage and extremely powerful computers, we could brute-force are way to a software solution.**
  - On-board DEM and scene classification reduce computational requirements but at the cost of additional memory requirements.
  - Work has been done on “multi-resolution trees (MRT)” allows one to use coarser data grids and may provide a compromise approach
- **Lidar return data will be very sparse over defined 2D field of view and provides challenge to predictor over entire field.**
  - The scene classifier simplifies task of estimating power by reducing problem dimensionality.
  - Heritage data.

Technical Challenges and Solution Path (2/2)

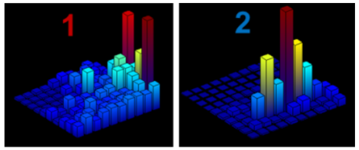


- **Incompatibility of different remote sensing data products.**
  - **Different resolutions, file types, processing levels, etc require use of software tools like Matlab Mapping toolbox and community developed tools including NASA and universities.**
- **We have no real model for MPC, and the same scene changes with time (seasonal).**
  - **Data driven approach**
  - **Scene classification creates a group of pixels with similar reflection or power return thereby creating a model**

MISSION ANALYSIS & DEFINITION

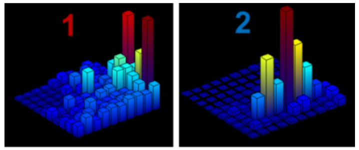
ESTABLISHING BASIC MODEL PARAMETERS

Mission Definition - DeltaSat



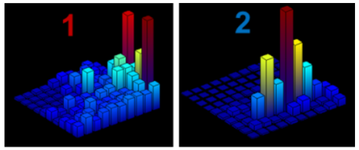
- Mission lidar parameters (wavelength(s), SNR requirements, etc) very dependent upon science requirements. However, we will generate generic lidar mission for ground DEM and foliage characterization with focus on changes from heritage data – DeltaSat.
  - Need for SNR calculations of lidar beams for lidar power map beam distribution.
  - On-board computation and processing
- Use of slightly modified version of lidar Radiometric Math Model used and validated on Calipso with extensive improvements for ground and foliage return signals (see Table next page)

Mission Level Definition and Requirements

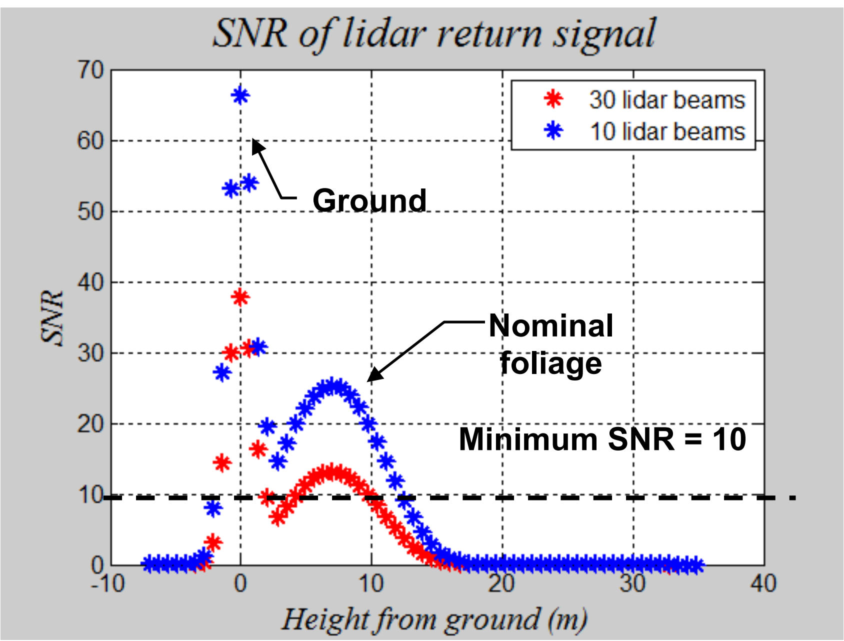


Parameter	Value	Units	Comments
Orbit altitude	440	km	
Spot size per beam	30	m	Landsat size – application driven
Across-track swath width	7.68/ 1.0	Km/ deg (instrument FOV)	
Solar background	0.0309	W/m^2 sr nm	
Background noise, Detector noise	1, 8	Photons/time bin/pixel	
Laser pulse energy/ Average power	1000/ 30	mJ/ W	Fibertek SQ laser – 1064 nm, eye-safe concern 1 beam
Maximum* SNR ground return [1 beam, 10 beams, 100 beams]	[243 66 15.7]	SNR	Maximum SNR of waveform return. 1 beam case not deflected.
Maximum* SNR foliage return [1 beam, 10 beams, 100 beams]	[96 25 3.3]	SNR	Maximum SNR of waveform return. 1 beam case not deflected.
Number pixels per footprint	1	-	Max SNR but poor resolution
iFOV	68	?rad	
Detector size	256 x 256	-	
Receiver telescope diameter	1	m	CALIPSO telescope design
Number of pixels across per degree FOV	256	-	
Range bins	61		
Resolution of lidar range	0.7	m	

Lidar Return SNR Depends Upon Surface and # of Beams



- **Example output from Lidar Radiometric Model – ground and nominal foliage return signal.**
- **Radiometric calculations for 1064 nm, but Landsat/ MODIS data used in PCAES modeling is 850 nm band.**



SNR = Signal photons in each bin/ Noise photons in each bin

Reflectance variation of surfaces

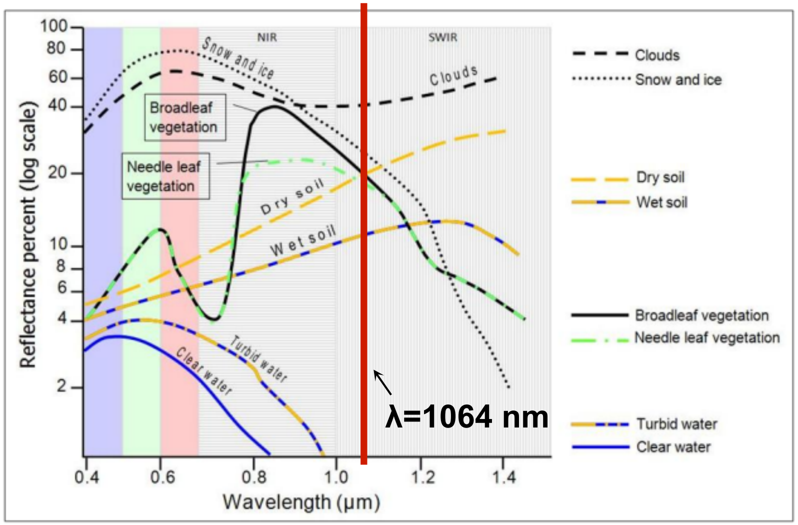
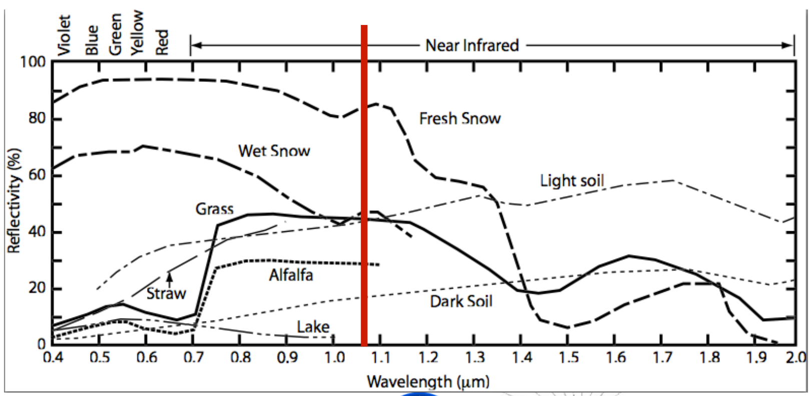
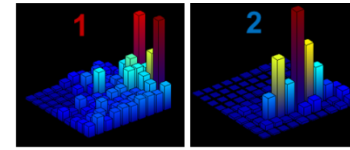


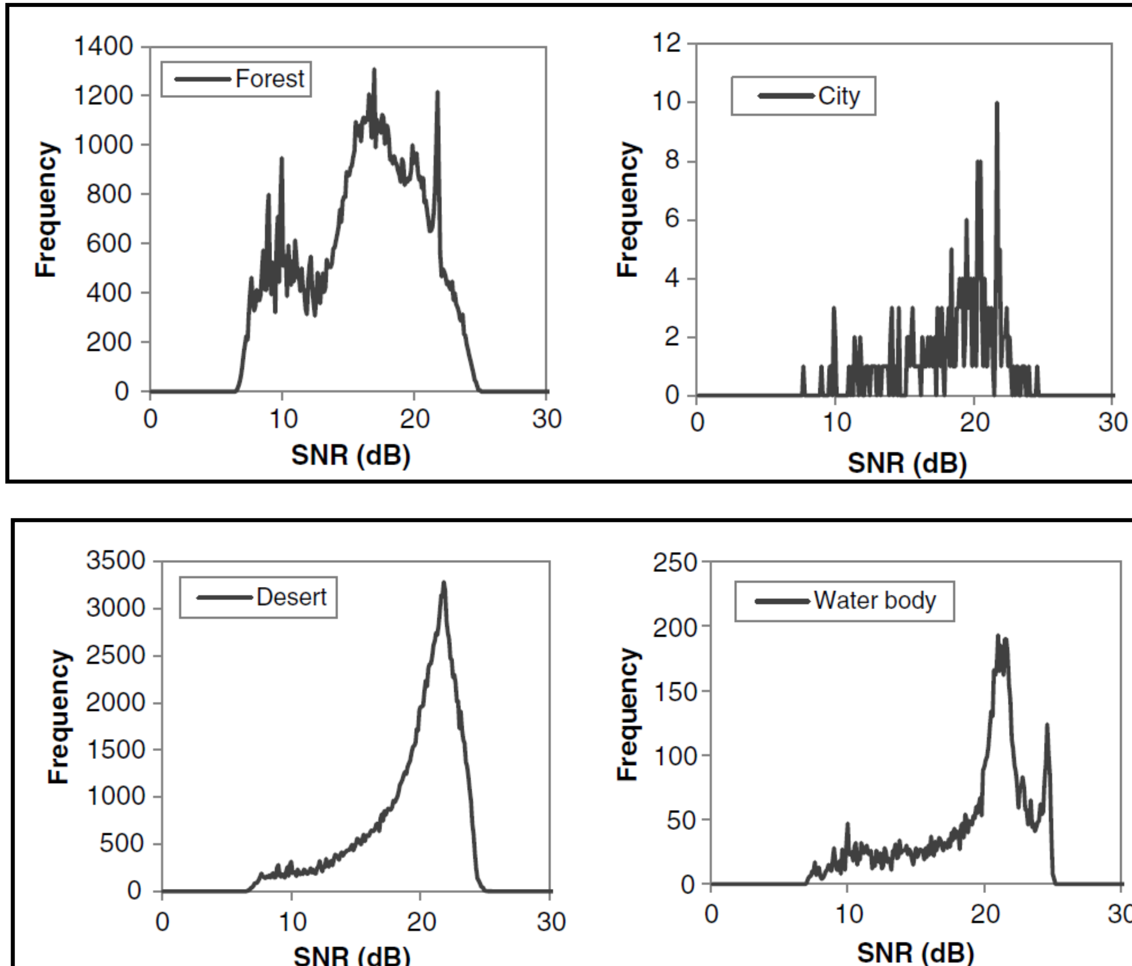
Figure 10. Spectral reflectance curves for the following paired landcover types (Aronoff, 2005):
- Clouds vs. Snow & ice
- Broadleaf vegetation vs. Needle leaf vegetation
- Dry soil vs. Wet soil
- Turbid water vs. Clear water



SNR for ICESat/ GLAS LIDAR

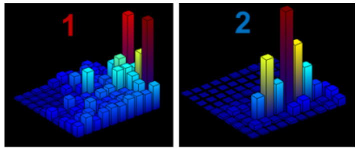


- Histogram of GLAS SNR for several types of terrain indicates 10 – 23 dB (10 to 200) range of returns.
- We assume minimum SNR = 10 for PCAES modeling.



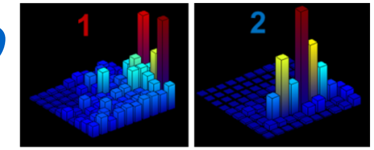
S. Nie, et al, "Signal-to-noise ratio-based quality assessment method for ICESat/GLAS waveform data", Opt. Eng., 53(10), Oct 2014.

PCAES Work Focused on Daytime Operation and Optimal Pixel Targeting



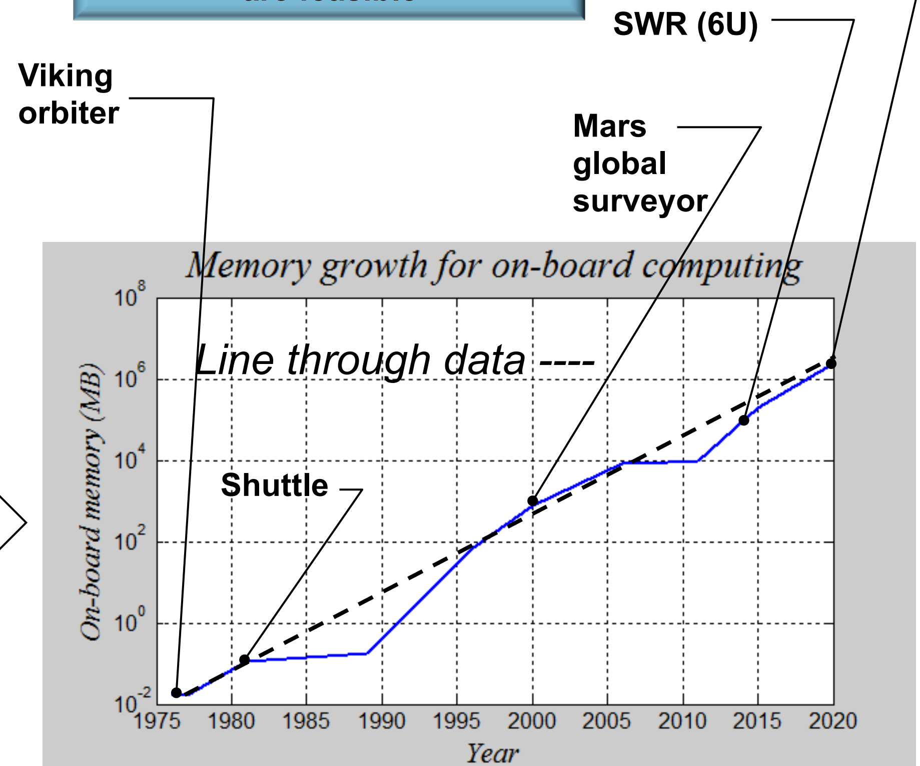
- Nighttime Operation – Large efficiency gains also
  - — Cloud detection works at night with mid-IR band.
  - — Ground registered heritage data bases.
  - — Moon glow imaging sensor
    - VIIRS DNB sensor band
      - — Panchromatic (0.5 – 0.9 µm)
  - — Active radar payload
- Optimal range gate control
  - — Much simpler problem to solve
  - — On-board global DEM map helps to provide initialization.

Including Heritage Databases into Estimator – On-Board Memory



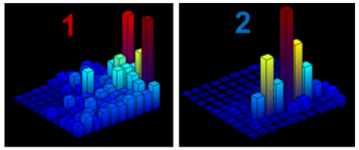
- On-board memory requirements (global land coverage – no compression) – 100 m/ 30 m pixels
  - BRDF heritage –90 GB/ 1.0 TB
  - DEM heritage – 75 GB/ 0.85 TB
  - Scene classification heritage - 22 GB/ 0.25 TB
  - *Total ~190 GB/ ~2.1 TB*
- Memory capabilities should be around 0.4 TB for 6U cubesats and 4 TB for full size spacecraft by 2020.
- ICESat-2 plans to carry multiple databases including global DEM.
  - 30 m to 1 km ground pixel size.

With data compression large on-board heritage databases are feasible

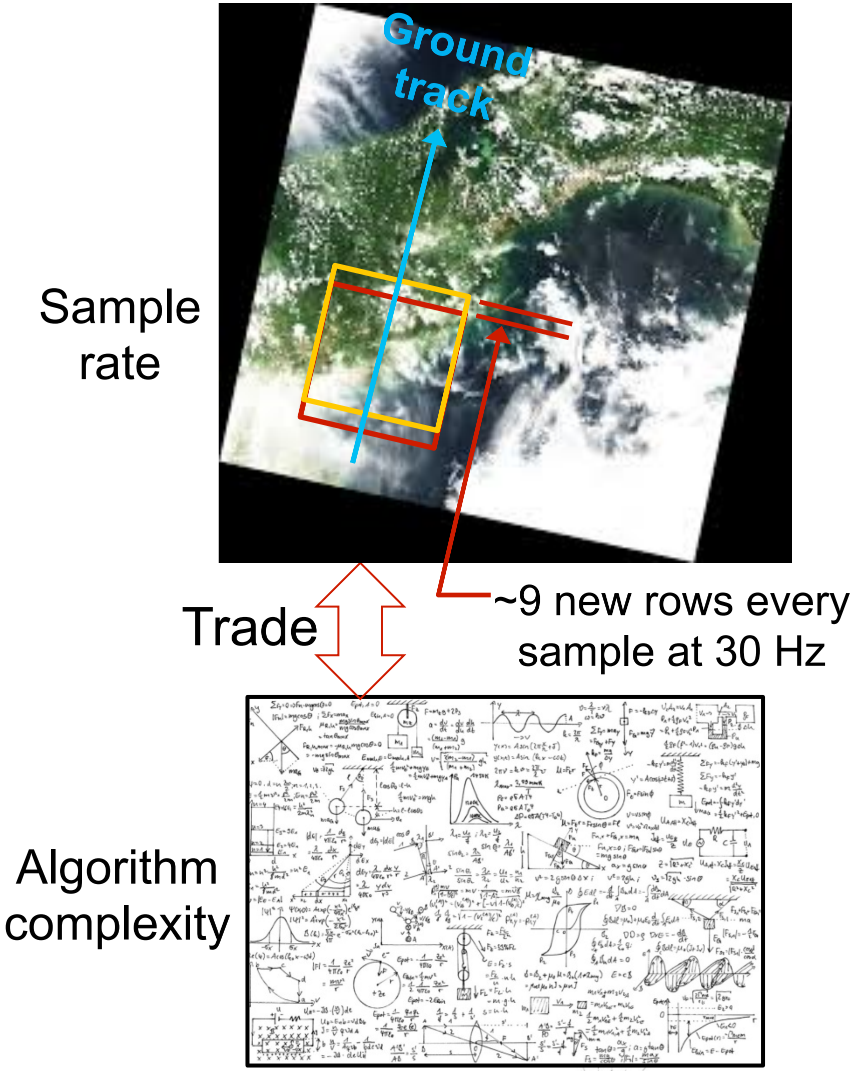


See http://www.cpushack.com/space-craft-cpu.html for years 1975 - 2011.
See SWRI and AIRBUS sites for 2015-2016 data.

Computational Resources

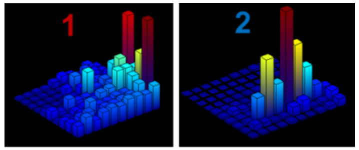


- Camera/ ROIC array size, AOM deflection range, and ground spot pixel size determine computation sample rate.
- With a 256x256 camera/ ROIC, and orbit parameters given, a new scene occurs 1.1 s.
- Sample rate goal set at 30 Hz for lidar and computation rate of FPGA.
  - Trade between computer update rate and algorithm complexity on-going.

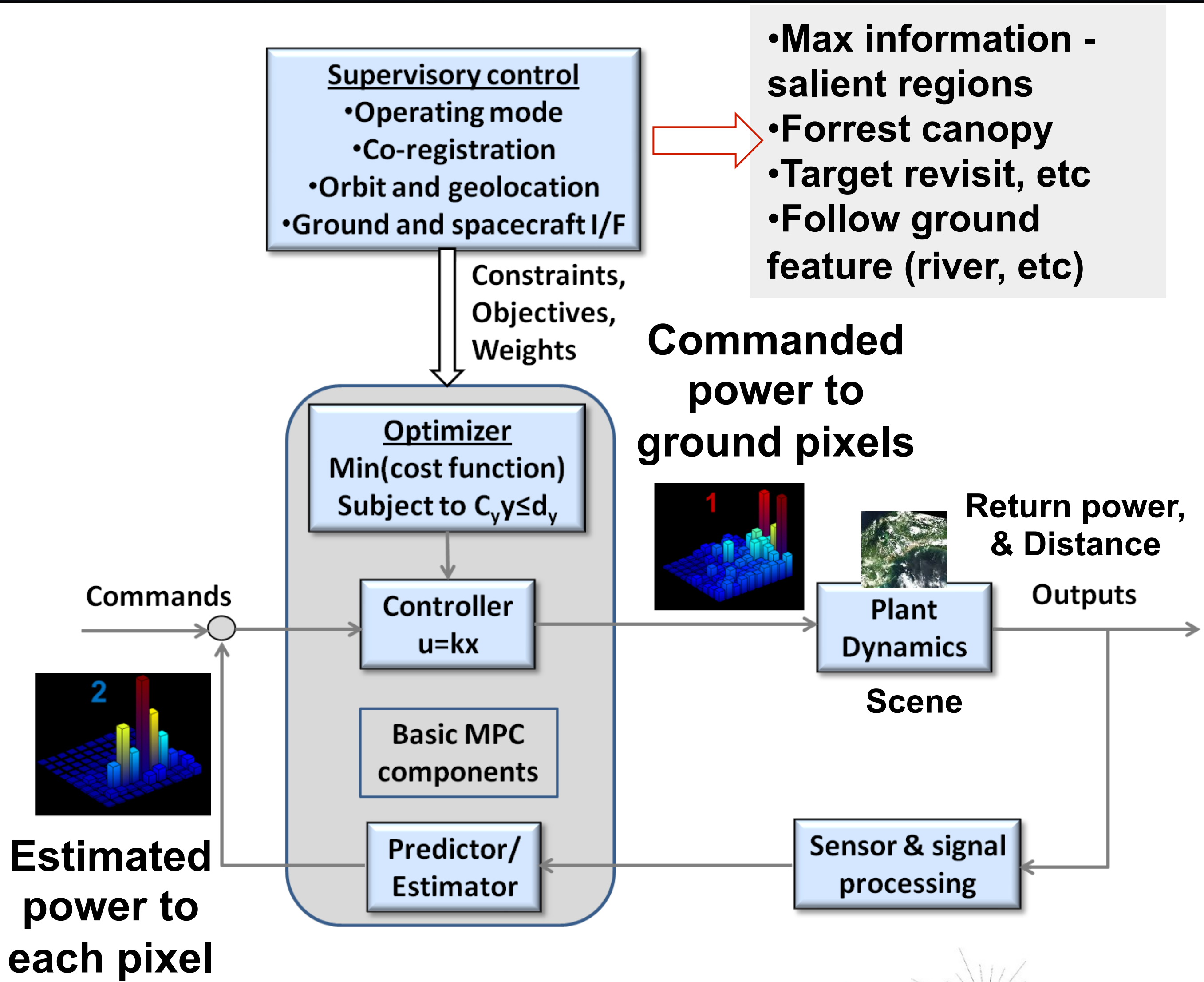


PCAES ARCHITECTURE

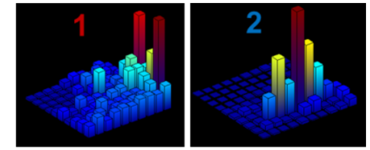
Simple MPC Architecture Diagram



- Two things are controlled – lidar power to ground pixels and dynamic range gate.
- Estimator predicts return power to ROIC receiver.
- Supervisory control passes down weights.
- Optimizer algorithm requires scene classification.



Unusual Characteristics of Our "Plant" Leads to Data Driven Approach



- We have to go to a data-driven control (DDC) approach because there is no conceivable mathematical model for the time-varying scene … well kind of true.
- Data driven approach uses the lidar return data as the model. We take it one step further by doing scene classification (system ID) – reduces model from 256x256 to perhaps 10-20 regions in the FOV.
- Our "actuators" are the lidar beamlets and the sun (daytime operation). We only control one and it only collects sparse returns over FOV (1% type numbers).

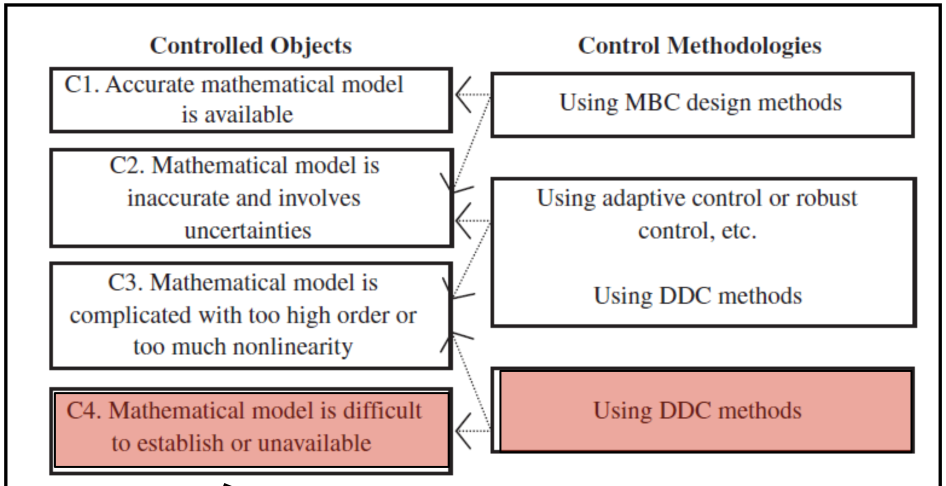
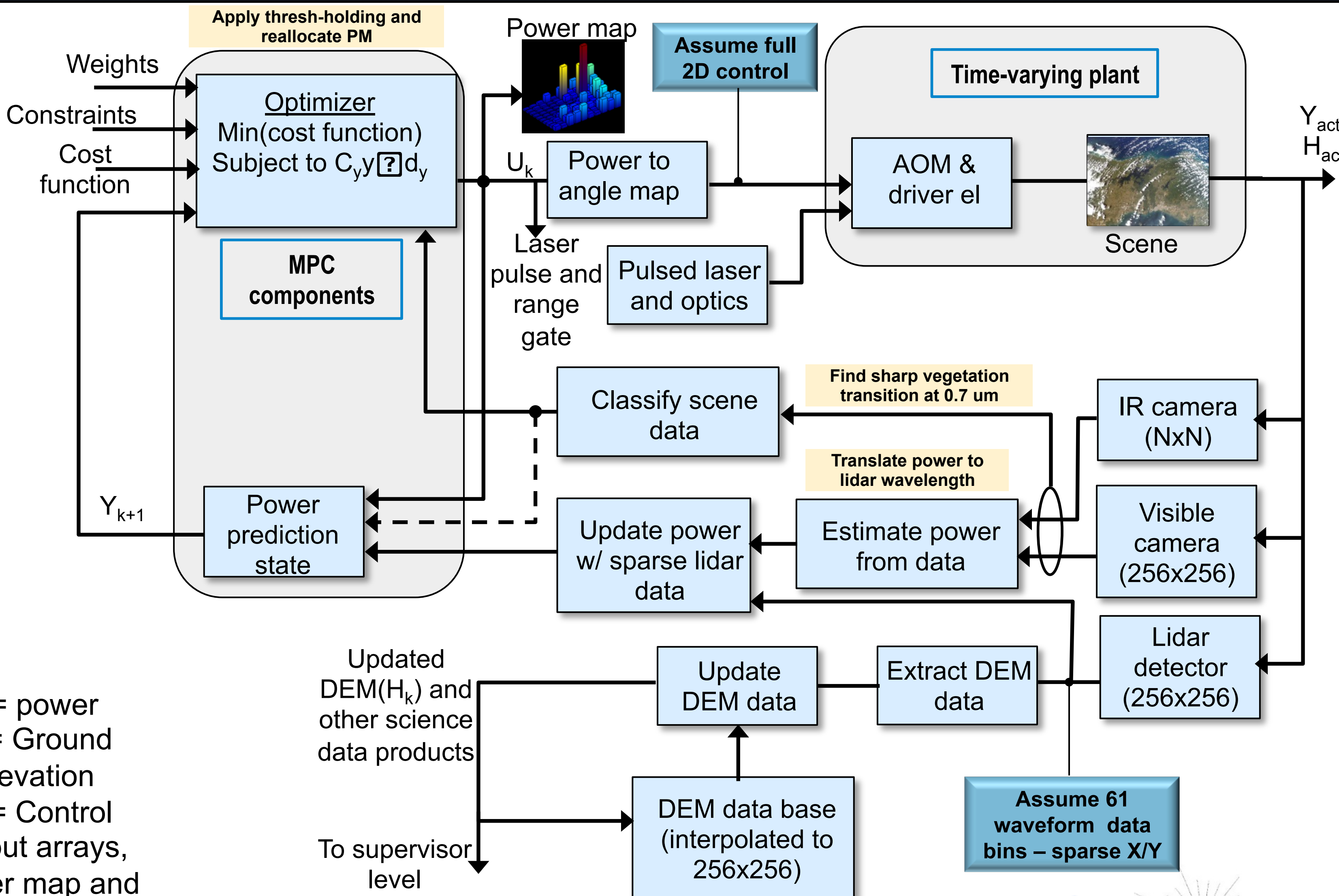


Fig. 3. Controlled objects of DDC.

ESFL MPC in this category

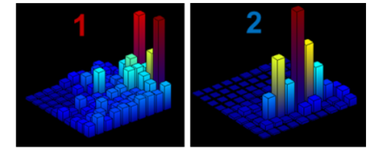
Z.-S. Hou, "From model-based control to data-driven control: Survey, classification and perspective", Inf. Sci., 235 (2013) 3-35.

Finer Definition of MPC Architecture – Better Understanding of MPC Loop and Data Flow

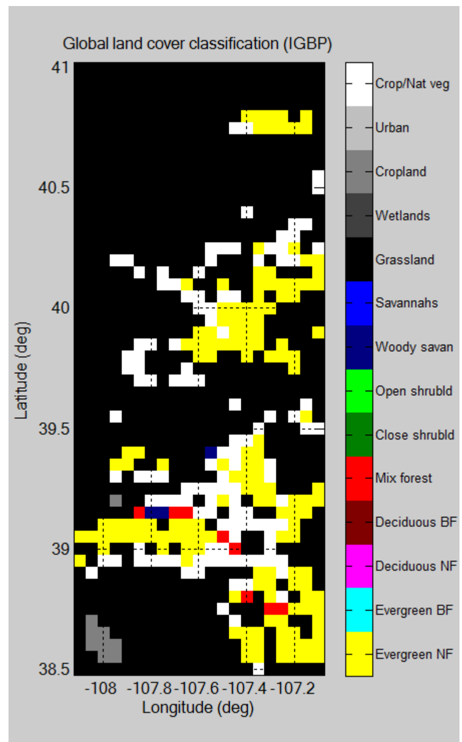


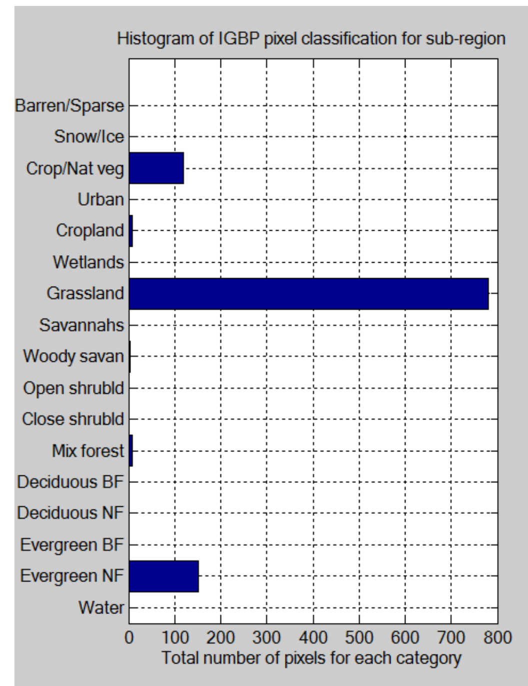
Y= power
H = Ground elevation
U = Control output arrays, power map and range gate

PCAES End-to-End Model



- **PCAES driver file:**
  - Reads in all data bases selected – if global, pulls out strip.
    - Reflectance data for selected wavelengths
    - MODIS reference classification
  - Interfaces with RT scene classification algorithm (in process)
  - Runs reflected power estimation algorithms
  - Inputs data from lidar radiometric math model.
  - Interfaces with optimization routine (in process)
  - Generates large variety of output plots

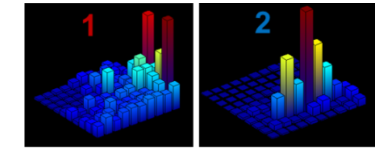




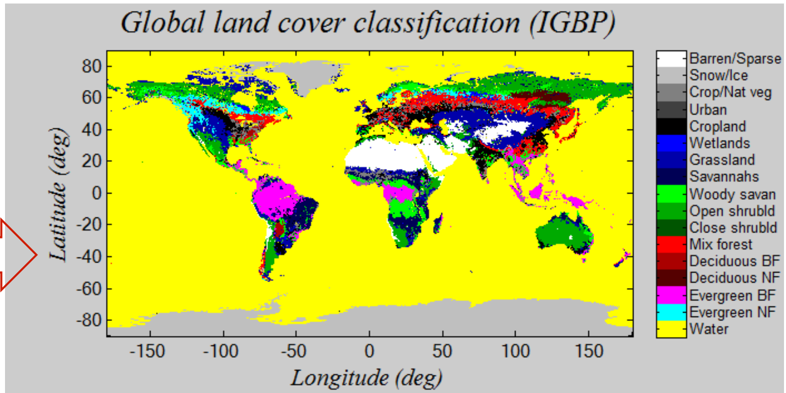
SCENE GENERATION

- MODEL AND LAB TESTING

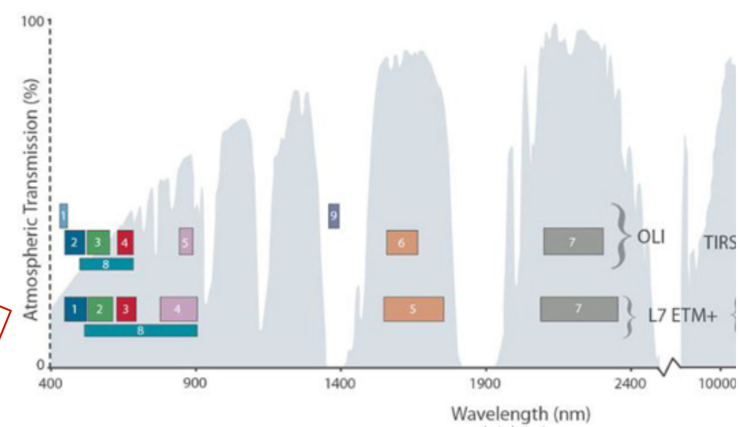
Abundance of Real Satellite Scenery Provided Best Path Forward (1/3)



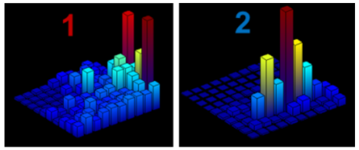
- **MODIS data products:**
  - Global BRDF and scene classification – 6.5 km pixels
    - Matlab SW extracts test strips and can pull 7 spectral bands
    - IGBP classification – 17 categories and we add in clouds.
  - US data base at 500 m and 1000 m pixels.
  - Downloaded from Earth Explorer.
    - Level 2 and 3 data – clouds removed and atmospheric corrected BRDF.
  - Also pulled in level 1b data that was not processed for clouds & aerosols.



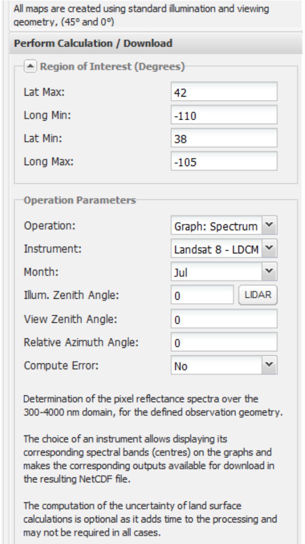
- **Landsat data base:**
  - 30 m pixel size data matches camera pixels
  - Variable amount of clouds.
  - Multiple spectral bands useful for classification

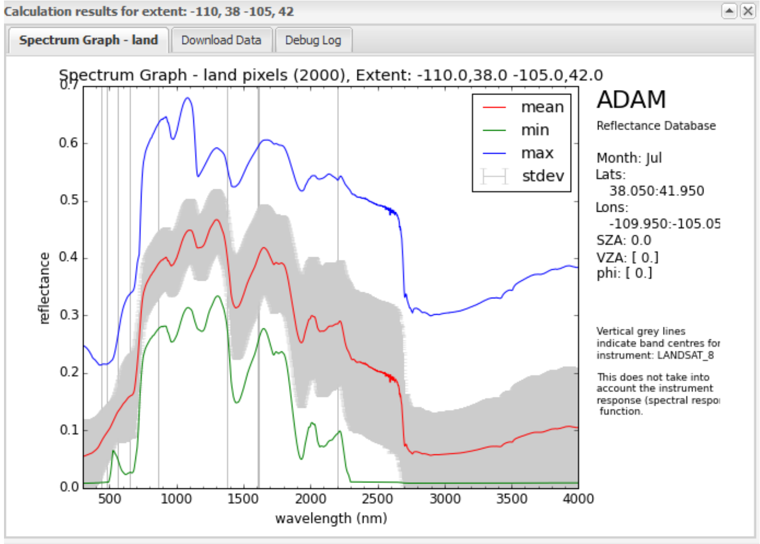


Abundance of Real Satellite Scenery Provided Best Path Forward (2/3)



- **ADAMS (A surface reflectance Database for ESA's earth observation Missions)**
  - **Provided reflectance (BRDF) data for global area but with 11 km pixels.**
  - **Generated variance and BRDF shape information**
  - **Allowed us to look at reflectance and sensitivity effects**

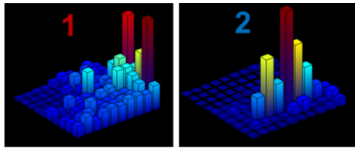




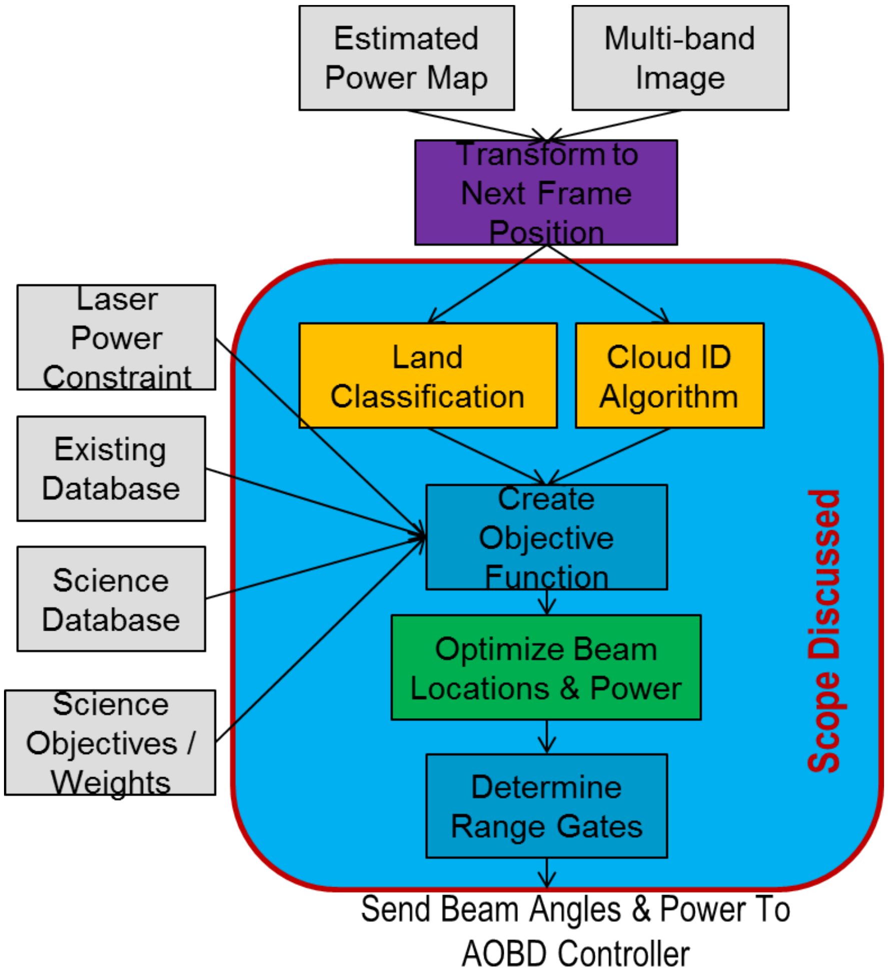
- **Issues:**
  - **Data is not always compatible – different projections, pixel scales, units, file types, etc.**

OPTIMIZATION & SCENE CLASSIFICATION

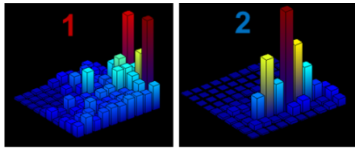
Overview Of Optimization & Classification



- Represents first iteration of optimization with single scene look ahead
- Inputs:
  - Objective function
    - Weights
    - Science objectives
  - Cloud map (by pixel)
  - Desired pointing locations for each science objective
  - Map of where data has been collected already
  - Map of estimated power required for each pixel
  - Constraints
- Outputs:
  - Power map (by pixel) for next frame
  - Range gates



Scene Classification Overview

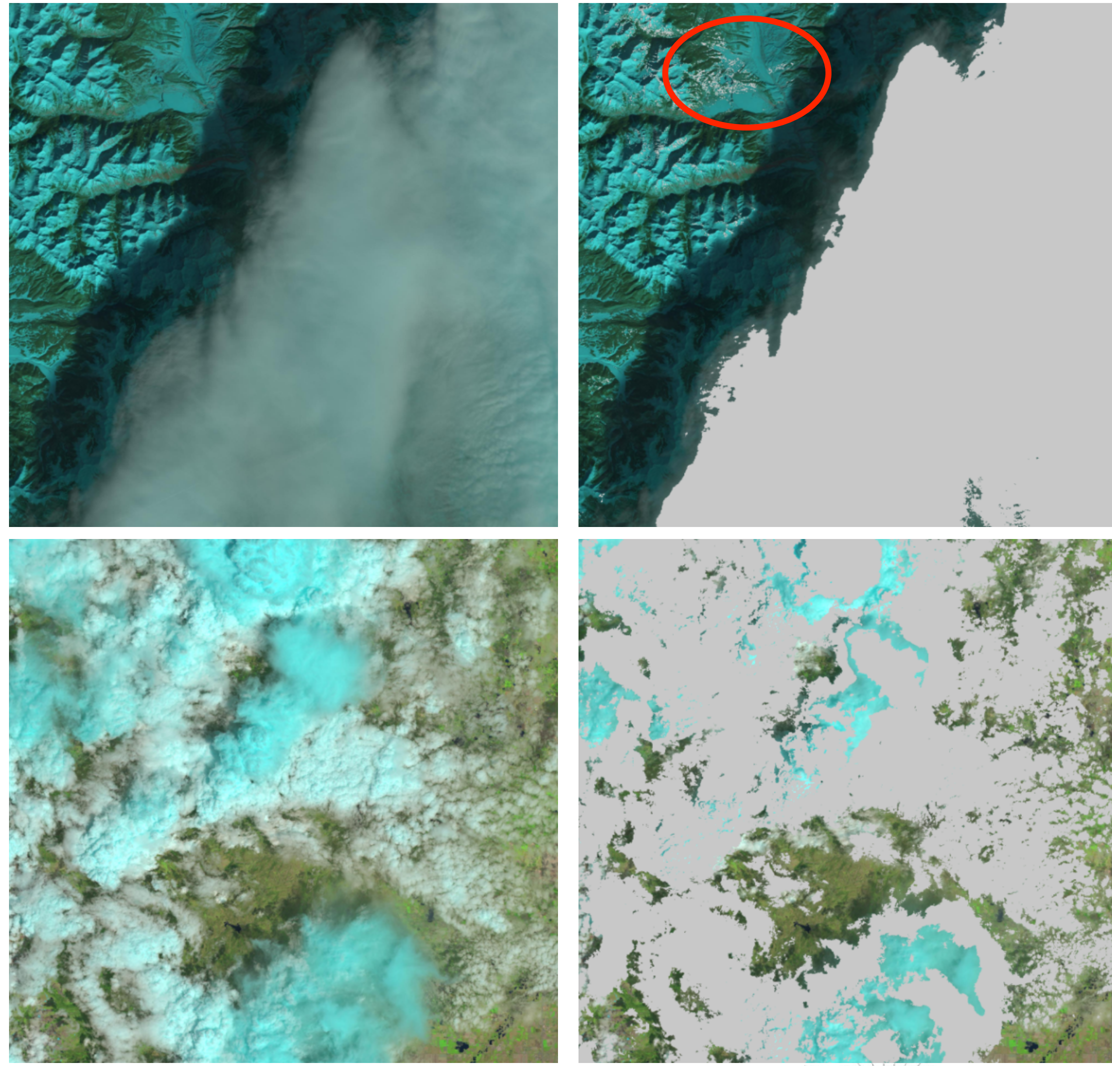


- Primary objective is to avoid clouds
  - Earth is 70% covered with clouds, so avoiding clouds is the biggest gain to be realized
- Secondary objective to identify difference in land type so science can be prioritized

- Attempt to limit the number of bands needed to avoid expensive instruments for the look-ahead camera
  - Current work uses 6 bands from Landsat-8
    - Green and SWIR1 for snow index
    - Blue and cirrus for cloud-vegetation differentiation
    - Red and NIR for vegetation index
  - Can be implemented with less precise instrument for rapid on-orbit classification

Cloud Identification

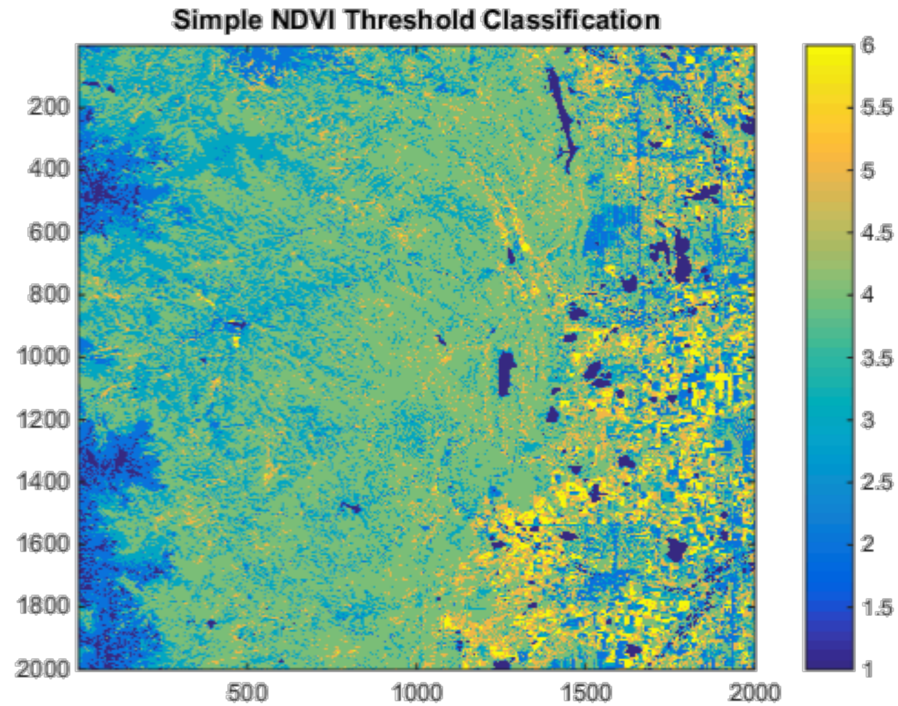
- **Clouds are difficult to separate from snow and ice**
  - Uses NDSI (snow index)
    - Green & SWIR1 bands
  - Uses blue and cirrus bands
    - Ratio separates clouds from vegetation easily
- **Upper images are a cloud bank over Colorado with snow on the mountains**
  - Only small bits of snow are labeled as cloud
- **Lower image is a mix of high and low altitude clouds over Colorado (no snow)**
  - High altitude clouds are frozen, so look like snow if the correct bands are not available
  - Misses at transition from ice clouds to vapor clouds and in wispy icy clouds



Vegetation Classification

- Early work used NDVI thresholds to determine the terrain type
  - Works, but is a bit crude
    - Requires tuning by region
  - Can't differentiate between scrubland and sparse forest
  - Different input images show that NDVI is more related to rain and subsequent growth rate
  - Computationally fast, and only requires two bands
    - Band 4 – Red
    - Band 5 – NIR
- NDVI = (NIR-red)/(NIR+red)
  - Data input in raw DN for my tests to simulate the raw data that would be available for on-orbit processing
- Many papers show improved methods, but rarely produce significantly better results

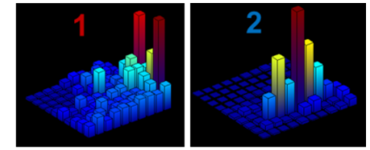
More actively growing vegetation

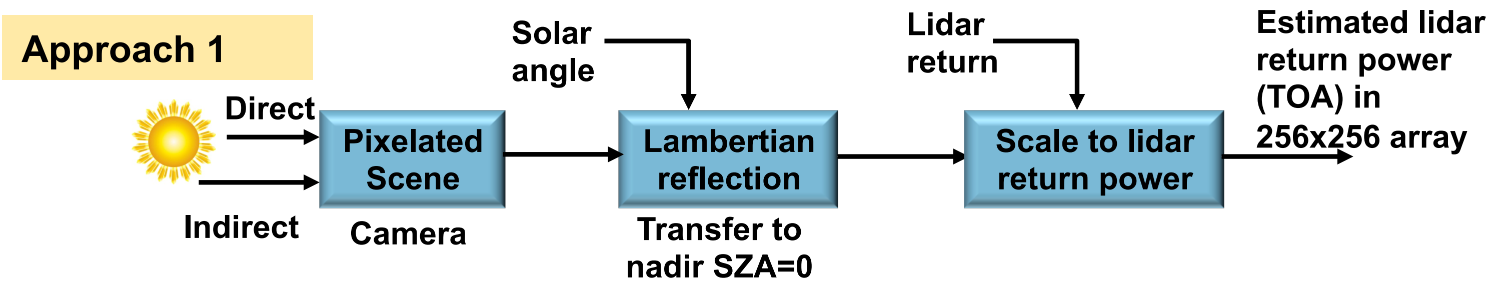


Less actively growing vegetation, water, and concrete/ asphalt

LIDAR RETURN POWER ESTIMATION

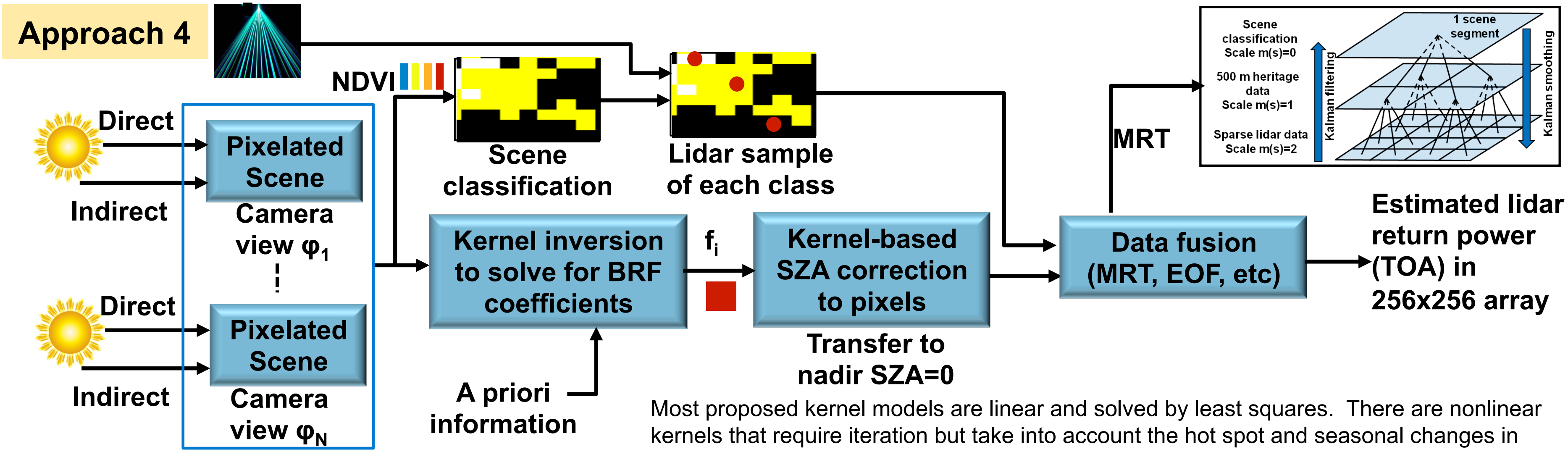
Approaches to Power Estimation (1/4)





ID	Description	Attributes	Issues
1	Estimates all pixels in camera array, lidar spectral band, reflection described by Lambertian function. Correction for the angle between VZA and SZA, and a single lidar return scale factor applied.	Very simple computationally. Does not use any lidar cycles to calibrate specific classifications as all the other approaches do but calibrates from science data already taken in lidar FOV. Updates all pixel values.	Assumption of Lambertian deviates as solar angle increases - BRDF not Lambertian function. No heritage or a priori data incorporated and doesn't improve collection efficiency over time. No path to update scene classification errors. Works poorly for high value data – mountainous regions and urban areas.

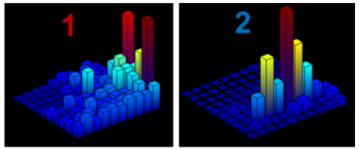
Approaches to Power Estimation (4/4)



Most proposed kernel models are linear and solved by least squares. There are nonlinear kernels that require iteration but take into account the hot spot and seasonal changes in the BRF by incorporating directly the NDVI using two bands [Latifovic2003]

ID	Description	Attributes	Issues
4	Same as approach 3 but uses multiple view camera so that there is enough data to compute kernel inversion equation to get actual kernel coefficients.	All above advantages as in 3, but also generates real-time BRF kernel coefficients which automatically incorporate aerosols and clouds. Can use BRF kernel with extra term to account for SZA=VZA effect (hotspot).	Large computational framework needs to be studied to see if it fits within FPGA capability. Requires additional global on-board heritage data bases. Inversion instabilities possible with poor or noisy data but this is usually stabilized with a priori information. Requires a camera with large multi-angle look-ahead capability .

Computing Predicted Power Output Using BRDF



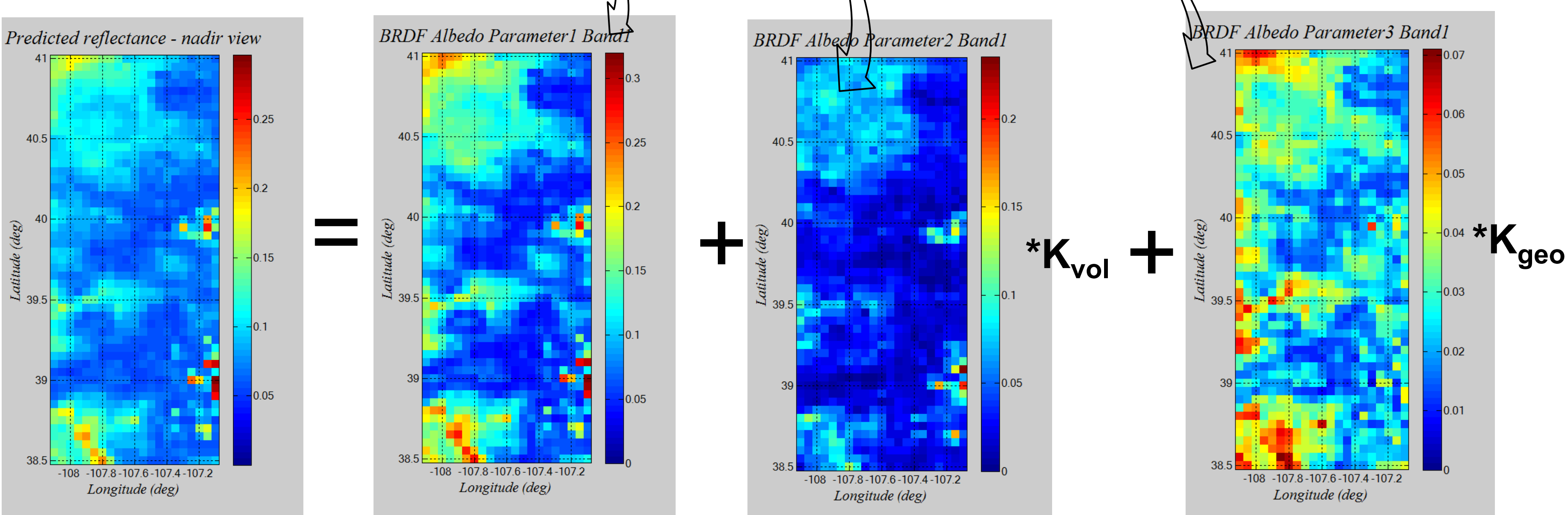
- **The equation below computes the BRF from three terms:**

$$R(\theta,\vartheta,\phi,\Lambda) = f_{iso}(\Lambda) + f_{vol}(\Lambda)K_{vol}(\Lambda) + f_{geo}(\Lambda)K_{geo}(\Lambda) \textit{ where}$$

K_{vol}, K_{geo}, *and* K_{fwd} *are the scattering kernels and the* f_i *terms are the coefficients.*

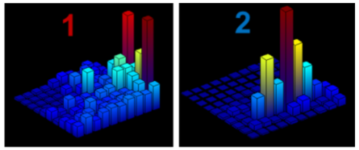
We have assumed kernel functions are constant (scalars) over small FOV.

- **The kernel functions can be pre-computed knowing the solar zenith angle, the view zenith angle and the relative azimuth angle. Below is a plot of selected sub-region with 3 coefficients for 645 nm band.**

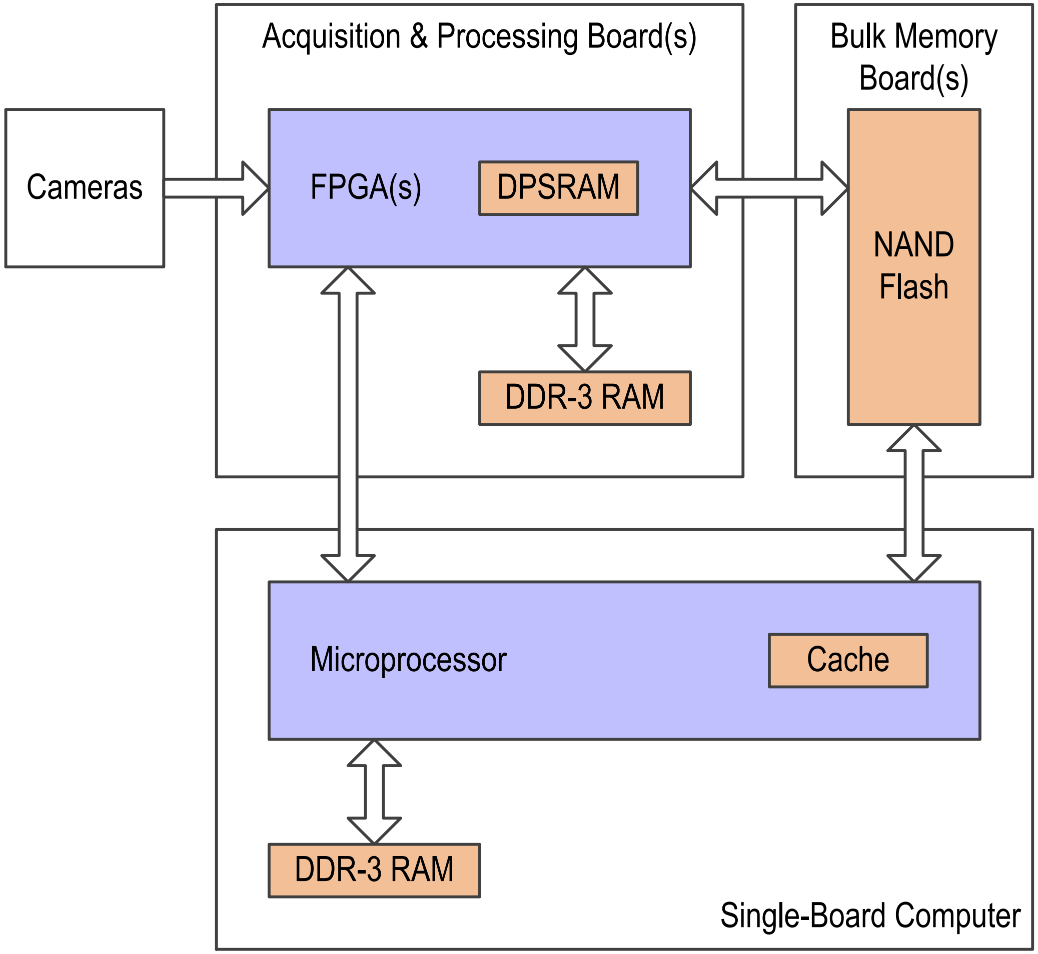


FPGA IMPLEMENTATION

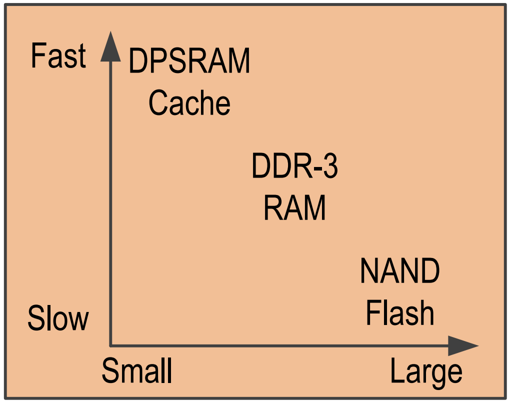
Processing Hardware Architecture



Hardware Architecture



Memory Technologies



Processing Technologies