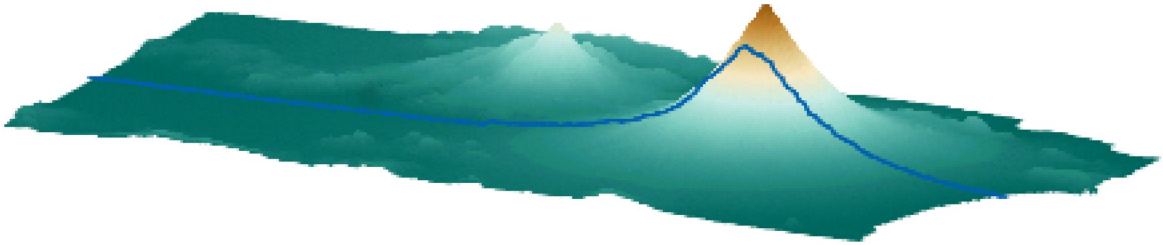
Rugged: an open-source sensor-to-terrain mapping tool

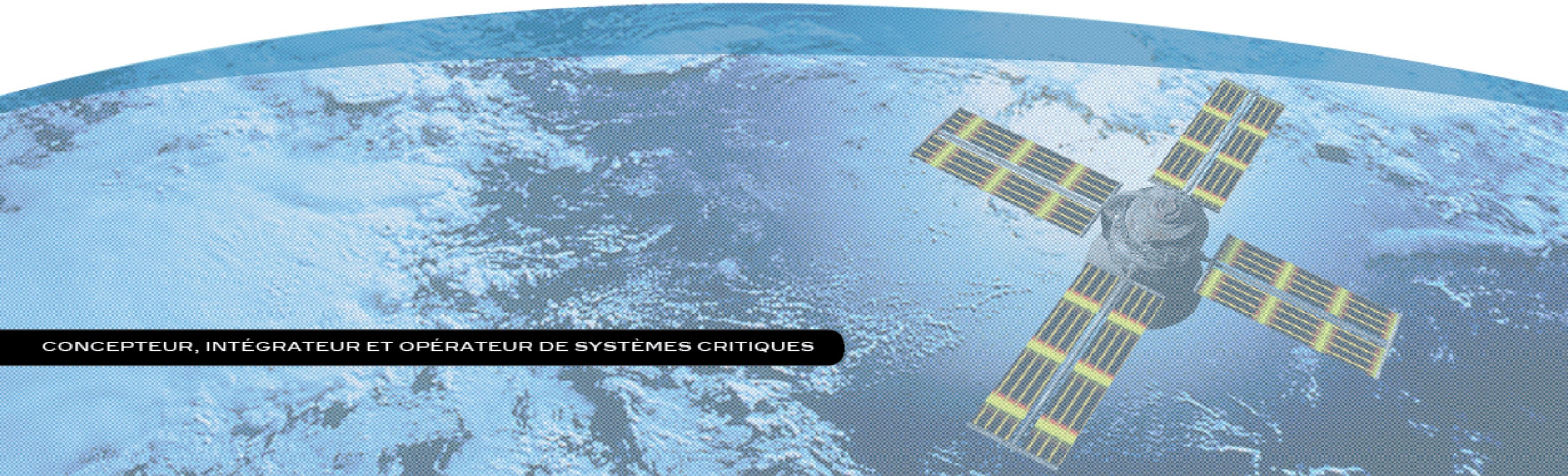
la force de l'innovation

Luc Maisonobe

Aude Espesset

Guylaine Prat





CONCEPTEUR, INTÉGRATEUR ET OPÉRATEUR DE SYSTÈMES CRITIQUES

AGENDA

la force de l'innovation

- Context
- Architecture
- Design
- Error budget
- Status

Why Rugged?

- What was needed
  - Rigorous modelling of sensor's line of sight for precise geolocation and orthorectification
  - Direct location (from sensor to ground)
    - **pixel, line → lat, long, h**
  - Reverse location (from ground to sensor)
    - **lat, long, h → pixel, line**
- What was not needed
  - A new image processing library

Examples of applications

- Orthorectification

  is divided in two main steps :

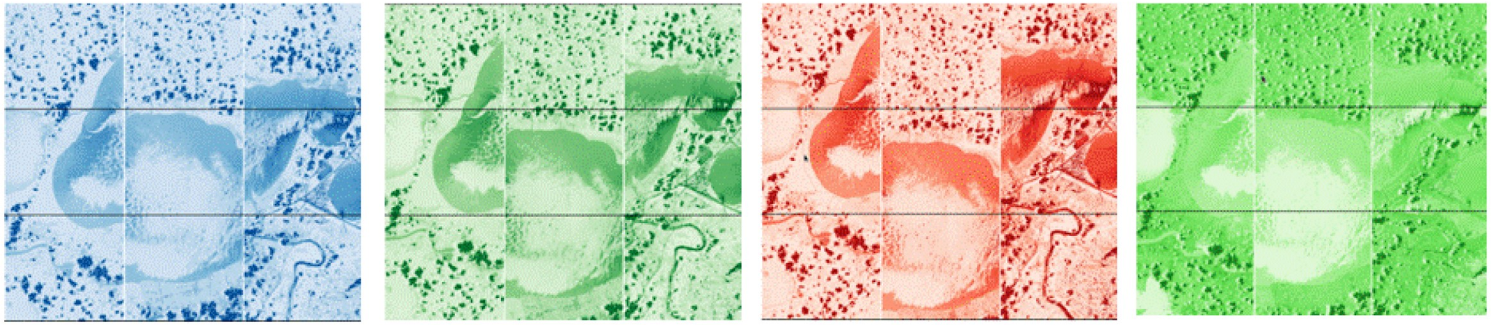
  - Computation of reverse location grid gives the mapping between the ground and the image pixels → based on LOS model, independent of the image
  - Image resampling → pure image processing algorithm (does not need the LOS model)

- Band registration quality control
  - Computation of colocation grids gives the mapping between the image coordinates on one band and the image coordinates on the other band → LOS model
  - Resampling one band on another → pure image processing
  - Correlation between bands to detect misalignment → pure image processing

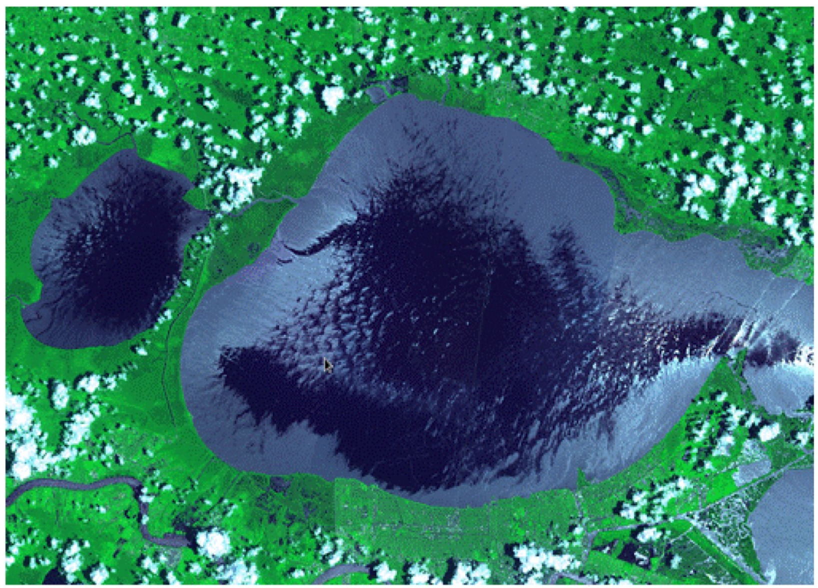
S2 IPF development context

- L1 operational processing chain
    - includes footprint geolocation, orthorectification, band registration, viewing model refinement
- Specific constraints
    - No preprocessing of auxiliary data → need transform from inertial to terrestrial frames
    - Performance
    - Planning → in-house solution preferable

Goal: perform this processing



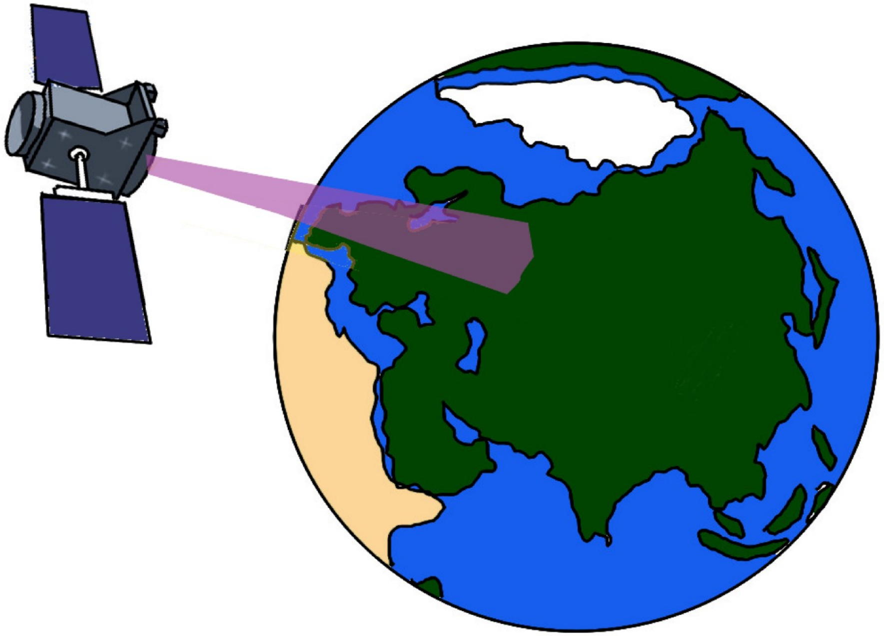
Raw data B02 (490nm) B03 (560 nm) B04 (665 nm) B08 (842 nm)
Vegetation red-edge



L1 product processed with Rugged
Colored composition
Red:B04, Green:B08,
Blue:B02

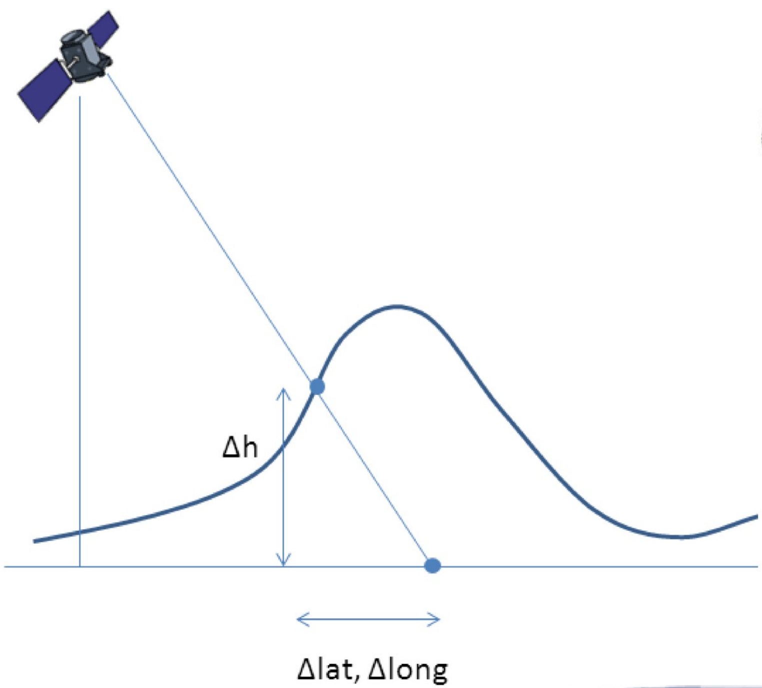
What have we got that works

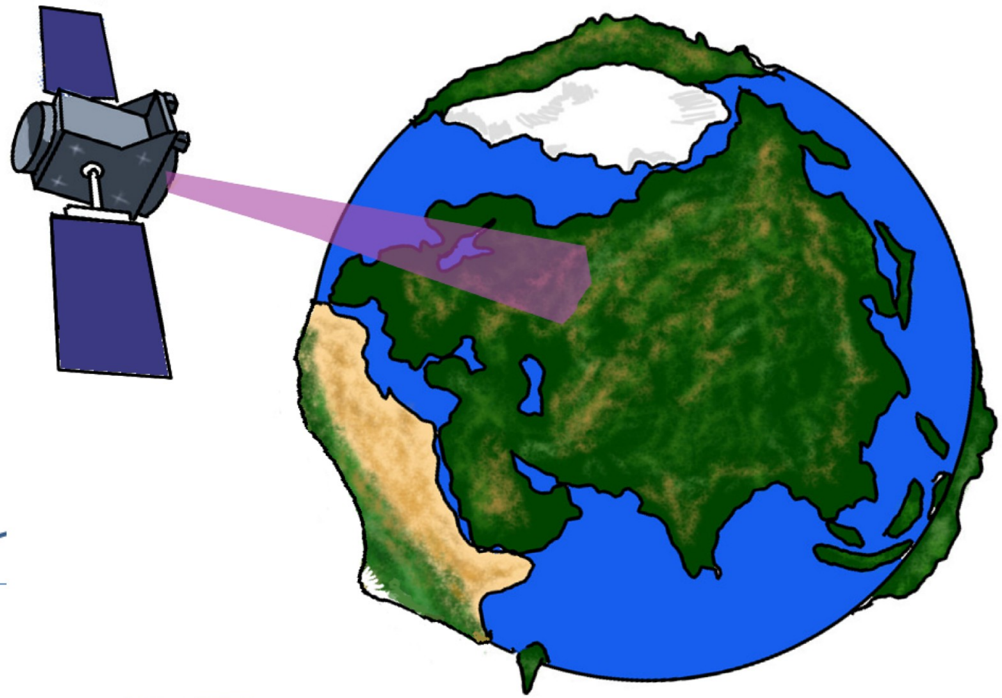
- Open-source flight dynamics library OREKIT, developed by CS
- Open-source mathematical library Apache Commons Math with many contributions by CS
- What can we do with them?
  - Transform between inertial and terrestrial frames
  - Line-of-sight modelling
  - Intersection of the line of sight with the Earth as ellipsoid



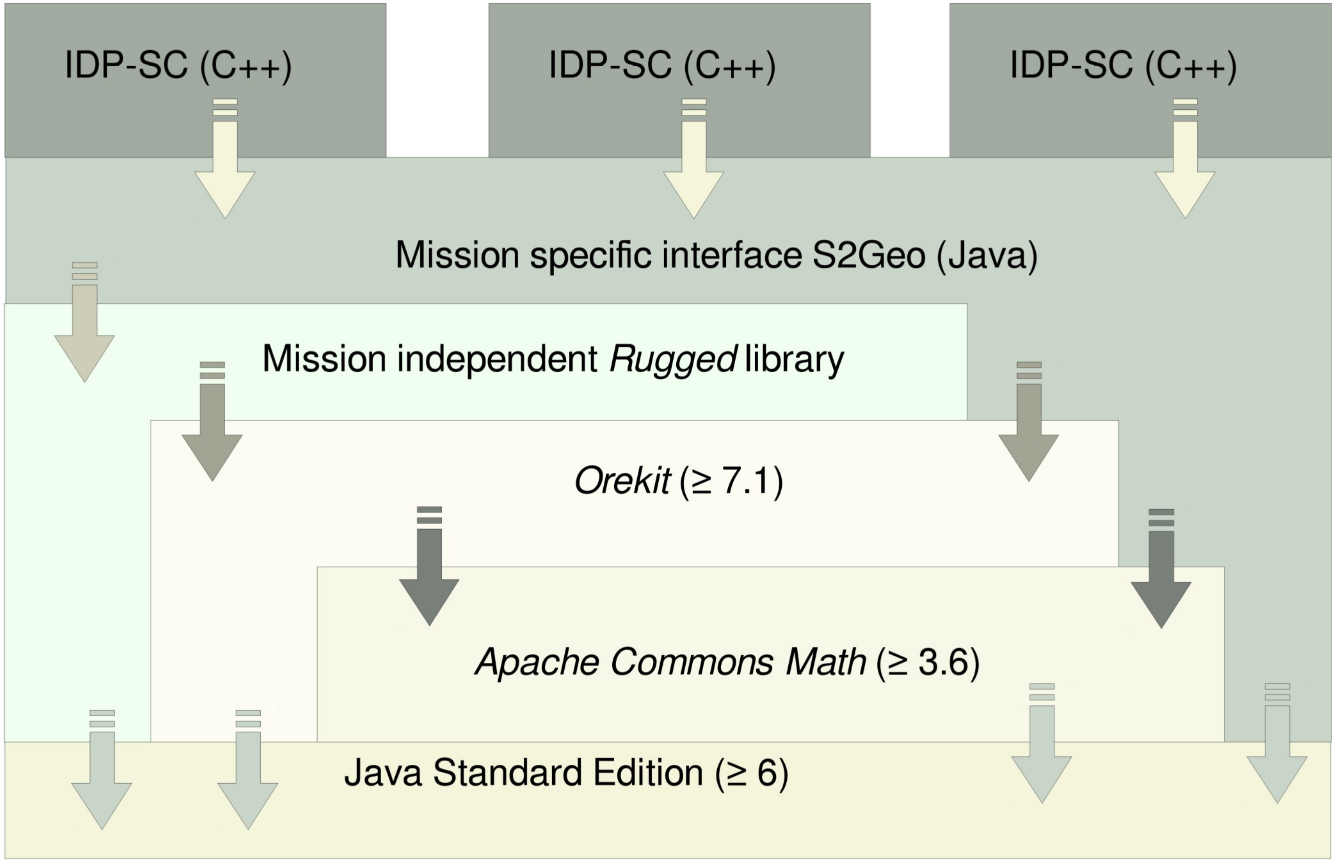
What is missing

- The Earth is not an ellipsoid
- Ignoring the rugged terrain leads to significant distortions in the images
- We must take into account a DEM (digital elevation model)





Rugged overall architecture (Sentinel 2 example)



Rugged functional breakdown (1/2)

la force de l'innovation

Topic	Layer	Comment
Sensor to ground mapping	Rugged	Base provided feature
Ground to sensor mapping	Rugged	Base provided feature
Individual pixels	Rugged	API supports any number of pixels
Optical path	Interface/Rugged	Lower level library provide tools to compute folded path
Line time-stamping	Interface/Rugged	Model defined in interface, Called from Rugged
Orbit/Attitude interpolation	Orekit	Both interpolation and full propagation supported
Orbit/Attitude file parsing	Interface	Could be delegated to Orekit if CCSDS ODM/ADM

Rugged functional breakdown (2/2)

Topic	Layer	Comment
Frames transform	Orekit	Many frames supported, sub-millimeter accuracy
IERS data correction	Orekit	Transparent for users, including $\delta\Delta\varepsilon$ and $\delta\Delta\psi$
Grid-post elevation model	Rugged	Only raster model supported
Triangulated Irregular Network elevation model	Not supported	Must be converted to raster For now
Geoid computation	Orekit	Can be computed for any gravity field
Time-dependent deformations	Interface/Rugged	Model defined in interface, Called from Rugged
DEM file parsing	Interface	DEM is mission-specific

Rugged design choices (1/2)

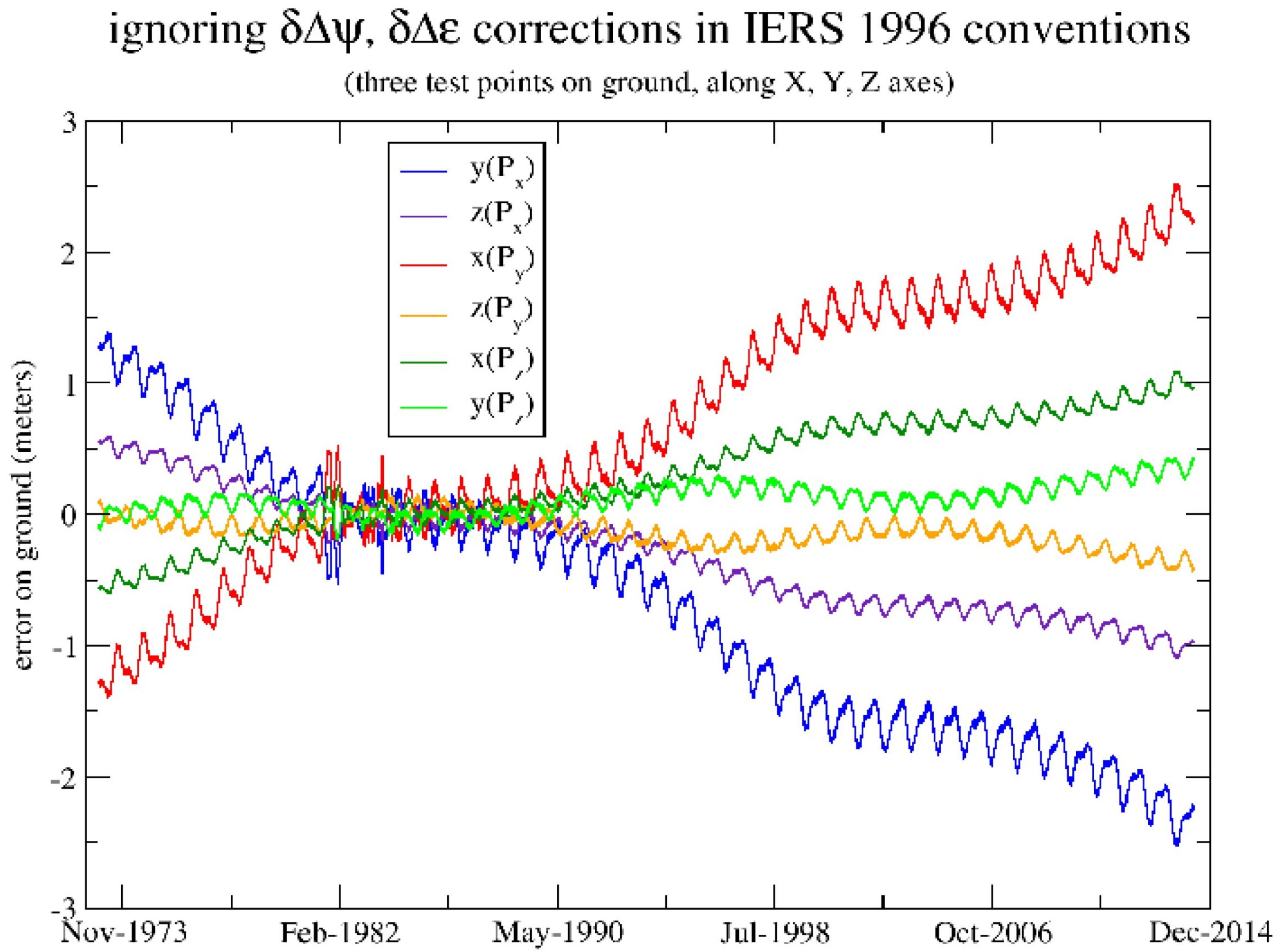
- Full accuracy frames
  - Selectable inertial frame (GCRF, EME2000, MOD, TOD, Veis1950 ...)
  - Selectable Body frame (ITRF, ITRF-equinox, GTOD ...)
  - Selectable ellipsoid (GRS80, WGS84, IERS96, IERS2003 ...)
  - IERS corrections: dtu1, x_p(u), y_p(v), lod, δΔε/δΔψ or dx/dy
- Both interpolation and propagation for orbit/attitude
  - Easier validation, generating our own data
  - Easier integration in the system
  - Don't use linear interpolation of quaternion components
- Fix light travel time for each pixel
- Line-of-sight
  - Linear in Cartesian 3D
  - Non-linear in geodetic coordinates (no flat-body hypothesis)

Rugged design choices (2/2)

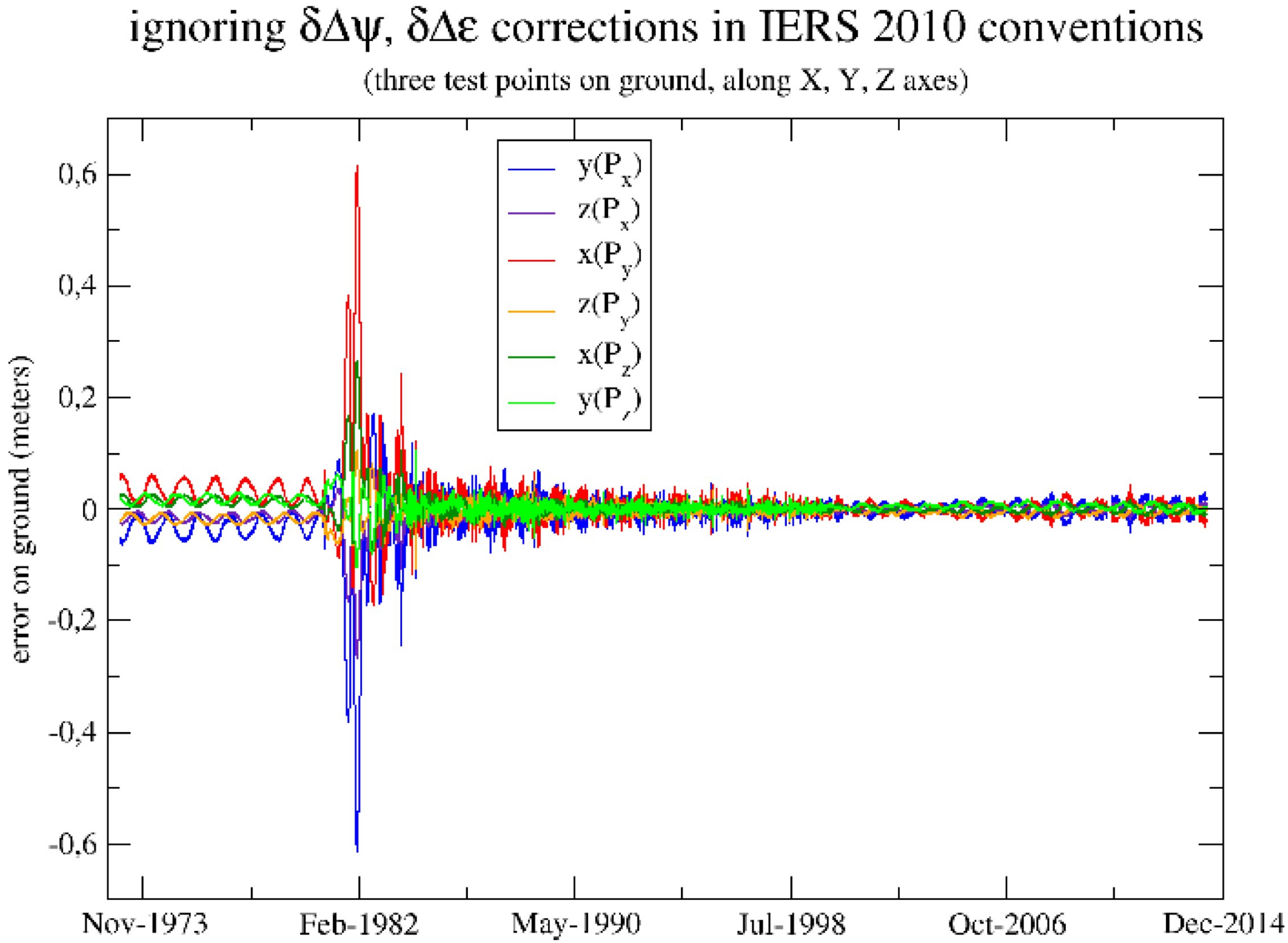
la force de l'innovation

- Selectable Digital Elevation Model Intersection algorithm
  - Baseline using Duvenhage algorithm
  - Basic (slow) scanning algorithm available for validation
  - Algorithm ignoring DEM available for special needs
  - Line-stepping is not in baseline
- Inverse location algorithm
  - Replace multi-dimensional optimizer by 2 solvers
  - Don't put the full direct location in the core
- Precompute costly transforms out of the pixels loop
  - Most important for Earth precession/nutation
  - Still possible to fix light travel time for each pixel
- Not limited to nadir observations
- Not limited to Earth

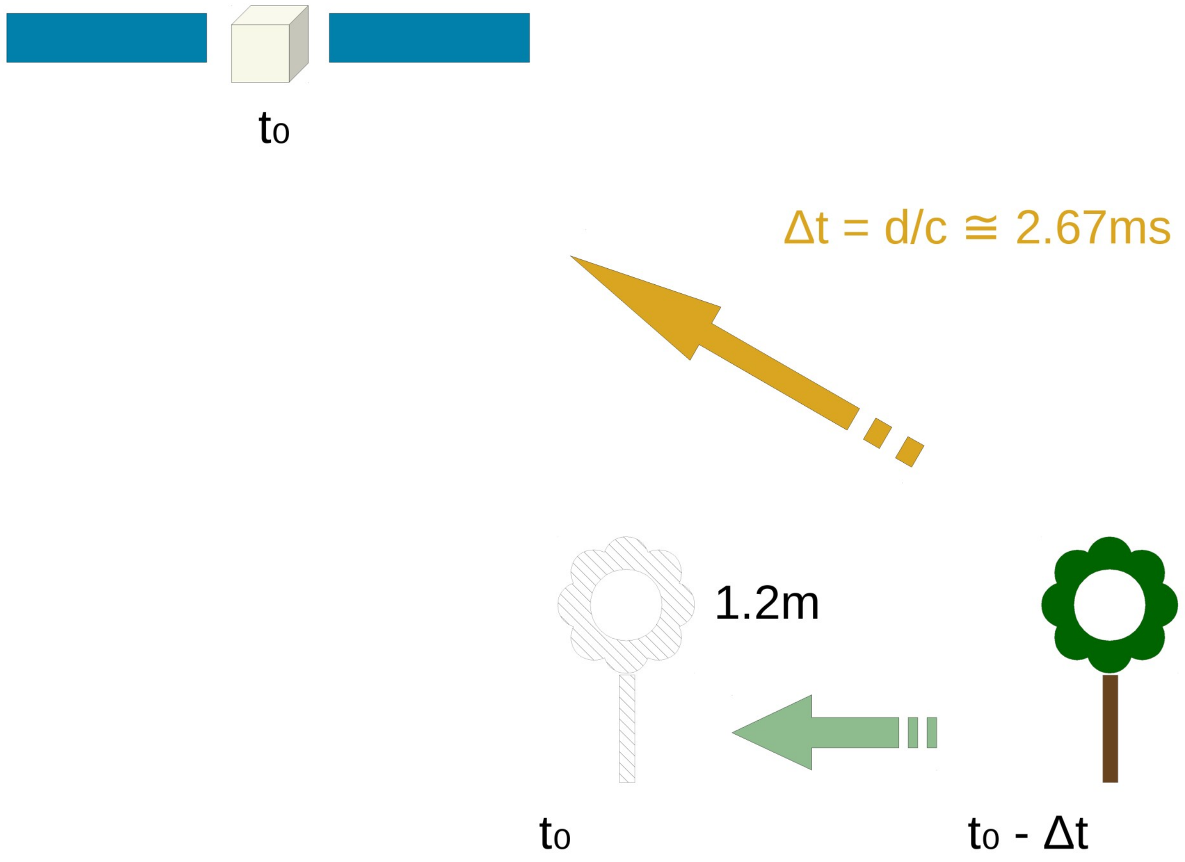
Earth rotation error analysis (1/2)



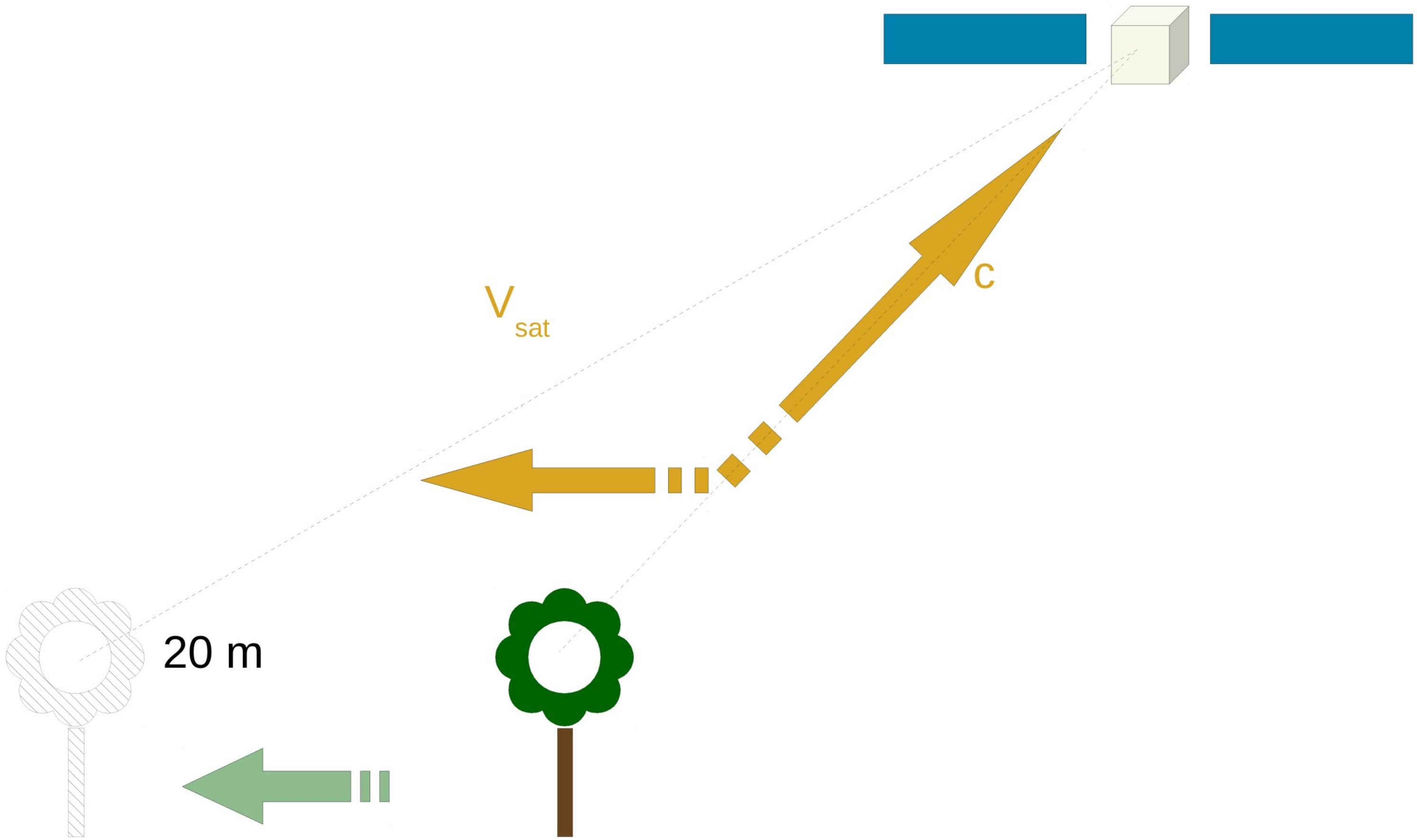
Earth rotation error analysis (2.2)



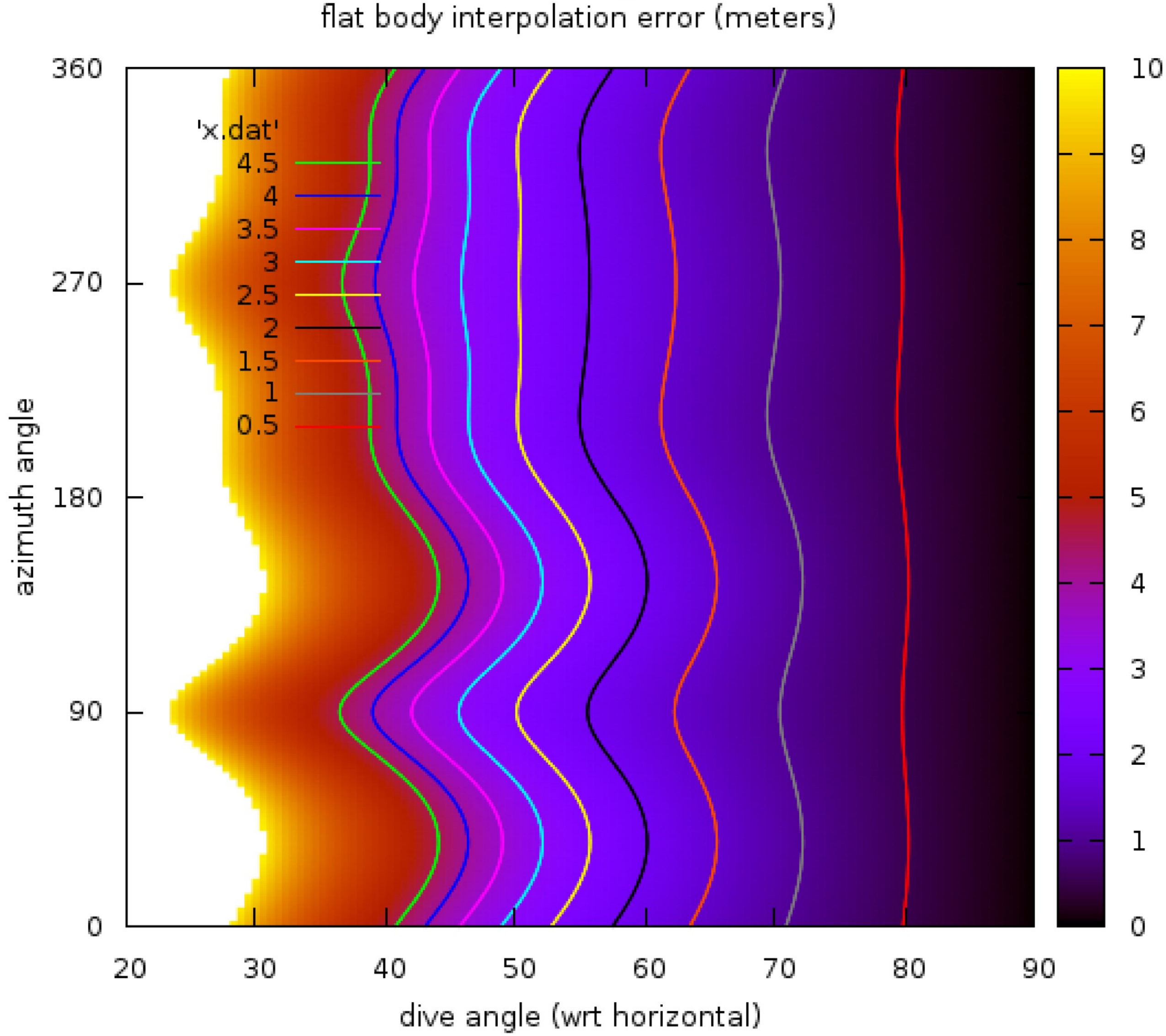
Light-time error analysis



Aberration of light error analysis



Earth curvature error analysis



Error budget

la force de l'innovation

origin	amplitude	location	comment
δΔε and δΔψ corrections for precession and nutation models	> 3m on simulated data, < 1m on GPS data	horizontal shift	up-to-date precession and nutation models are also available
quaternion interpolation	negligible	line-of-sight direction	the effect is important for step sizes above 1 minute
instrument position	1.5m	along track	coupled with attitude
light travel time	1.2m	East-West	pixel-dependent, can be switched off
Aberration of light	20m	along track	velocity dependent, can be switched off
flat-body	0.8m	across line-of-sight	error increases a lot for large fields of view

Rugged current status (2016-03-16)

- Technical aspects
  - Direct/Inverse location at the heart of an operational system
  - Performance benchmarks show good performances
  - First operational success
  - Calibration designed, development initiated
  - Refraction to be added
- Open-source aspects
  - Freely available since 2015-01-12
  - Community to be built, several candidates for PMC
  - Governance under discussion
  - Meritocratic model: user → contributor → committer → PMC member

First Sentinel-2 image (June 2015)