Satellite Derived Bathymetry Primer

Shallow Survey 2018
St. John's, NL, Canada, October 1-3

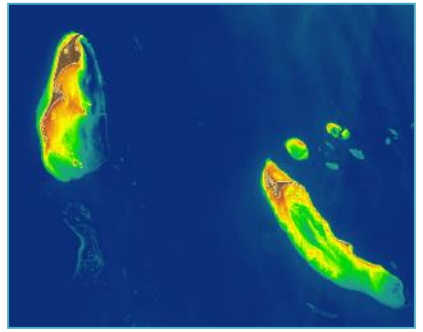
Véronique Jégat
Manager Bathymetric Projects, IIC Technologies Inc.

Patrick Salembier
GIS developer, IIC Technologies Inc.

Sam Lieff
Planet Labs Geomatics Corp.

IIC Technologies and Satellite Derived Bathymetry

- Reconnaissance tool for bathymetric survey planning
- Capacity Building & Training (IIC Academy)
- Chart adequacy analysis
- On-demand local SDB extraction



Reconnaissance

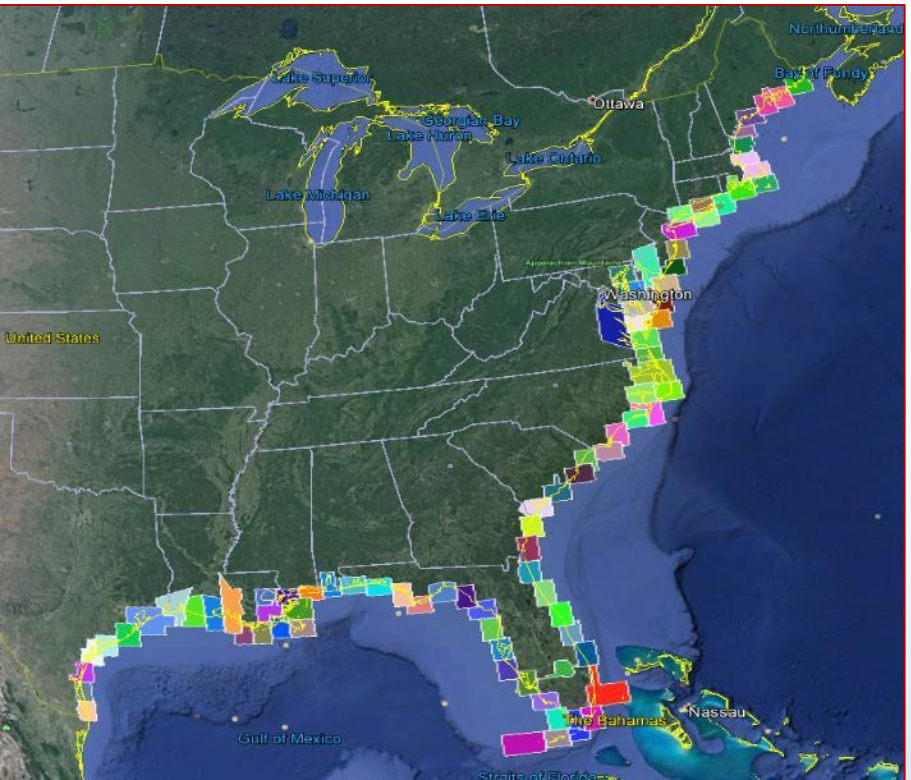
FIG/IHO/ICA S-5 (Hydrographic Surveys) Cat B & S-8 (Nautical Cartography) Cat B

Chart adequacy and production

Examples of SDB projects

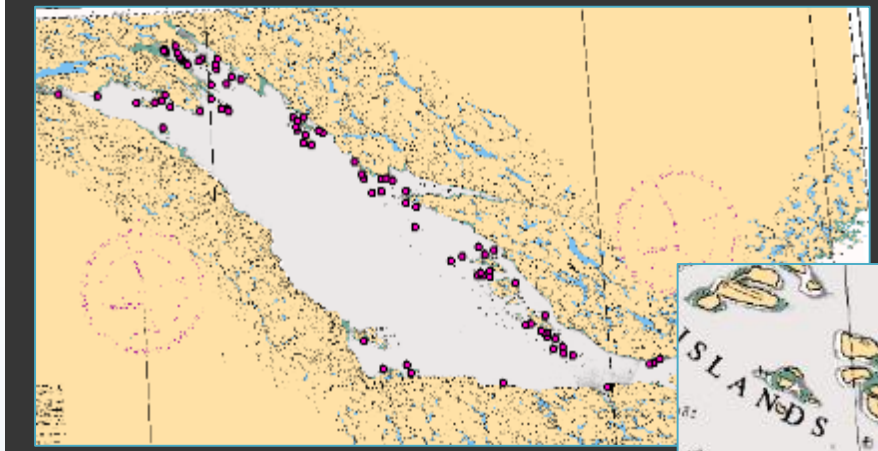
Chart Adequacy

- NOAA USA East and Gulf Coast ENCs
- SDB Extraction (LiDAR benchmark)

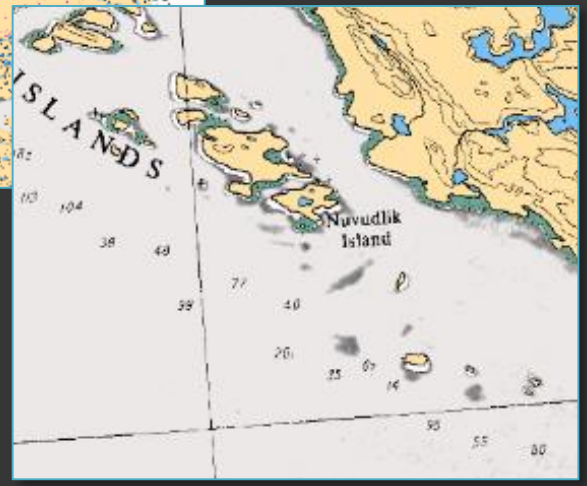


Shoals detection in Arctic waters

- Wager Bay, Nunavut, Canada

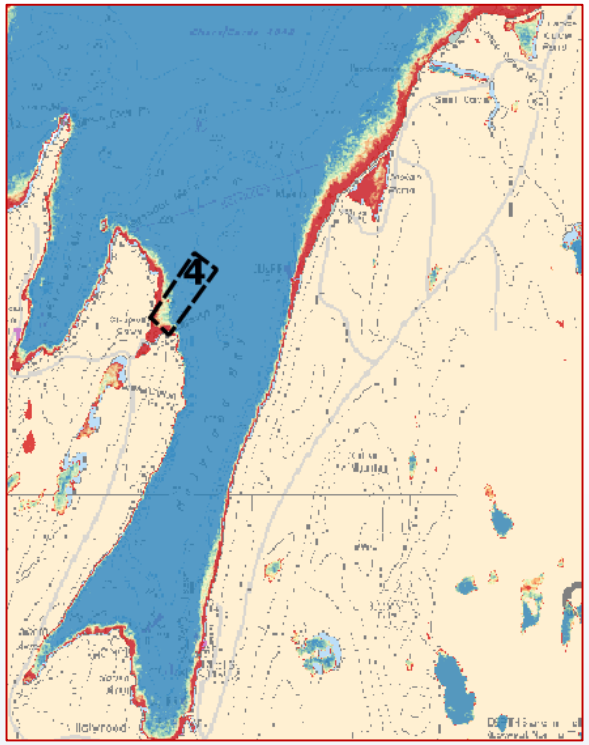


Digitized Shoals



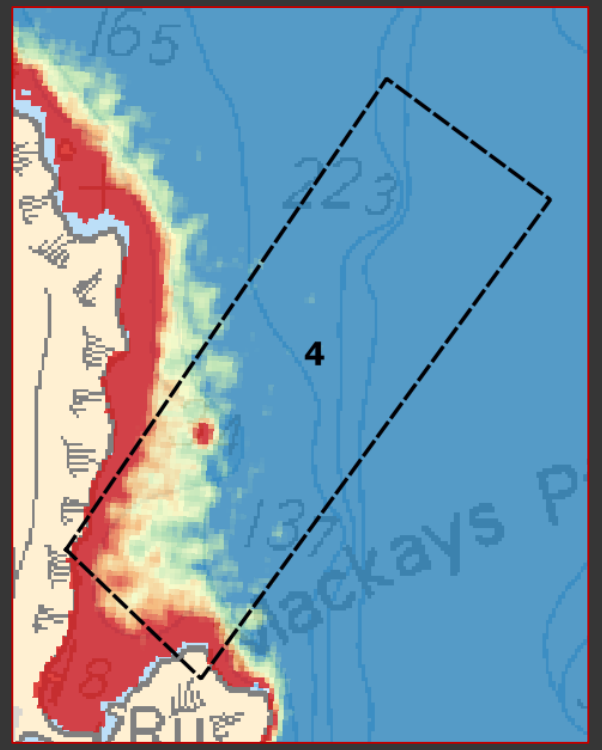
SDB Extract

Proof of Concept: Simple Approach



Sentinel-2

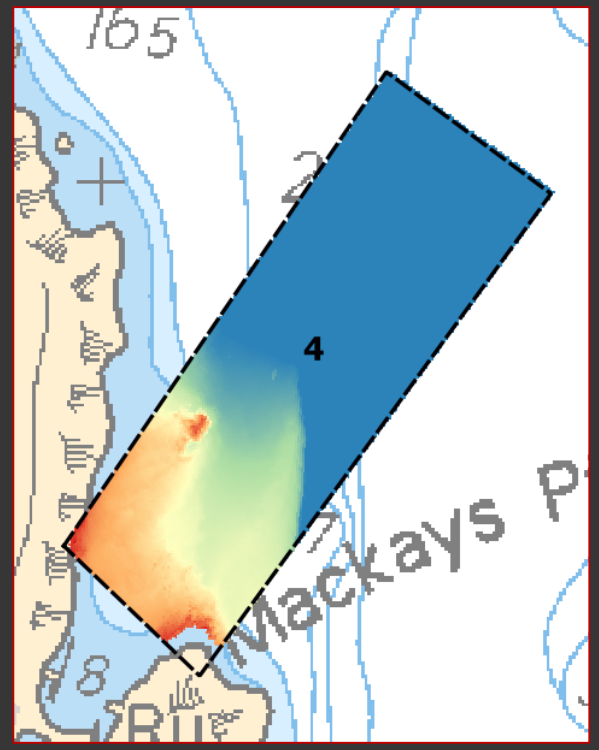
Close-up



Sentinel-2

2018-Apr-17

Not vertically referenced



EM2040

2017-Sep-05

Shallow survey 2018 dataset

Overview

Step 1

- ❖ Select Image
- ❖ Download

- ✓ Simplify using Planet Explorer online tool
- ✓ Access wide source of images

Step 2

- ❖ SDB extraction algorithm

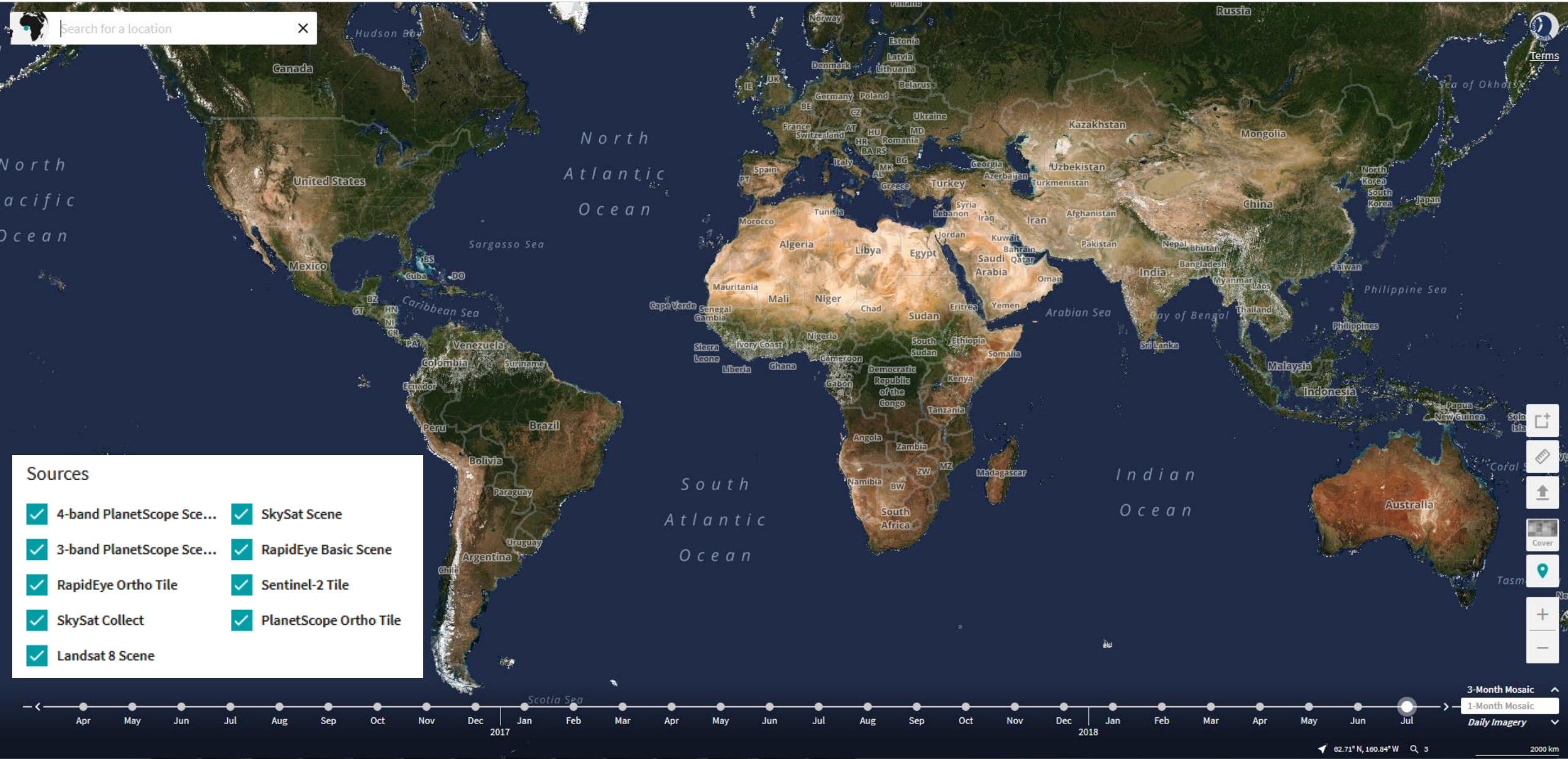
- ✓ Code in Python and implement in open source QGIS
- ✓ Use Planet API to combine Step 1 and 2

Step 3

- ❖ Compare results in GIS
- ❖ Assess best dates and sources for SDB

Planet Explorer

IIC TECHNOLOGIES

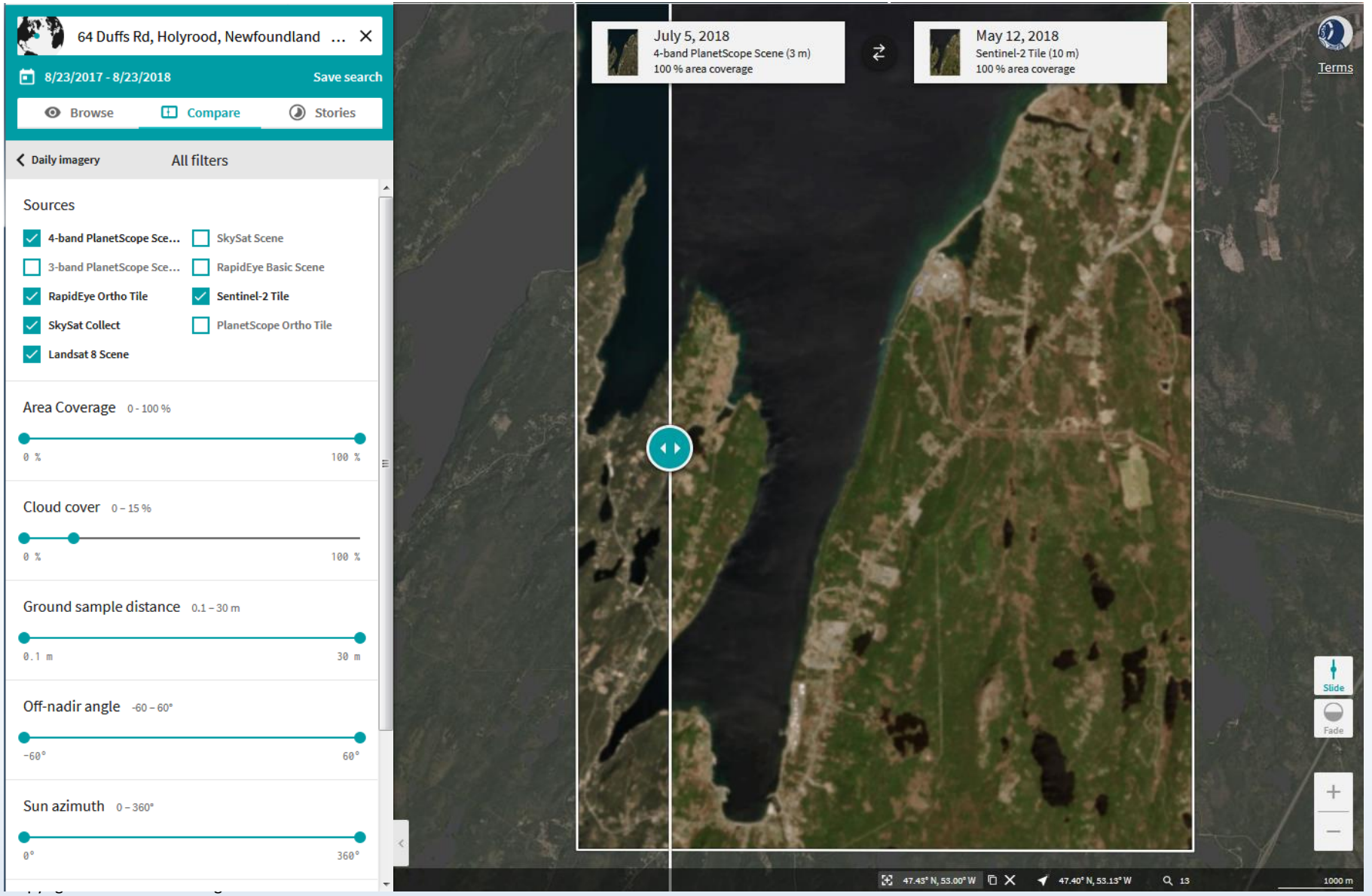


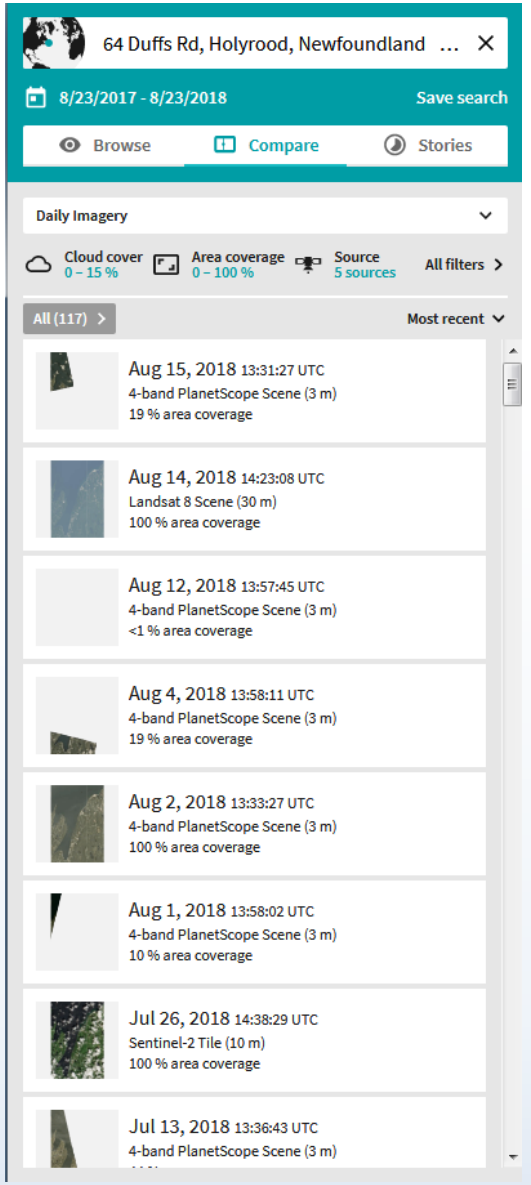
Satellite sources

Term	LandSat 8	Sentinel 2	Planet Scope	Rapid Eye	Sky Sat
Cadence	~ Twice monthly	~Weekly	Daily	~Weekly	Weekly Tasked
Approx. pixel size	~30m	~10m	~3m	5m	~0.9m
Number of bands	9: Aerosol + RGB + NIR +2 SWIR + Pan + Cirrus	12: RGB + 3 Red Edge + NIR + 3 SWR + Vapor + Aerosol	4: RGB + NIR	5: RGB + Red Edge + NIR	5: RGB + Pan + NIR

Source: Planet Labs Geomatics

Filters and temporal comparison





Scripting approach

Version 1

- QGIS:
  - Open background BSB
  - Define AOI
  - PY script:
    - Operator inputs
    - Search images (Planet API)
    - Download image (Planet API)
    - SDB algorithm
  - Inspect results

Version 2

- Planet Explorer:
  - Define AOI
  - Set Filters
  - Compare and select image (obtain Planet ID)
- QGIS:
  - PY script :
    - Download using Planet ID (Planet API)
    - SDB algorithm
  - Inspect results

QGIS scripting

IIC TECHNOLOGIES

QGIS

Version 3.2

Earlier version may not work

Set-up

Import libraries

SDB_script.py

```
#import standard Qgis libs
from qgis.core import *
from qgis.PyQt.QtCore import *
from qgis.PyQt.QtGui import *
#import lib for processing
import processing
#lib for debugging
import sys
import os

#this code is for Planet API integration
```

QGIS scripting

IIC TECHNOLOGIES

Smoothing

Saga simple filters (optional)

```
#Run the simplefilter saga algo for B2, and add to the project
Blue_smooth_output = processing.run('saga:simplefilter', {'INPUT': B2Layer, 'MODE':0, 'METHOD':0,'RAD
Blue_smooth_path = Blue_smooth_output['RESULT']
fileInfo = QFileInfo(Blue_smooth_path)
baseName = fileInfo.baseName()
Blue_smooth = QgsRasterLayer(Blue_smooth_path, baseName)
#iface.addRasterLayer(Blue_smooth_path, "Blue_smooth")

#run the simplefilter saga algo for B3, and add to the project
Green_smooth_output = processing.run('saga:simplefilter', {'INPUT': B3Layer, 'MODE':0, 'METHOD':0,'RA
Green_smooth_path = Green_smooth_output['RESULT']
fileInfo = QFileInfo(Green_smooth_path)
baseName = fileInfo.baseName()
Green_smooth = QgsRasterLayer(Green_smooth_path, baseName)
```

Compute Algorithm

QGIS Raster Calculator
Land/Water separation based on NIR band (optional)

```
#Doing the raster valc with qgis method
#first, set the layer names with band nums attached and expression to be used
B6Name=B6Layer.name() + '@1'
Blue_smoothName = Blue_smooth.name() + '@1'
Green_smoothName = Green_smooth.name() + '@1'
expression = '(\"'+B6Name+'\" < '+str(LWthreshold)+') * ln(\"'+Blue_smoothName+'\")/ln(\"'+Green_smoothN
LANDSAT_SDB_alg_script_qgis = processing.run('qgis:rastercalculator', {'EXPRESSION':expression,'LAYERS':
iface.addRasterLayer(outRast, 'LANDSAT_SDB_alg_script_qgis')

```

Planet API

- ✓ Python 3.6
- ✓ Node.js
- ✓ cURL
- ✓ requests
- ✓ retrying
- ✓ jq
- ✓ geojsonio-cli
- ✓ Scipy
- ✓ Gdal
- ✓ rasterio
- ✓ pandas
- ✓ seaborn
- ✓ traitlets
- ✓ shapely
- ✓ IPython
- ✓ Ipyleaflet

Preliminary

Planet API Key (subscription)

Development Environment Setup

Install multiple tools
(API supports Python 2.7+)

Planet API

IIC TECHNOLOGIES

```
import os
import requests
from requests.auth import HTTPBasicAuth
import json
import urllib
import sys
#from PyQt5.QtWidgets import QApplication, QWidget, QInputDialog, QLineEdit

from qgis.core import *
from qgis.PyQt.QtCore import *
from qgis.PyQt.QtGui import *
from qgis.PyQt.QtWidgets import QInputDialog, QLineEdit
```

```
def makeFilter(AoiGeometry, startDate, endDate, cloudPerc):
    print(startDate, endDate)
    geo_json_geometry = {
        "type": "Polygon",
        "coordinates": AoiGeometry["features"][0]["geometry"]["coordinates"][0]
    }

    # filter for items the overlap with our chosen geometry
    geometry_filter = {
      "type": "GeometryFilter",
      "field_name": "geometry",
      "config": geo_json_geometry
    }

    # filter images acquired in a certain date range
    date_range_filter = {
      "type": "DateRangeFilter",
      "field_name": "acquired",
      "config": {
        "gte": startDate,
        "lte": endDate
      }
    }
```

```
    # filter any images which are more than 50% clouds
    cloud_cover_filter = {
      "type": "RangeFilter",
      "field_name": "cloud_cover",
      "config": {
        "lte": cloudPerc
      }
    }

    # create a filter that combines our geo and date filters
    # could also use an "OrFilter"
    Planet_filter = {
      "type": "AndFilter",
      "config": [geometry_filter, date_range_filter, cloud_cover_filter]
    }
    return Planet_filter
```

Set-up

Import libraries

Set Filters

Start/End dates
Cloud cover
Satellite selection

Planet API

```
#def for generating list of bands to download
def getData(filter, location):

    def makeList(prompt):
        assetList = []
        while True:
            assetName, okPressed = QInputDialog.getText(None, "Get text",prompt, QLineEdit.Normal, "")
            if assetName == "-1":
                break
            assetList.append(assetName)
        return assetList
```

```
    for element in download_list:
        #out_file_path = "C:/Users/User/Documents/QGIS_SDB/manual_working/PL_API_Downloads/" +
        element+".txt"
        download_url = dataset_assets[element]["location"]
        print("Downloading, please wait...")
        urllib.request.urlretrieve(download_url, location +"/"+element+".tiff")
        print("Finished downloading "+element)
```

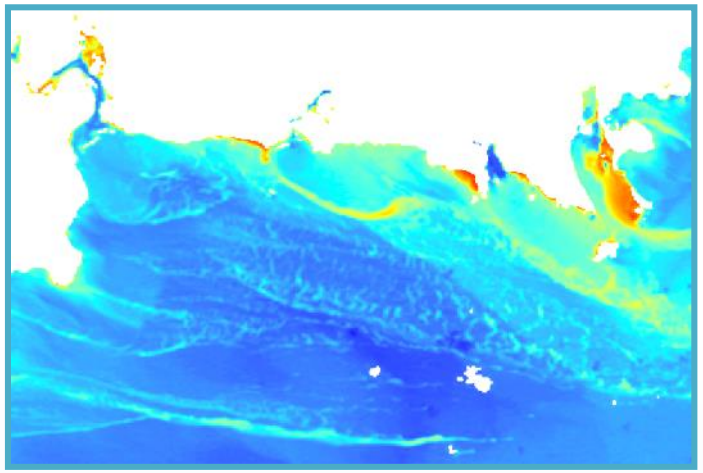
Search Images

List images matching filters
Make a selection

Download

Download selection

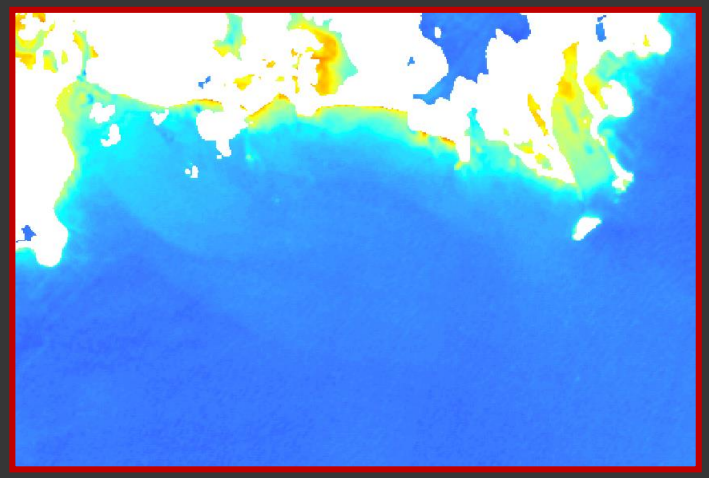
Temporal comparison



February 2015

Visible ice

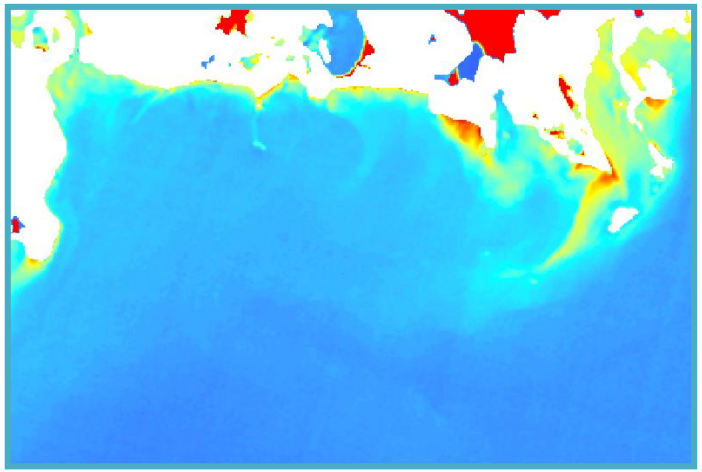
Water turbidity



April 2015

Some cloud cover

Overall best results



May 2015

Biological growth

Increased turbidity

Further plans

- **Short term**
  - Fine tune search and preview options (version 1)
  - Create QGIS plug-in
- **Mid term**
  - Implement new algorithms
  - Adapt to other purpose (e.g. shore line extraction)
- **Other projects**
  - Plug-in for CARIS BDB?
  - Adapt to ArcGIS?

NEW PATHS. NEW APPROACHES

Thank You

Véronique Jégat
vero.jegat@iictechnologies.com
www.iictechnologies.com