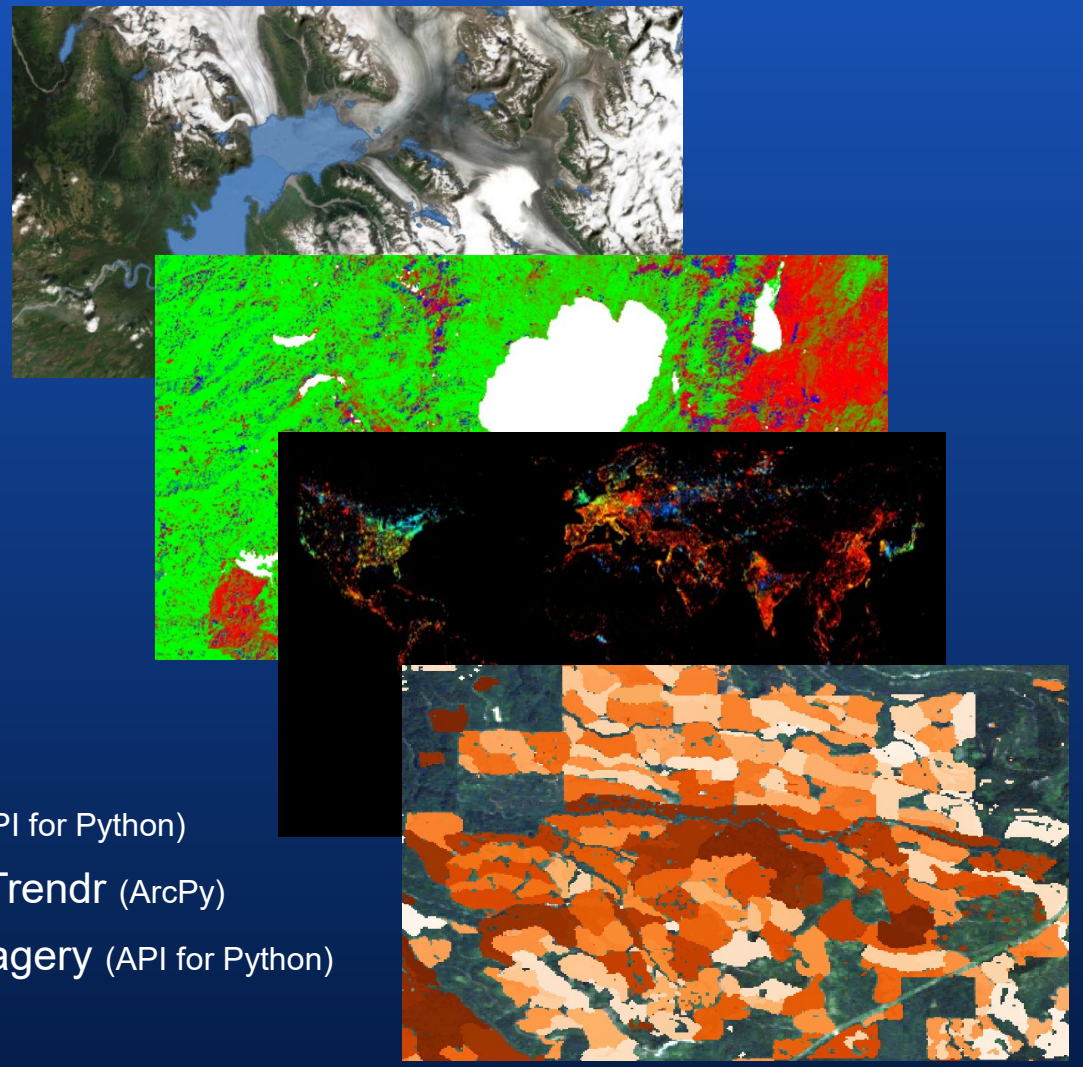
Advanced Analysis of Satellite Imagery Using Python

Hong Xu, Shengan Zhan

2021 ESRI
DEVELOPER SUMMIT

Agenda

- **ArcGIS image analysis capabilities**
- **ArcGIS python programming environment**
  - ArcPy and ArcGIS API for Python
  - Image analysis APIs
  - Raster collection and multidimensional raster workflows
- **Use case demos**
  - Mapping glacial lakes using Landsat time series (ArcPy)
  - Analyzing land cover changes using linear spectral unmixing (API for Python)
  - Detecting forest disturbance and recovery patterns using LandTrendr (ArcPy)
  - Modeling and predicting urban growth using nighttime lights imagery (API for Python)



ArcGIS Image Analysis Capabilities

Raster functions, Geoprocessing tools, and analytics tools

Image Classification

- Random forest
- Support vector machine
- Maximum likelihood
- Spectral linear unmixing

Deep Learning

- Object detection
- Semantic segmentation
- Object classification

Multidimensional Analysis

- Trend and predict
- Anomaly, aggregate
- Detect change
  - CCDC
  - LandTrendr (2.7)
- Categorical raster (2.7)

Spectral Analysis/Index

- NDVI Cig, Cire, EVI, GEMI, GNDVI, PVI, GVITM, VARI IronOxide, MSAVI, MTVI2, NDVIre, NDWI, SAVI, SRre, BandArithmatic, etc.

Spatial Analysis

- Surface
- Distance
- Hydrology
- Map algebra
- Etc.

Extendable

- Python raster function
- NumPy
- Xarray
- SciPy

Image Analysis with Python in ArcGIS

ArcGIS API for Python and ArcPy

ArcPy

- Desktop application
- Work with local and web layers
- Processing on local machine
- Parallel processing
- Output local data

Import arcpy
from arcpy.ia import *

ArcGIS API for Python

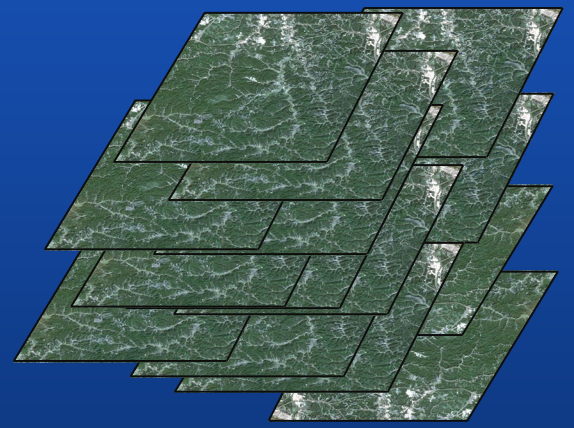
- Web application
- Work with web layers
- Distributed computing image server
- Output web layers

import arcgis
from arcgis.raster import *

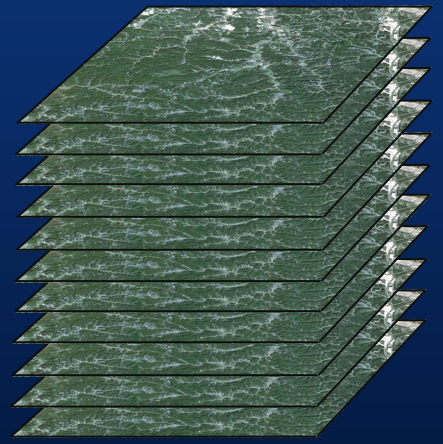
APIs for
Raster data access
and image analysis

Getting Familiar with the Concepts

Mosaic dataset and multidimensional raster



- **Mosaic Dataset**
  - Data model managing collections of images
  - Reference images and catalog metadata
  - 20+ satellite data: Landsat, MODIS, Sentinel, etc.
  - Scalable and flexible

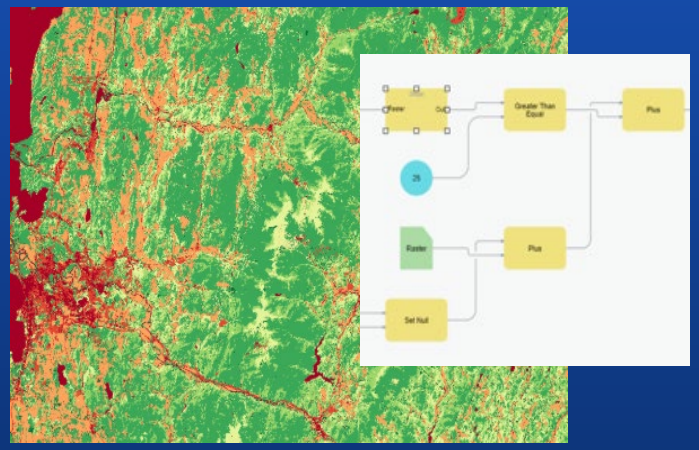


- **Multidimensional raster**
  - Well-structured raster dataset for multidimensional data (time, depth)
  - manages variables and dimensional arrays
  - Examples: time series imagery, ocean temperature, etc
  - CRF, NetCF, etc.

Getting Familiar with the Concepts

Raster functions and image server

- **Raster functions**
  - Algorithm definition processing raster on-the-fly
  - Many out-of-the-box functions
  - Chained together to create complex model
  - Persist as a Raster Function Template(RFT)

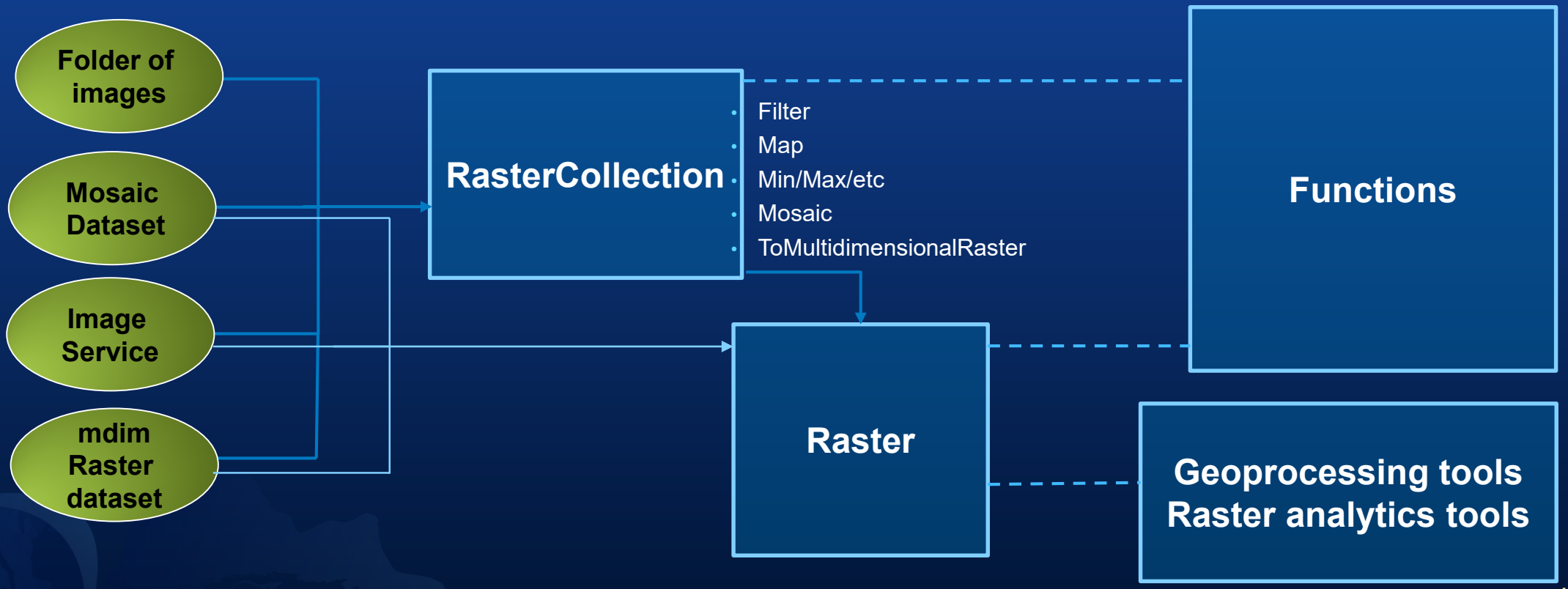


- **Image Service/Image Server**
  - A server serves raster data
  - On-the-fly processing using raster functions
  - Distributed computing

Raster and Raster Collection Objects

For both ArcPy and ArcGIS API for Python

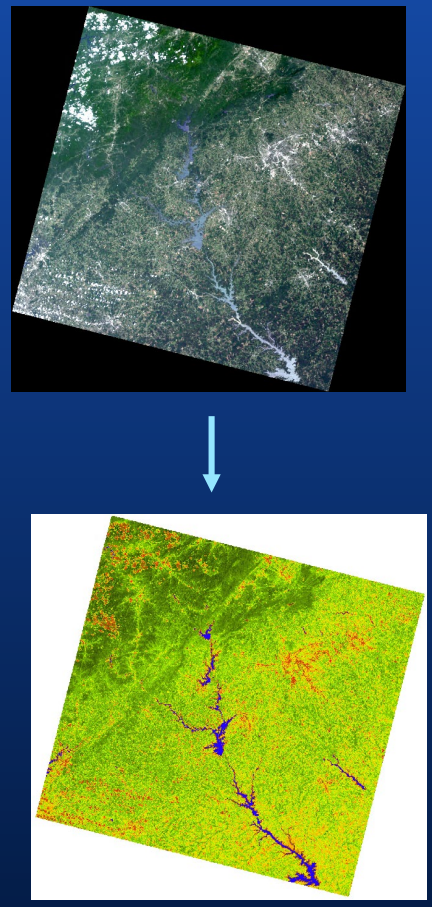
- RasterCollection for accessing a collection of images
- Raster for accessing a single raster dataset or a multidimensional raster



Processing a Single Image Using Raster Object

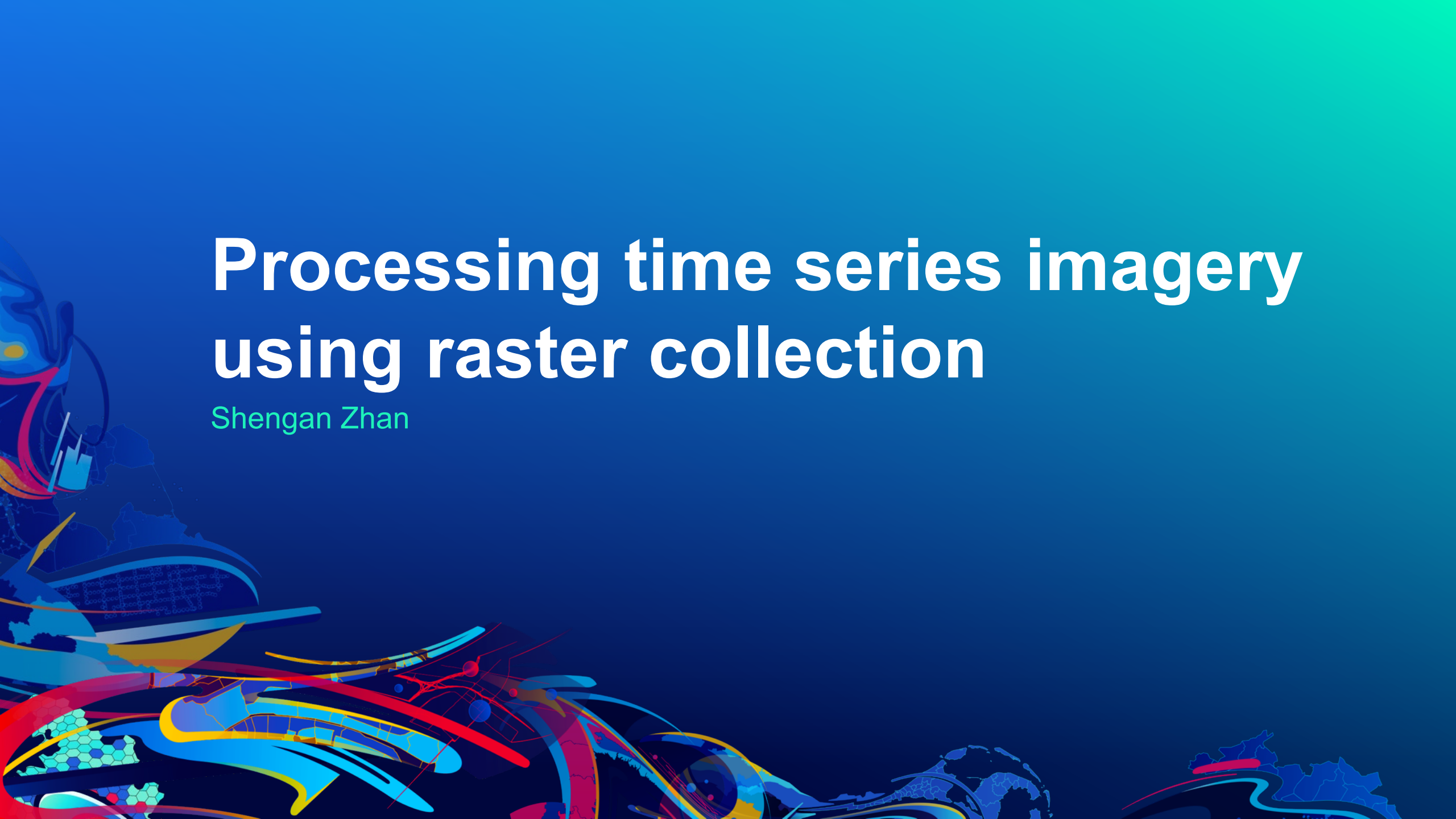
- **Create NDVI from a Landsat scene**

```
#import arcpy package
import arcpy

#create a raster object from a landsat scene
image = arcpy.Raster(r"c:\LC80180362013164LGN00_MTL.txt\Multispectral")

#mask out the black borders
mask_image = arcpy.ia.Mask(image, no_data_values = 0 )

#create an NDVI image using NDVI function
ndvi_image = arcpy.ia.NDVI(mask_image, nir_band_id=5, red_band_id=4)

#display using a colormap
render_ndvi_image = arcpy.ia.Render(ndvi_image, rendering_rule=
                            {'min':-0.2, 'max':0.8}, colormap = "NDVI")
render_ndvi_image
```

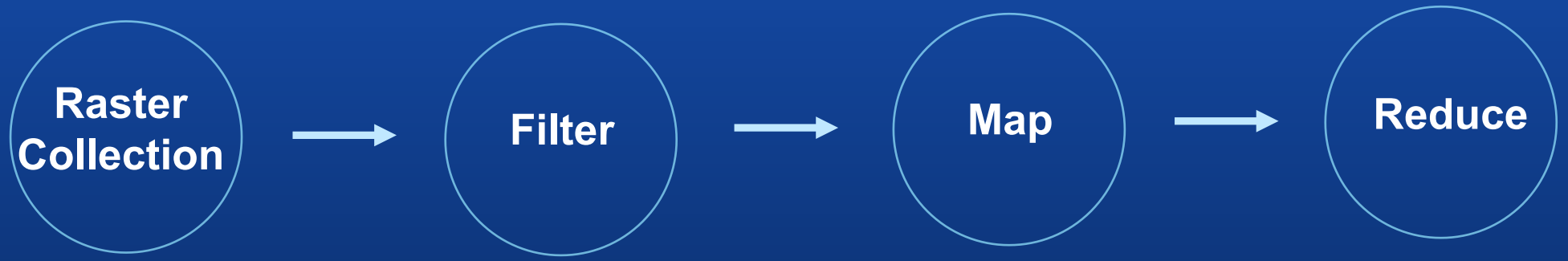


Processing time series imagery using raster collection

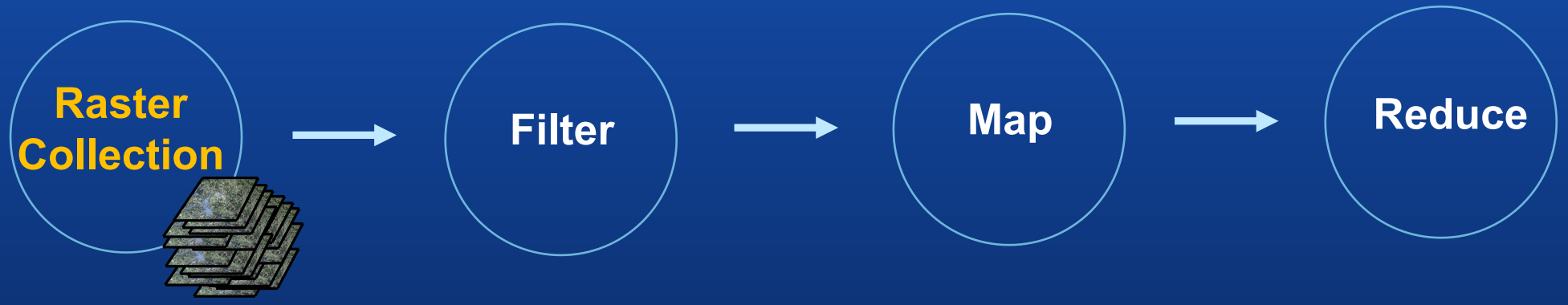
Shengan Zhan



Processing Using Raster Collection



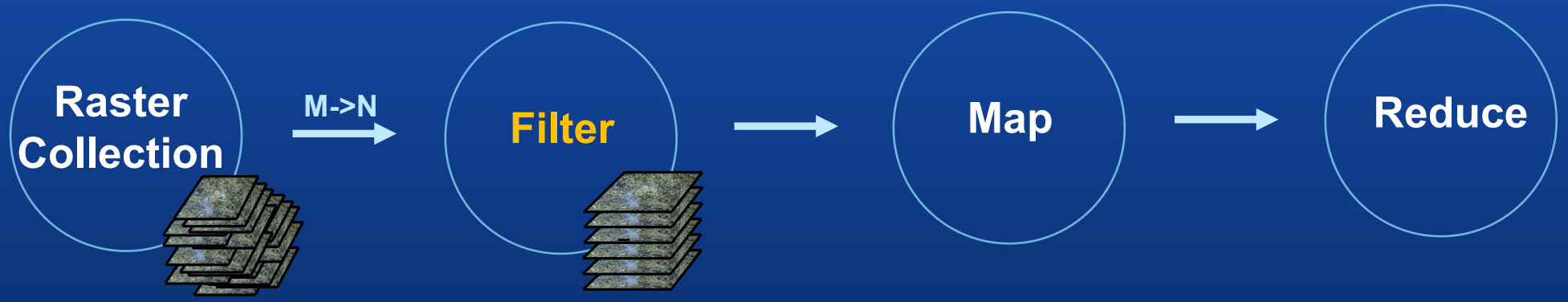
Processing Using Raster Collection



- Create from
  - List of images
  - mosaic dataset
  - Image service
  - multidimensional raster

```
import arcpy
# from a list of rasters
in_rasters = [r"D:\data\L7_2000.tif", r"D:\data\L7_2001.tif", r"D:\data\L7_2002.tif"]
rc=arcpy.ia.RasterCollection(in_rasters, {"Variable": "Landsat7",
                                          "StdTime": ["2000-01-01", "2001-01-01", "20002-01-01"]})

#create from a mosaic dataset
rc=arcpy.ia.RasterCollection(r"D:\data\Landsat7.gdb\CA_1990_2000")

#create from an image service
rc=arcpy.ia.RasterCollection("https://localhost/server/rest/services/CA/ImageServer")

#create from multidimensional raster
rc=arcpy.ia.RasterCollection(r"D:\data\CA_raster.crf")
```

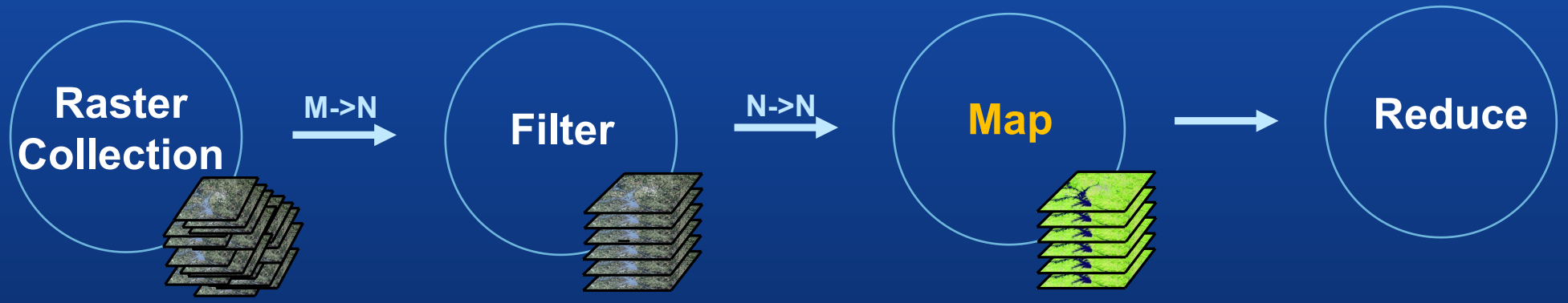
Processing Using Raster Collection



- Filter items by
  - Spatial extent
  - Time extent
  - Geometry
  - Where clause and attributes
  - Raster properties

```
filtered_rc = rc.filterByTime(end_time = '2009-01-01T00:00:00', time_field_name = 'AcquisitionDate')
filtered_rc = rc.filterByGeometry(line)
filtered_rc = rc.filterByAttribute("SensorName", "CONTAINS", "Landsat7")
filtered_rc = rc.filterByCalendarRange(calendar_field="DAY_OF_YEAR", start=1, end=5)

filtered_rc = rc.filterByGeometry(study_area).filterByTime(time_field_name = 'AcquisitionDate',
                        start_time = '1990-01-01 00:00:00', end_time = '1995-12-31 00:00:00')
```

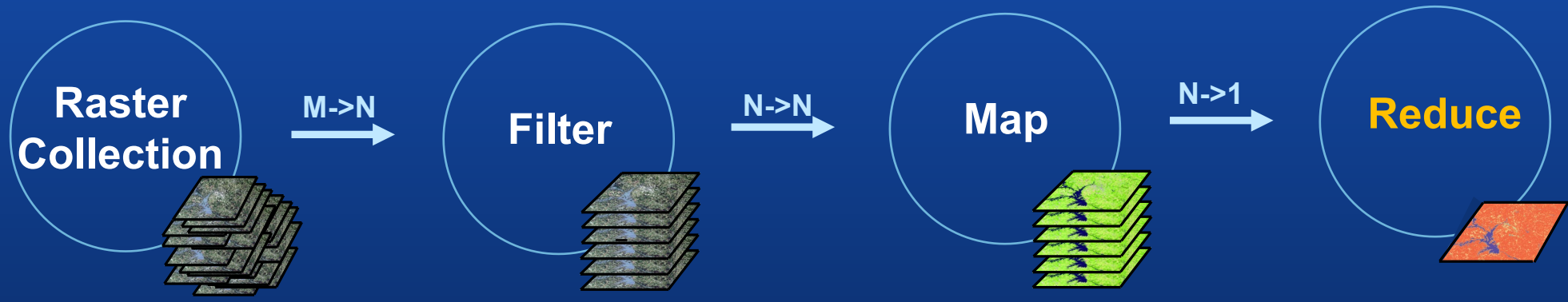
Processing Using Raster Collection



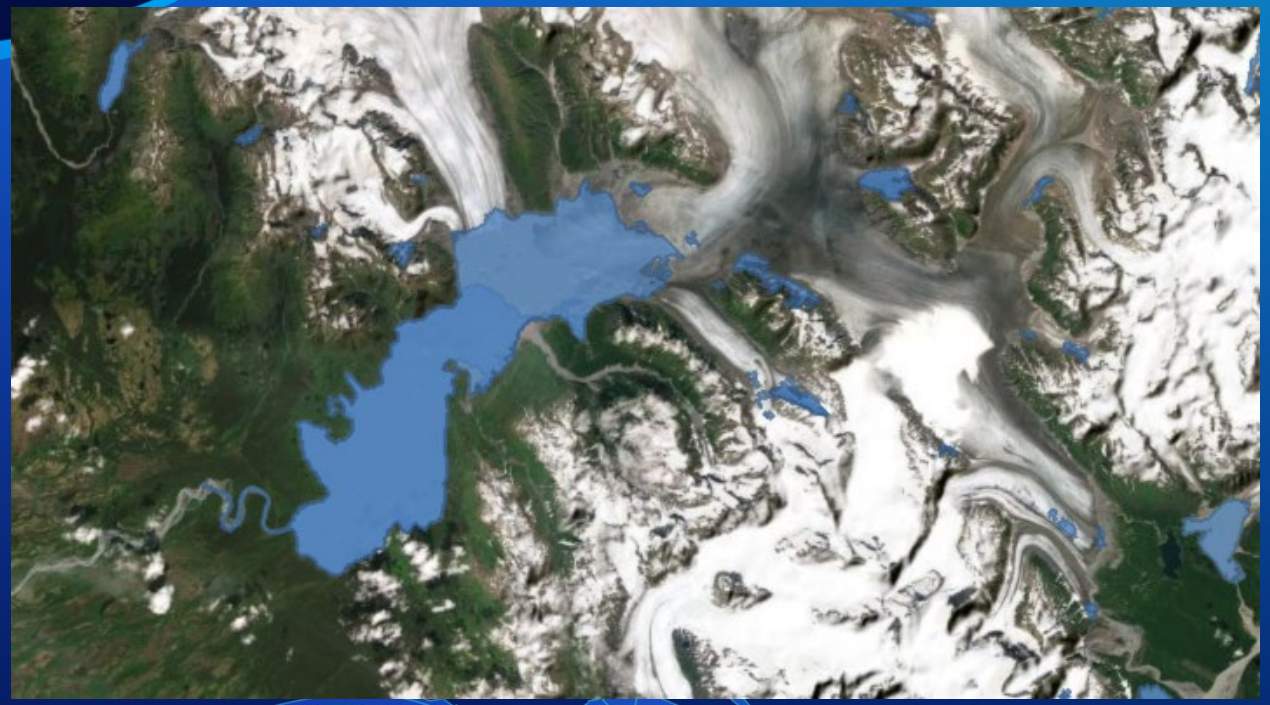
- Map a function for each item in the RasterCollection

```
def calcNDVI(item):
    ndvi = arcpy.ia.NDVI(item["Raster"], nir_band_id = 4, red_band_id = 3)
    return {"raster": ndvi, "Name": item["Name"], "AcquisitionDate": item["AcquisitionDate"]}

ndvi_rc = filtered_rc.map(calcNDVI)
```

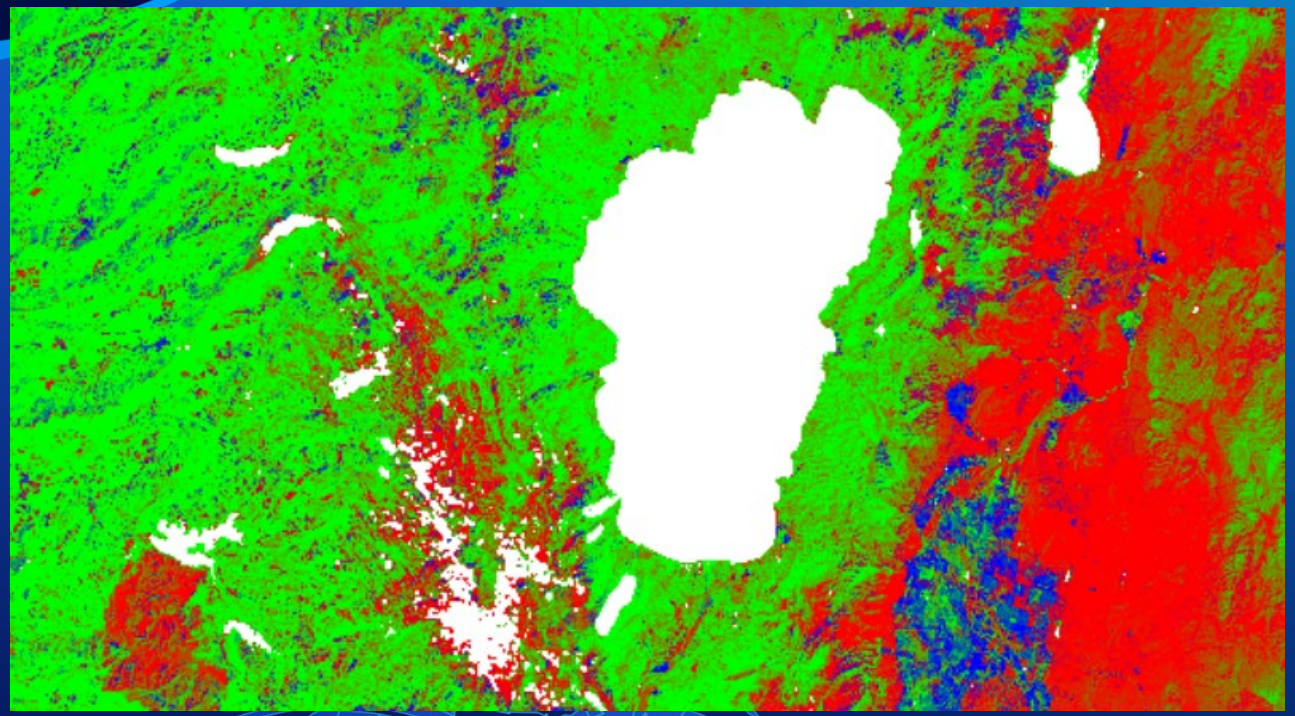
Processing Using Raster Collection



- **Reduce summarizing pixels**
  - RasterCollection::Min()
  - Max, Mean, Sum, Median, Majority
- **Reduce by mosaicking pixels**
  - RasterCollection::Mosaic, QualityMosaic



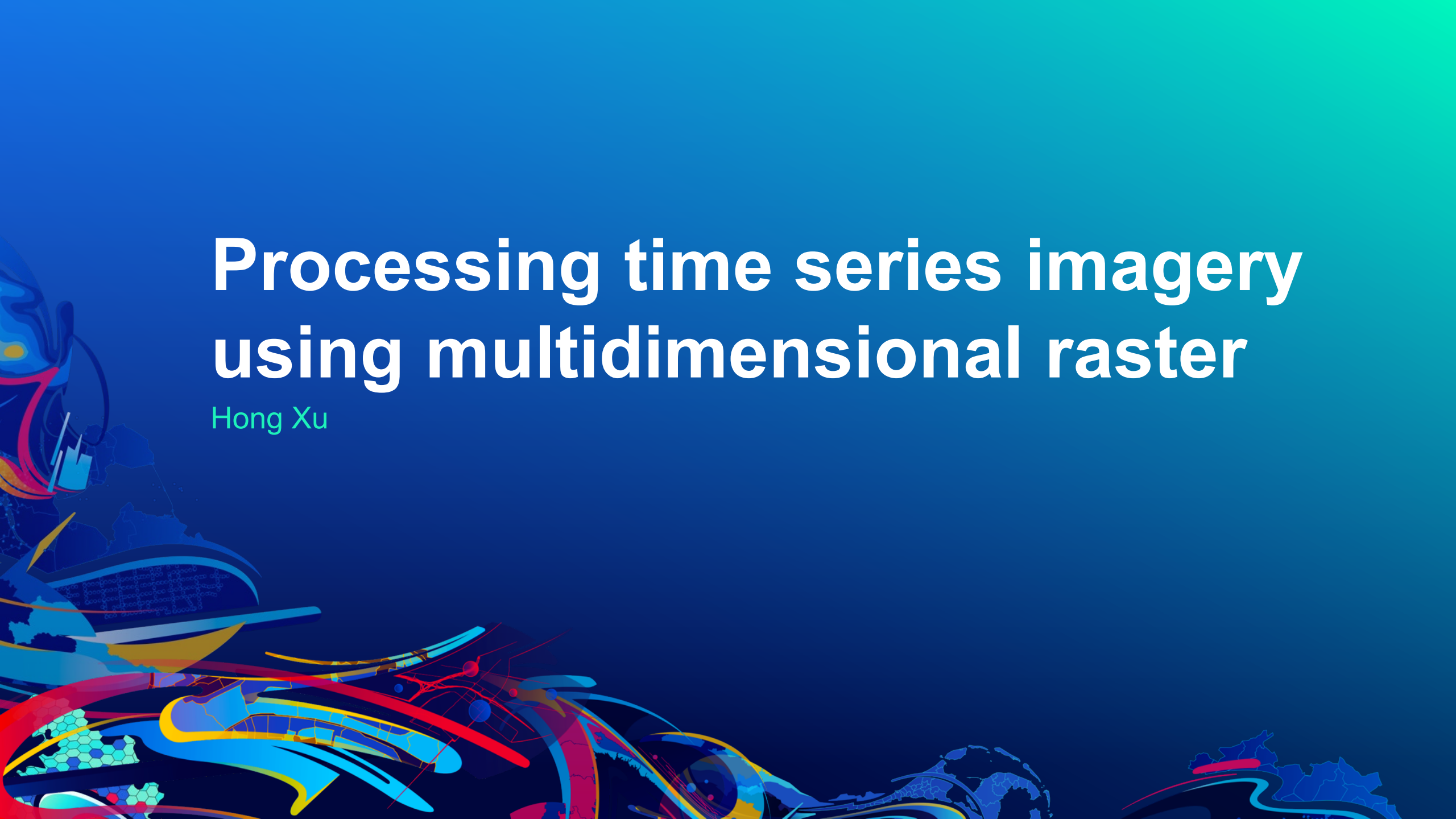
Mapping glacial lakes using Landsat time series



Analyzing land cover changes using linear spectral unmixing

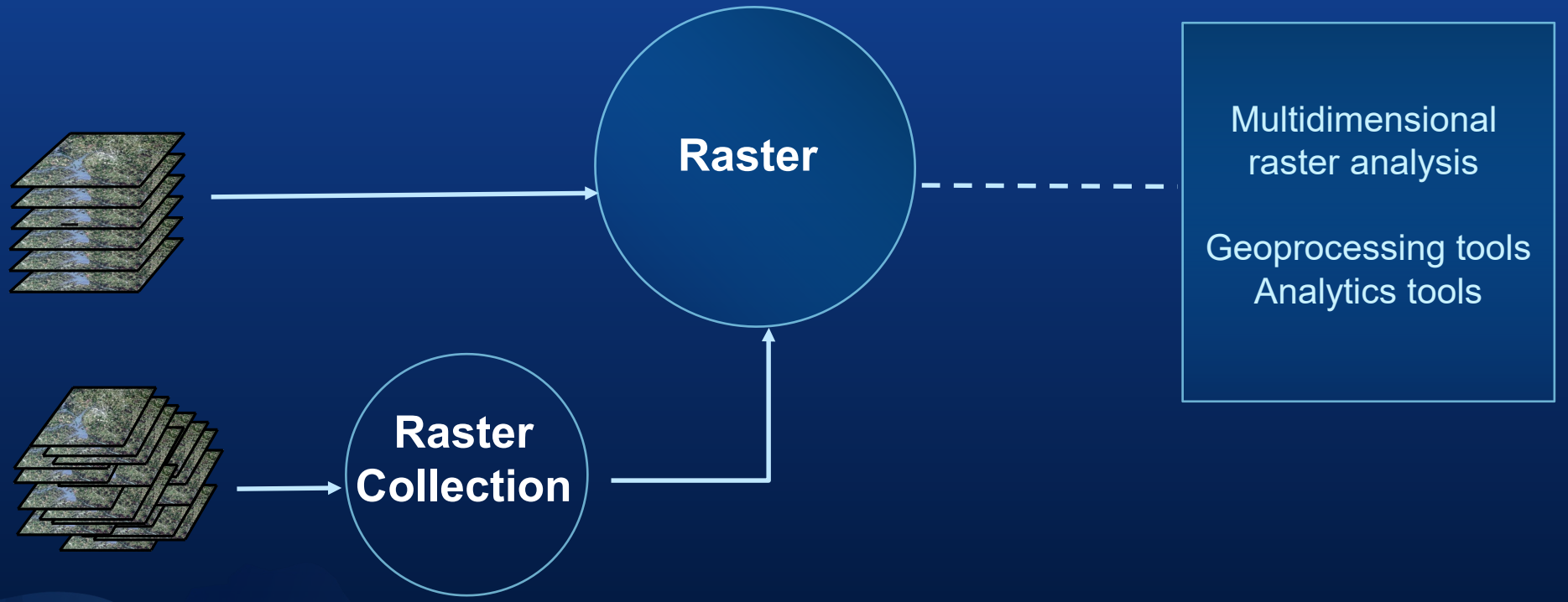
Processing time series imagery using multidimensional raster

Hong Xu



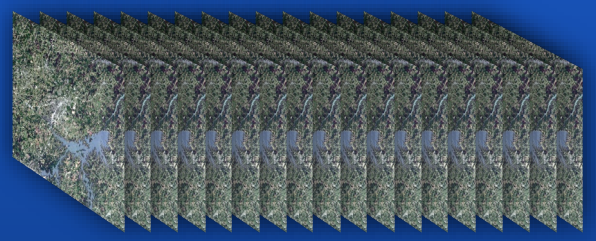
Processing Using Multidimensional Raster

- **Create a multidimensional raster directly from dataset**
- **Create from raster collection**
  - RasterCollection::ToMultidimensionalRaster



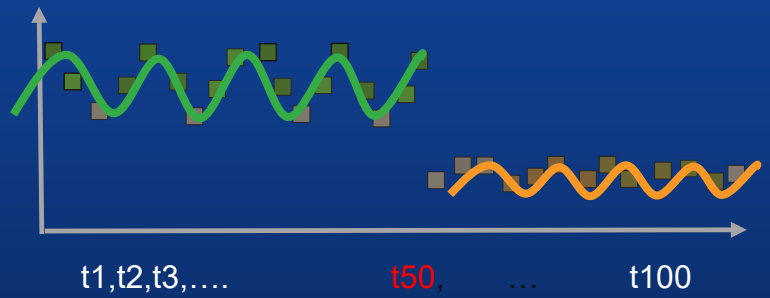
Multidimensional Raster Analysis Capabilities

Change Detection with CCDC and LandTrendr

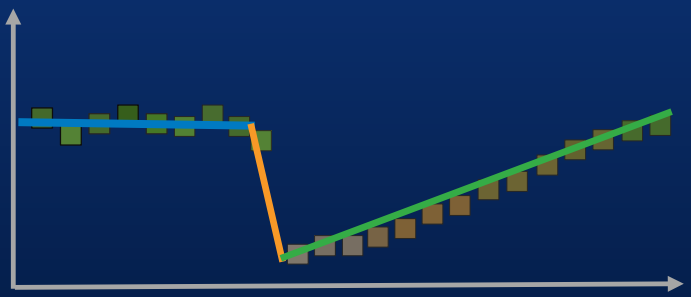


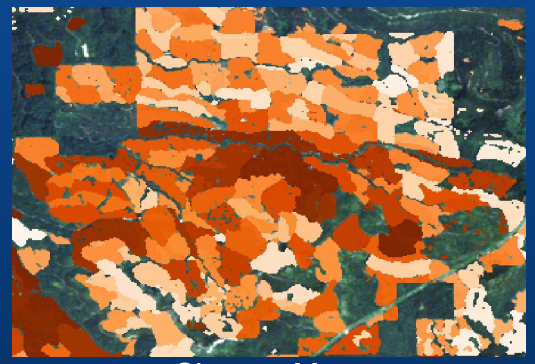
Input time series

- **CCDC**
  - Continuous Change Detection and Classification
  - Harmonic regression
  - Take account of seasonality

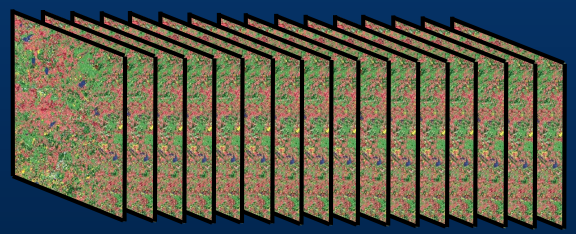


- **LandTrendr**
  - Landsat-based Detection of Trends and Disturbances
  - Temporal segmentation
  - Yearly data





Change Map

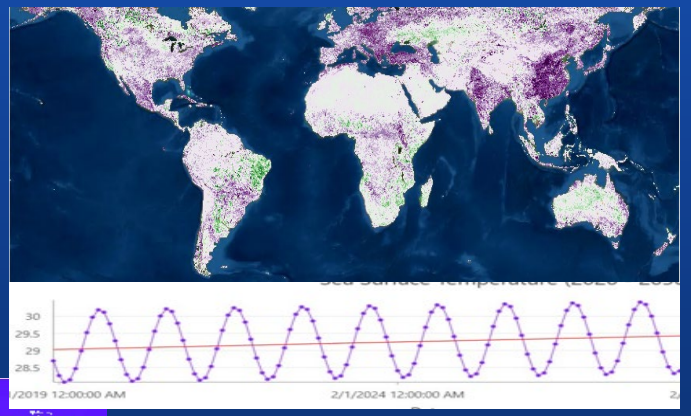


Time Series Classification

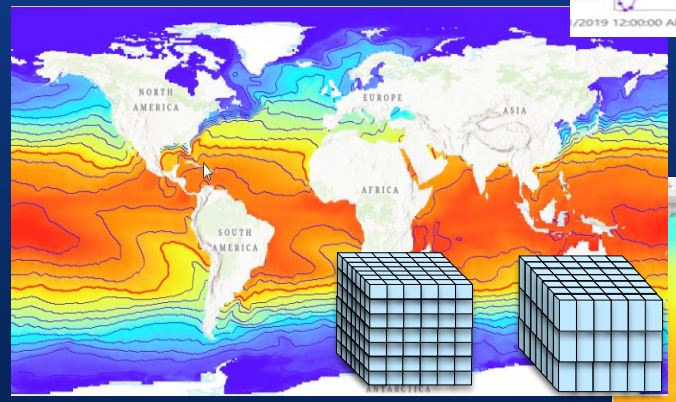
Multidimensional Raster Analysis Capabilities

Tend, predict, aggregate, and anomaly

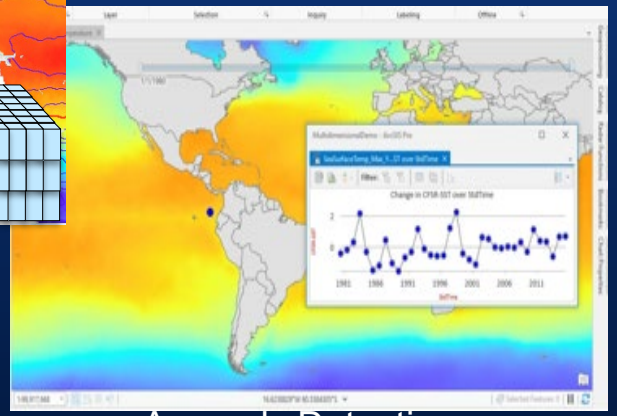
- Generate Trend
  - Linear
  - Harmonic
  - Mann-Kandall test
  - Seasonal Kandall test
- Predict
- Aggregate data
- Compute anomaly



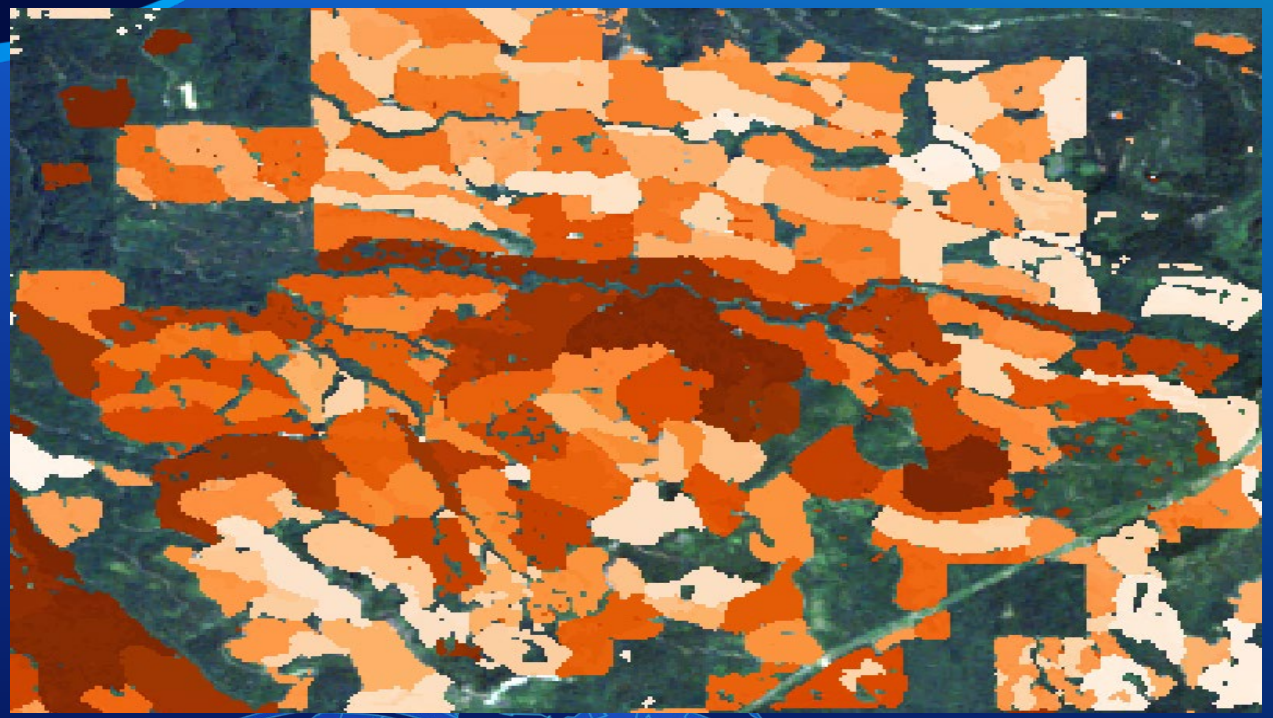
Trend Map



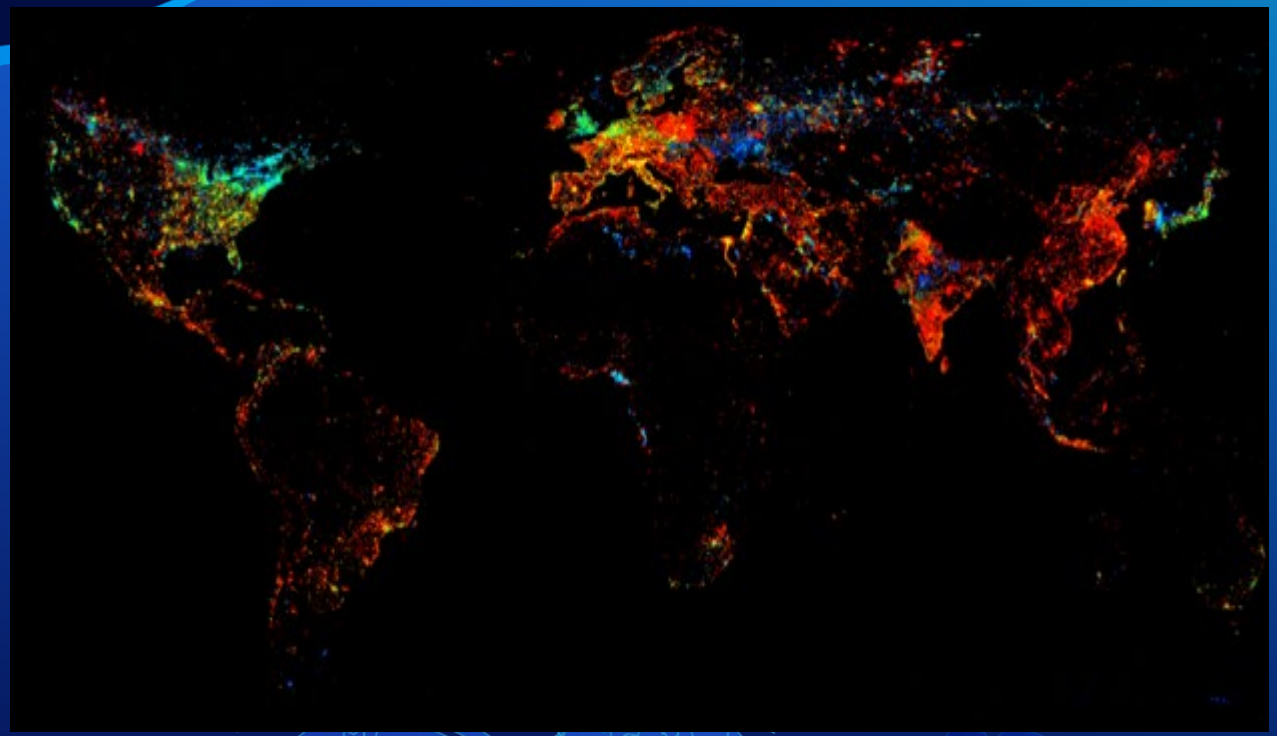
Data Aggregation



Anomaly Detection



Detecting forest disturbance and recovery patterns using Landtrendr



Modeling and predicting urban growth using nighttime lights imagery

Creating a Multidimensional Raster for Landsat Collection

- **Create with Geoprocessing tools**
  - Create a mosaic dataset
  - Add rasters to mosaic dataset using Landsat Raster type
  - Build Multidimensional Info

  https://www.esri.com/arcgis-blog/products/arcgis-pro/analytics/create-multidimensional-raster-data/

- **Data quality recommendation for time series analysis**
  - Surface reflectance
  - ARD or Level 2 products
  - Remove cloud and shadow
- https://www.esri.com/arcgis-blog/products/arcgis-pro/imagery/clean-up-your-landsat-imagery-removing-cloud-and-cloud-shadow/

Additional Resources

Sessions in Dev Summit

Working with Multidimensional Data in ArcGIS

Advanced Custom Raster Processing Using Python

Detect Change of Raster Data

Prediction and Trend Analysis of Raster Data

Help references

ArcPy APIs and examples

- https://pro.arcgis.com/en/pro-app/arcpy/image-analyst

ArcGIS API for Python examples

- https://developers.arcgis.com/python/api-reference/arcgis.raster.toc.html

Please provide your feedback for this session by clicking on the session survey link directly below the video.

Copyright © 2021 Esri. All rights reserved.