Oracle Spatial & Graph SIG

ORACLE SPATIAL SUMMIT 2016

January 26–28, 2016

With BIWA 2016: The Oracle Big Data + Analytics Conference

# High Performance Raster Database Manipulation and Data Processing with Oracle Spatial and Graph


Qingyun (Jeffrey) Xie
Senior Manager of Software Development
Oracle Corporation


ORACLE

# Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

# Program Agenda

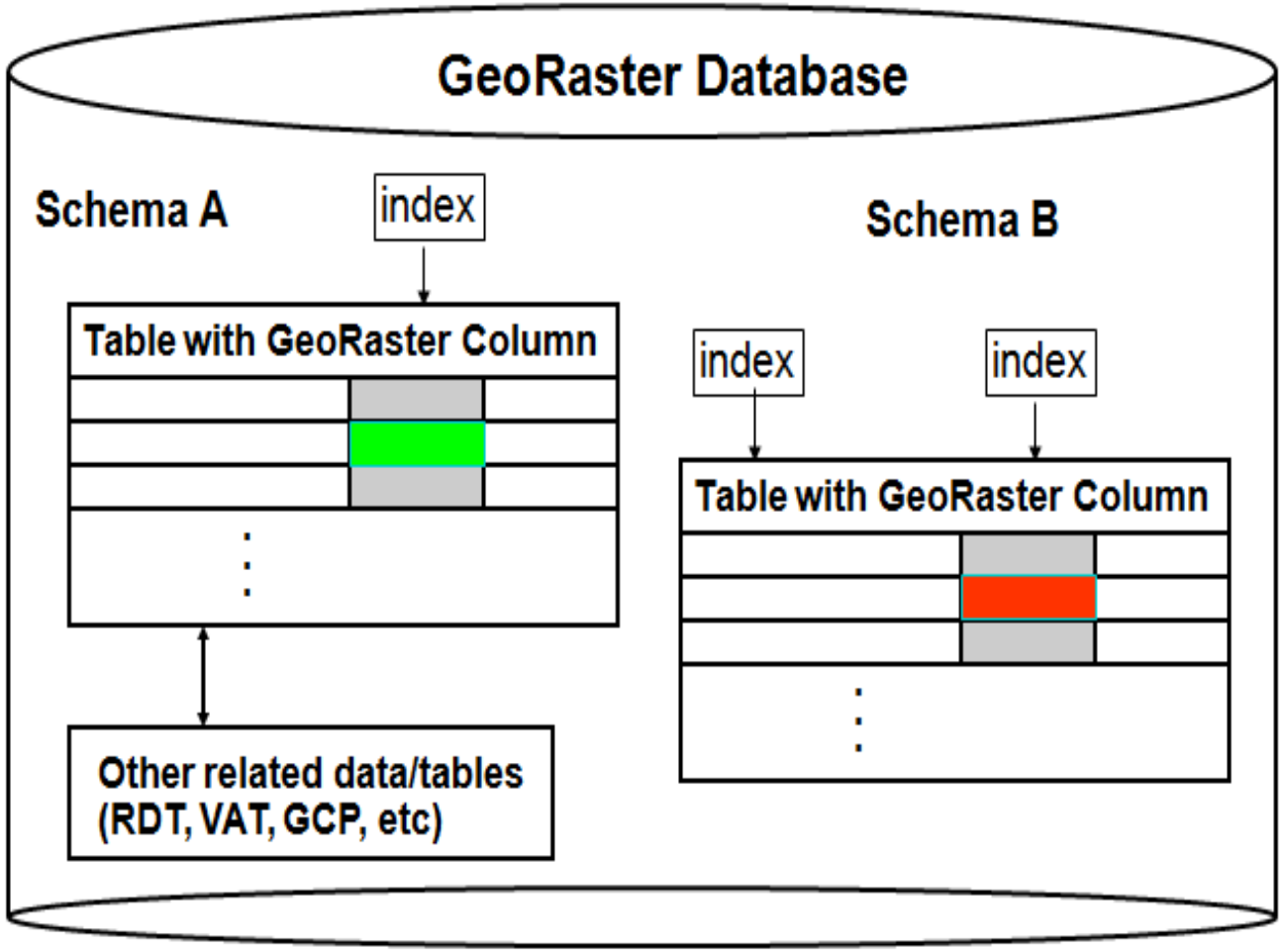
1. GeoRaster Overview and Major Components
2. GeoRaster Workflow and Sample Use Cases
3. High Performance Raster Data Processing
4. Application Development Framework
5. Demonstration and Summary

# Program Agenda

1. GeoRaster Overview and Major Components
2. GeoRaster Workflow and Sample Use Cases
3. High Performance Raster Data Processing
4. Application Development Framework
5. Demonstration and Summary

# What is Oracle Spatial GeoRaster

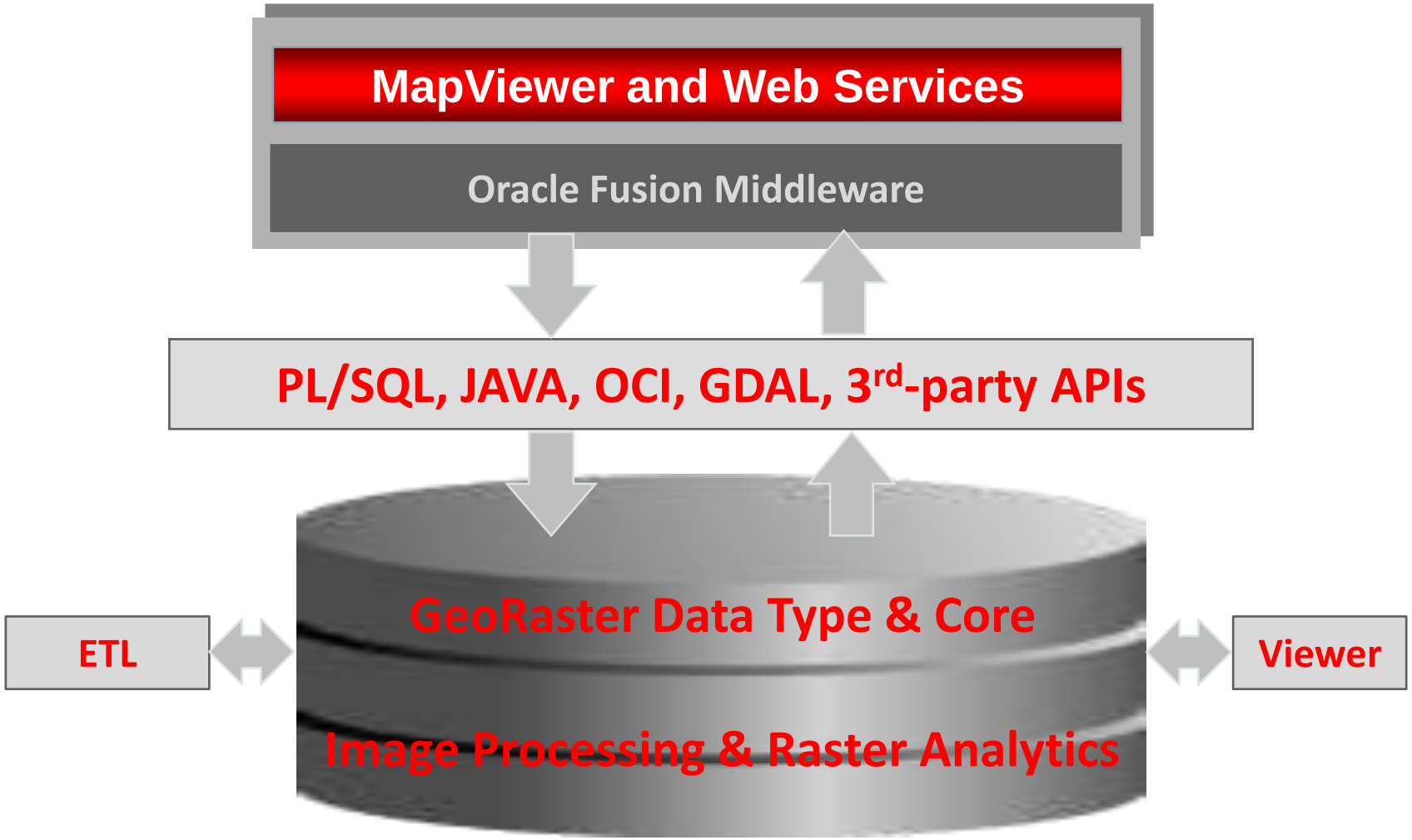
1. **A native data type, which encapsulates both metadata and cell data to store rasters:**
   - Satellite imagery and aerial photographs
   - Gridded rasters and DEM's
   - An XML schema to store Metadata
2. **A raster database management system, which lets you:**
   - Load, store, compress, and index rasters
   - Query, update, backup, and export rasters
   - Supports advanced features such as multi-user, security, standby, and multi-tenancy.
3. **An In-database processing/analysis engine, including:**
   - Image processing and mosaicking
   - Raster algebra and analytics



# Major Components of GeoRaster

## Five Major Components:

1. The GeoRaster Data Type and Core Database Features
2. In-database Processing and analytics Engines
3. GeoRaster Tools (ETL and Viewer)
4. MapViewer and Web Services (WMS)
5. Application Programming Interfaces



# GeoRaster Data Type and Core Database Features

1. **The native SDO_GEORASTER data type encapsulates both metadata and cell data to store rasters:**

```
CREATE TYPE sdo_georaster AS OBJECT (
    rasterType NUMBER,
    spatialExtent SDO_GEOMETRY,  -- stores footprint
    rasterDataTable VARCHAR2(32), -- stores actual raster cell data but its manipulation is automated and transparent to users
    rasterID NUMBER,
    metadata SYS.XMLType );          -- stores metadata
```

2. **The GeoRaster database management system provides almost 200 raster manipulation operations and inherits almost all Oracle enterprise RDBMS features:**
   - Load, store, and index rasters
   - Search, query, update, and export rasters
   - Pyramid, compress, decompress, subset, merge, mosaick rasters
   - Multi-user, concurrency, security, backup, recovery, replication, standby, and multitenancy
   - High scalability:
     - Virtually no size limit for each raster object
     - Unlimited number of raster objects
     - Scalable multi-tasking and multi-user with computer clustering

ORACLE

# GeoRaster Image Processing Engine

- Advanced **Georeferencing**
- **Reprojection**
- **Rectification**
- **Orthorectification**
- **Scaling**
- **Pyramiding**
- **Masking**
- **Stretching**
- **Segmentation**

- **NDVI** (Normalized Difference Vegetation Index)
- **TCT** (Tasseled Cap Transformation)
- Bands **Merging**
- Large-scale Image **Mosaicking**
- Large-scale Image **Appending**
- Large-scale **Virtual Mosaic**

(Xie, et al. In-database Image Processing in Oracle Spatial GeoRaster. In: *the Proceedings of ASPRS 2013 Annual Conference*, Baltimore, Maryland, March 24-28, 2013)

ORACLE

# GeoRaster Raster Analytics Engine

- GeoRaster provides in-database Raster Algebra for raster analytics and modeling
- The Raster Algebra Language is provided as an extension to the PL/SQL language
- It includes an expression language for raster algebra operators
  - general arithmetic, casting, logical, and relational operators
- It includes raster algebra functions
  - cell value based conditional queries;
  - cell value based conditional updates;
  - arithmetic operations;
  - raster classification/segmentation
  - statistical analysis

(Xie, et al. In-database Raster Analytics: Map Algebra and Parallel Processing in Oracle Spatial GeoRaster. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B4, 2012*, XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia)

ORACLE

# The Raster Algebra Expressions

```
arithmeticExpr:
        unaryArithmeticExpr
       | binaryArithmeticExpr
       | functionArithmeticExpr
       | booleanExpr
       | castingExpr
       | constantNumber
       | identifier
       | (arithmeticExpr)

booleanExpr:
        unaryBooleanExpr
       | binaryBooleanExpr
       | arithmeticExpr comparisonOp
        arithmeticExpr
       | (booleanExpr)

unaryArithmeticExpr:
      (arithmeticUnaryOp arithmeticExpr)

binaryArithmeticExpr:
        arithmeticExpr arithmeticBinaryOp
        arithmeticExpr)
```

```
functionArithmeticExpr:
        numericFunction (arithmeticExpr)

castingExpr:
        rangeType (arithmeticExpr)

unaryBooleanExpr:
        booleanUnaryOp booleanExpr

binaryBooleanExpr:
        booleanExpr booleanBinaryOp
        booleanExpr

arithmeticBinaryOp:
        +
        | -
        | *
        | /
comparisonOp:
        =
        | <
        | >
        | >=
        | <=
        | !=
```

# The Raster Algebra Expressions (cont.)

```
arithmeticUnaryOp:
        +
      | -

booleanBinaryOp:
        &
      | |

booleanUnaryOp:
      !

rangeType:
        castint
      | castonebit
      | casttwobit
      | castfourbit
      | casteightbit

numericFunction:
        abs
      |sqrt
      | exp
      | log
      | ln
      | sin
      | cos
      | tan
      | sinh
      | cosh
      | tanh
      | arcsin
      | arccos
      | arctan
      | ceil
      | floor

constantNumber:
      double number

identifier:
        {ID,band}
      | {band}

ID:
      integer number

band:
      integer number
```

# GeoRaster Viewer and ETL Tools

- **GeoRaster Viewer**
  - It's a developer and DBA tool for displaying GeoRaster objects and metadata, as well as virtual mosaics.
  - Use it to display any GeoRaster object and visually exam cell values and metadata
- **GDAL-based GeoRaster ETL Tool**
  - GUI to create GDAL-based batch loading and exporting description files (XML)
  - GUI to concurrently load many batches of image and raster files
  - All GDAL supported file formats
- **JAI-based GeoRaster Loader and Exporter**
  - Lightweight tools for conveniently load and export a limited number of image and raster files one at a time.
  - Only support some standard image formats, so using the GDAL-based ETL is recommended
- **Third Party ETL tools**
  - Many third parties provide ETL and viewing tools for GeoRaster
  - For example, Safe Software FME, PCI GeoRaster ETL, etc.

ORACLE

# MapViewer and Web Services

- **Oracle Fusion Middleware MapViewer**
  - MapViewer and MapBuilder fully support GeoRaster
  - Defines GeoRaster themes and Virtual Mosaic themes
  - Supports zoom-in, zoom-out, masking, reprojection, enhancements, and overlay with vector themes
  - Tile Server caching can speed up display dramatically
- **WMS – Web Map Service**
  - WMS 1.1.1 and 1.3.0
  - Supports the GetMap, GetFeatureInfo, and GetCapabilities requests

# GeoRaster Application Programming Interfaces

- **GeoRaster PL/SQL API**
  - Almost 200 functions and procedures
  - Enables query, editing, update, compressing, processing and analysis of both raster data and metadata
- **GeoRaster Java API**
  - A complete mapping of GeoRaster objects into pure Java objects
  - A complete mapping of GeoRaster PL/SQL APIs in pure Java
  - An image rendering package to fully leverage the power of JAI for visualization
- **OCI - Oracle Call Interface**
  - The GeoRaster Data Model and Format is completely open enabling bit level access and manipulation
- **MapViewer XML, Java and AJAX-based JavaScript API**
- **Third-party APIs**
  - OSGeo GDAL C++/C/Java/Python API
  - PCI EASI and many other APIs

# Program Agenda

1. GeoRaster Overview and Major Components
2. **GeoRaster Workflow and Sample Use Cases**
3. High Performance Raster Data Processing
4. Application Development Framework
5. Demonstration and Summary

# Typical GeoRaster Workflow

**Create and Build**

- Setup database
- Create GeoRaster and RDT tables
- Load rasters and images
- Validate GeoRasters
- Create Spatial Index and other indices

**Manage and Manipulate**

- Search and query
- Delete and Copy
- Edit and update
- Optimize, compress, and pyramid
- Subset and merge
- Append and mosaic
- Backup and recover
- Replicate and standby
- Multitenancy

**Process and Analyze**

- GCP georeference
- Rectify
- orthorectify
- Scale
- Stretch
- Physical mosaic
- Virtual mosaic
- Raster algebra
- Analytical query
- Cartographic modeling

**Visualize and Export**

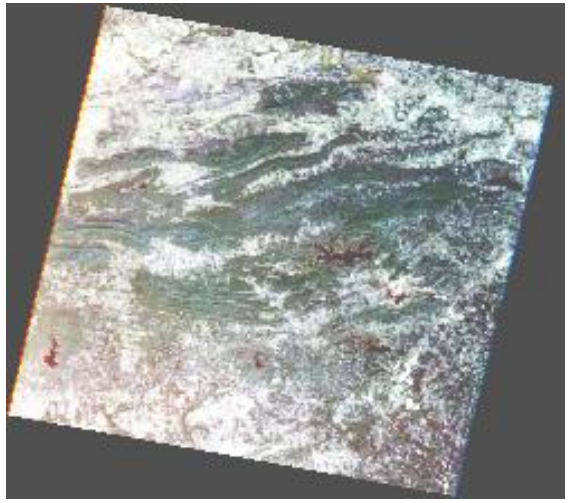
- MapViewer
- WMS
- Export
- Distribute

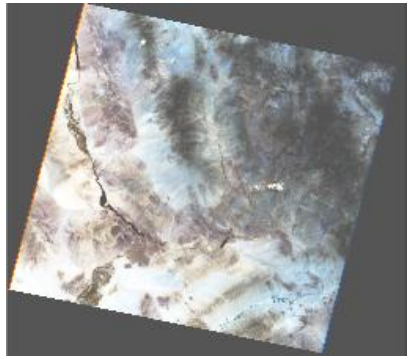
# Sample Use Case 1:

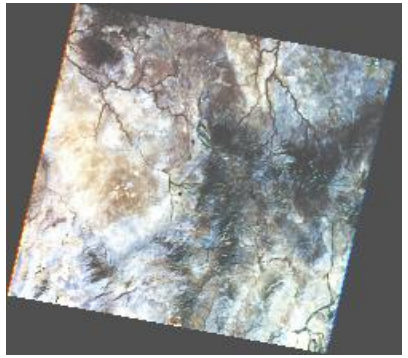
# Landsat Image Mosaic of Continent USA

# Use Case 1: Landsat Image Mosaic of Continent USA

- **Goal:**
  - Quickly create large scale **physical mosaics** of the whole continent USA
- **Source images:**
  - 454 Landsat 5 Level 1T TM images, from *the U.S. Geological Survey*.
  - Each band is a Geotiff file, so in total 3178 files
  - The total size of the image set is about 180GB
  - The images are in 10 different UTM zones
  - Each image looks rotated and has black boundary areas
  - All images overlap neighboring images







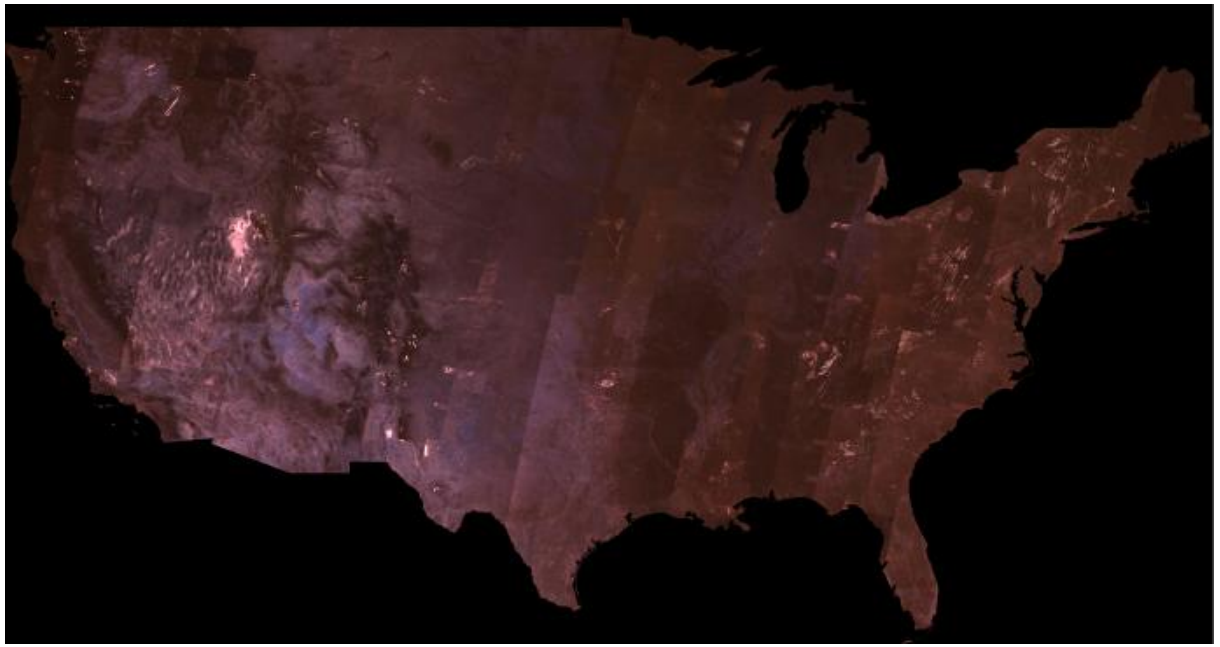
(Imagery Courtesy of USGS)

# Use Case 1: Landsat Image Mosaic of Continent USA (cont.)

- **Processing steps:**
  - **Load**: gdalbuildvrt – to merge 7 bands into 1 image (can also be done using sdo_geor.mergeLayers); gdal_translate or GeoRasterETL
  - **Index**: create spatial index based on spatial extent (footprint)
  - **Mosaick**: sdo_geor_aggr.mosaicSubset – **a single call** to mosaick all images into one
  - **Pyramid**: sdo_geor.generatePyramid
- **Output mosaic:**
  - About 124 GB in size
  - Can be in different projections with different resolutions
  - Mosaics at different pyramid levels can be created as well
  - The mosaics are stored as single GeoRaster objects – called **physical mosaic**

ORACLE

# Use Case 1: Resulting Mosaic in WGS 84

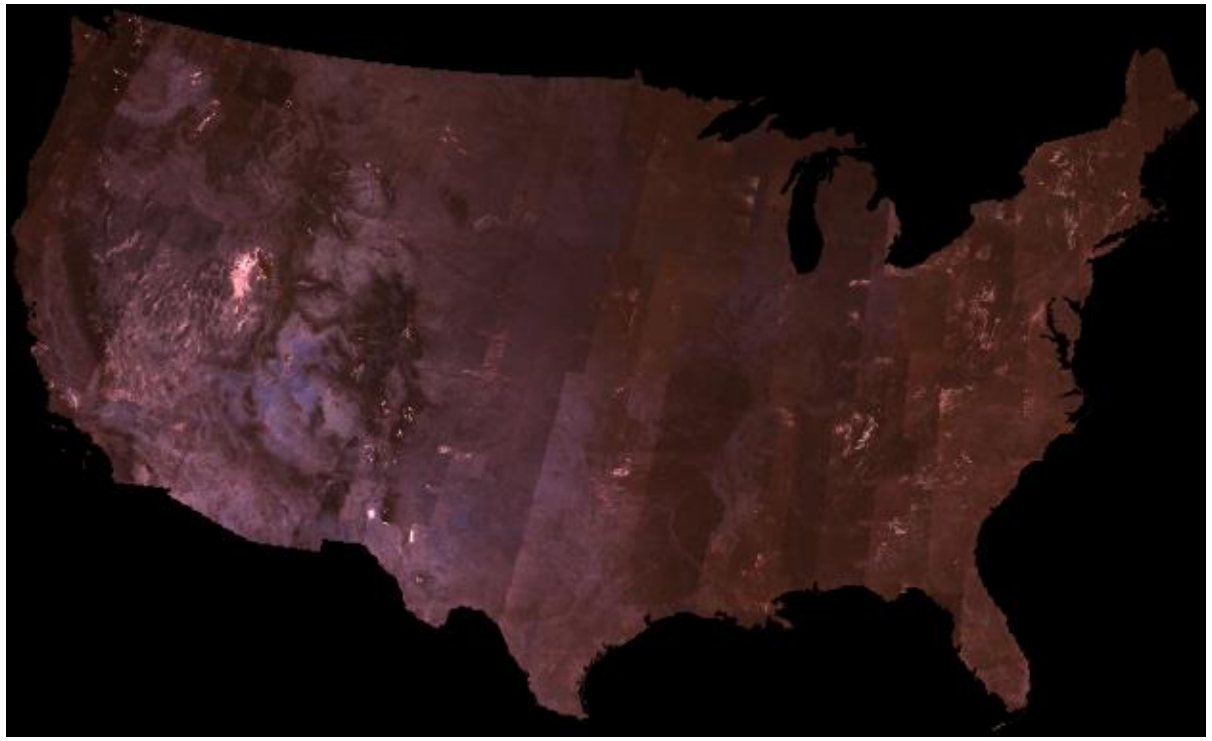


Mosaic Overview

Mosaic Overview - Stretched

(Note, the source images are in different seasons of many years, so no color balancing is done on this mosaic. The mosaic is in SRID 4326. Imagery Courtesy of USGS)

# Use Case 1: Resulting Mosaic in Albers Conical Equal Area



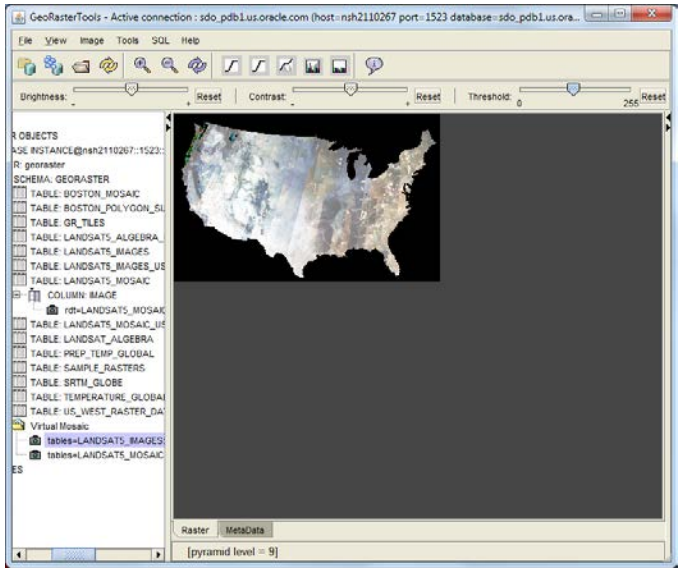
Mosaic Overview

Mosaic Overview - Stretched

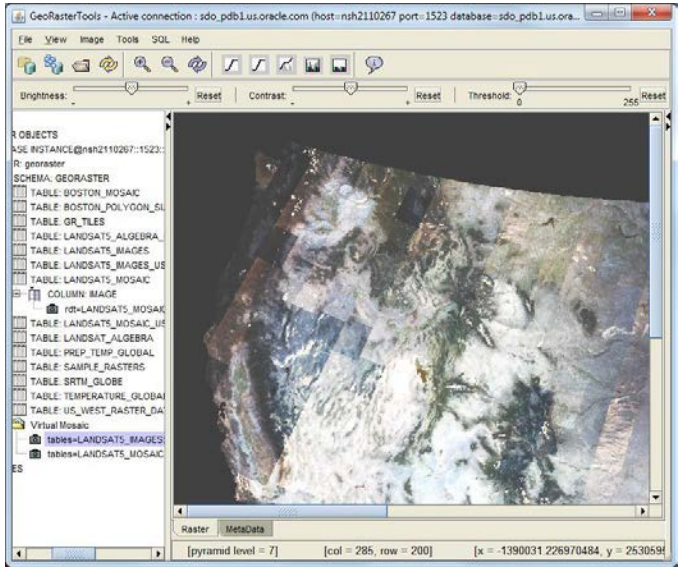
(Note, the source images are in different seasons of many years, so no color balancing is done on this mosaic. The mosaic is in SRID 5070. Imagery Courtesy USGS)

# Use Case 1: Optionally Create A Virtual Mosaic of the USA

- Users can zoom to display the **physical mosaic** very quickly.
- Users can also create many **virtual mosaics** based on the loaded image set or a subset of it and then dynamically zoom to display the source images covering the whole USA in different ways. For example,

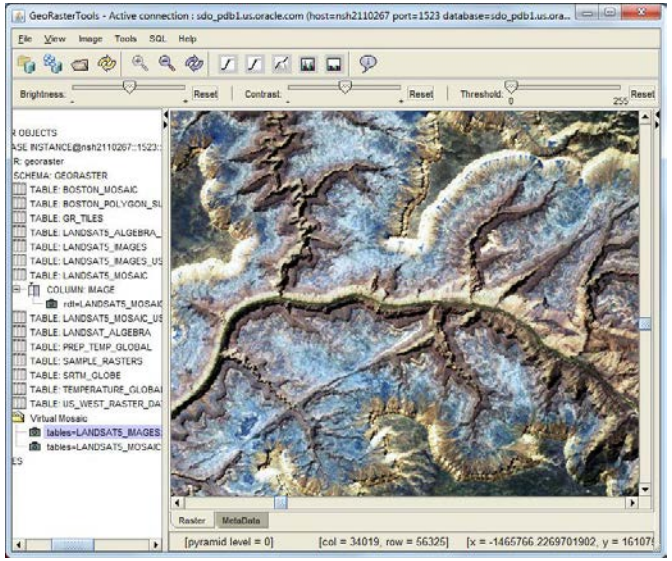


**Overview**



**Zoom In**

(Imagery Courtesy of USGS)



**Zoom In Further**

# GeoRaster Large-scale Mosaicking Capabilities

- **sdo_geor_aggr.mosaicSubset** to create persisted **physical mosaic**
- **sdo_geor_aggr.getMosaicSubset** to query **virtual mosaic** on-the-fly
- Parallelized and scalable
- Source images can be tables, views or a SQL statement
- Source images can be unrectified or in different CS and resolution
- Supports internal resampling, reprojection or rectification
- Supports gaps, no data, and overlapping regions
- Supports 8 common point rules (max, min, avg, LATEST, OLDEST ...)
- User defined priority for overlapping regions (Date or SQL ORDER BY)
- Color balancing

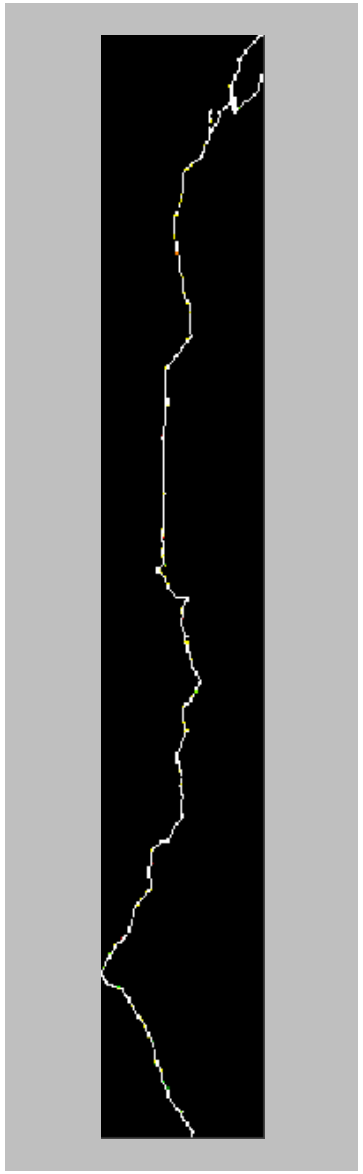
ORACLE

# Sample Use Case 2:

# Railroad Image Management

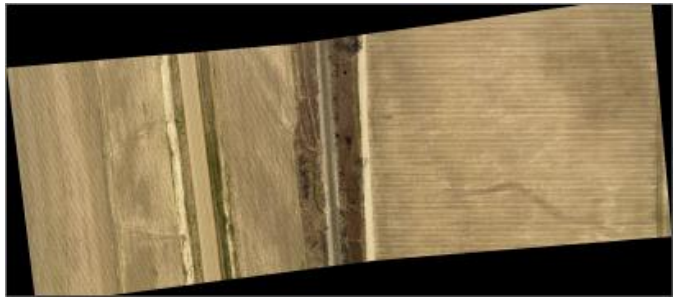
# Use Case 2: Railroad Image Management

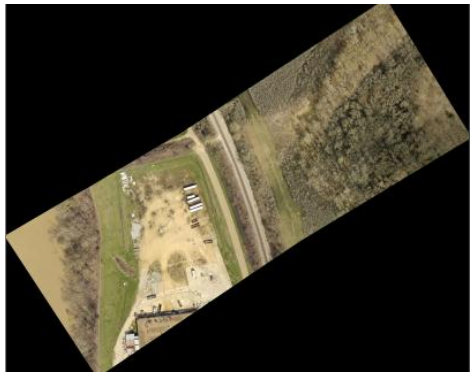
- **Goal:**
  - Store and manage terabytes of very high resolution imagery along railroad
  - Visualize the image dataset seamlessly as a virtual mosaic without preprocessing
  - High efficiency and high performance
  - Note this is a testing project, not a production use case
- **Testing Area:**
  - The target area covers the rail road in the whole state of **Mississippi**
  - Data Courtesy of **Canadian National Railway Company**



# Use Case 2: Railroad Image Management (Cont.)

- **Source Datasets:**
  - 4586 images in total. 160 ~ 400MB each, average about 300MB
  - 1.382 TB in total – uncompressed
  - The images are in geodetic space, very high resolution airborne photos averaging 2.5 – 3 cm
  - Most of the images have slightly different resolutions and have very different rotations
  - Neighboring images overlap each in drastically different ways. Some places have up to 8 overlapping images





(Data Courtesy of Canadian National Railway Company)

# Use Case 2: Railroad Image Management (Cont.)

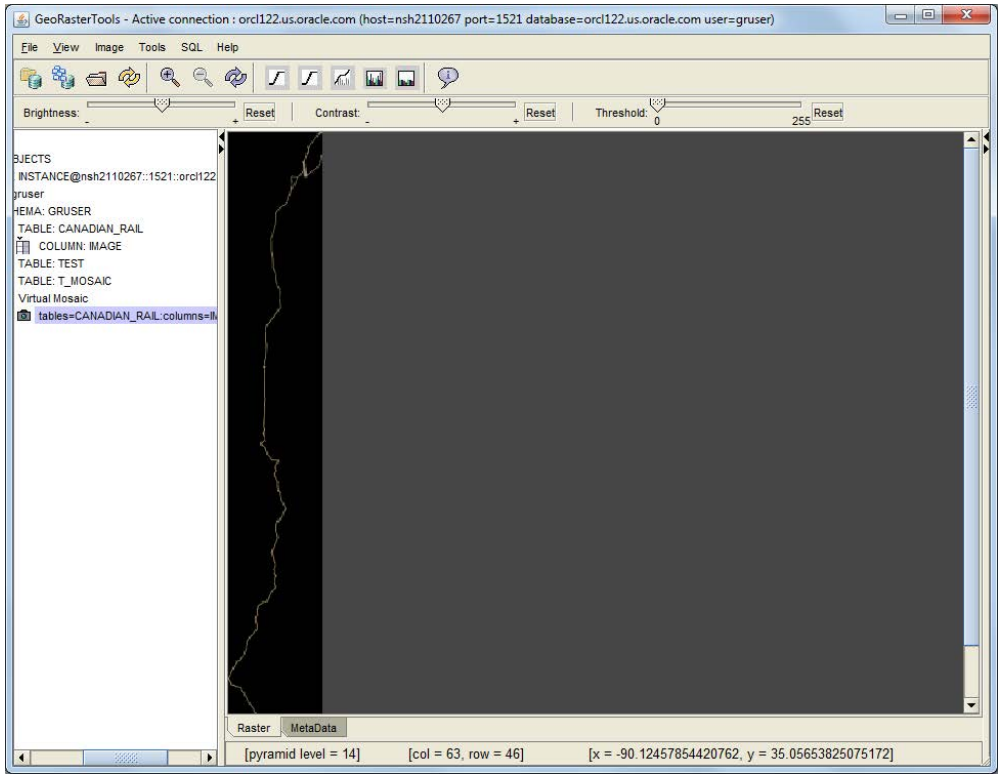
- **Processing Steps:**
  - Load all images into one GeoRaster table, concurrently in 5 batches: GeoRaster ETL
  - Generate pyramids:
    - Generate 7 pyramid levels for each image: sdo_geor.generatePyramid
    - Divide the railroad into 204 segments, create a mosaic at pyramid level 7 for each segment, then remove the 7$^{th}$ pyramid level from the original images: sdo_geor_aggr.mosaicSubset and sdo_geor.deletePyramid
    - Generate 7 pyramid levels for the mosaics of each segment
    - Create a single mosaic from the 7$^{th}$ pyramid level of the 204 segment mosaics and then remove the 7$^{th}$ pyramid level from each segment mosaic: sdo_geor_aggr.mosaicSubset and sdo_geor.deletePyramid
    - Generate full pyramids for the top mosaic
  - Create and populate the MIN_X_RES$ and MAX_X_RES$ columns in the GeoRaster table
  - Create spatial index on the GeoRaster column and optionally B-tree index on the MIN_X_RES$ column
  - Define the virtual mosaic as the whole table, i.e., the table name plus the column name in the GeoRaster viewer or MapViewer

# Use Case 2: Railroad Image Management (Cont.)

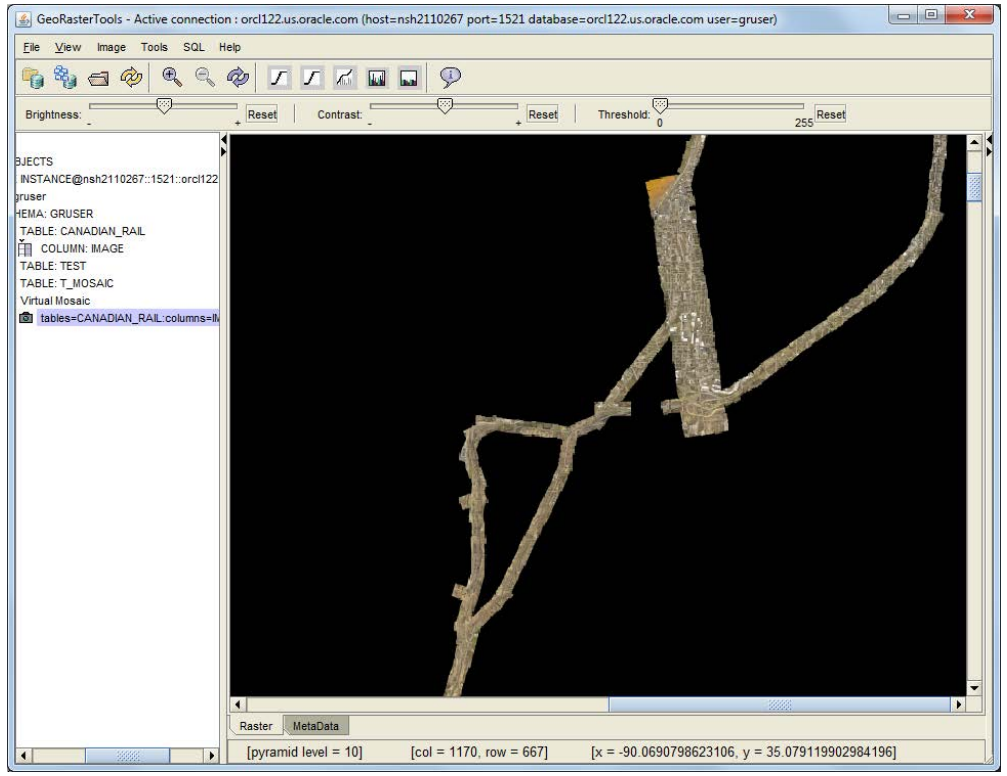
- **Results:**
  - A virtual mosaic of 4586 images without the need of any preprocessing or creating a large physical mosaic
  - With full pyramids, the actual storage size without compression is about **1.828 TB**.
  - The virtual mosaic has 10,258,311 rows, 1,513,411 columns, and 3 bands. That's equivalent to 46,575,122,126,463 pixels (i.e, **46.58 TB**) if they are stored as a physical storage.
  - Visualize the whole image set seamlessly as if they are a physical mosaic
  - Local display is VERY fast
  - Newly collected images can be easily loaded into the table and be mosaicked into display
- **Conclusion:**
  - GeoRaster can be easily used to build a single database to efficiently manage images for ALL railroads in North America or even in the whole world with great performance
  - Benefits: fast visualization, saving disk space, flexible display, dynamic update

ORACLE

# Use Case 2: Displaying Railroad Image Virtual Mosaic



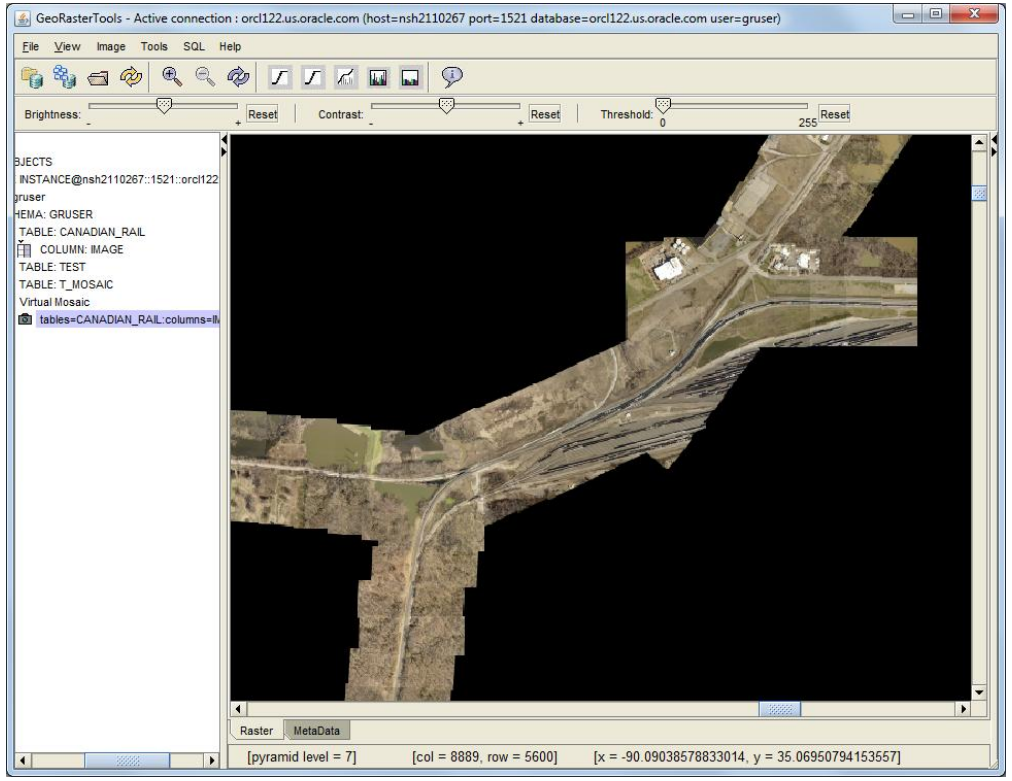

Pyramid Level 14



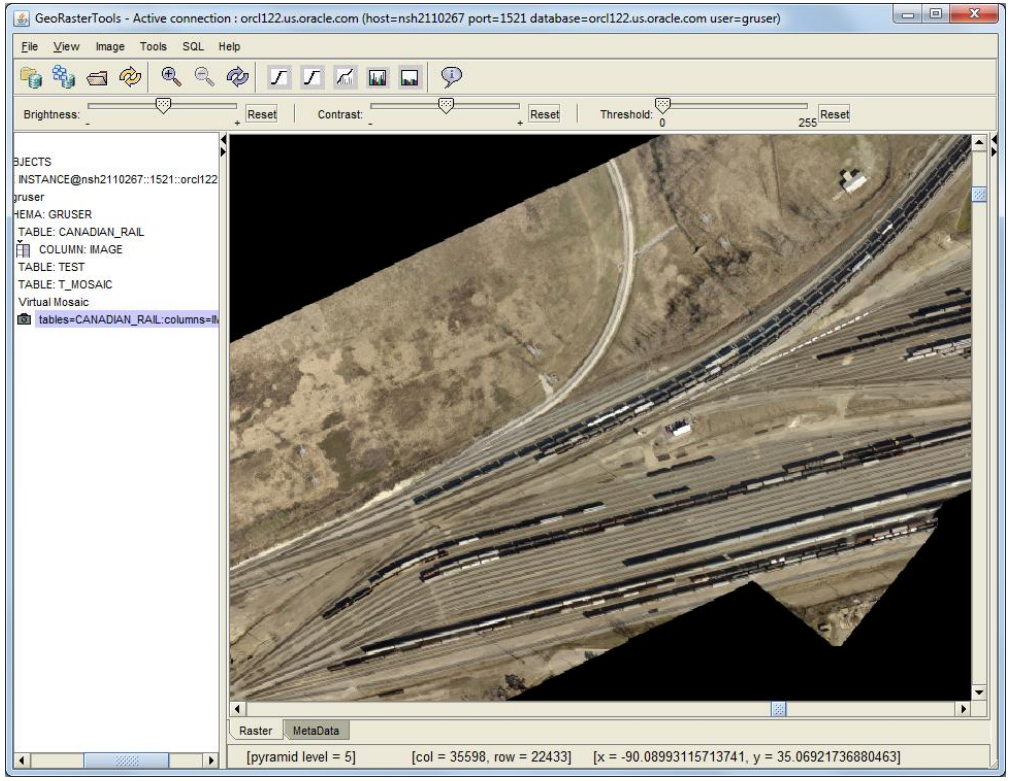

Pyramid Level 10

(Data Courtesy of Canadian National Railway Company)

# Use Case 2: Displaying Railroad Image Virtual Mosaic



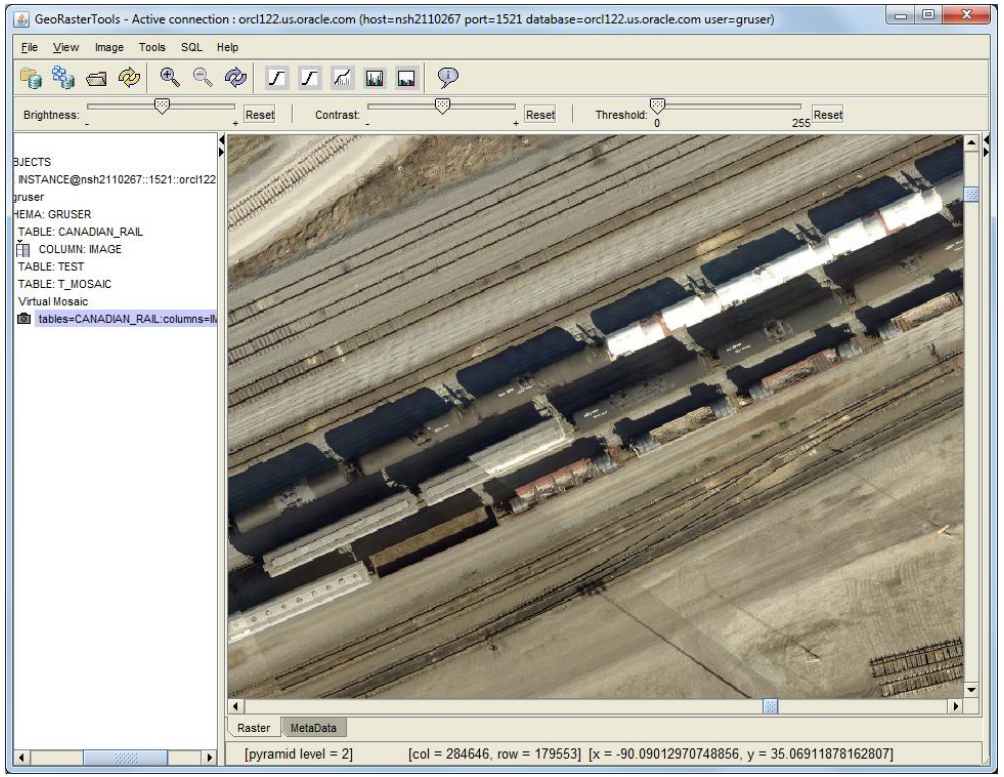

Pyramid Level 7



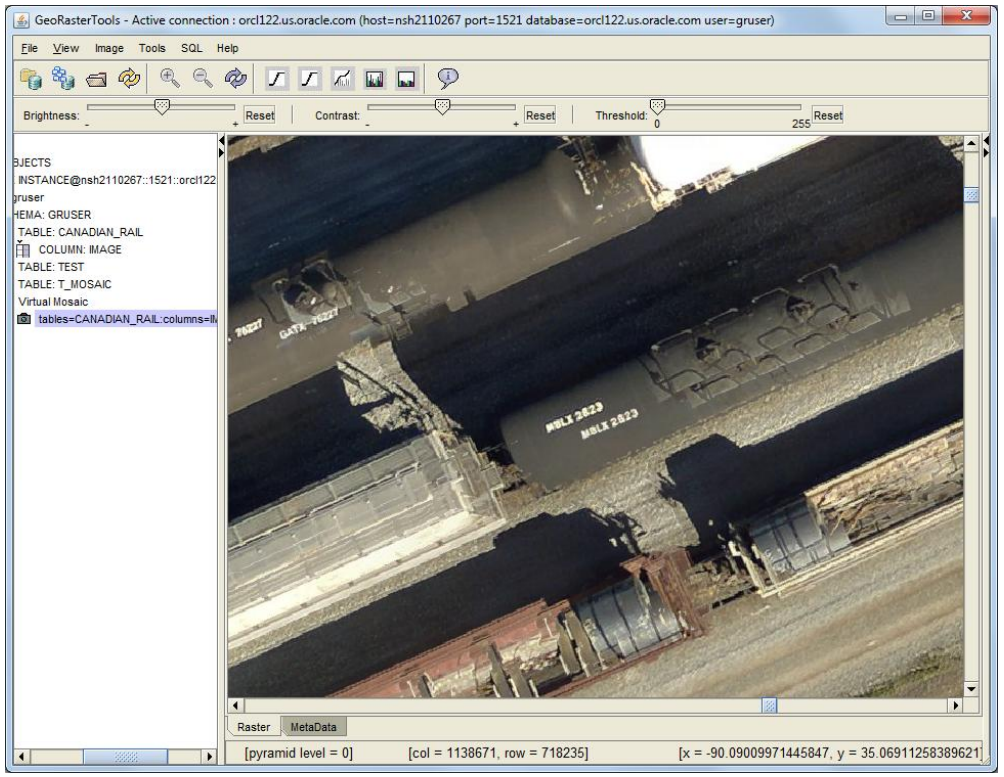

Pyramid Level 5

(Data Courtesy of Canadian National Railway Company)

# Use Case 2: Displaying Railroad Image Virtual Mosaic




Pyramid Level 3




Pyramid Level 0

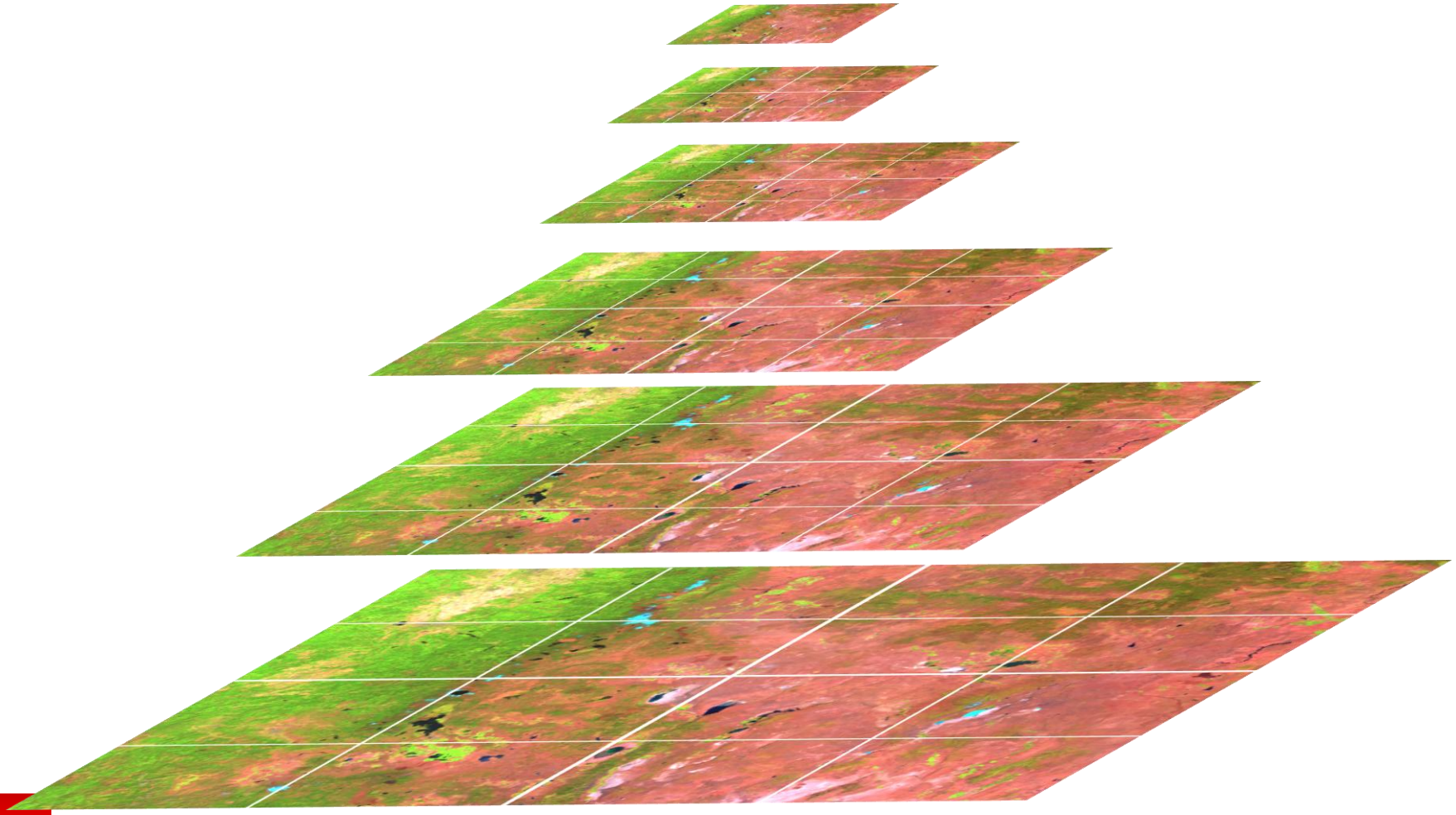
(Data Courtesy of Canadian National Railway Company)

# More on GeoRaster Virtual Mosaic

1. Further Explanation
2. How to Optimize
3. Benefits

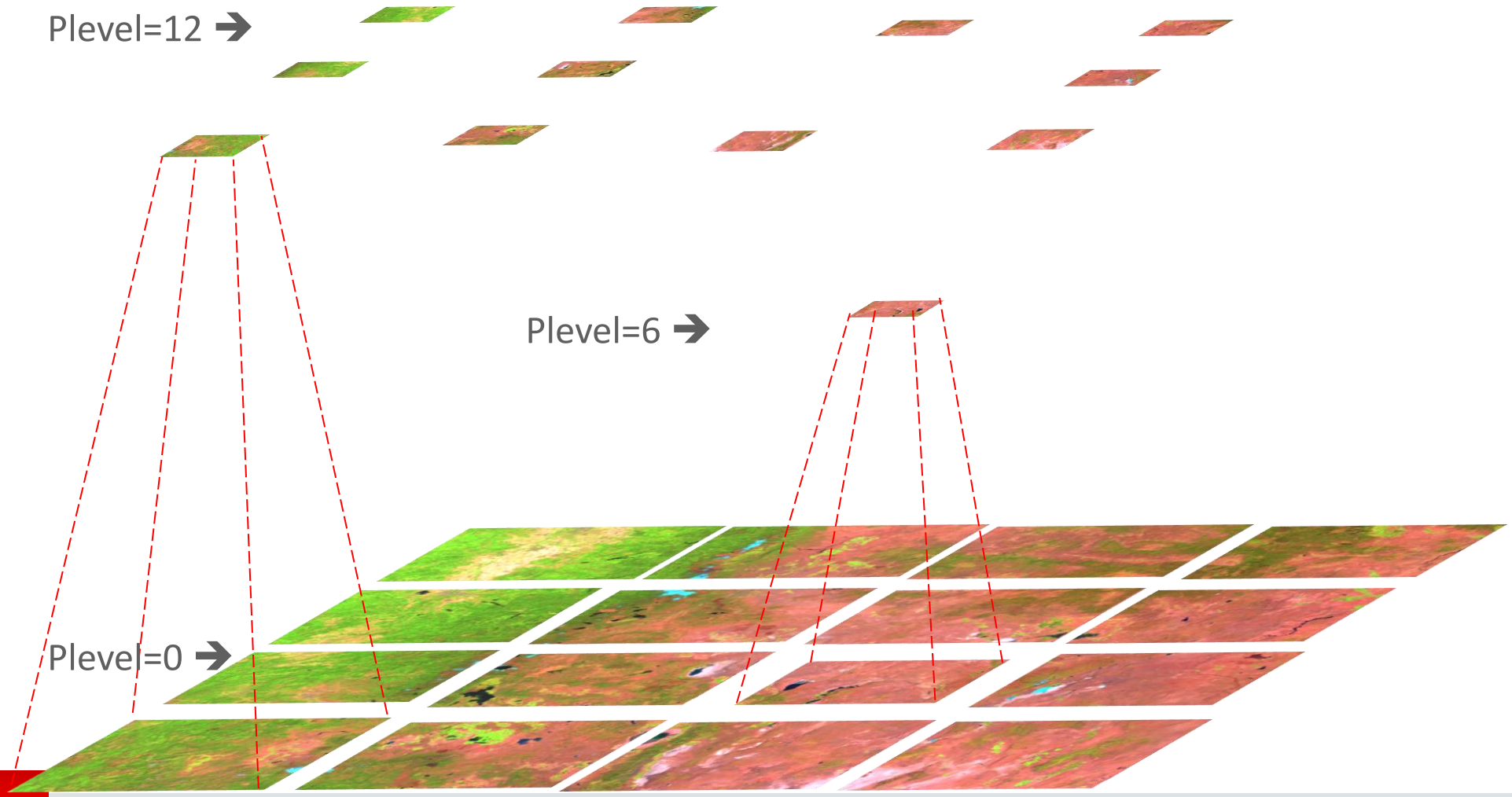
# Large Scale Virtual Mosaic and Visualization

- as if it’s a single image and the source images can be stored in different tables



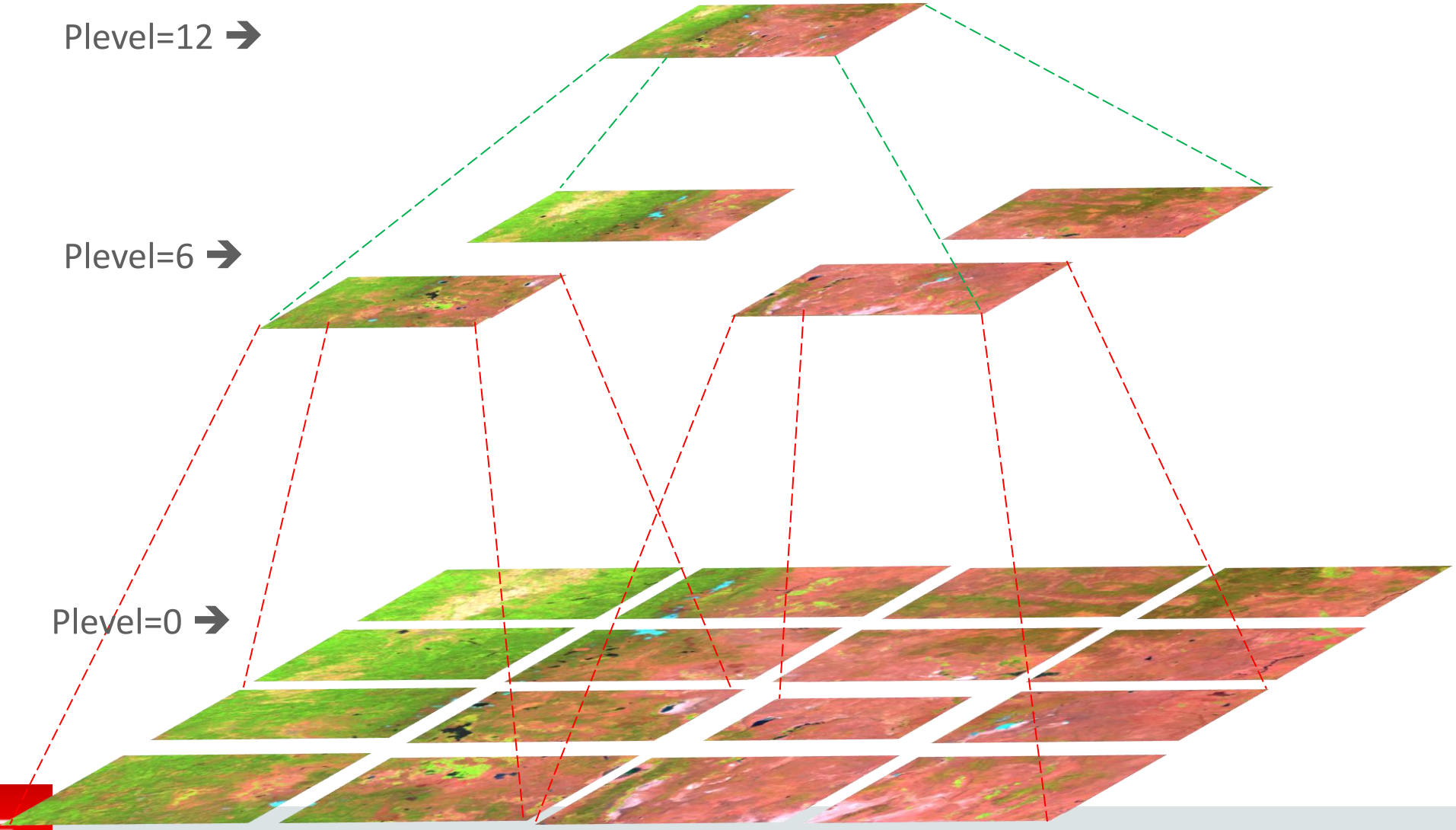
# Large Scale Virtual Mosaic without Optimization

## -- only pyramids are built on source images and zooming out is slow



# Optimizing Large Scale Virtual Mosaic for Fast Display

-- pre-generating multi-level regional physical mosaics and add resolution columns



# Benefits of Using Virtual Mosaics

1. **Save disk space:** Store large volume of imagery without making too many extra copies or filling in gaps
2. **Flexible modeling:** The same images can be displayed or removed in different virtual mosaics; Multiple mosaics on the same image set; different mosaicking/resampling approaches can be applied
3. **Dynamic update:** New images coming in and can be immediately displayed on the mosaic
4. **High speed:** no reason to mosaick DOQ's yet display them seamlessly as fast as physical mosaics
5. **Avoid resampling source data yet display them seamlessly:** Users may not want to resample and mosaick DEM's, particularly for large regions

ORACLE

# Sample Use Case 3:

# Agricultural Analysis of US West

# Use Case 3: Agricultural Analysis of US West

- **Goal:**
  - Create a geo database for the 5 states in US West: Washington, Oregon, Idaho, Nevada and California
  - Evaluate feasibility for certain agricultural activities
  - This is just a sample case to demonstrate the capabilities; The analysis and models are **hypothetical**
- **Source Datasets:**
  - Landsat 5 TM imagery: 133 images, 30-meter resolution, covering the US West. About 53 GB
  - Global SRTM DEM: 1 layer, 874 tiles, 3 arc second (92 meters) resolution, covering the whole world. 144001 x 432001 x 2 bytes = 124 GB
  - Global temperature data: 396 layers, a collection of 33 years of global temperature data, 1 layer for each month. About 336 MB.

# Use Case 3: Agricultural Analysis of US West (cont.)

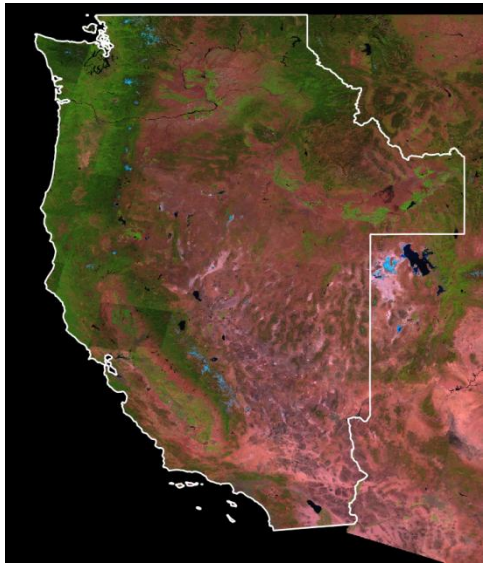
- **Processing Steps:**
  1. **Create the database:**
     - Load landsat images – 133 georaster objects: gdalbuildvrt, GeoRaster ETL
     - Load and mosaic 874 SRTM files into 1 georaster object – gdalbuildvrt, gdal_translate
     - Load temperature dataset – 396 georaster objects: GeoRaster ETL
     - Load state boundary and create the five-state polygon for the study area: sdo_geom.sdo_union
  2. **Create a raster stack for the study area:**
     - Mosaic and crop landsat images – one georaster with 7 bands: sdo_geor_aggr.mosaicSubset. The mosaic is 76263 x 50434 x 7 = 27 GB, cropped mosaic is 41170 x 34222 x 7 bands = 9.9 GB
     - Crop SRTM dataset – one georaster with 1 layer: sdo_geor.subset
     - Merge Temperature dataset – one georaster with 396 layers: sdo_geor.mergeLayers
     - Enlarge the SRTM and crop it: sdo_geor.rectify, 41170 x 34222 x 4 bytes = 5.7 GB
     - Enlarge the temperature dataset and crop it: sdo_geor.rectify, 41170 x 34222 x 4 bytes = 5.7 GB
     - Note, cropping may be done after step 3

# Use Case 3: Agricultural Analysis of US West (cont.)

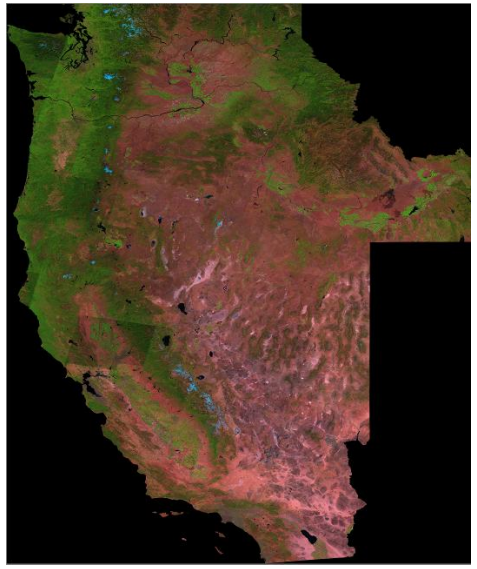
- **Processing Steps:**
  3. **Analyze rasters to generate thematic layers**
     - NDVI, TCT(Greenness and Wetness): sdo_geor.rasterMathOp
     - Slope, Aspect, Roughness: gdaldem
     - Average Temperature, Temperature Anomaly: sdo_geor.rasterMathOp
     - In total, 57 GB in size. Note, step 3 can be done before step 2
  4. **Apply cartographic and agricultural models**
     - A variety of cartographic and agricultural models are applied on the raster stack
     - A cropping area can be used to limit the analysis for any AOI
     - This can also be called content-based analytical database queries , persistently or on-the-fly
- **Results:**
  - Agricultural feasibility map for the whole region or for small regions on-demand
  - Farm and forest assessment map, etc. etc.

ORACLE

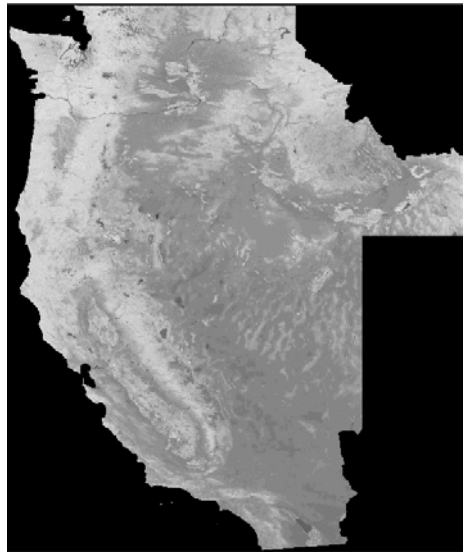
# Use Case 3: NDVI and TCT



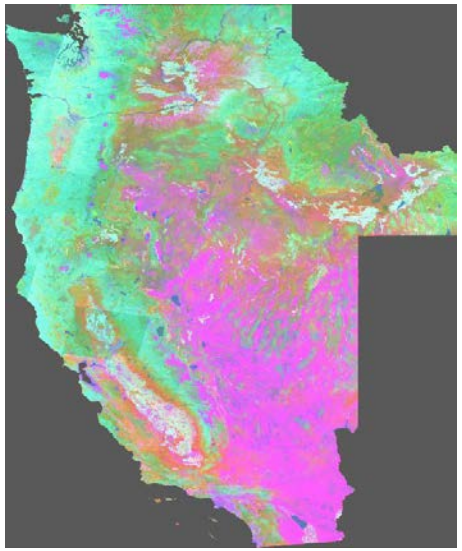
Target Area



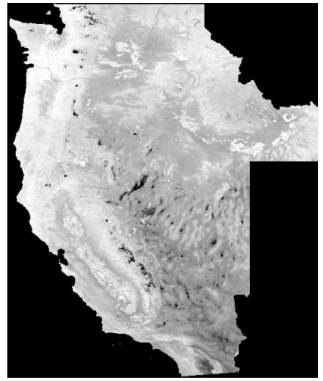
Mosaic



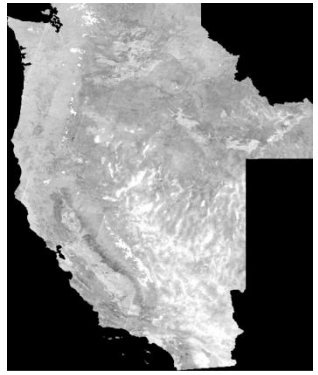
NDVI



TCT



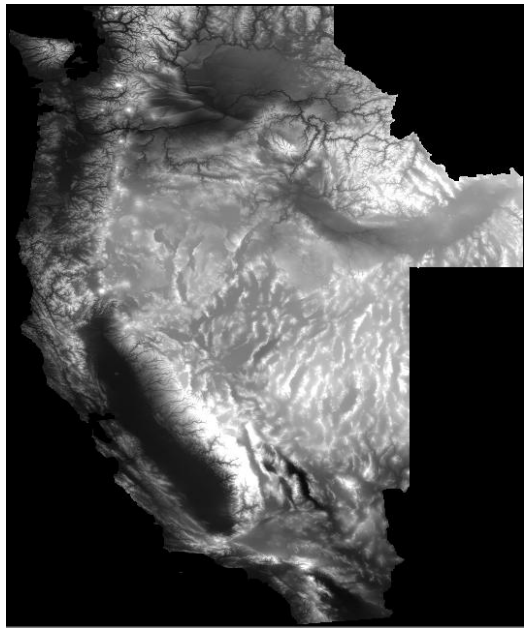
Greenness



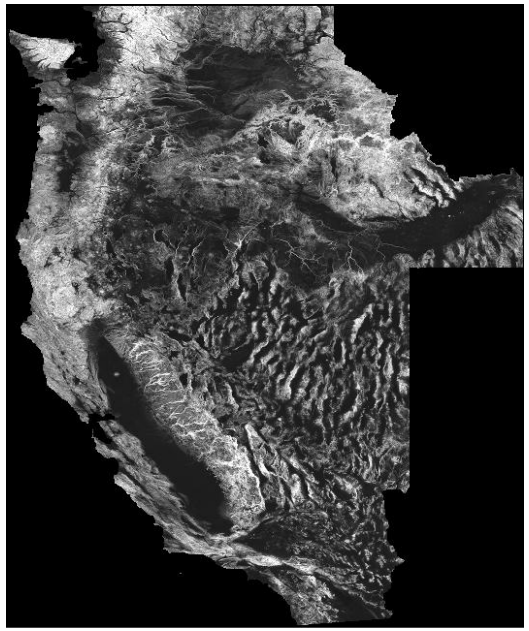
Wetness

(Landsat TM Image Courtesy of USGS)

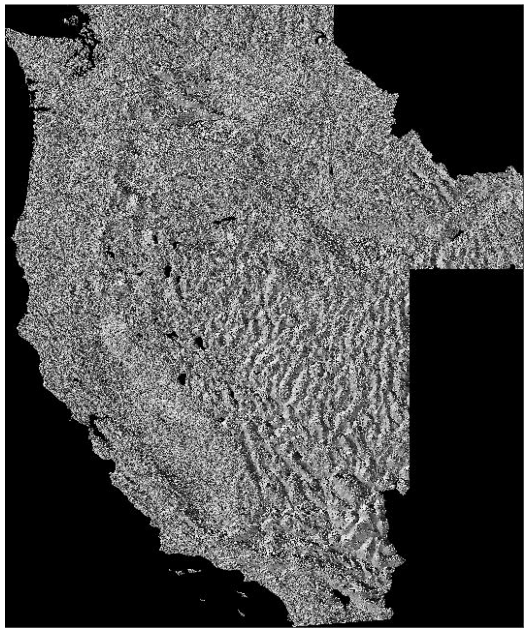
# Use Case 3: Terrain Analysis



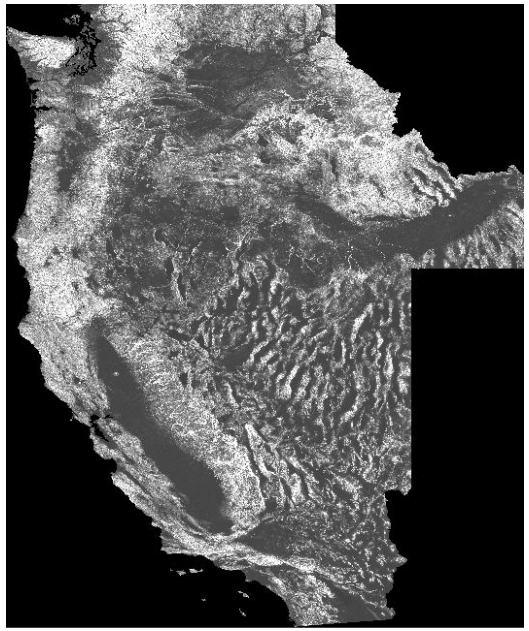
SRTM DEM



Slope



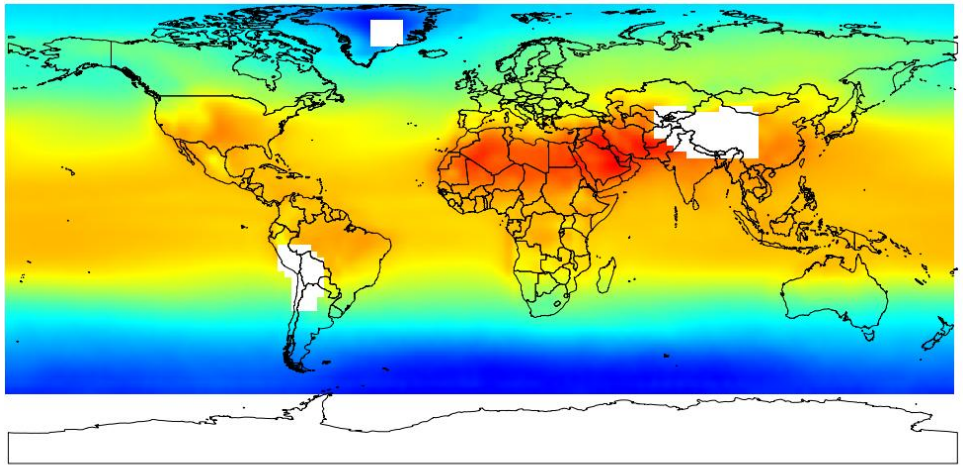
Aspect



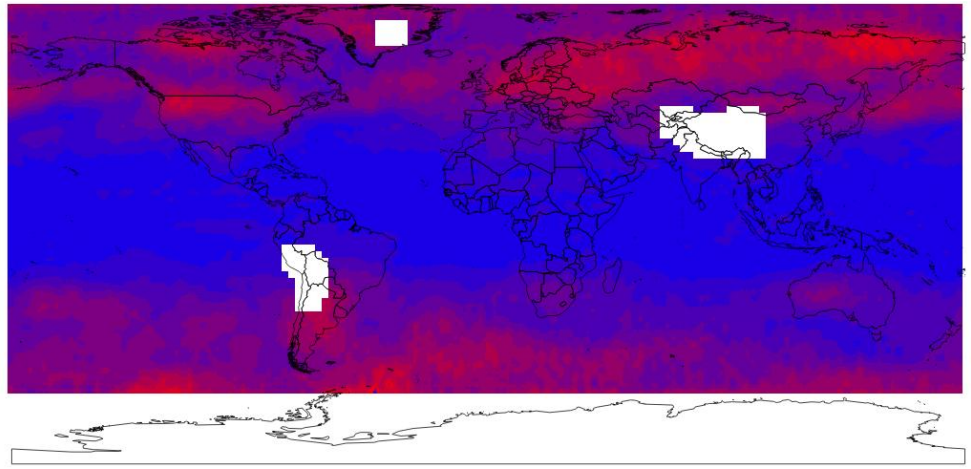
Roughness

(SRTM Data Courtesy of JPL, NASA)

# Use Case 3: Historical Temperature Analysis



June Average Temperature over 33 Years



June Temperature Mean Absolute Deviation Over 33 Years

June Average Temperature over 33 Years

(Source Temperature Data Courtesy of Remote Sensing Systems - http://www.remss.com/)

# Use Case 3: Terrain Based Feasibility Modeling

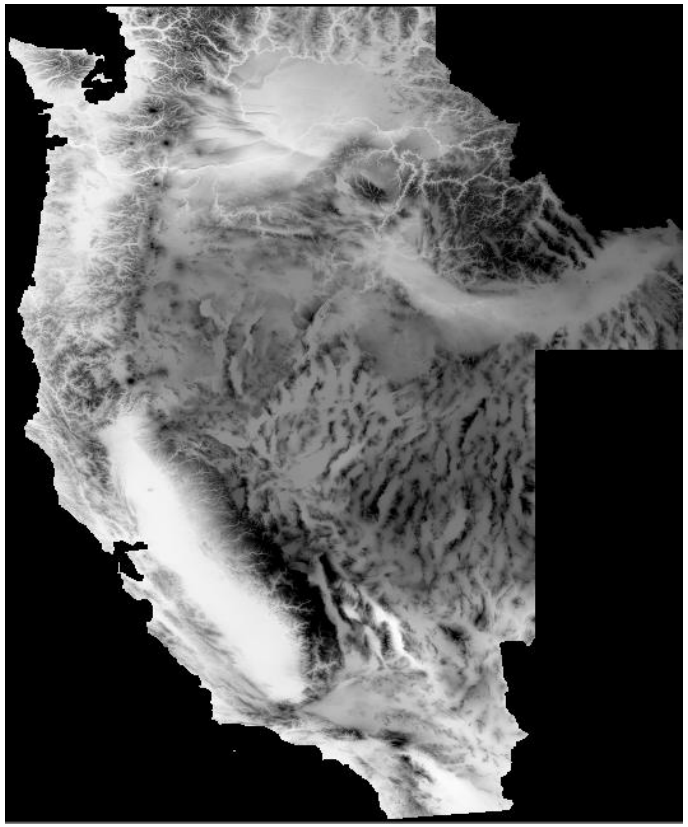
Note, this model is purely hypothetical and only for demonstration purpose.

**INPUT :**

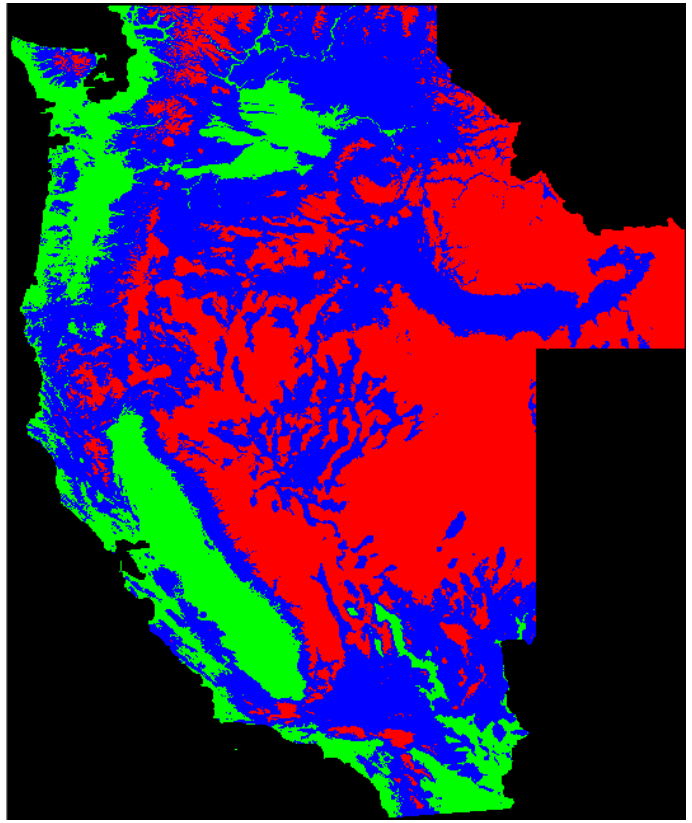
{1,0} - DEM
{2,0} - SLOPE
{3,0} - ROUGHNESS

**MODEL :**

( 1.0 - ( ( **{1,0}** + 291.0 ) / ( 4624.0 ) ) * 90.0 ) +
( 1.0 - ( ( **{2,0}** ) / (77.2764 ) ) * 10.0 ) +
( 1.0 - ( ( **{3,0}** ) / ( 905.0 ) ) * 5.0 )



Model Output



Classification

# Use Case 3: Agricultural and Forest Assessment

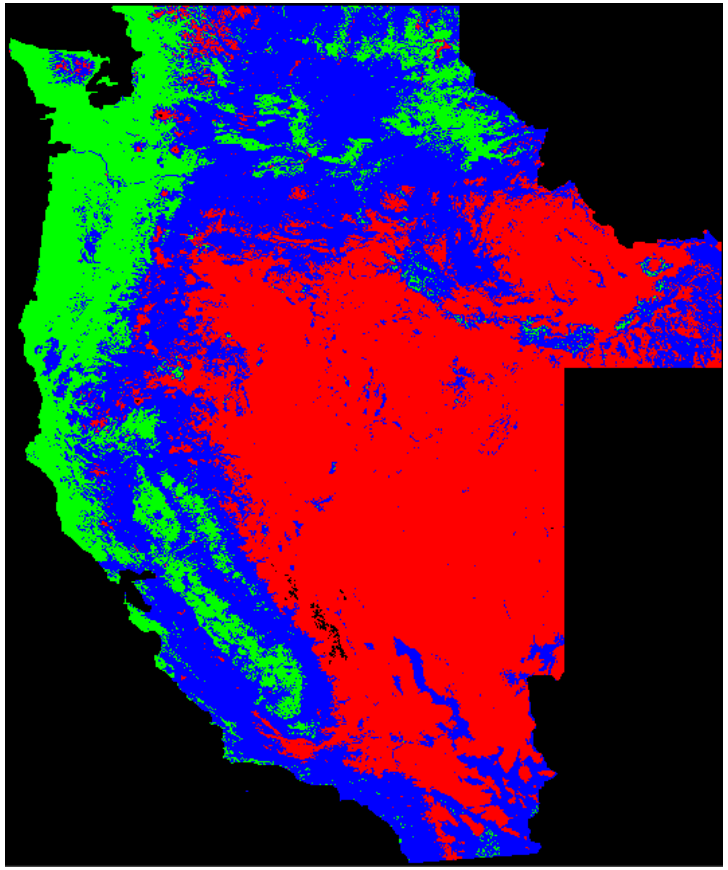
Note, this model is purely hypothetical and only for demonstration purpose.

**INPUT :**

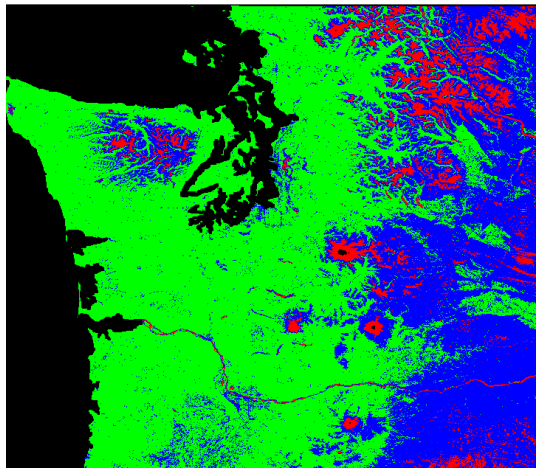
{0,0} - NDVI
{1,0} - DEM
{2,0} - SLOPE
{3,0} - ROUGHNESS

**MODEL :**

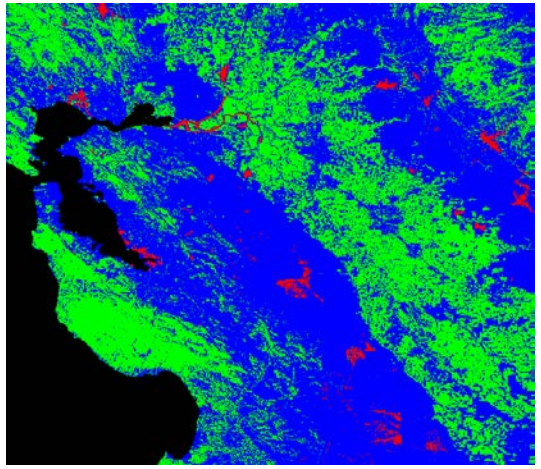
( **{0,0}** + 1.0 ) / ( 2.0 ) * 150.0 ) +
( 1.0 - ( ( **{1,0}** + 291.0 ) / ( 4624.0 ) ) * 90.0 ) +
( 1.0 - ( ( **{2,0}** ) / (77.2764 ) ) * 10.0 ) +
( 1.0 - ( ( **{3,0}** ) / ( 905.0 ) ) * 5.0 )



Model Output



Seattle, WA



San Francisco, CA

# Program Agenda

1. GeoRaster Overview and Major Components
2. GeoRaster Workflow and Sample Use Cases
3. **High Performance Raster Data Processing**
4. Application Development Framework
5. Demonstration and Summary

# GeoRaster High Performance Processing

1. **Locality Computing**

   In-database processing

   Moving image processing to database (not the reverse)

2. **Parallel Processing**

   Leveraging multi-CPU or multi-core computers

3. **Concurrent Processing**

   Optimized multi-tasking and batch processing

# Locality Computing (In-Database Processing)

- **Reasons:**
  A. Image and raster data are Big Data – data movement issue
  B. Image and raster data processing is data intensive
  C. Analytical queries usually involves multiple layers and results in one layer or less data – only results need to be shipped
- **Benefits:**
  A. Greater Performance
  B. True Security
  C. Greater Scalability
- **Conclusion:**
  All processing can be efficiently done in the database!

ORACLE

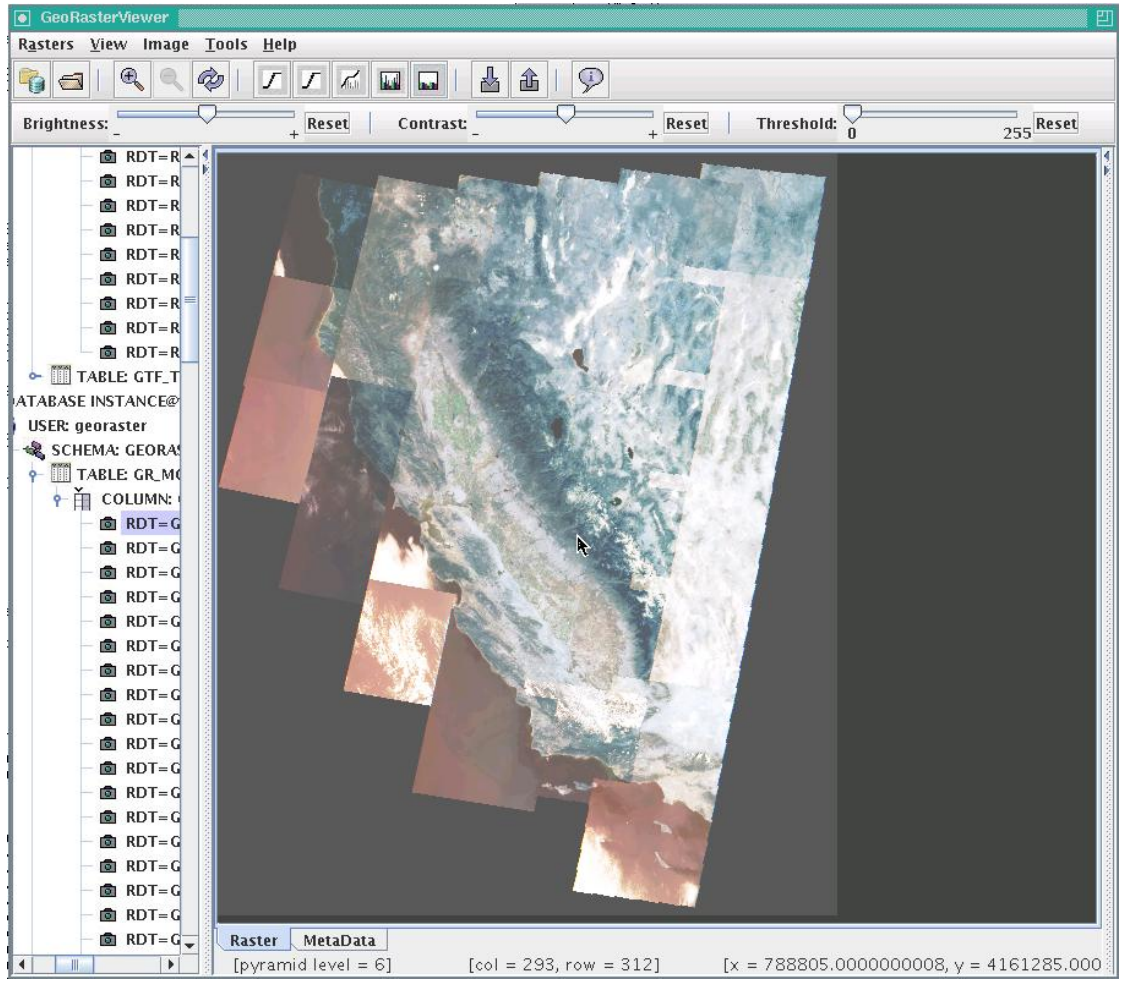
# Testing: Mosaic of Landsat Images

Source images:

31 Landsat 5 Level 1T TM images, from *the U.S. Geological Survey*. The total size of the image set is 11.9 GB. 15 images are in UTM zone 11N and 16 images are in UTM zone 10N projection. They overlap each other.

Output mosaic:

about 7.7GB in size



# Performance: GeoRaster Mosaicking vs GDAL Mosaicking

Test Data: the same 31 Landsat TM images. Total size 11.9GB

Test Machine: x4170 M2 Sun Server with 24 CPUs and 8 SAS disks.

GeoRaster is much faster:
- serial: 2.3x faster
- parallel: 11.3x faster

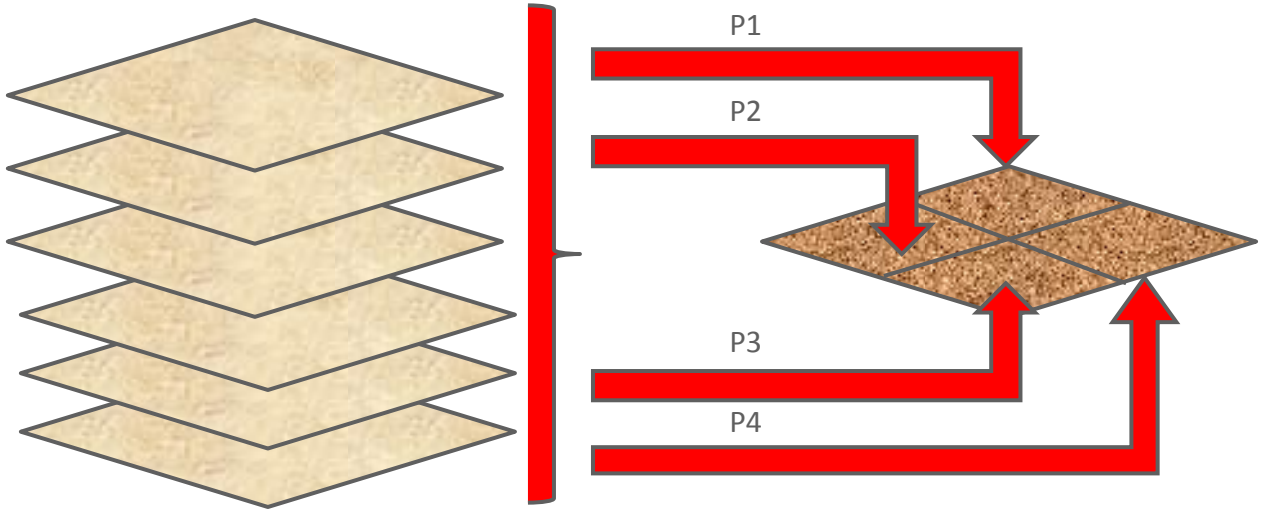
Table. GDAL and GeoRaster Image Mosaicking Time in Minutes

| | Serial | Parallel |
|---|---|---|
| GDAL mosaic | 43.12 | 42.1 |
| GeoRaster mosaic | 18.65 | 3.73 |

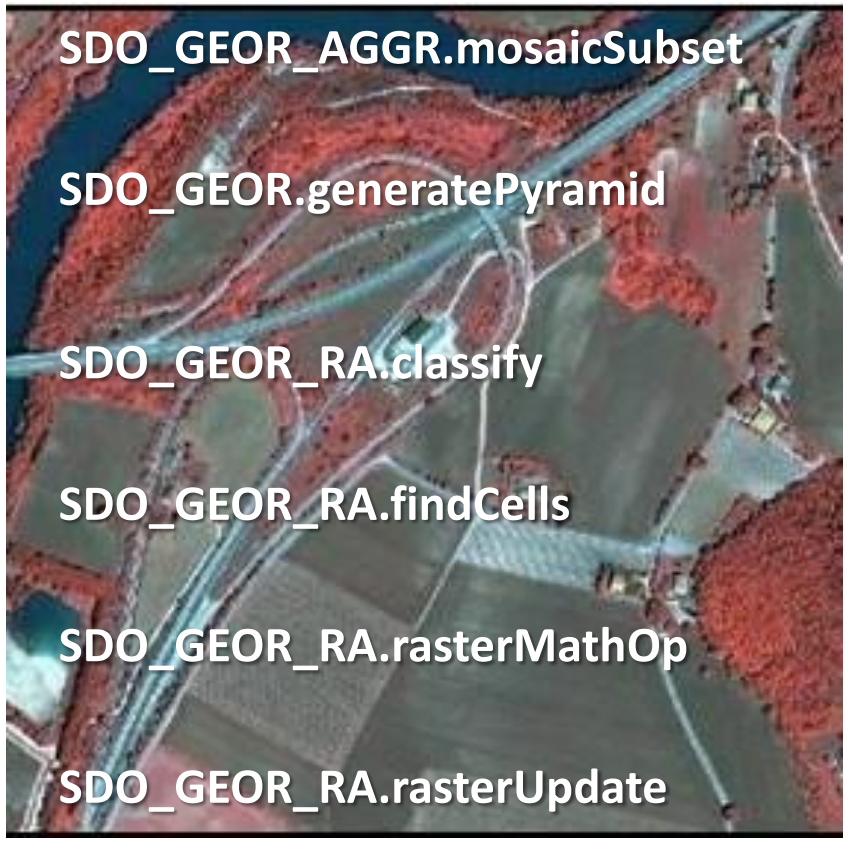
(Xie, et al. In-database Image Processing in Oracle Spatial GeoRaster. In: *the Proceedings of ASPRS 2013 Annual Conference*, Baltimore, Maryland, March 24-28, 2013)

# Parallel Processing

1. Leverage the Oracle parallel execution engine
2. The output region is divided into sub regions and the sub regions are processed in parallel
3. Degree of Parallelism (DoP) can be controlled



# Parallelized GeoRaster Operations




Raster functions THAT CAN PARALLELIZE

SERIAL Operations perform up to 3x faster than 11g Release

Scale to over 100 times faster on highly parallel systems

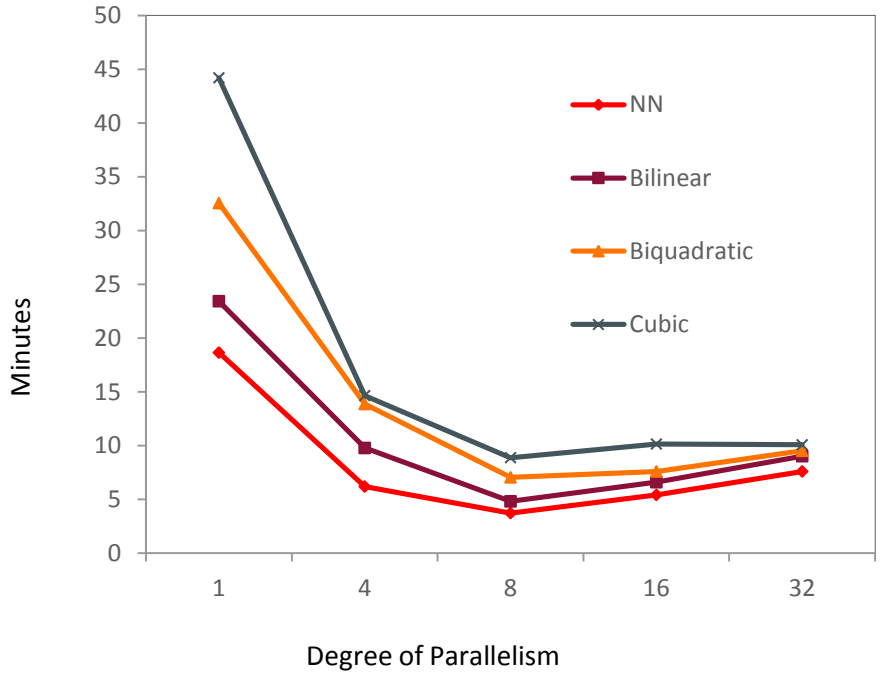
# Parallelized Mosaicking Performance

Test Data: the same 31 Landsat TM images. Total size 11.9GB

Test Machine: x4170 M2 Sun Server with 24 CPUs and 8 disks

Speed up:

- Up to 5x faster than serial
- 18.65m down to 3.75m (NN)
- 44.2m down to 8.88m (Cubic)
- many more times faster (if more overlappings)




**Parallelized Mosaicking with Different DOP**

(Xie, et al. In-database Image Processing in Oracle Spatial GeoRaster. In: *the Proceedings of ASPRS 2013 Annual Conference*, Baltimore, Maryland, March 24-28, 2013)

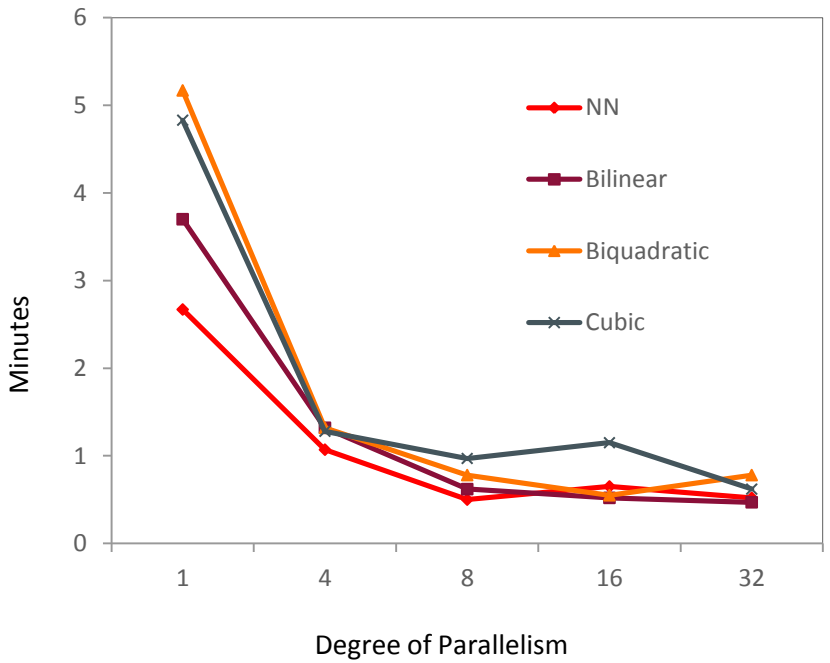
# Parallelized Pyramiding Performance

Test Data: the mosaic of 31 TM images. Total size 7.7GB

Test Machine: x4170 M2 Sun Server with 24 CPUs and 8 disks

Speed up:

- Up to 10x faster than serial
- takes only 0.47m to 0.62m
- about 30x faster than 11g




**Parallelized Pyramiding with Different DoP**

(Xie, et al. In-database Image Processing in Oracle Spatial GeoRaster. In: *the Proceedings of ASPRS 2013 Annual Conference*, Baltimore, Maryland, March 24-28, 2013)

# Concurrent Processing (Multitasking)

1. Easy to script batch processing with PL/SQL
2. Optimized load balancing for concurrent processes in Oracle database
3. Leveraging modern IT infrastructure
4. Either single machine or a cluster of machines (RAC) or Exadata database machine

(Xie, et al. Using Enterprise Grid Computing Technologies to Manage Large-scale Geoimage and Raster Databases. In: *the Proceedings of ASPRS 2006 Annual Conference*, Reno, Nevada, May 1 – 5, 2006)

ORACLE

# Concurrent Mosaicking Performance

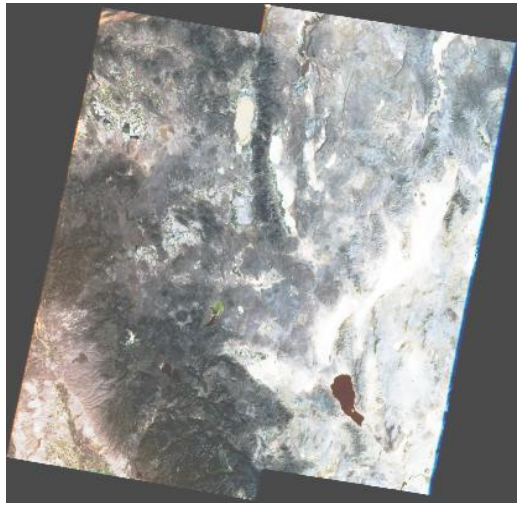
Test Data: the mosaic of 4 TM images (from USGS). 2 images are in UTM 11N and 2 are in UTM 10N. Total size 1.52 GB. Mosaic size is 1.2 GB

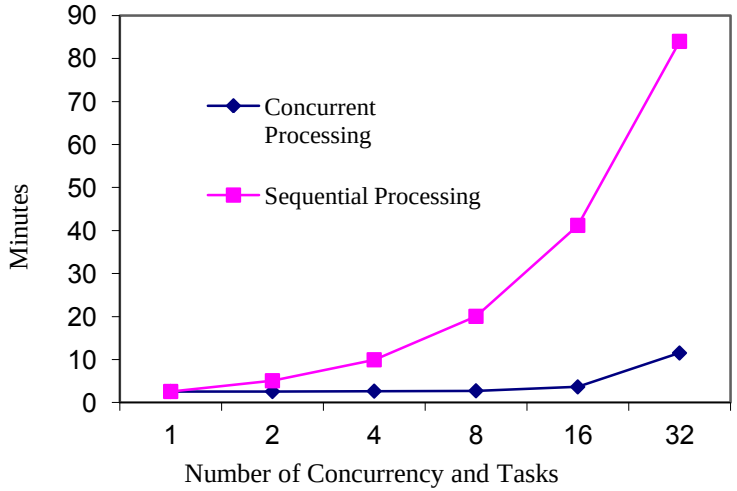
Test Machine: x4170 M2 Sun Server with 24 CPUs and 8 disks

Speed up:

- Up to 11x faster than serial
- only 3.65m to create 16 mosaics

(Xie, et al. In-database Image Processing in Oracle Spatial GeoRaster. In: *the Proceedings of ASPRS 2013 Annual Conference*, Baltimore, Maryland, March 24-28, 2013)






**Serial vs Concurrent Mosaicking**

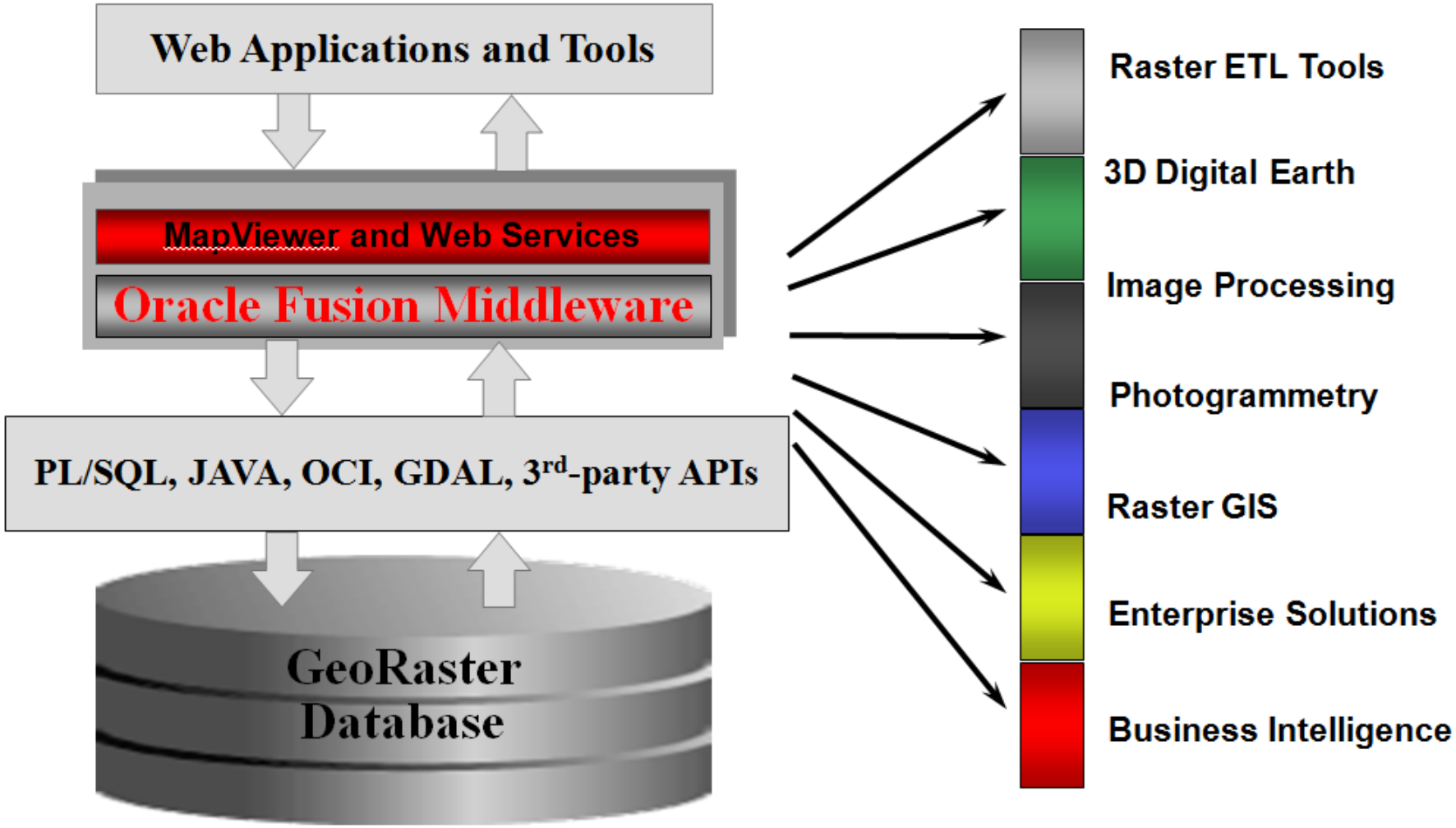
# Oracle Exadata Database Machine

- Database Grid
  - Intel-based database servers
  - Oracle Linux or Solaris 11
  - Oracle Database 11g R2
  - 10 Gig Ethernet (to data center)
- Storage Grid
  - Intel-based storage servers
  - Up to 336 terabytes raw disk
  - 5.3 terabytes Flash storage
  - Exadata Storage Server Software
- InfiniBand Network
  - Internal connectivity (40 Gb/sec)

# Program Agenda

1. GeoRaster Overview and Major Components
2. GeoRaster Workflow and Sample Use Cases
3. High Performance Raster Data Processing
4. **Application Development Framework**
5. Demonstration and Summary

# GeoRaster Stack and Application Framework



# Program Agenda

1. GeoRaster Overview and Major Components
2. GeoRaster Workflow and Sample Use Cases
3. High Performance Raster Data Processing
4. Application Development Framework
5. **Demonstration and Summary**

# DEMONSTRATION

Simple GeoRaster Workflow

Batch Loading

Virtual Mosaic and Physical Mosaic

Cartographic Modeling

# Summary – GeoRaster in a Nutshell

- **A Powerful Raster Database Management System**
  - Integrated raster data model and unique design
  - Highly **scalable** and truly secure enterprise database management
  - Special raster data query and manipulation capabilities and massive data processing
- **A High Performance Engine for Image Processing and Raster Analytics**
  - In-database processing (greater performance, better scalability and true security)
  - Parallel processing (greater performance)
  - Concurrent Processing (greater performance, multitasking becomes easy)
- **A Powerful Platform for Various Applications**
  - Raster databases, image archives, and SDI
  - Image Server for the web and 3D Earth visualization
  - Terrain analysis, environmental modeling, forest management, precision farming, insurance
  - And numerous other applications

ORACLE

# The Spatial and Graph SIG

- The SIG promotes interaction and communication that can drive the market for spatial technology and data
- Members connect and exchange knowledge via online communities and at annual conferences and events

- Meet us at the Summit
  - Morning Receptions
    - Tuesday and Wednesday / 7:45 to 8:30 a.m. / Registration Area
  - Birds of a Feather Session
    - Wednesday / 12 to 1 p.m. / Auditorium – Look for “Spatial and Graph SIG” table
- Join us online
  - LinkedIn (search for “LinkedIn Oracle Spatial”)
  - Google+ (search for “Google+ Oracle Spatial”)
  - IOUG SIG (sign up for free membership through www.ioug.org)
  - OTN Spatial – Communities (search for “Oracle Spatial and Graph Community”)
- Contact the Board at oraclespatialsig@gmail.com

# Thank You!