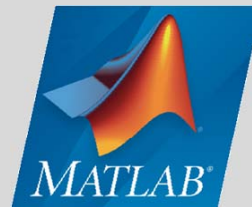
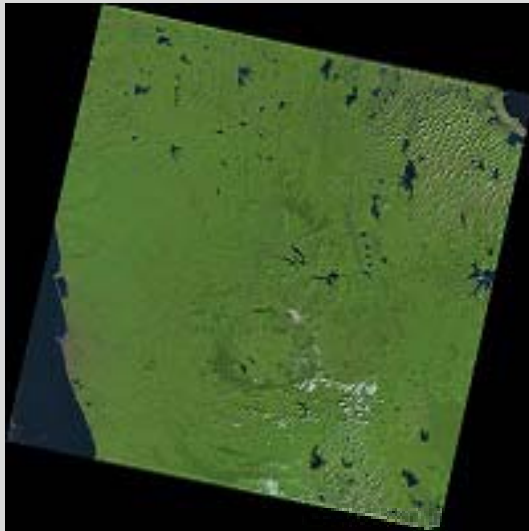


# 132 -GIS Seminar – 2015.10.22

Division of Spatial Information Science

## Handling Landsat Images with Matlab



	834	835	836	83
3987	165	162	154	83
3988	169	163	152	
3989	167	165	151	
3990	168	173	154	
3991	158	163	150	
3992	153	166	164	1
3993	161	176	169	14
3994	170	180	168	17
3995	173	179	178	175
3996	176	169	162	170
3997	179	163	151	162
3998	177	161	157	156
3999				161



Malinda Siriwardana, Prof. Yuji Murayama

University of Tsukuba  
Graduate School of Life and Environmental Science



# Introduction

- What is Landsat ?



The image shows a screenshot of the USGS Landsat website homepage. At the top, there is a browser address bar with the URL <http://landsat.usgs.gov/>. Below the address bar is a navigation bar with the USGS logo and the tagline "science for a changing world". The main content area features a large image of a satellite in space. Below the image, there is a section titled "Landsat Missions" with two main navigation buttons: "Home" and "Mission Headlines". Under "Home", there are links for "About Landsat", "Gallery", and "Science". Under "Mission Headlines", there is a notice dated "October 20, 2015" regarding "Possible Connectivity Issues Thursday, Oct 22, 2015". The notice states: "This site may have intermittent access Thursday October 22, 2015 between 6 pm and 9 pm CDT during network upgrades. We apologize for any inconvenience this may cause."

**USGS**  
*science for a changing world*

## Landsat Missions

**Home**      **Mission Headlines**

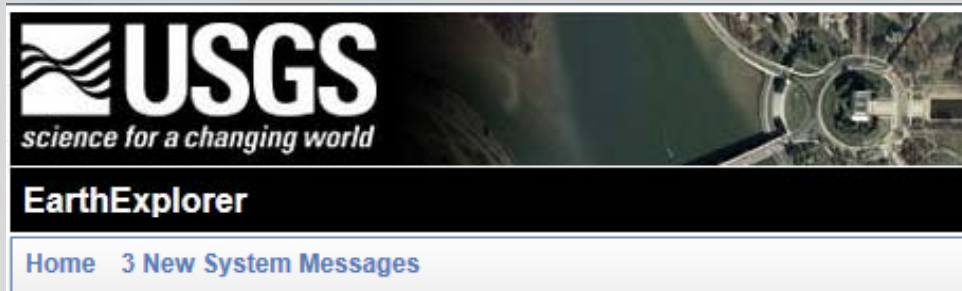
**About Landsat**  
**Gallery**  
**Science**

**October 20, 2015 – Possible Connectivity Issues Thursday, Oct 22, 2015**

This site may have intermittent access Thursday October 22, 2015 between 6 pm and 9 pm CDT during network upgrades. We apologize for any inconvenience this may cause.

# Introduction

- Where can I download?



**Get Data**

View and Get Color Images - [LandsatLook Viewer](#)

Browse and Download Data - [GloVis](#)

Search and Bulk-Download Data - [EarthExplorer](#)

Search Criteria | Data Sets | Additional Criteria | Results

### 1. Enter Search Criteria

To narrow your search area: type in an address or place name, enter coordinates or click the map to define your search area (for advanced map tools, view the [help documentation](#)), and/or choose a date range.

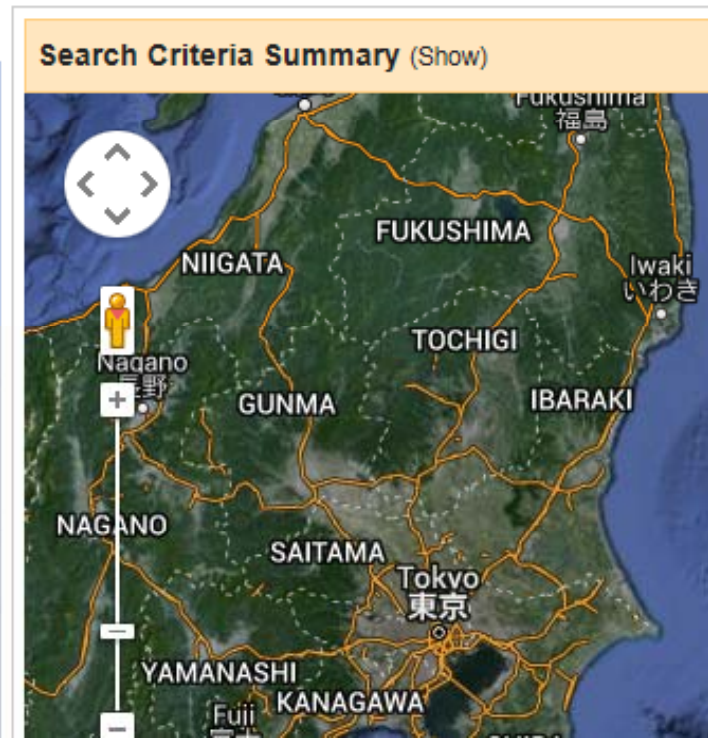
**Address/Place** | Path/Row | Feature | Circle

Show Clear

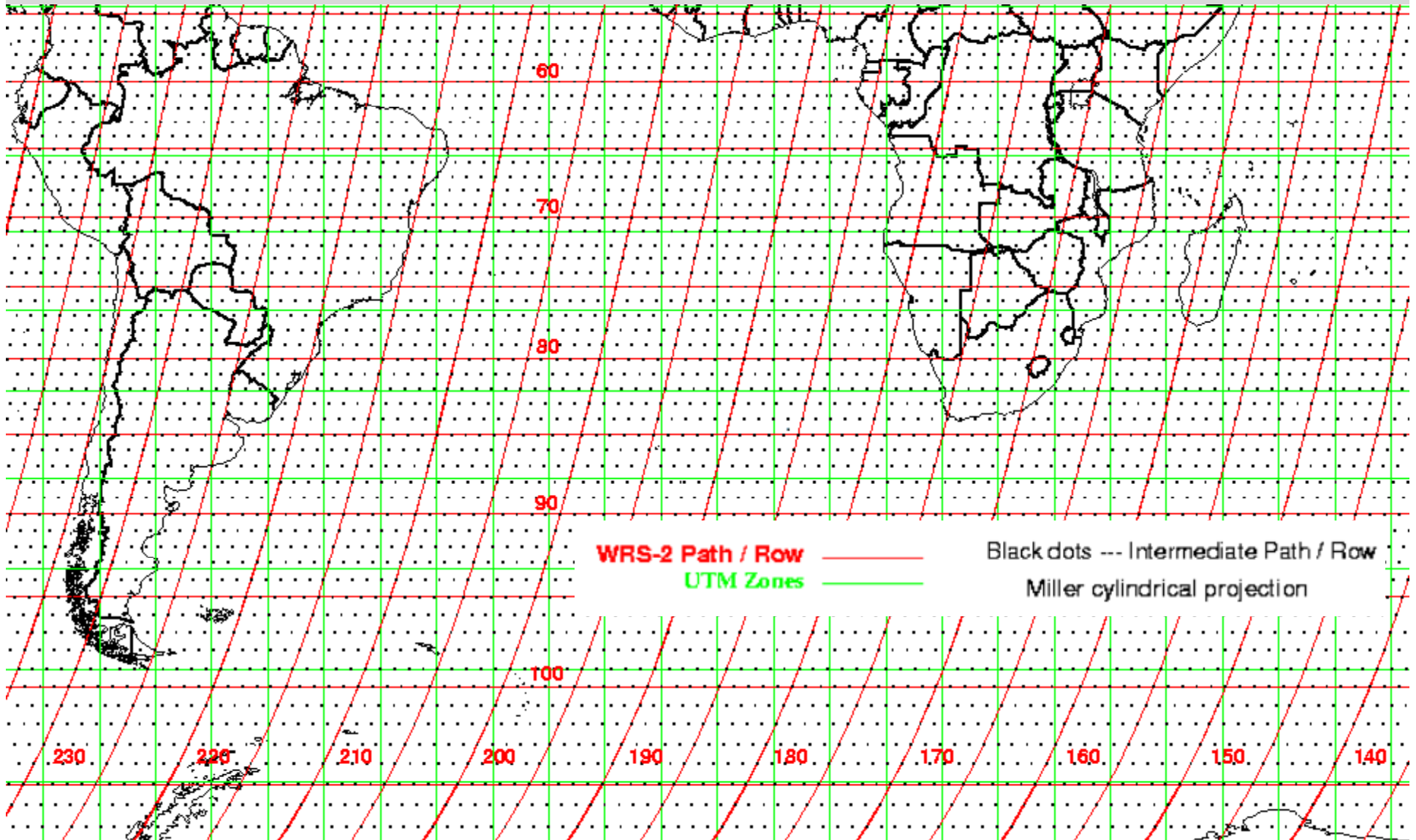
**Coordinates** | Predefined Area | Shapefile | KML

Degree/Minute/Second | Decimal

No coordinates selected



# Seen Selection – Path and Row





# Seen Selection

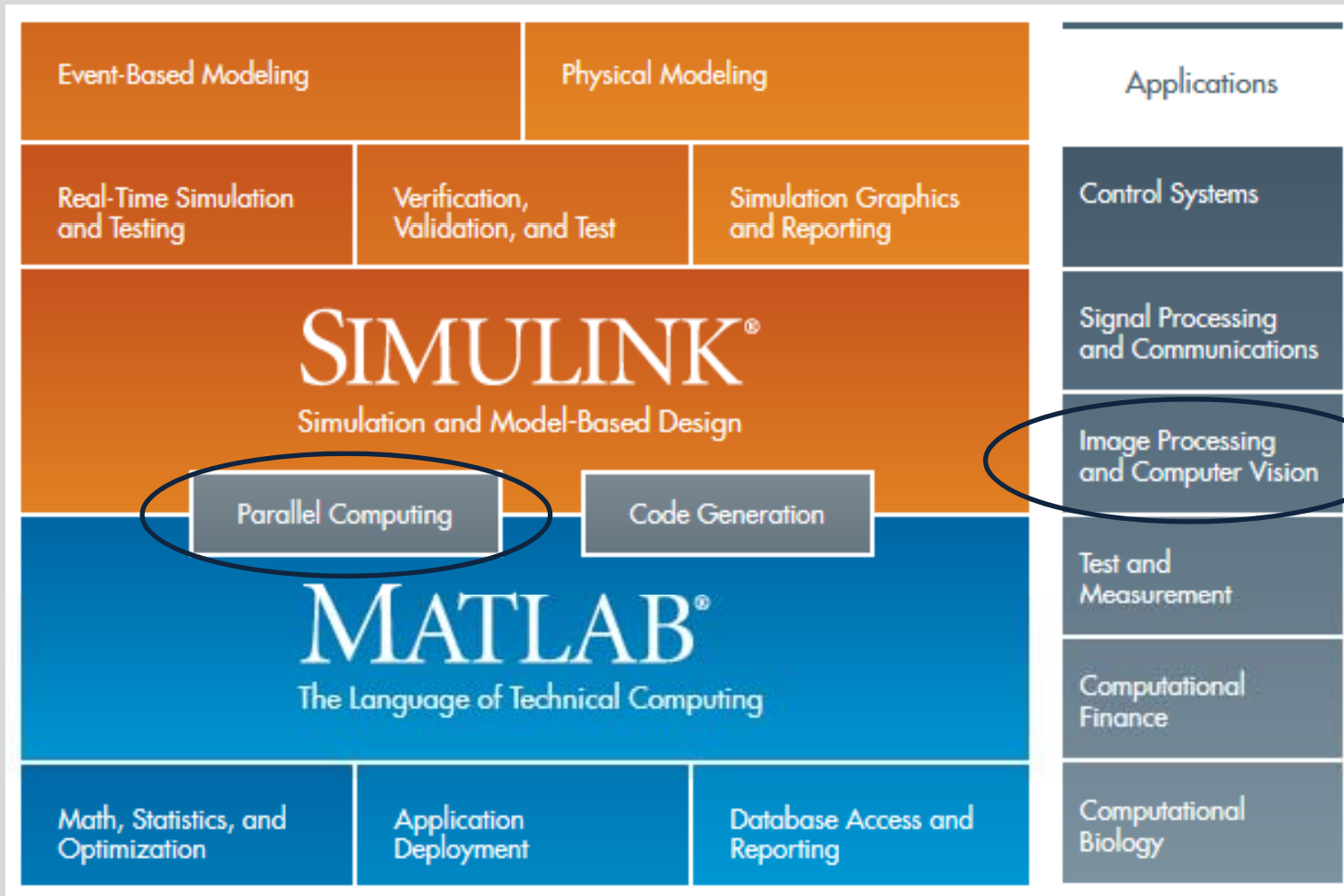
The image shows a screenshot of the Google Earth web interface. The main view is a 3D globe with a white grid overlay. The globe is centered on the Indian subcontinent. The interface includes a search bar at the top left with the text "ex: Pizza near Clayville, NY" and a "Search" button. Below the search bar is a "Places" panel with a tree view showing "My Places" and "Temporary Places". The "My Places" section includes "Sightseeing Tour" (with a sub-note "Make sure 3D Buildings layer is checked"), "Riyadh\_Kindom\_Sq", "testN", and "testS". The "Temporary Places" section includes "WRScornerpoints descendin...".

At the bottom left, there is a "Layers" panel with a tree view showing "Primary Database", "Voyager", "Go on Safari", "Edition 2", "Download", "Borders and Labels", "Places", "Photos", "Roads", "3D Buildings", "Ocean", "Weather", "Gallery", "Global Awareness", and "More".

The bottom right of the globe displays copyright information: "US Dept of State-Geographer #", "© 2015 Google", "© 2009 GeoBasis-DE/BKG", and "Data SIO, NOAA, U.S. Navy, NGA, GEBCO". The "Google Earth" logo is partially visible on the right. At the very bottom right, the coordinates "29°35'22.79" N 143°23'23.58" E" and "eye alt 9808.10 km" are shown.

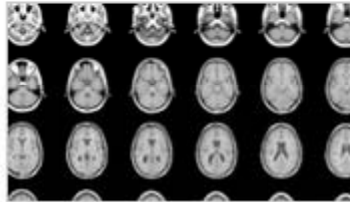
# Matlab

- Components



# Matlab – Image processing toolbox

## Capabilities



### Exploration and Discovery

Use functions and apps to acquire, visualize, analyze, and process images in many data types.

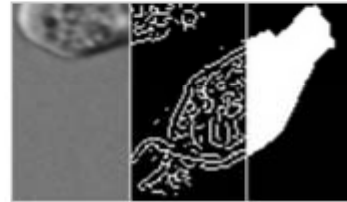
» [Learn more](#)



### Image Enhancement

Increase the signal-to-noise ratio and accentuate image features by modifying the colors or intensities of an image.

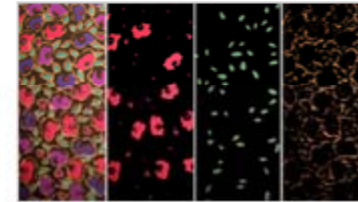
» [Learn more](#)



### Image Analysis

Perform image analysis by extracting meaningful information from images, such as finding shapes, counting objects, identifying colors, or measuring object properties.

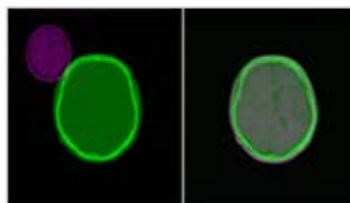
» [Learn more](#)



### Image Segmentation

Explore different approaches to image segmentation, including progressive methods, automatic thresholding, edge-based methods, and morphology-based methods.

» [Learn more](#)  
🕒 [Watch video 5:11](#)



### Image Registration and Geometric Transformations

Perform image registration, which is important in remote sensing, medical imaging, and other applications in which images must be aligned to enable quantitative analysis or qualitative comparison.



### Large Image Processing and Performance Acceleration

Work with large images that are difficult to process and display with standard methods.

» [Learn more](#)



### Target Hardware

Generate C, C++, and HDL code directly from MATLAB by using Image Processing Toolbox with MATLAB Coder, Vision HDL Toolbox, and HDL Coder.

» [Learn more](#)



# Matlab – Mapping toolbox



**Mapping Toolbox**  
Analyze and visualize geographic information

[Overview](#) [Features](#) [Code Examples](#) [Videos](#) [Webinars](#) [Related Products](#) [What's New](#) [Product Tri](#)

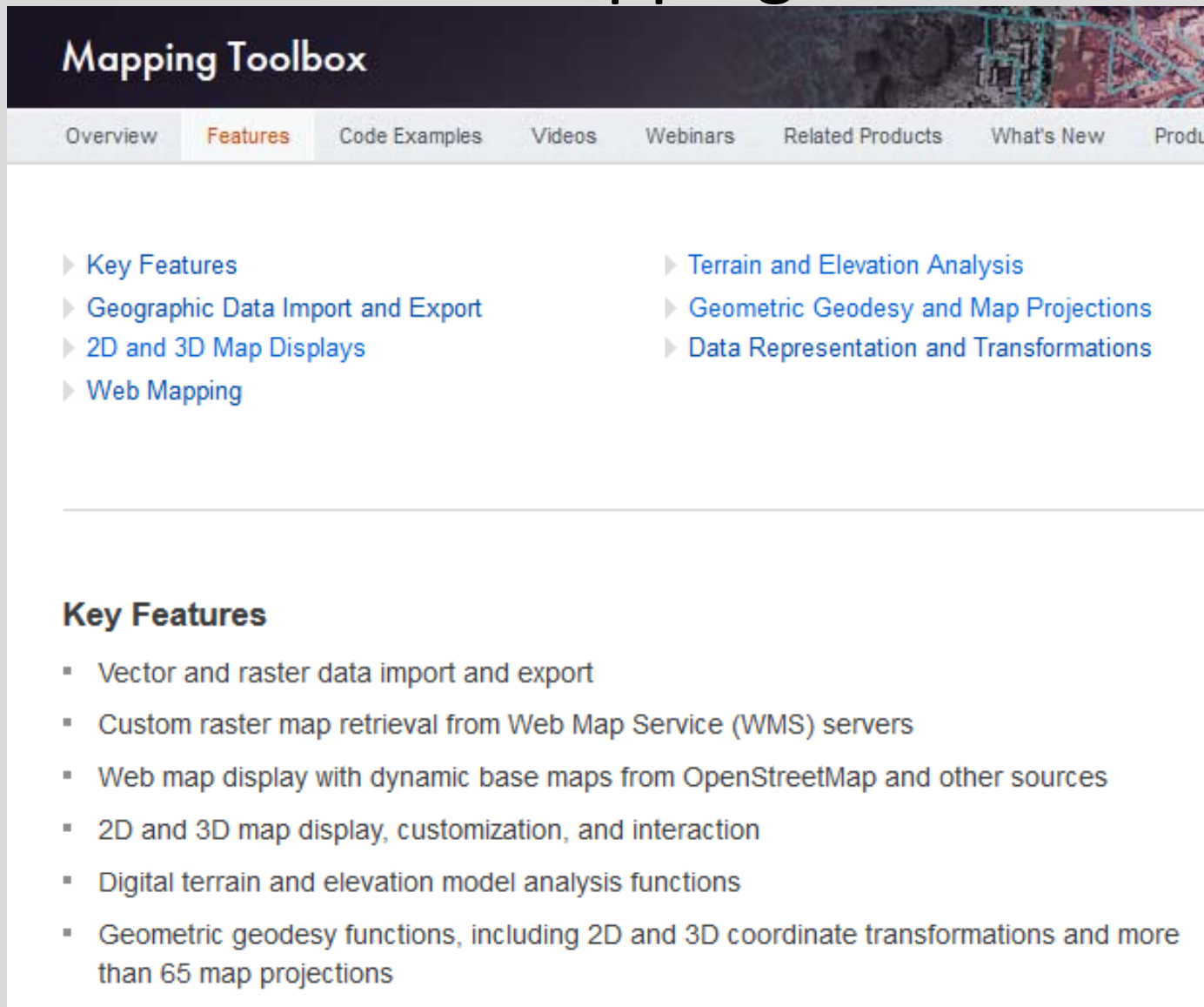
Mapping Toolbox™ provides algorithms, functions, and an app for analyzing geographic data and creating map displays in MATLAB®. You can import vector and raster data from a wide range of file formats and web map servers. The toolbox lets you subset and customize data using trimming, interpolation, resampling, coordinate transformations, and other techniques. Geospatial data can be combined with base map layers from multiple sources in a single map display. You can export data in file formats such as shapefile, GeoTIFF, and KML. By incorporating mapping functions into MATLAB programs, you can automate frequent tasks in your geospatial workflow.

- ▶ Key Features
- ▶ Geographic Data Import and Export
- ▶ 2D and 3D Map Displays
- ▶ Web Mapping
- ▶ Terrain and Elevation Analysis
- ▶ Geometric Geodesy and Map Projections
- ▶ Data Representation and Transformations





# Matlab – Mapping toolbox



**Mapping Toolbox**

Overview **Features** Code Examples Videos Webinars Related Products What's New Products

- ▶ Key Features
- ▶ Geographic Data Import and Export
- ▶ 2D and 3D Map Displays
- ▶ Web Mapping
- ▶ Terrain and Elevation Analysis
- ▶ Geometric Geodesy and Map Projections
- ▶ Data Representation and Transformations

---

## Key Features

- Vector and raster data import and export
- Custom raster map retrieval from Web Map Service (WMS) servers
- Web map display with dynamic base maps from OpenStreetMap and other sources
- 2D and 3D map display, customization, and interaction
- Digital terrain and elevation model analysis functions
- Geometric geodesy functions, including 2D and 3D coordinate transformations and more than 65 map projections

# File formats and data products supported by Mapping Toolbox

- Raster file formats, such as GeoTIFF, USGS DEM, DEM, DTED, Arc ASCII Grid, GTOPO30, ETOPO, and worldfile
- Vector file formats, such as ESRI® shapefiles, KML, GPX, VMAP0, and GSHHS
- Selected data products, such as AVHRR and EGM96

Relevant data formats supported by MATLAB include:

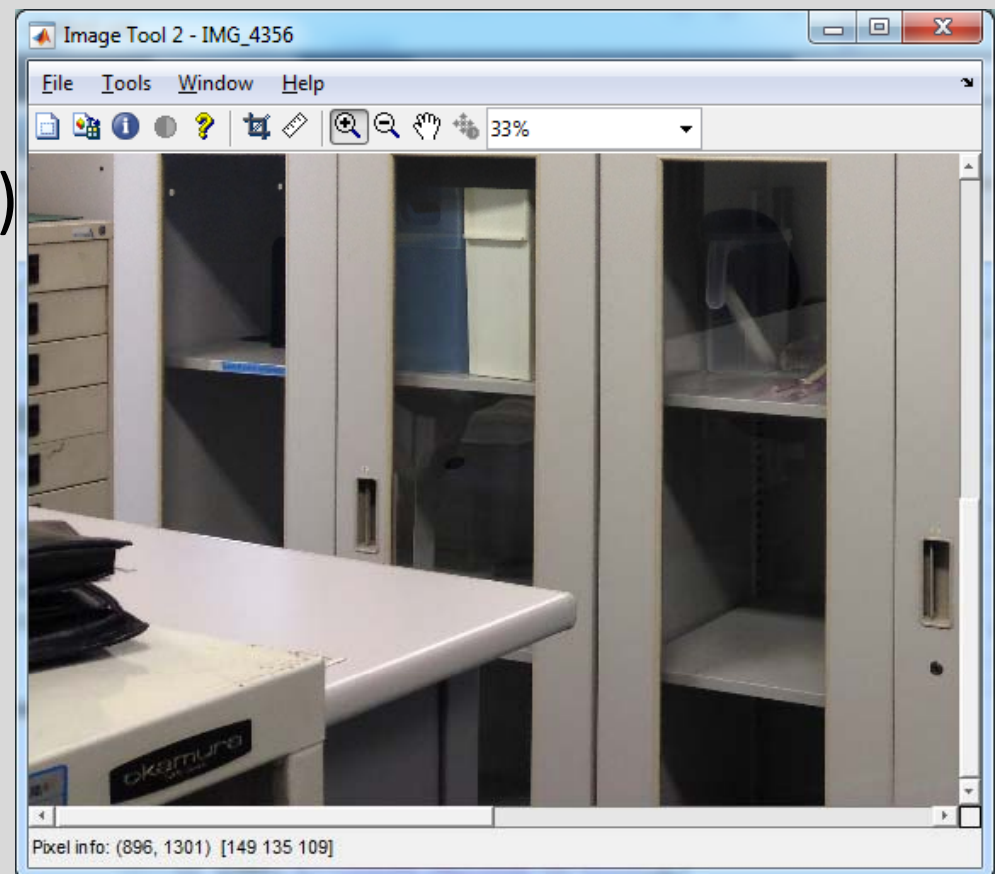
- Image file formats, such as TIFF, JPEG, PNG, and JPEG2000
- Scientific data formats, such as NetCDF, HDF5, HDF4, HDF-EOS, and multiband files (BIP, BIL, BSQ)
- Network data access through OPeNDAP URL address

Relevant image file formats supported by Image Processing Toolbox include:

- NITF and HDR

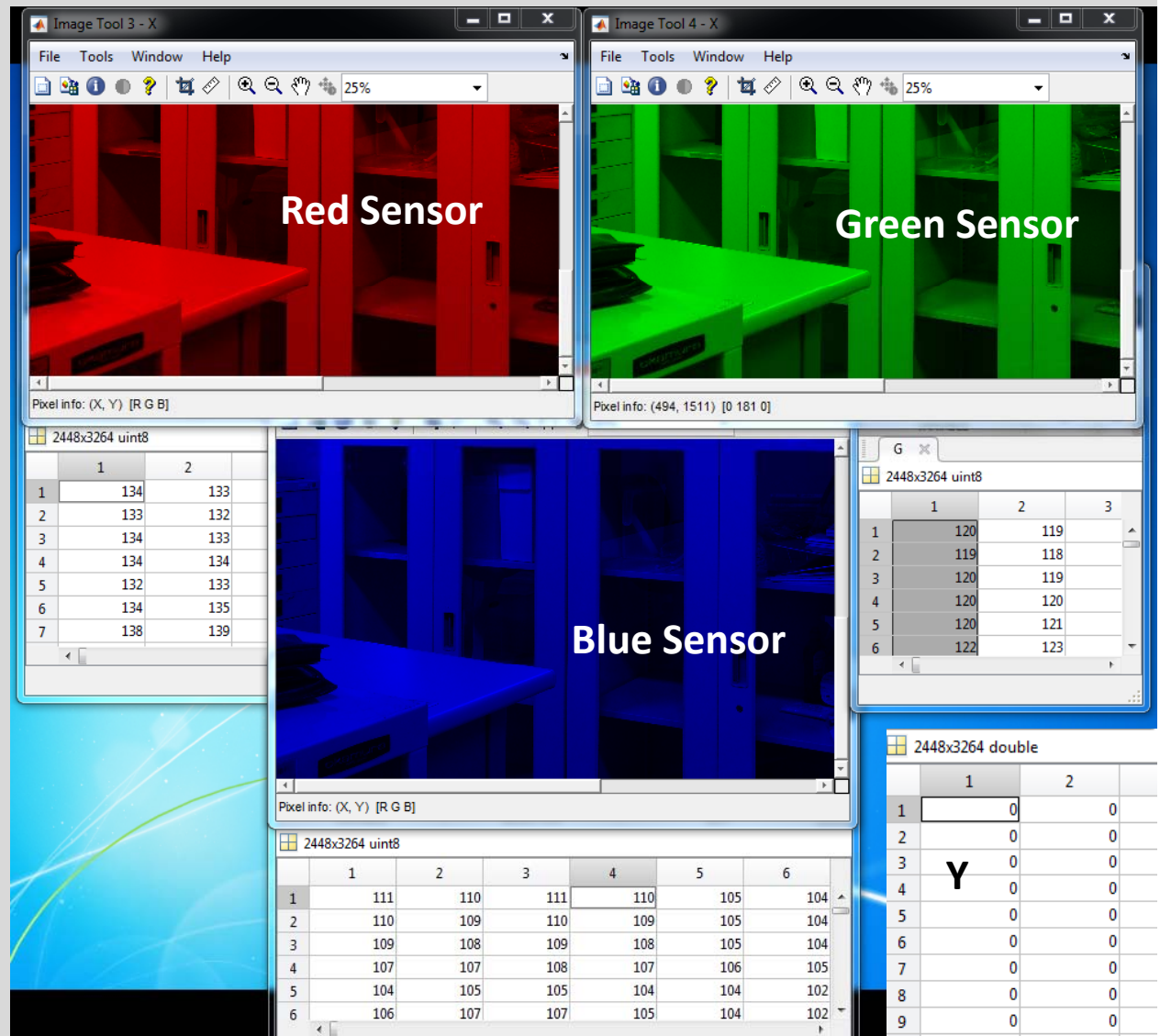
# Read an image with Matlab

- Bands ?
- R, G, B?
- Let's read an image  
`A = imread('img_name')`



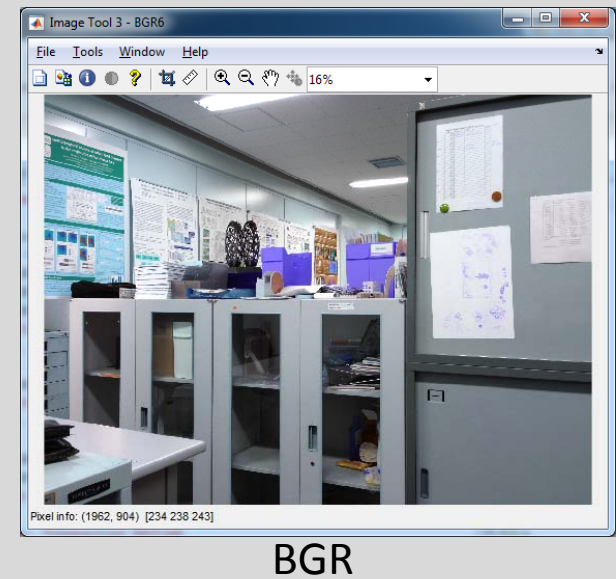
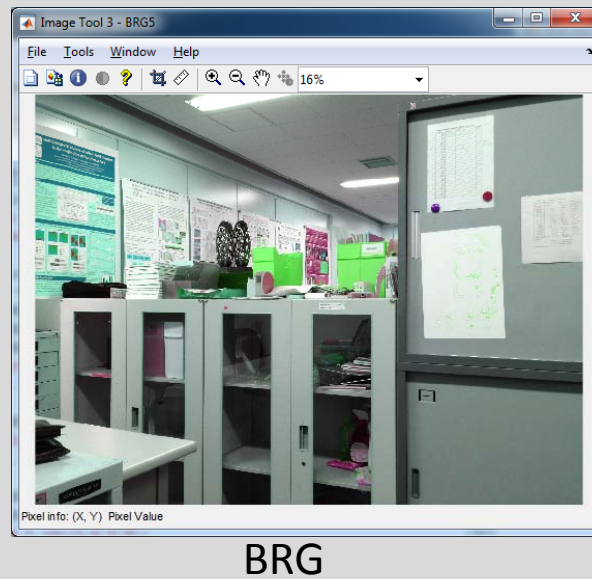
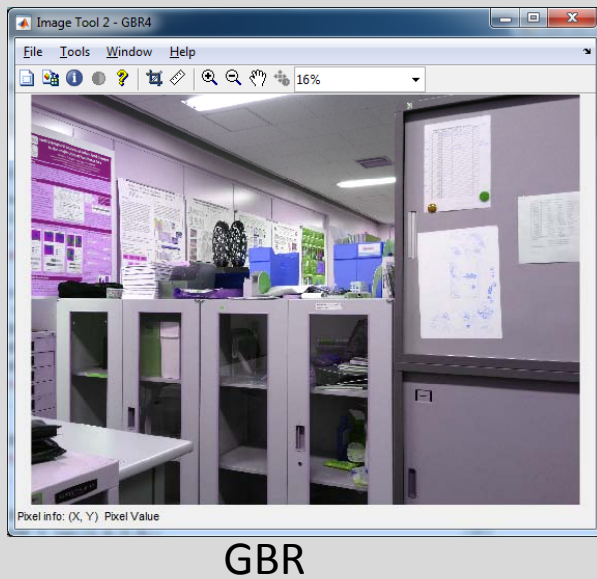
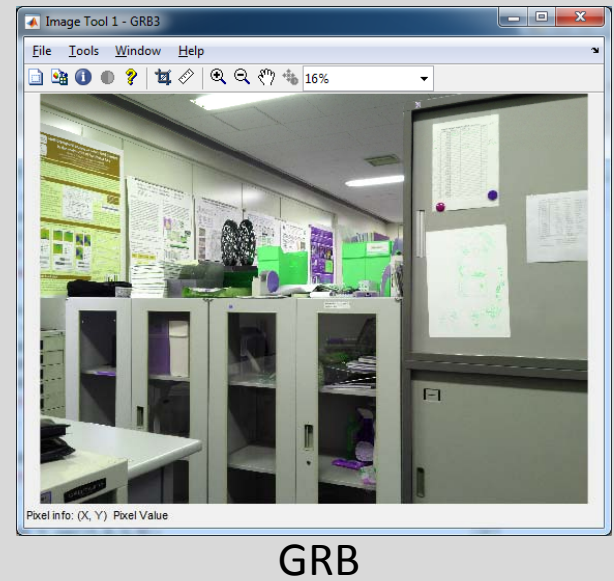
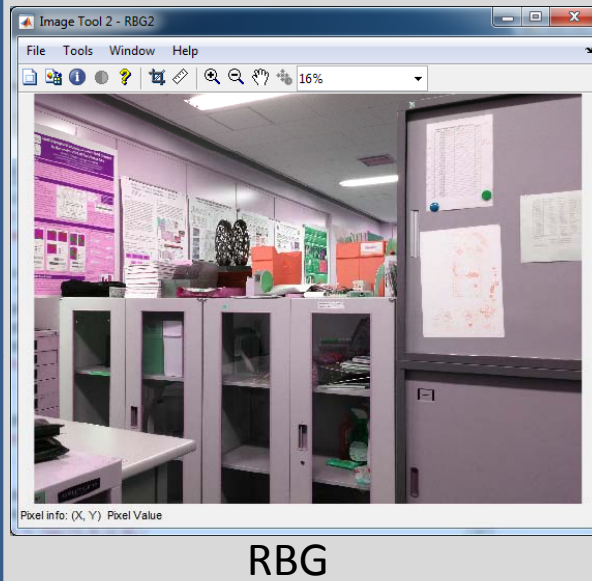
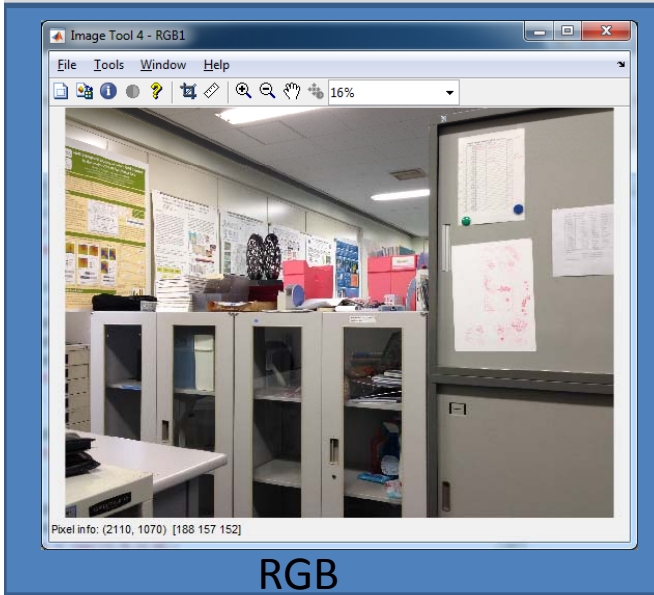
# True color image and band separation

- `imread('img_name')`
- `R = A(:,:,1);`
- `G = A(:,:,2);`
- `B = A(:,:,3);`
- `Y = zeros(size(R));`
- `X = cat(3,R,G,B);`
- `imshow(X)`
  
- `Rsum = sum(sum(R));`
  - 1.1534760400000000e+09
- `Gsum = sum(sum(G));`
  - 1.0933113380000000e+09
- `Bsum = sum(sum(B));`
  - 1.0376060820000000e+09



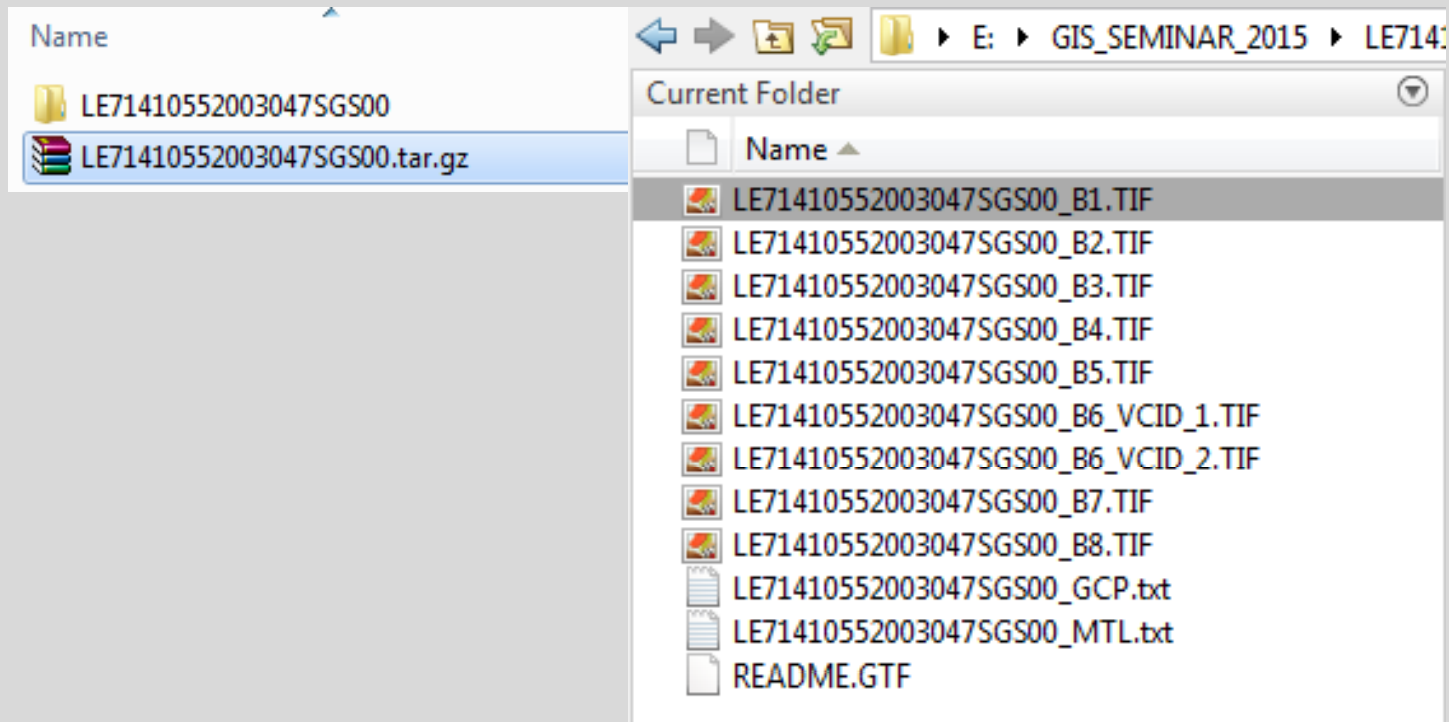


# False Color images



# Read Landsat Images

- `L1 = imread('LXX***###****###***##');`
- `L1= imread('LE71410552003047SGS00_B1.TIF')`



# Read Landsat images

The screenshot shows the MATLAB R2015a environment. The Command Window contains the following code:

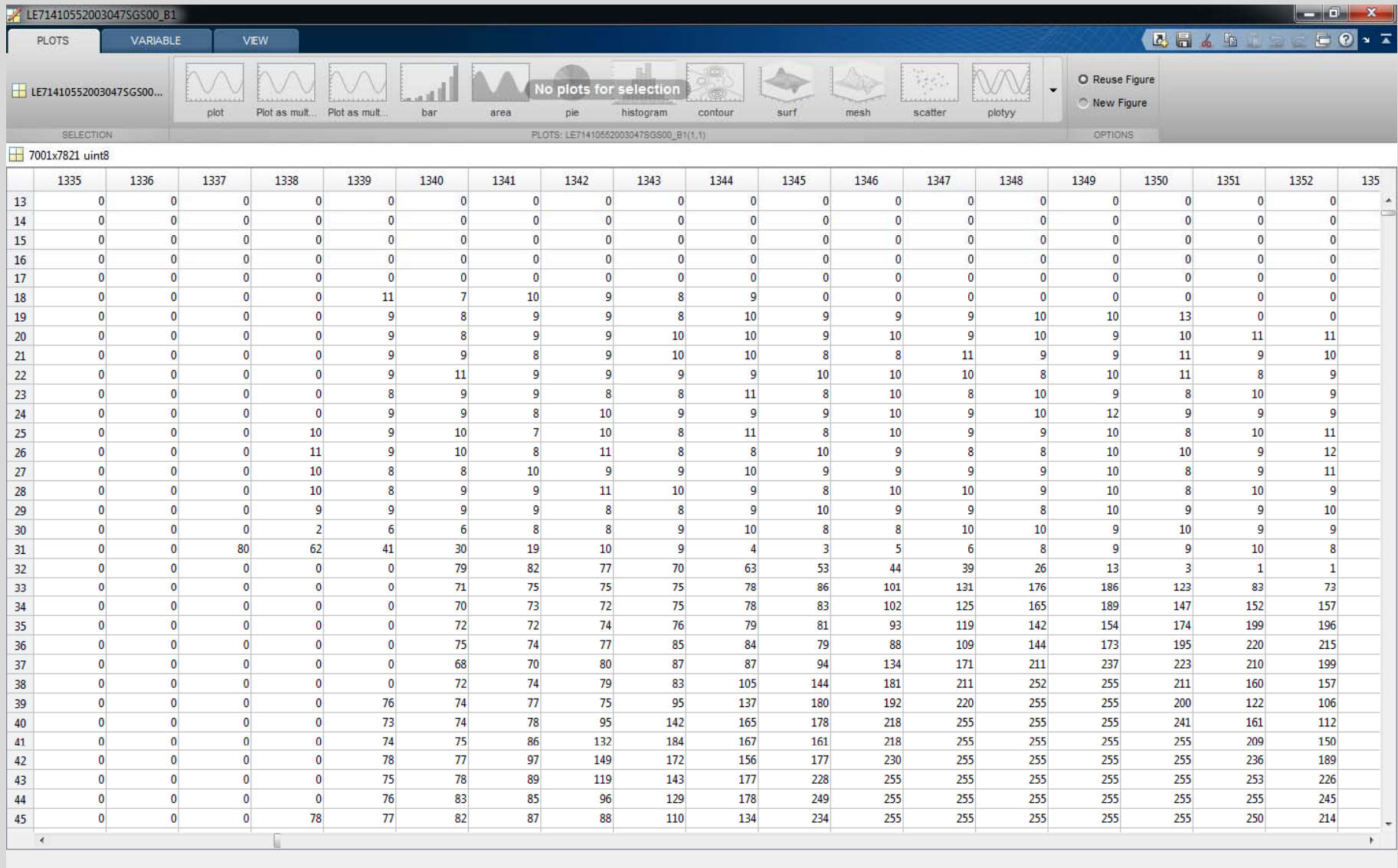
```
>> uiopen('E:\GIS_SEMINAR_2015\LE71410552003047SGS00')
>> imread('E:\GIS_SEMINAR_2015\LE71410552003047SGS00')
>> L1 = imread('E:\GIS_SEMINAR_2015\LE71410552003047SGS00')
>> uiopen('E:\GIS_SEMINAR_2015\LE71410552003047SGS00')
>> uiopen('E:\GIS_SEMINAR_2015\LE71410552003047SGS00')
```

The Workspace window shows the following variables:

Name	Value
L1	7001x7821
LE71410552003047SGS00_B1	7001x7821

The Image Tool window displays the image at 2.83% zoom. The status bar shows "Pixel info: (X, Y) Intensity" and "Display range: [0 255]".

# What's inside?





# More about Landsat files

1x1 struct with 40 fields

Field	Value
Filename	'E:\GIS_SEMINAR_2015\LE714105
FileModDate	'19-Aug-2015 18:41:29'
FileSize	54811178
Format	'tif'
FormatVersion	[]
Width	7821
Height	7001
BitDepth	8
ColorType	'grayscale'
FormatSignature	[73,73,42,0]
ByteOrder	'little-endian'
NewSubFileType	0
BitsPerSample	8
Compression	'Uncompressed'
PhotometricInterpretation	'BlackIsZero'
StripOffsets	1x7001 double
SamplesPerPixel	1
RowsPerStrip	1
StripByteCounts	1x7001 double
XResolution	[]
YResolution	[]
ResolutionUnit	'Inch'
Colormap	[]

Imfinfo('landsat\_img')

PlanarConfiguration	'Chunky'
TileWidth	[]
TileLength	[]
TileOffsets	[]
TileByteCounts	[]
Orientation	1
FillOrder	1
GrayResponseUnit	0.0100
MaxSampleValue	255
MinSampleValue	0
Thresholding	1
Offset	54754830
SampleFormat	'Unsigned integer'
ModelPixelScaleTag	[30,30,0]
ModelTiepointTag	[0,0,0,341100,906000,0]
GeoKeyDirectoryTag	1x28 double
GeoAsciiParamsTag	'UTM Zone 44 N with WGS84'

# More about Landsat files

- MTL\_parser.m
  - Written in Nov 2012 by Evan Miles, Scott Polar Research Institute, University of Cambridge
- Generate metadata in a structure
- MTL file - LE71410552003047SGS00\_MTL.txt
- Usege: MTL\_parser('Landsat\_img');

# More about Landsat files

The image displays three screenshots of Landsat metadata files, each showing a different set of fields and values. The first screenshot shows a file with 10 fields, the second with 7 fields, and the third with 45 fields. The fourth screenshot shows a file with 9 fields, and the fifth shows a list of quantization parameters for 7 bands.

Field	Value
METADATA_FILE_INFO	1x1 struct
PRODUCT_METADATA	1x1 struct
IMAGE_ATTRIBUTES	1x1 struct
MIN_MAX_RADIANCE	1x1 struct
MIN_MAX_REFLECTANCE	1x1 struct
MIN_MAX_PIXEL_VALUE	1x1 struct
PRODUCT_PARAMETERS	1x1 struct
RADIOMETRIC_RESCALING	1x1 struct
THERMAL_CONSTANTS	1x1 struct
PROJECTION_PARAMETERS	1x1 struct

Field	Value
ORIGIN	'Image courtesy of the U.S. Geological.
REQUEST_ID	'0101508186856_00001'
LANDSAT_SCENE_ID	'LE71410552003047SGS00'
FILE_DATE	'2015-08-19T18:41:24Z'
STATION_ID	'SGS'
PROCESSING_SOFTWARE_VERSION	'LPGS_12.6.1'
DATA_CATEGORY	'NOMINAL'

Field	Value
DATA_TYPE	
ELEVATION_SOURCE	
OUTPUT_FORMAT	
EPHEMERIS_TYPE	
SPACECRAFT_ID	
SENSOR_ID	
SENSOR_MODE	
WRS_PATH	
WRS_ROW	
DATE_ACQUIRED	
SCENE_CENTER_TIME	
CORNER_UL_LAT_PRODUCT	
CORNER_UL_LON_PRODUCT	
CORNER_UR_LAT_PRODUCT	
CORNER_UR_LON_PRODUCT	
CORNER_LL_LAT_PRODUCT	
CORNER_LL_LON_PRODUCT	
CORNER_LR_LAT_PRODUCT	
CORNER_LR_LON_PRODUCT	
CORNER_UL_PROJECTION_X_PRODU	
CORNER_UL_PROJECTION_Y_PRODU	
CORNER_UR_PROJECTION_X_PRODU	
CORNER_UR_PROJECTION_Y_PRODU	
CORNER_LL_PROJECTION_X_PRODU	
CORNER_LL_PROJECTION_Y_PRODU	
CORNER_LR_PROJECTION_X_PRODU	
CORNER_LR_PROJECTION_Y_PRODU	
PANCHROMATIC_LINES	
PANCHROMATIC_SAMPLES	

Field	Value
MAP_PROJECTION	'UTM'
DATUM	'WGS84'
ELLIPSOID	'WGS84'
UTM_ZONE	44
GRID_CELL_SIZE_PANCHROMATIC	15
GRID_CELL_SIZE_REFLECTIVE	30
GRID_CELL_SIZE_THERMAL	30
ORIENTATION	'NORTH_UP'
RESAMPLING_OPTION	'CUBIC_CONVOLUTION'

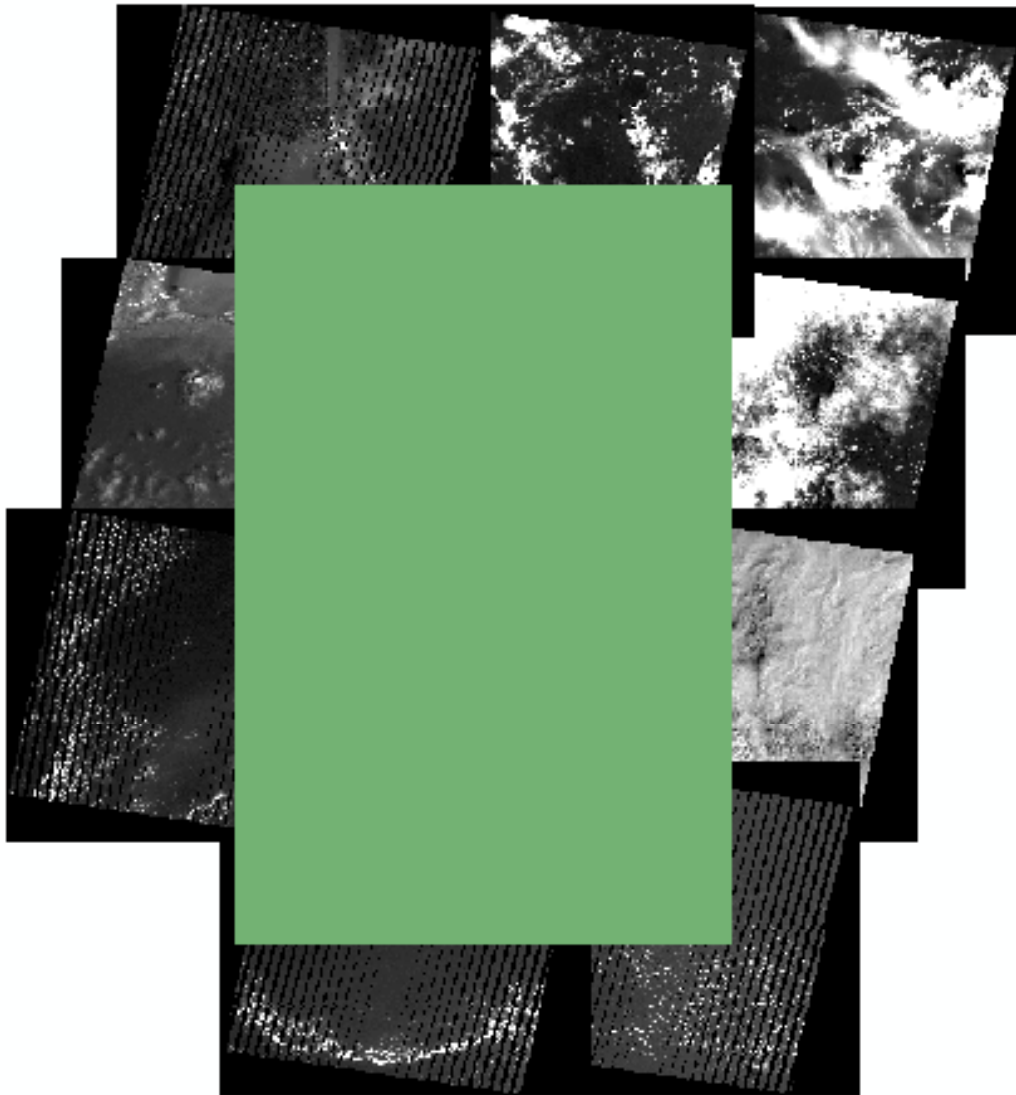
QUANTIZE_CAL_MAX_BAND_1	255
QUANTIZE_CAL_MIN_BAND_1	1
QUANTIZE_CAL_MAX_BAND_2	255
QUANTIZE_CAL_MIN_BAND_2	1
QUANTIZE_CAL_MAX_BAND_3	255
QUANTIZE_CAL_MIN_BAND_3	1
QUANTIZE_CAL_MAX_BAND_4	255
QUANTIZE_CAL_MIN_BAND_4	1
QUANTIZE_CAL_MAX_BAND_5	255
QUANTIZE_CAL_MIN_BAND_5	1
QUANTIZE_CAL_MAX_BAND_6_V...	255
QUANTIZE_CAL_MIN_BAND_6_VC...	1
QUANTIZE_CAL_MAX_BAND_6_V...	255
QUANTIZE_CAL_MIN_BAND_6_VC...	1
QUANTIZE_CAL_MAX_BAND_7	255
QUANTIZE_CAL_MIN_BAND_7	1

# Intersect extractor

- Clip Landsat images (each band) according to a given project size.
- Input - folder name - containing all valid Landsat images within the project region
- Global project parameters need to be set
- Total number of Landsat scenes used – 12684



# Project\_Landsat Intersect extractor

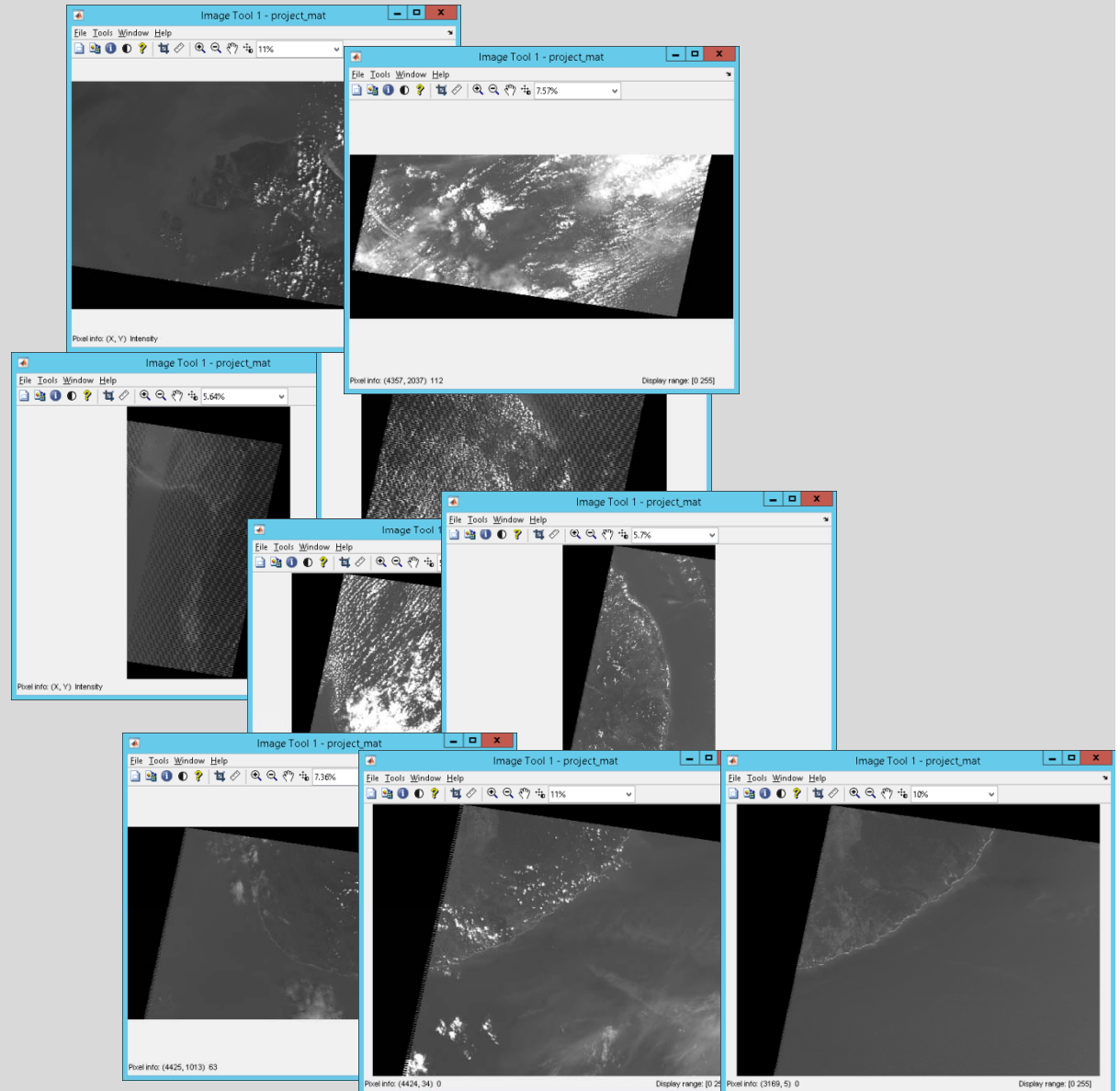
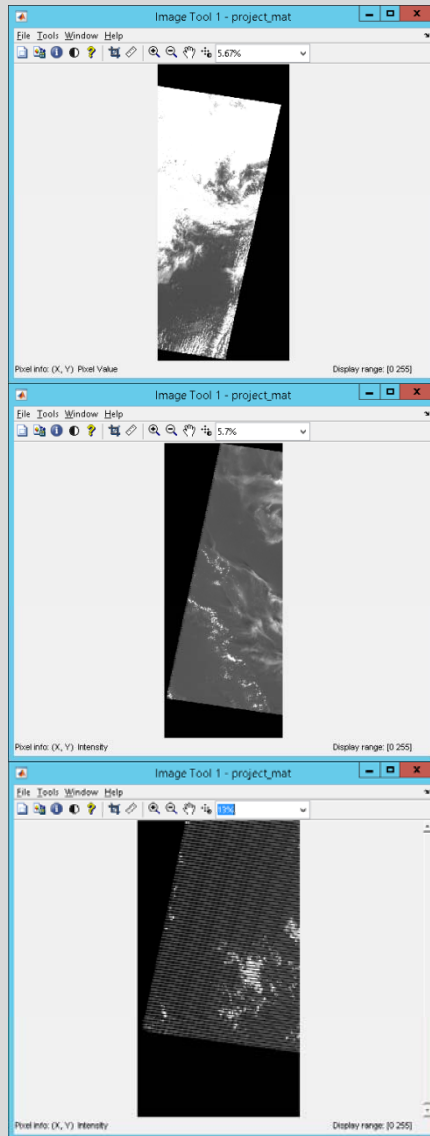


Path	Raw
140	053
140	054
140	055
140	056
141	053
141	054
141	055
141	056
142	053
142	054
142	055

# Project\_Landsat Intersect extractor

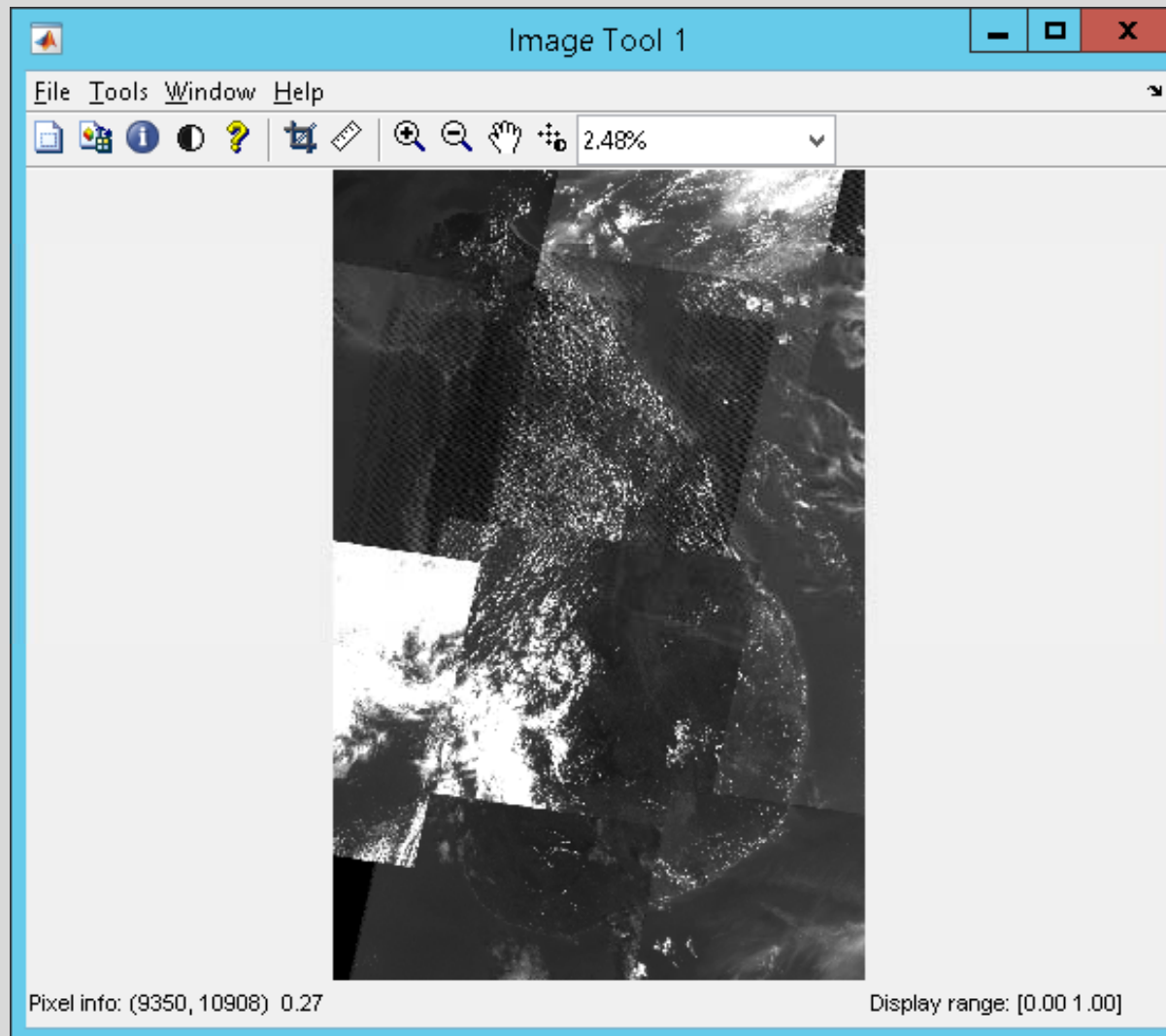
- On each clipped image other algorithms can be preformed
  - Cloud removal (Fmask), TOA correction etc.
- Reproduce Tiff metadata and export.

# Mosaic Creator

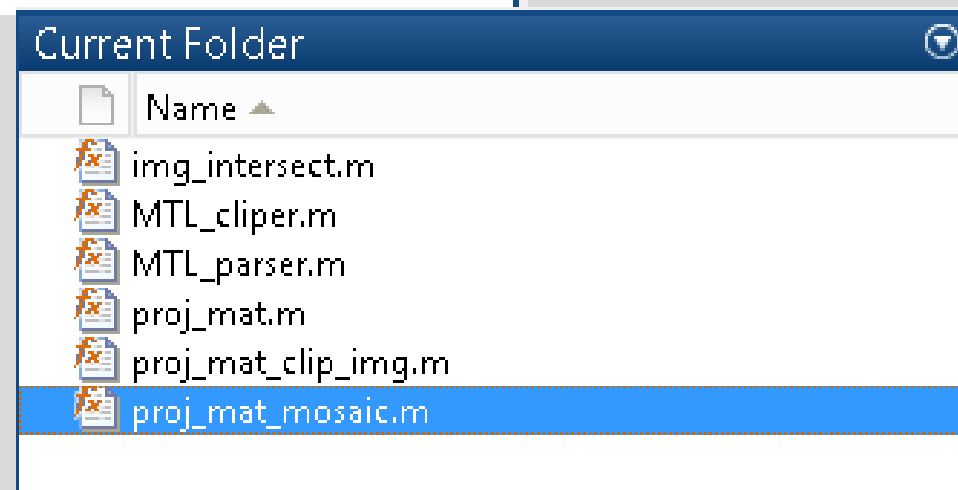
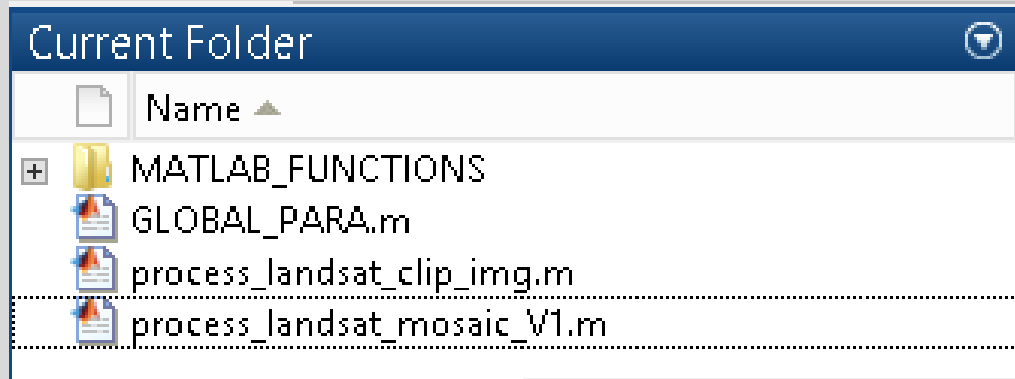


# Mosaic Creator

- All clipped images are added together



# Scripts so far ...





# Scripts so far ...

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%% Function - create proj_mat
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  % Create project mat for each landsat image
5
6  function [] = proj_mat(infile, outfile)
7  %%% INCLUDES start %%%
8  GLOBAL PARA
9  %%% INCLUDES end %%%
10
11  img_land = imread(infile);
12  img_info = imfinfo(infile);
13
14  img_X_bl = img_info.ModelTiepointTag(4); % x remains same
15  img_Y_bl = img_info.ModelTiepointTag(5) - img_info.Height*30; % y calc
16
17  proj_pv = [PROJ_TL(1), PROJ_BR(2), 10500*30, 16000*30];
18  img_pv = [img_X_bl, img_Y_bl, img_info.Width*30, img_info.Height*30];
19
20  intersect_area = rectint(proj_pv, img_pv);
21
22  if (intersect_area > 0)
23
24      valid_img = true;
25
26  end
27
28  if (valid_img)
29
30      fprintf('valid file available, processing data .....\\n');
31      % find intersect position vector
32      intersect_pv = img_intersect(proj_pv, img_pv); % xx_pv - intersect
```

Thank you!!!