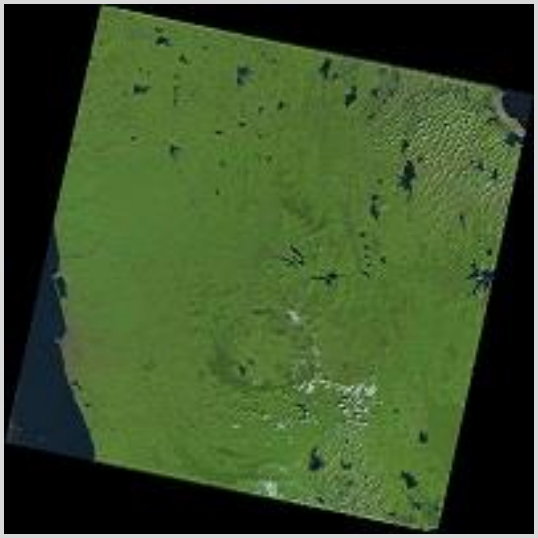
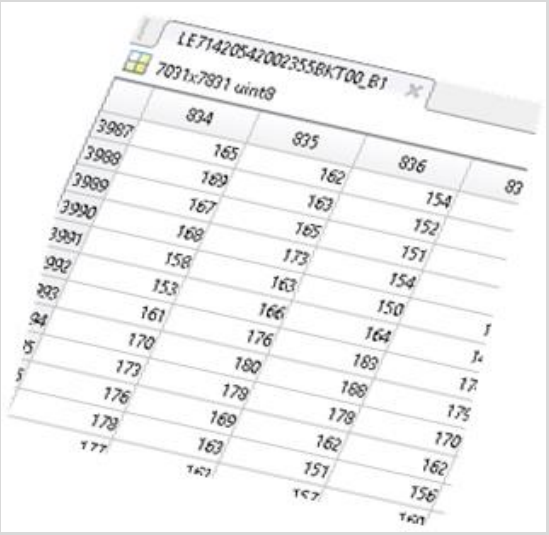
132 -GIS Seminar – 2015.10.22

Division of Spatial Information Science

Handling Landsat Images with Matlab





Malinda Siriwardana, Prof. Yuji Murayama

University of Tsukuba
Graduate School of Life and Environmental Science

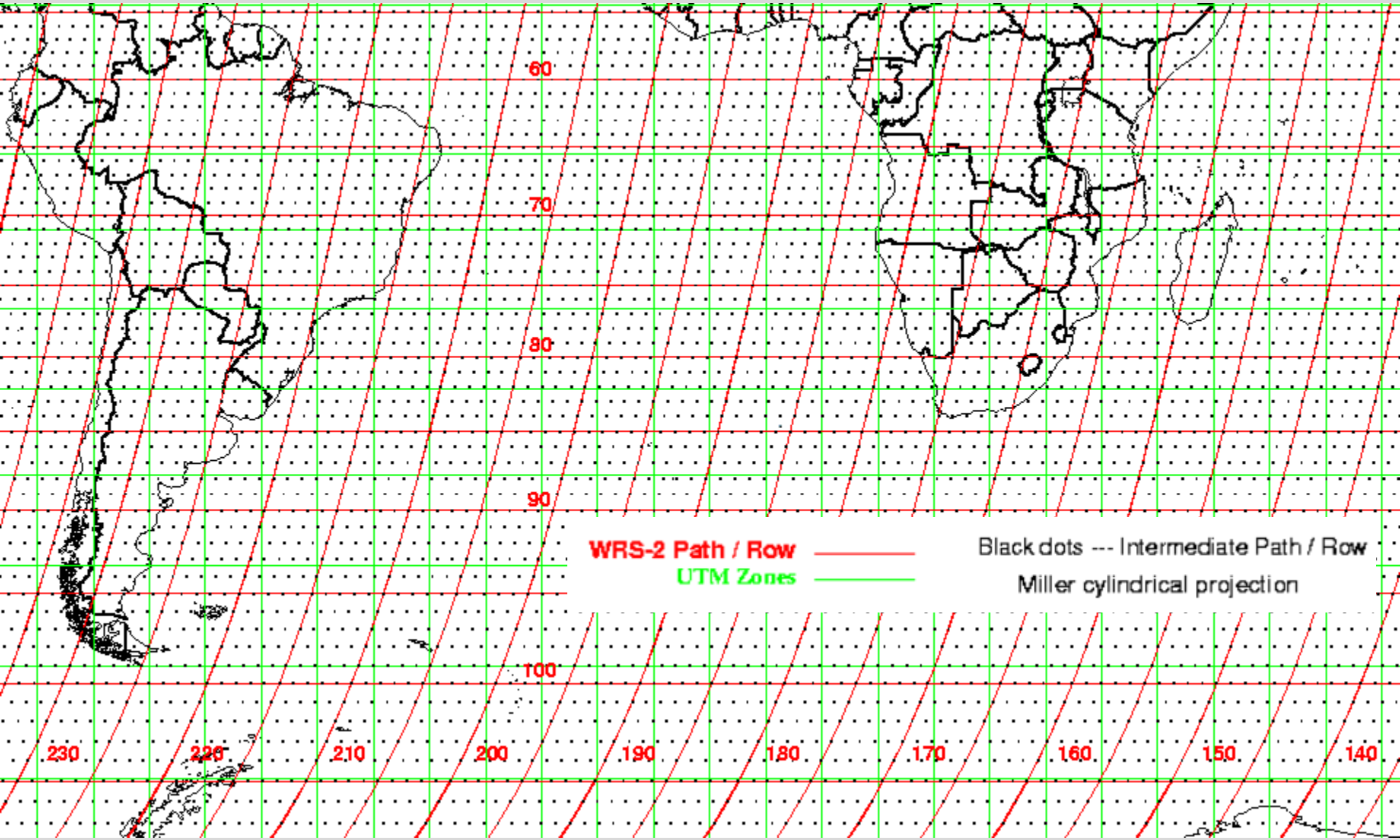
Introduction

- What is Landsat ?

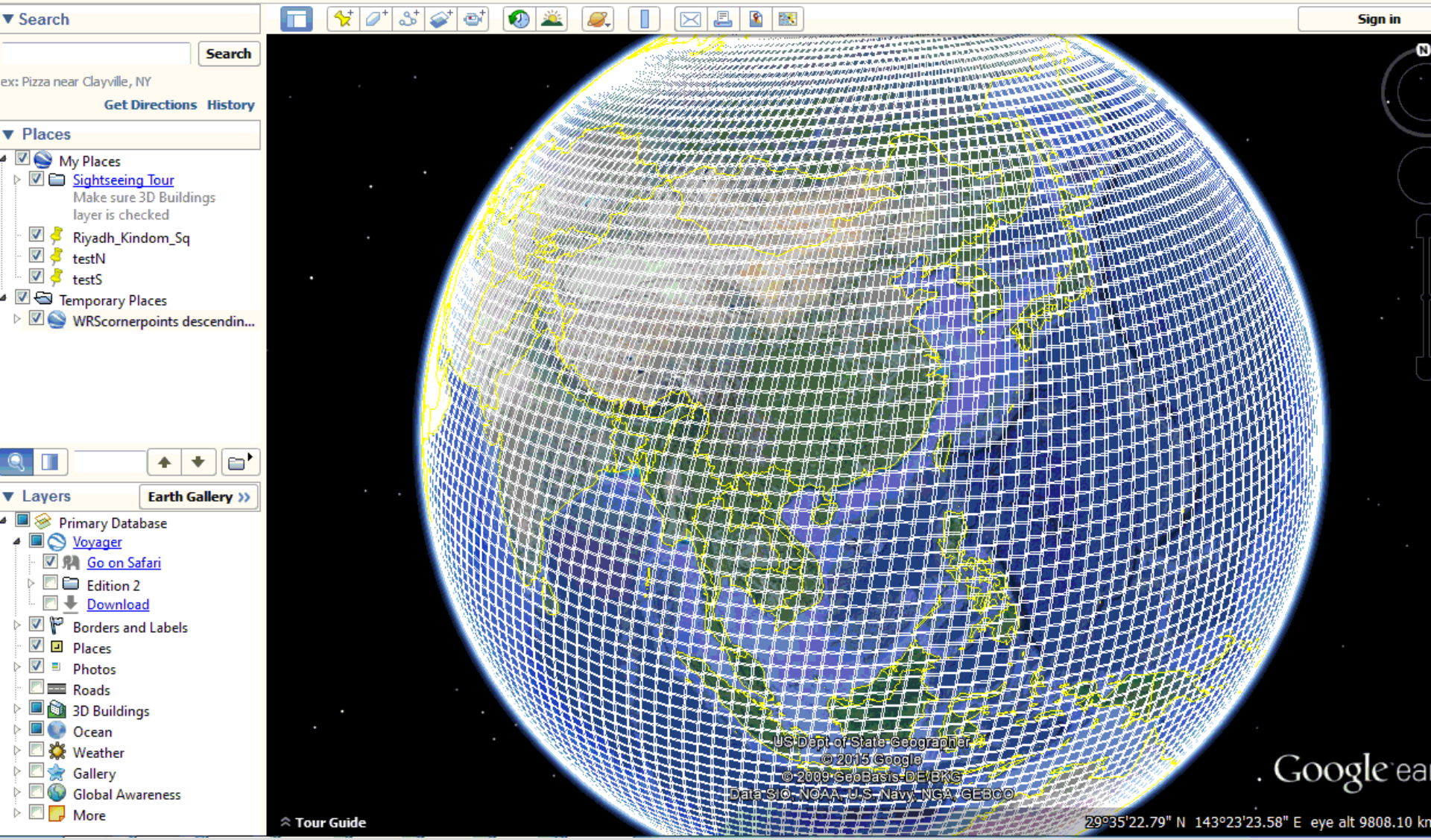
Introduction

- Where can I download?

Seen Selection – Path and Row

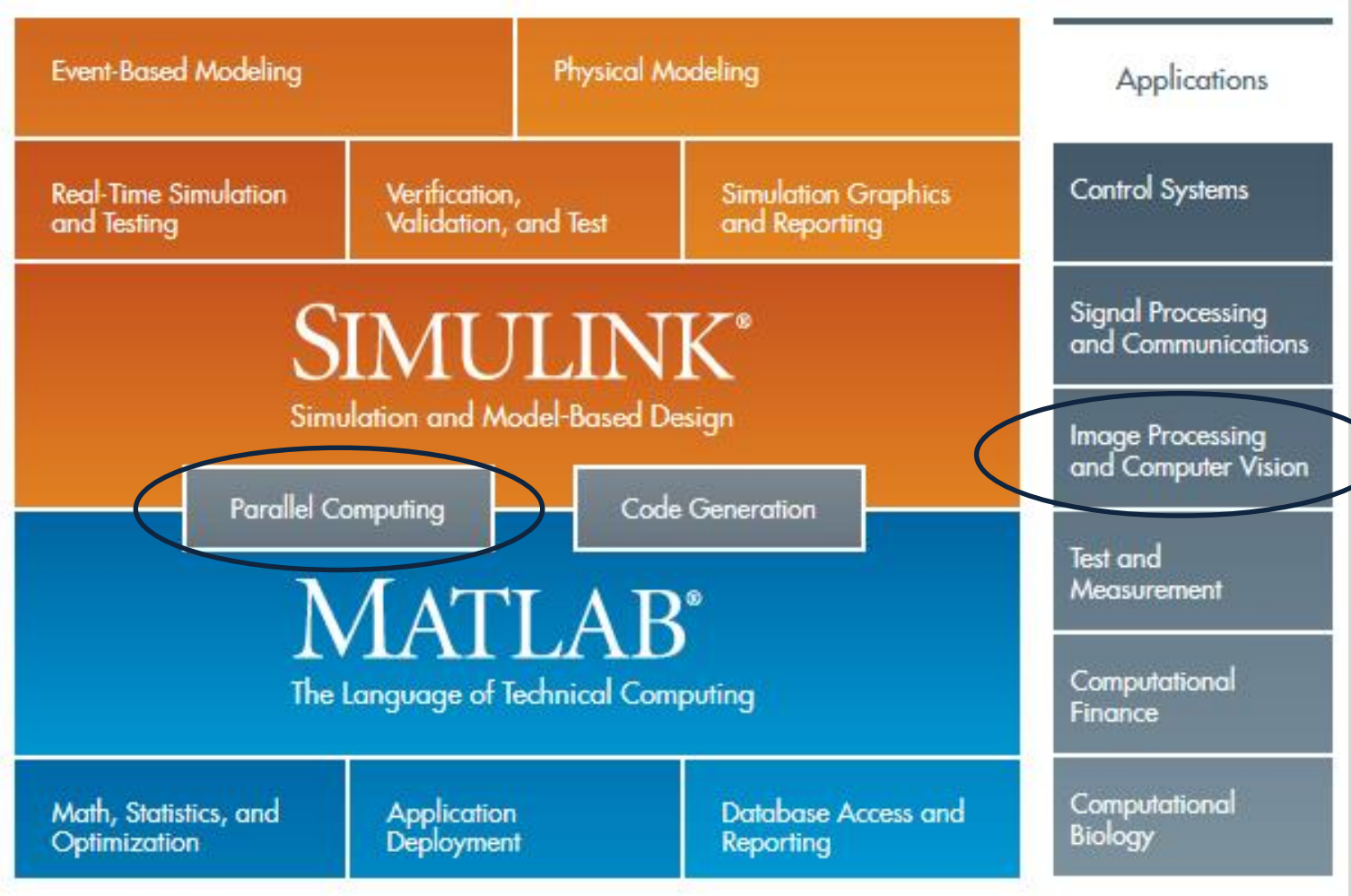


Seen Selection



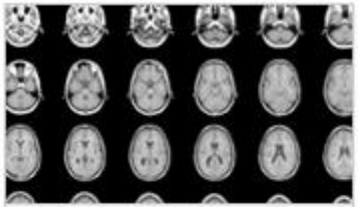
Matlab

- 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

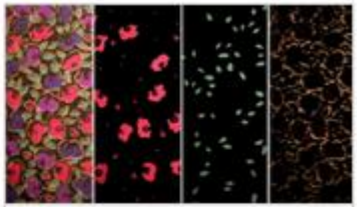


Image Segmentation

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

» Learn more

Watch video 5:11

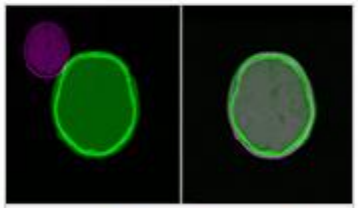


Image Registration and Geometric Transformations

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

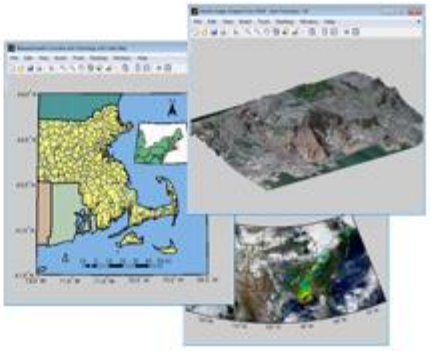
Mapping Toolbox

Analyze and visualize geographic information

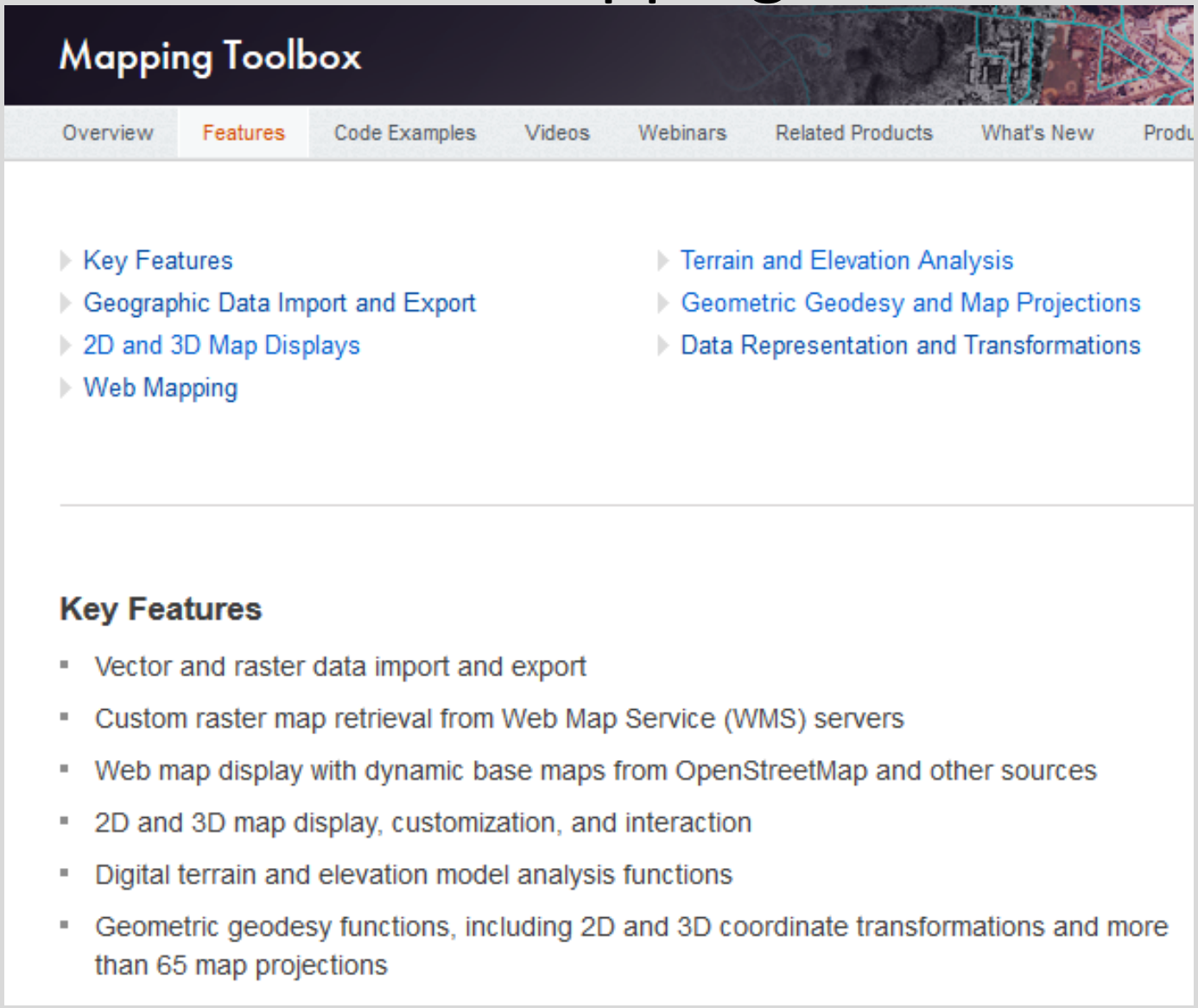
Overview | Features | Code Examples | Videos | Webinars | Related Products | What's New | Product Tri

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



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

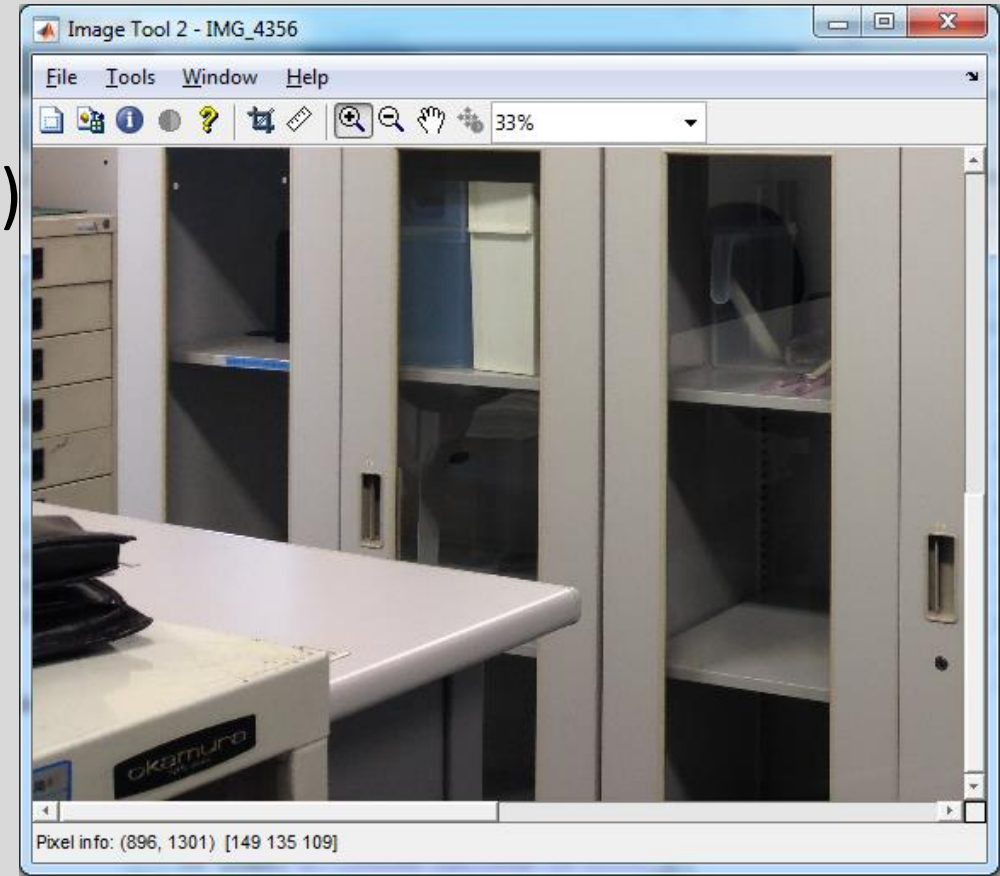
Relevant image file formats supported by Image Processing Toolbox include:

- NITF and HDR

Read an image with Matlab

- Bands ?
- R, G, B?
- Let’s read an image

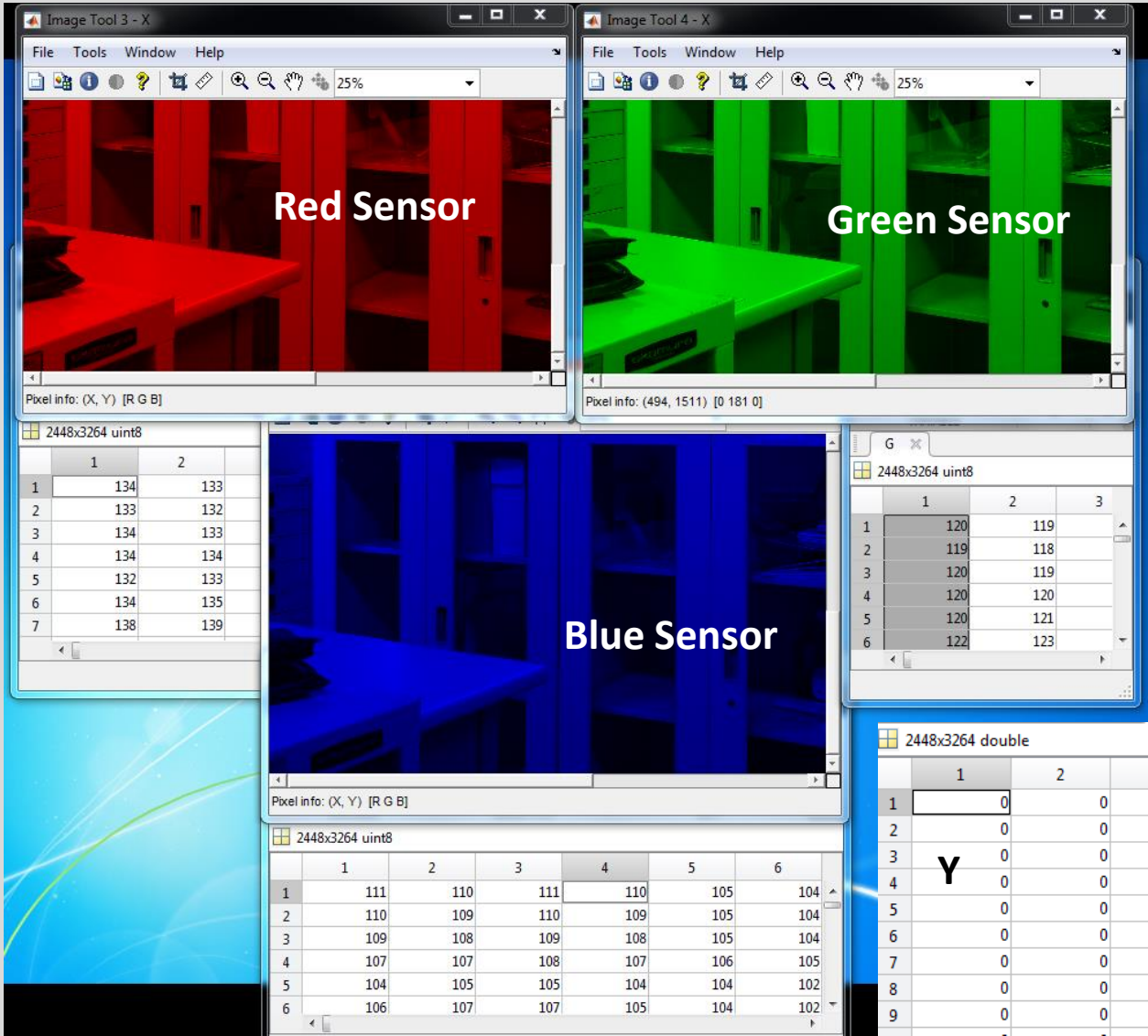
  A = imread(‘img_name’)



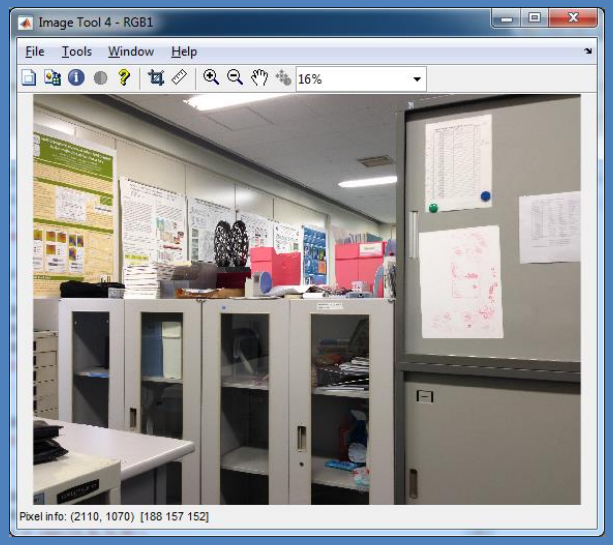
True color image and band separation

- imread('img_name')
- R = A(:,:,1);
- G = A(:,:,2);
- B = A(:,:,3);
- Y = zeros(size(R));
- X = cat(3,R,G,B);
- imshow(X)

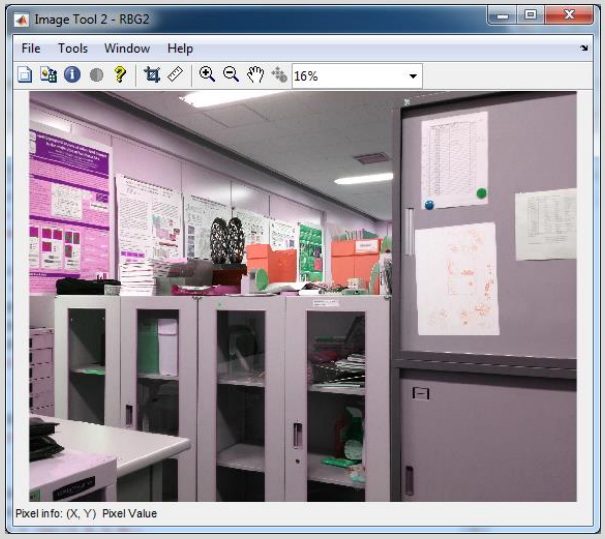
- Rsum = sum(sum(R));
  - 1.153476040000000e+09
- Gsum = sum(sum(G));
  - 1.093311338000000e+09
- Bsum = sum(sum(B));
  - 1.037606082000000e+09



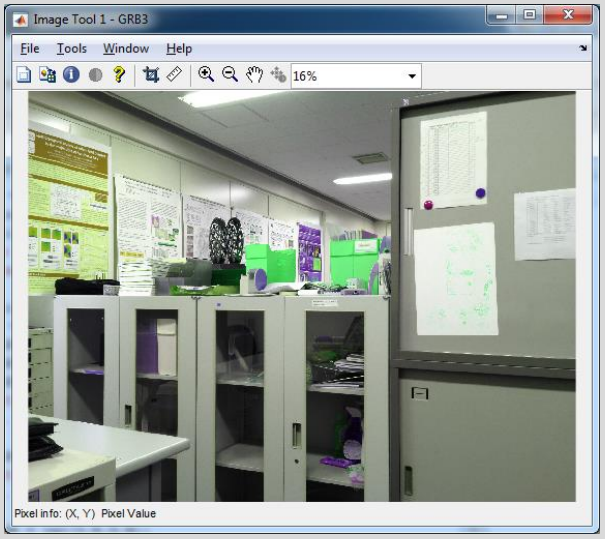
False Color images



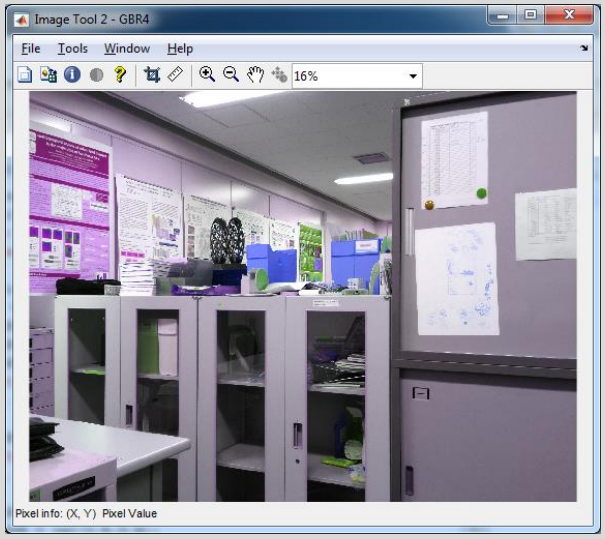
RGB



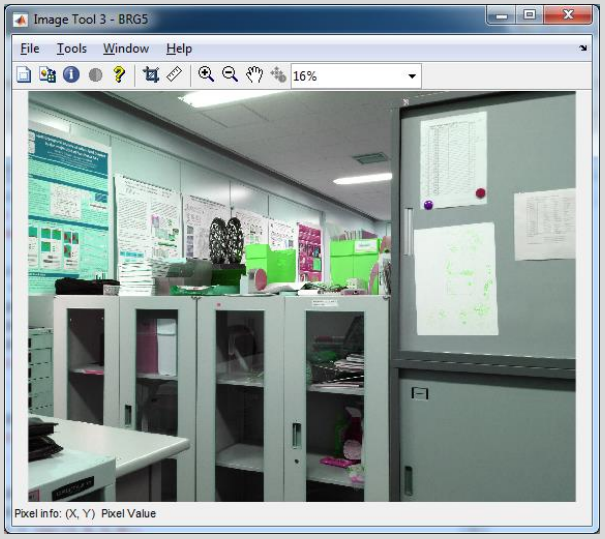
RBG



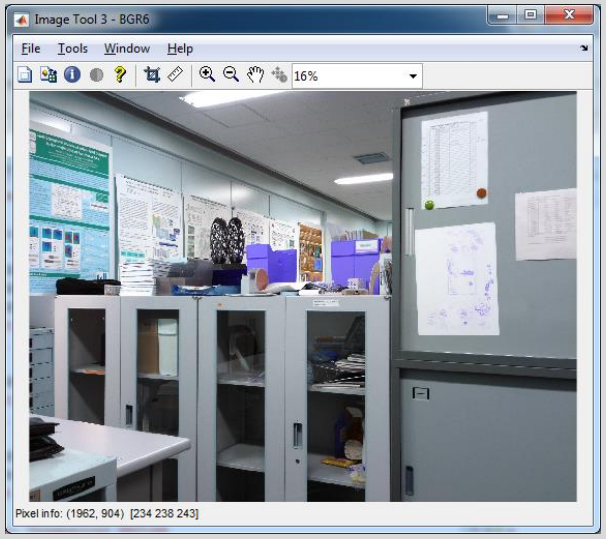
GRB



GBR



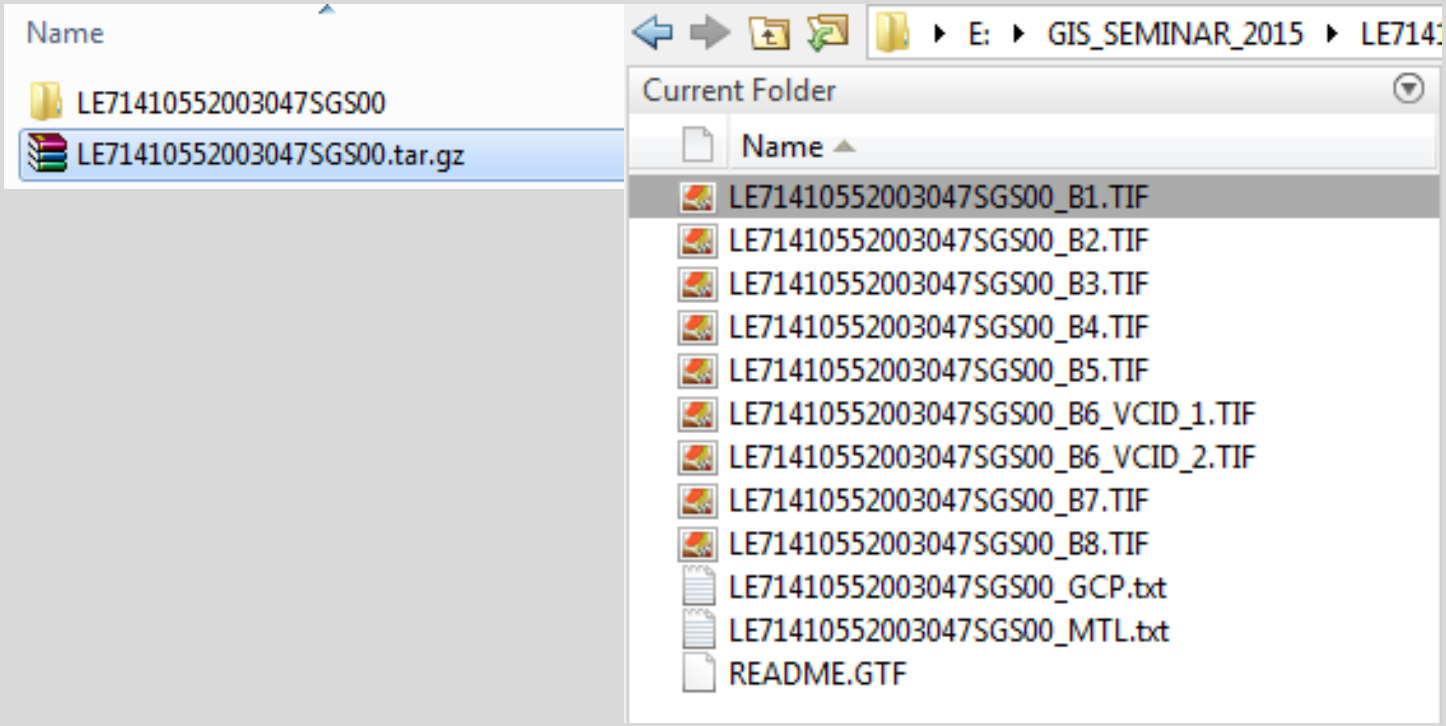
BRG



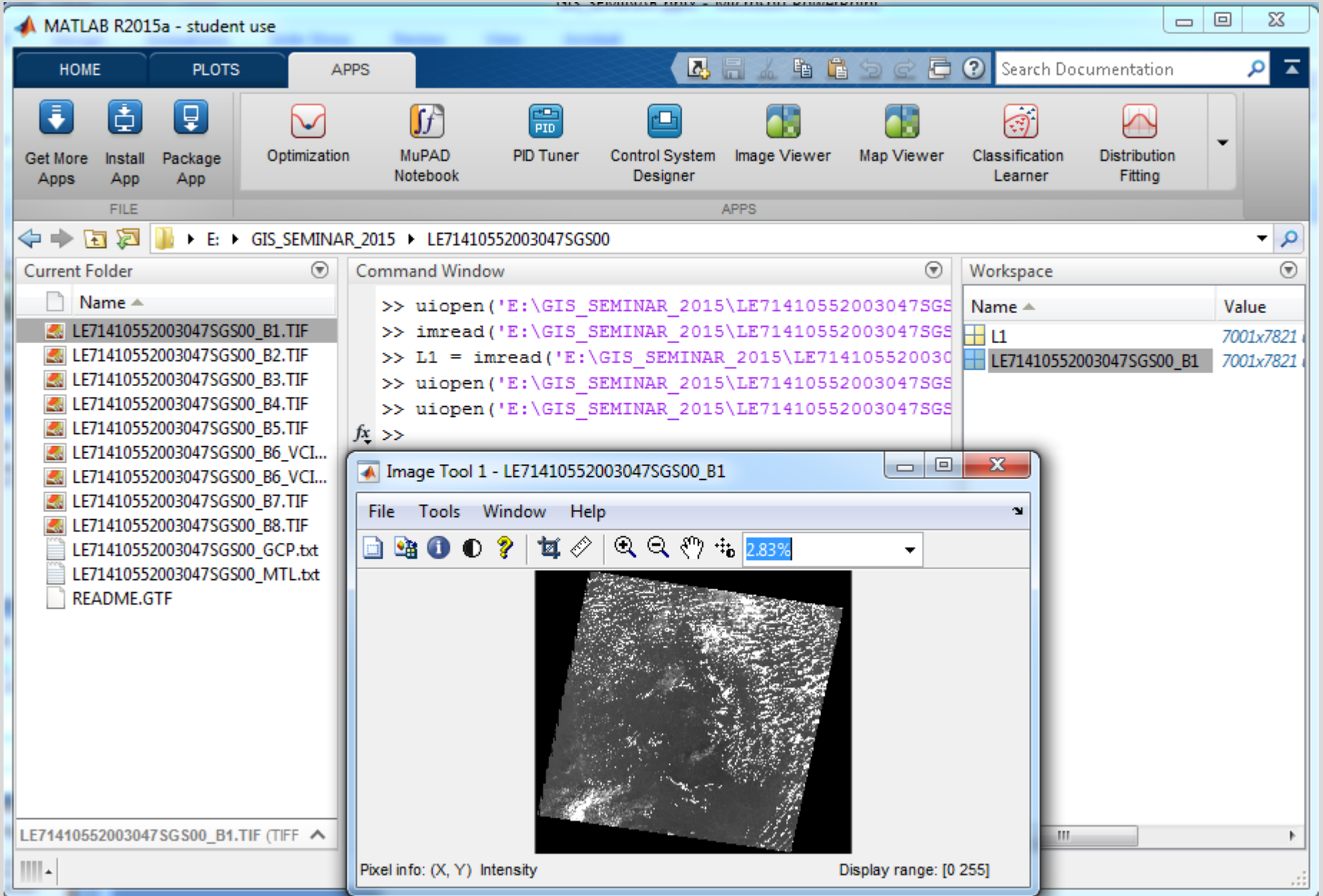
BGR

Read Landsat Images

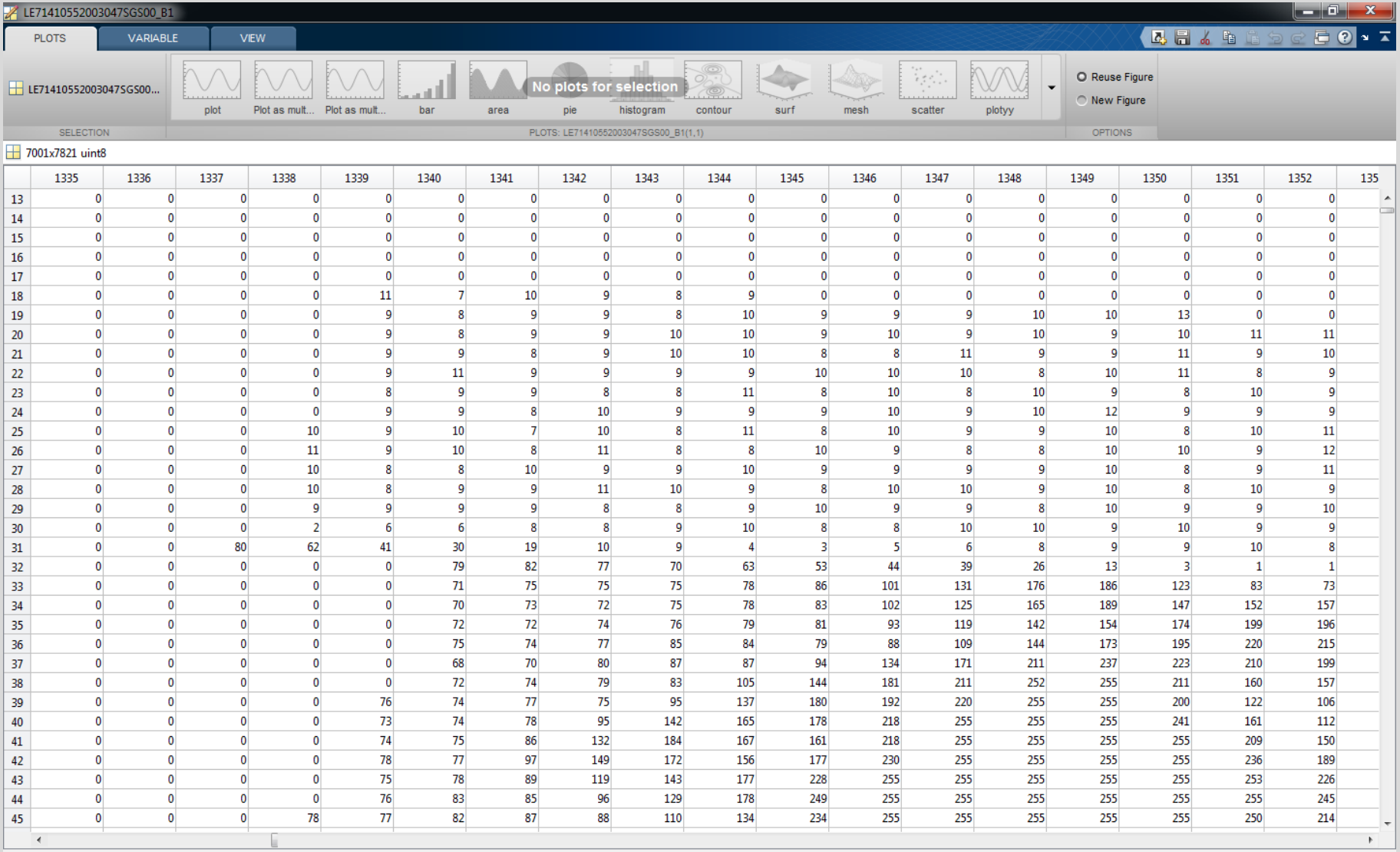
- L1 = imread('LXX***###*****###****##');
- L1= imread('LE71410552003047SGS00_B1.TIF')



Read Landsat images



What's inside?



	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	135
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18	0	0	0	0	11	7	10	9	8	9	0	0	0	0	0	0	0	0	
19	0	0	0	0	9	8	9	9	8	10	9	9	9	10	10	13	0	0	
20	0	0	0	0	9	8	9	9	10	10	9	10	9	10	9	10	11	11	
21	0	0	0	0	9	9	8	9	10	10	8	8	11	9	9	11	9	10	
22	0	0	0	0	9	11	9	9	9	9	10	10	10	8	10	11	8	9	
23	0	0	0	0	8	9	9	8	8	11	8	10	8	10	9	8	10	9	
24	0	0	0	0	9	9	8	10	9	9	9	10	9	10	12	9	9	9	
25	0	0	0	10	9	10	7	10	8	11	8	10	9	9	10	8	10	11	
26	0	0	0	11	9	10	8	11	8	8	10	9	8	8	10	10	9	12	
27	0	0	0	10	8	8	10	9	9	10	9	9	9	9	10	8	9	11	
28	0	0	0	10	8	9	9	11	10	9	8	10	10	9	10	8	10	9	
29	0	0	0	9	9	9	9	8	8	9	10	9	9	8	10	9	9	10	
30	0	0	0	2	6	6	8	8	9	10	8	8	10	10	9	10	9	9	
31	0	0	80	62	41	30	19	10	9	4	3	5	6	8	9	9	10	8	
32	0	0	0	0	0	79	82	77	70	63	53	44	39	26	13	3	1	1	
33	0	0	0	0	0	71	75	75	75	78	86	101	131	176	186	123	83	73	
34	0	0	0	0	0	70	73	72	75	78	83	102	125	165	189	147	152	157	
35	0	0	0	0	0	72	72	74	76	79	81	93	119	142	154	174	199	196	
36	0	0	0	0	0	75	74	77	85	84	79	88	109	144	173	195	220	215	
37	0	0	0	0	0	68	70	80	87	87	94	134	171	211	237	223	210	199	
38	0	0	0	0	0	72	74	79	83	105	144	181	211	252	255	211	160	157	
39	0	0	0	0	76	74	77	75	95	137	180	192	220	255	255	200	122	106	
40	0	0	0	0	73	74	78	95	142	165	178	218	255	255	255	241	161	112	
41	0	0	0	0	74	75	86	132	184	167	161	218	255	255	255	255	209	150	
42	0	0	0	0	78	77	97	149	172	156	177	230	255	255	255	255	236	189	
43	0	0	0	0	75	78	89	119	143	177	228	255	255	255	255	255	253	226	
44	0	0	0	0	76	83	85	96	129	178	249	255	255	255	255	255	255	245	
45	0	0	0	78	77	82	87	88	110	134	234	255	255	255	255	255	250	214	

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	[]

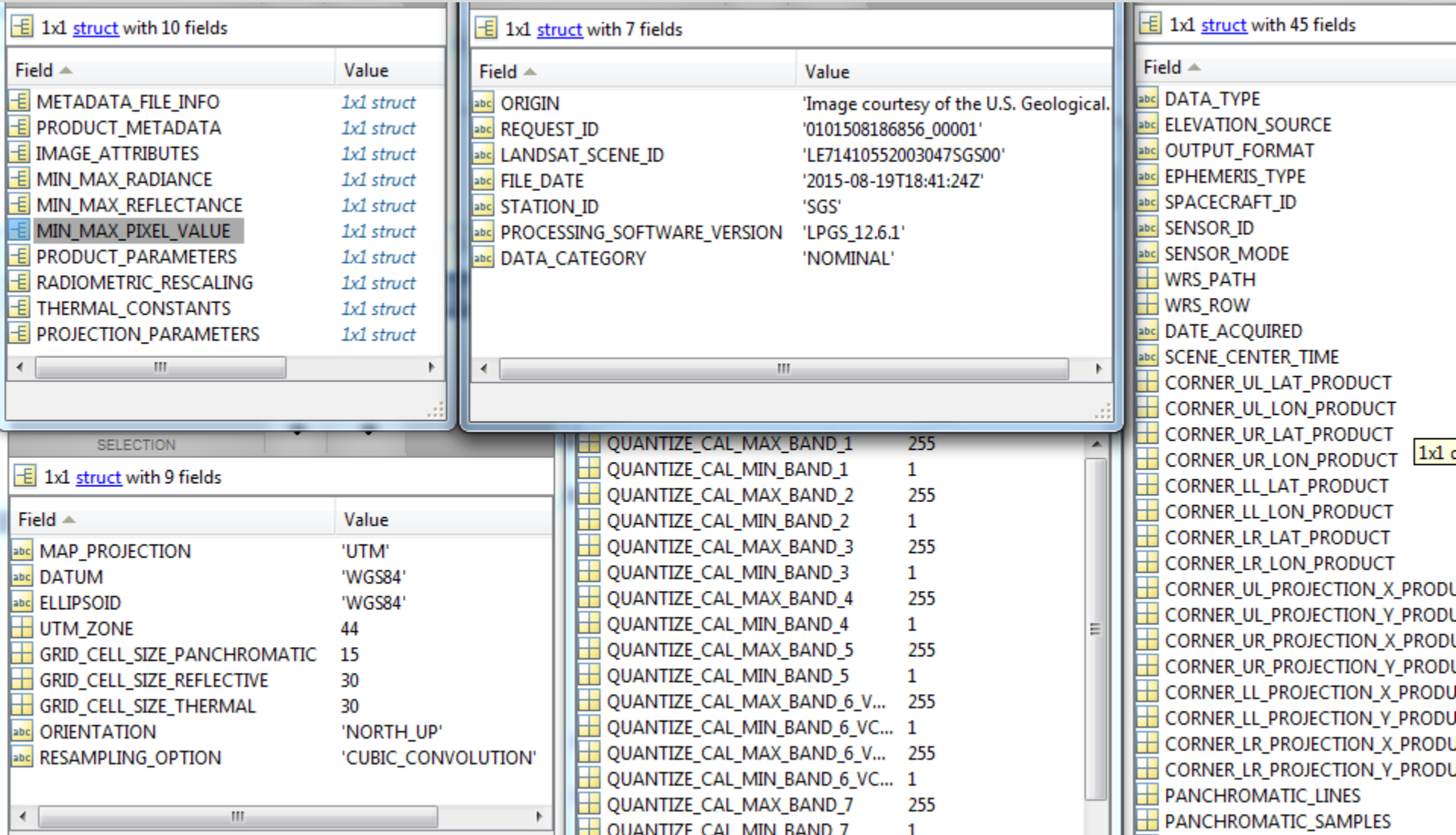
Imfinfo('landsat_img')

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');

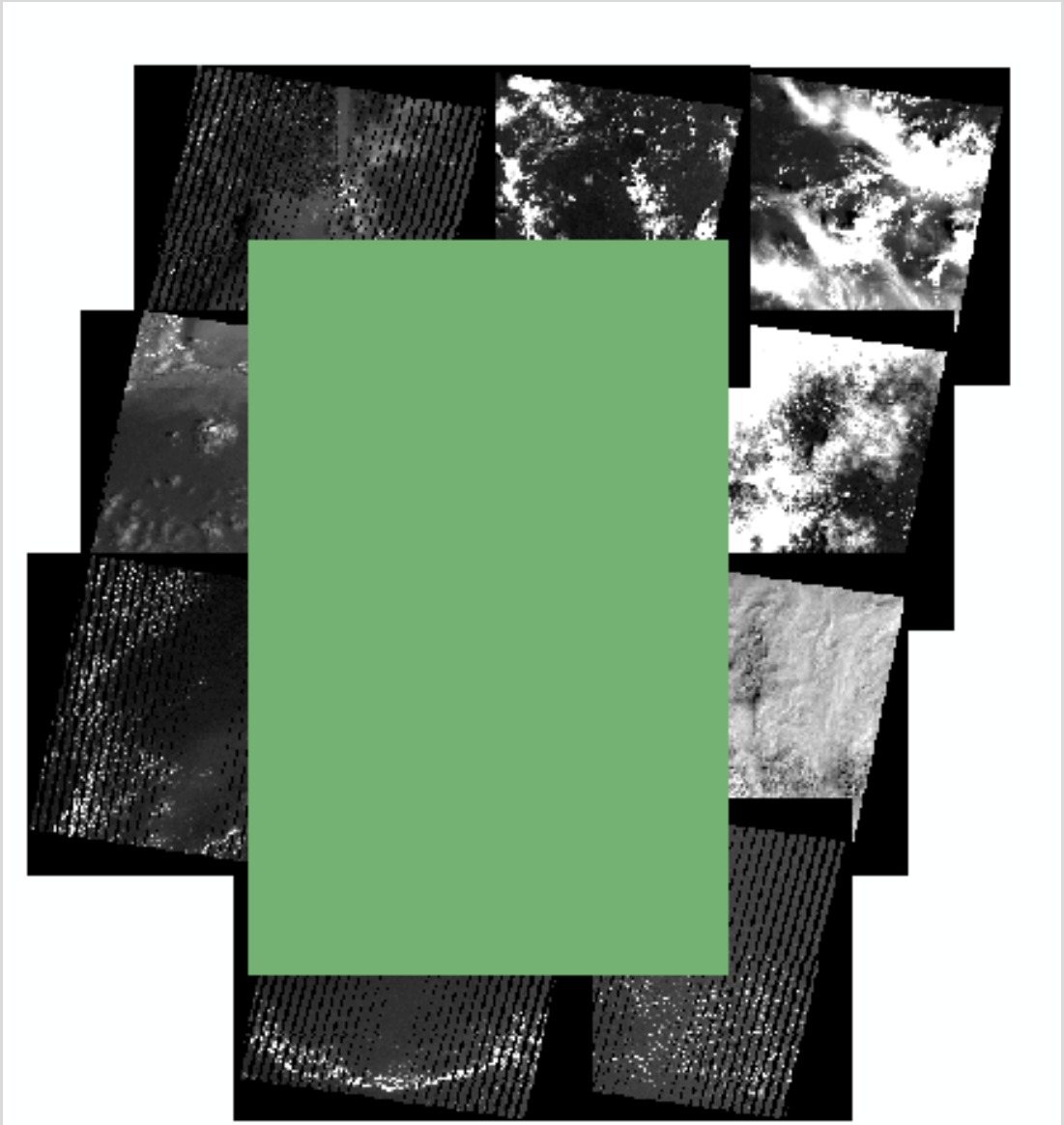
More about Landsat files



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

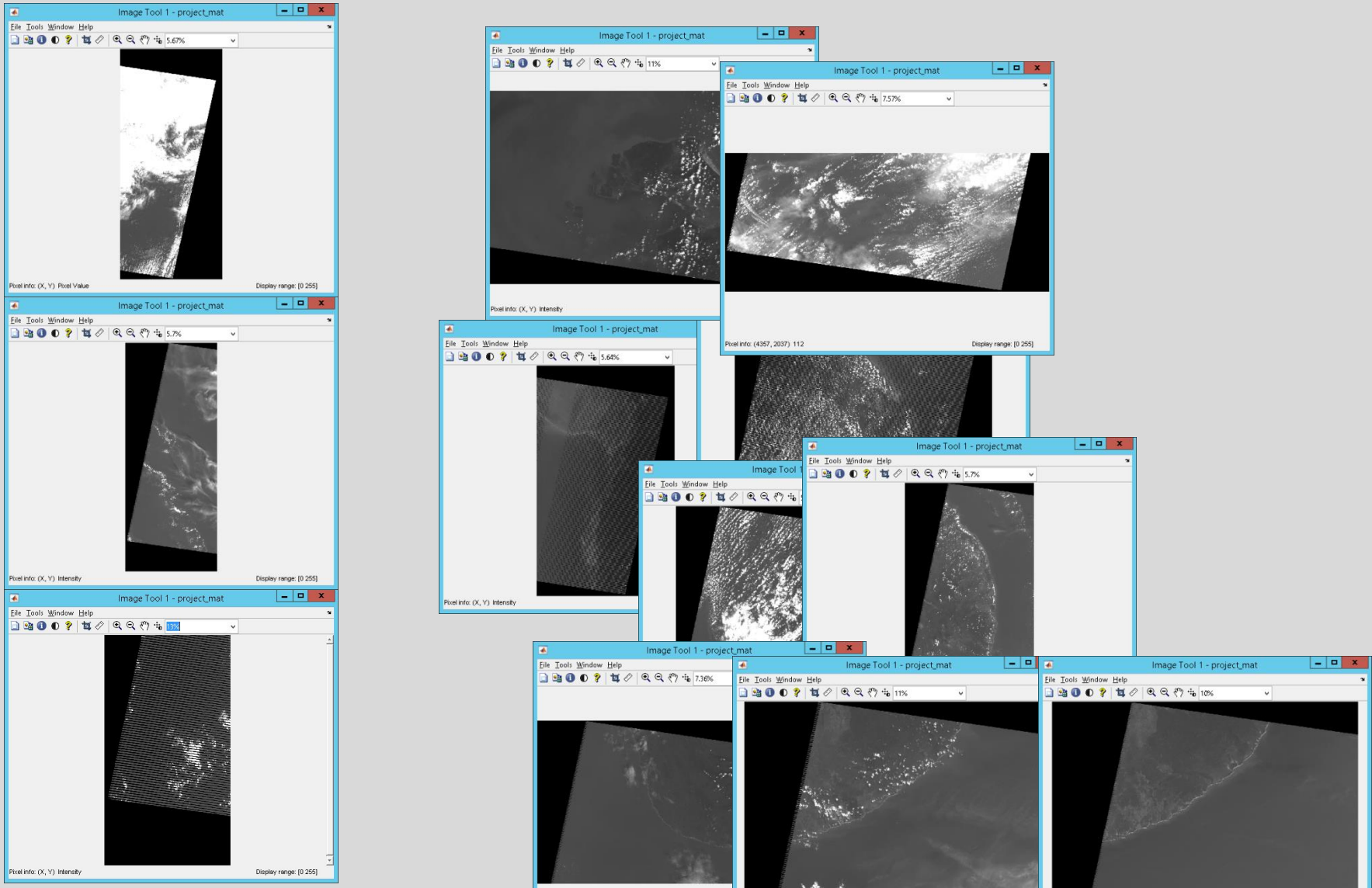


Path	Raw
140	053
140	054
140	055
140	056
141	053
141	054
141	055
141	056
142	053
142	054
142	055

Project_Landsat Intersect extractor

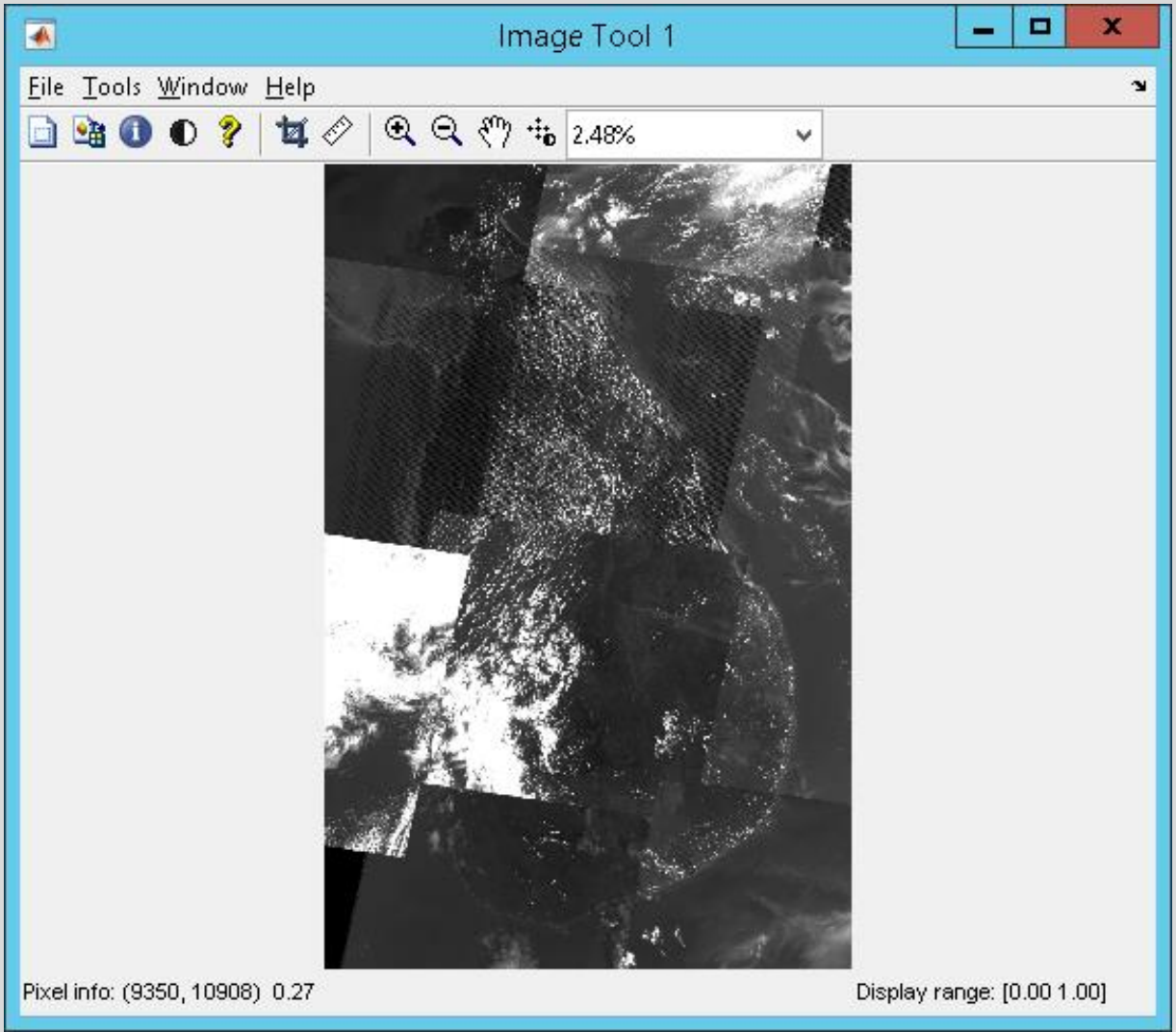
- On each clipped image other algorithms can be preformed
  - Cloud removal (Fmask), TOA correction etc.
- Reproduce Tiff metadata and export.

Mosaic Creator

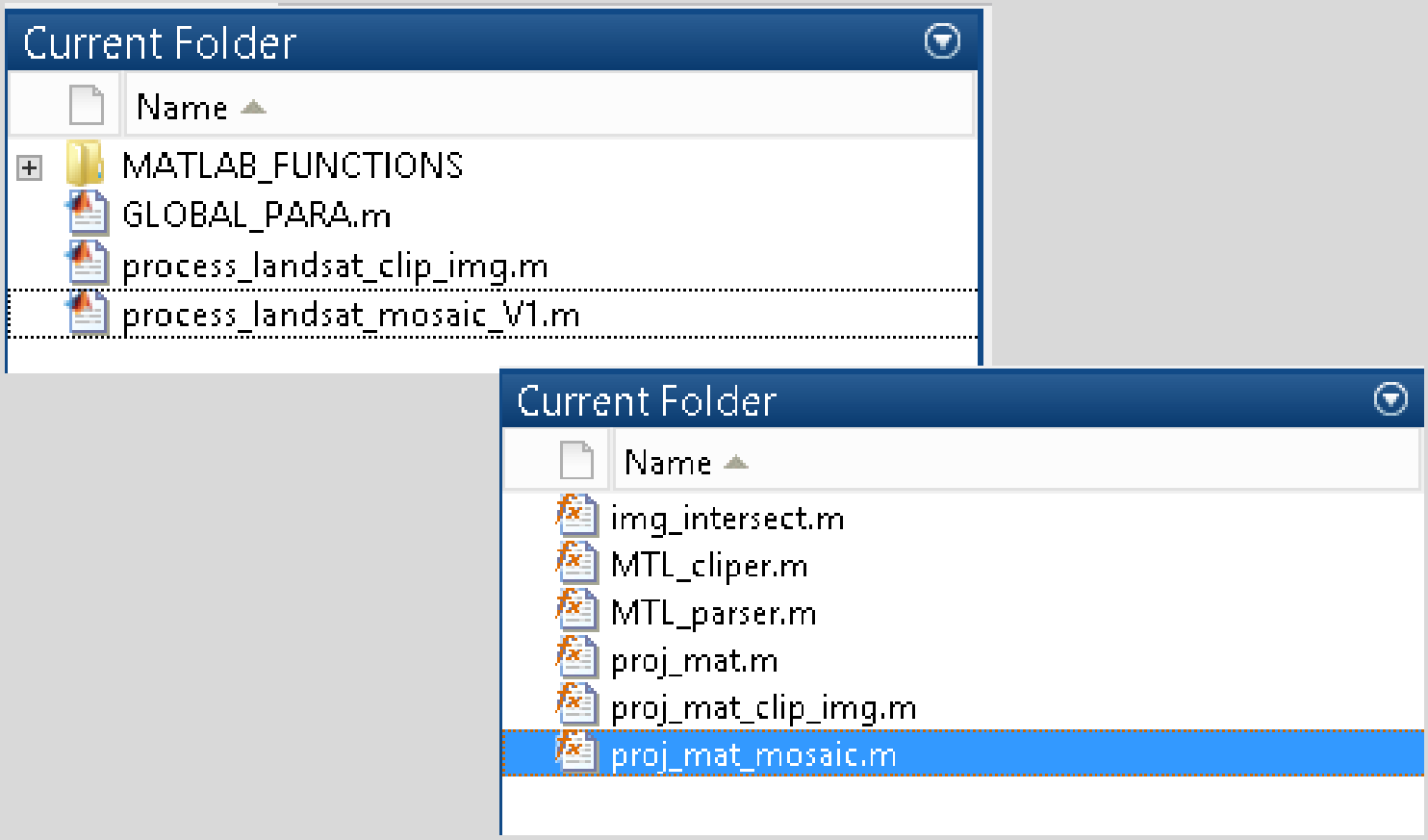


Mosaic Creator

- 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(infilename, outfilename)
7   %%%%% INCLUDES start %%%%%
8   GLOBAL_PARA
9   %%%%% INCLUDES end   %%%%%
10
11  img_land = imread(infilename);
12  img_info = imfinfo(infilename);
13
14  img_X_bl = img_info.ModelTiepointTag(4); % x reamains 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!!!