

# STARSpan:

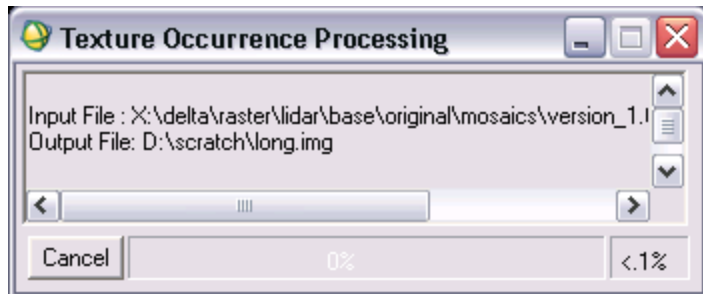
*Bridging vector and raster data*

Jonathan A. Greenberg

Carlos Rueda

CSTARS, UC Davis

# RS is not for the impatient...



Fun game! How many times can you check your email before the process finished?

The amount of time it takes to analyze and process large amount of raster data can preclude investigation into multiple types of analyses, including ones that are not canned as part of a software package.

# Obstacles to RS

- Extracting image data associated with vector field data for external data analysis processing is non-trivial, particularly with data spread across multiple datasets.
- Current RS programs require index & textural transforms and classification of entire image before validation can be performed.
- Validation of classification/model application to raster datasets often requires both quantitative (confusion matrix) and qualitative (visual) assessments.

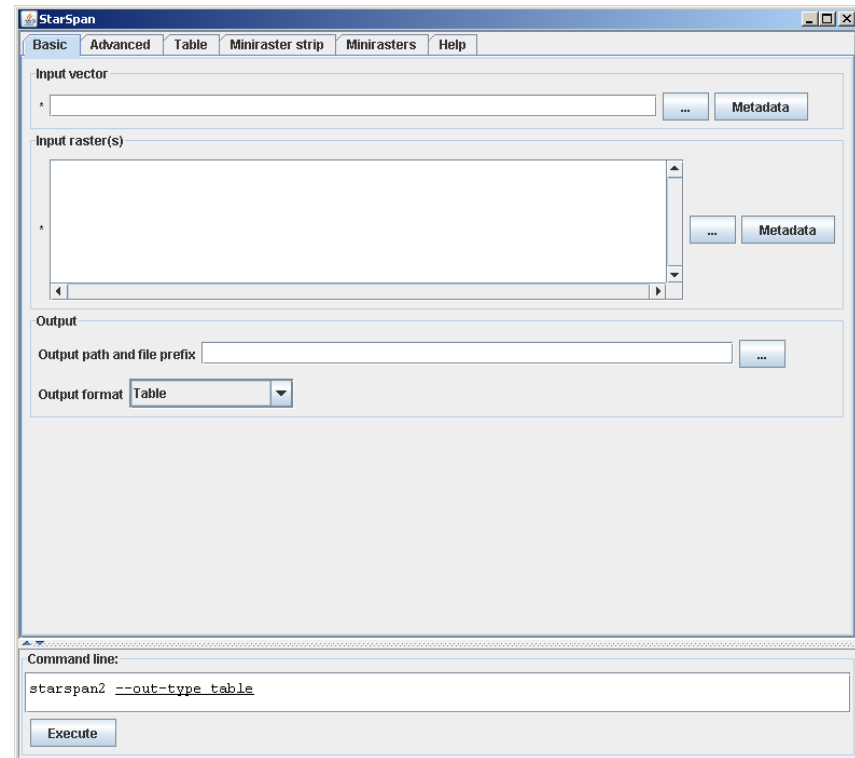
# Current Approaches

Getting data out of imagery for use in third-party stats packages

- Arc: Zonalstats: slow, requires rasterizing vector layers, not scalable to larger images, can only be run on one file at a time.
- ENVI: Output ROIs to ASCII: requires cleaning up the output, can only be run on one file at a time.
- GRASS: v.rast.stats: slow, runs on one file at a time.

# STARSpan in a nutshell

- Inputs: raster image(s), vector coverage
- Outputs: DB with vector columns and new columns containing the pixel values from the raster image(s)



# Table Extraction

- Outputs: vector DB fused with extracted image pixel values (raw or summarized) in ASCII text format.
- Uses: performing analyses of spectral properties based on field measurements using external data analysis packages (e.g. JMP, Excel, R, See5).

# Table Extraction

The screenshot displays the ENVI 4.5 software interface. The main window shows a false-color satellite image of a river and surrounding land. A red box highlights a specific area, which is magnified in the '#1 Zoom [8x]' window. The 'Vector Information' window is open, displaying the following data:

Vector Information	
File	
Type:	Point (Npts:1) (Record:1033)
FID_1:	153
SPECIES_1:	EMR-Pennywort
COVER_1:	80-100
SPECIES_2:	.....
COVER_2:	.....
SPECIES_3:	.....
COVER_3:	.....
SPECIES_4:	.....
COVER_4:	.....
SPECIES_5:	.....
COVER_5:	.....
PATCH_LEN:	S (3-5m)
PATCH_WID:	S (3-5m)
PATCH_ORIE:	0
LOCATION:	side
DEPTH_TO_S:	-99.00000000000000
SECCHI_DEP:	-99.00000000000000
SECCHI_B:	No
WATER_SAMP2:	0
WATER_SAMP:	0
REVISIT_:	No
COMMENTS:	
GPS_DATE:	39247
GPS_TIME:	09:51:50am
DATAFILE:	1A061407.cor
POINT_ID:	202
TEAM:	1A
PHOTOS:	X:\delta\data\field_photos\2007\team1\
LIFE_FORM:	emergent
CHK_ATTRB:	1
CHK_LOC:	1
FLAG:	0
EM_FLAG:	0
EMERGENT_F:	0
FLIGHT_ID:	sdw_2007_p1_40_geo3_ort
SAV_MX_FL:	0

The '#1 Spectral Profile: 200706\_p1\_40\_ref1...' window shows a graph of Spectral Profile with the following axes:

- Y-axis: Value (0 to 5000)
- X-axis: Wavelength (0.5 to 2.0)

The graph shows a prominent peak at approximately 0.8 micrometers, with a secondary peak at approximately 1.6 micrometers. A vertical red line is drawn at the first peak.

Excel spreadsheet with columns A-BS and rows 1-99. Cell B93 (row 93, column B) is highlighted in yellow.

Vector Information dialog box showing file type: Point (Npts:1) (Record:1033) and various metadata fields like FID, SPECIES, COORDINATES, etc.

Cursor Location / Value dialog box showing coordinates and map data: Disp #1 (6368,478) Scrn: R: 244 G: 46 B: 111, Projection: UTM, Zone 10 North, Map: 618521.00E, 4214112.00N Meters, Data: R: 5184 G: 437 B: 783



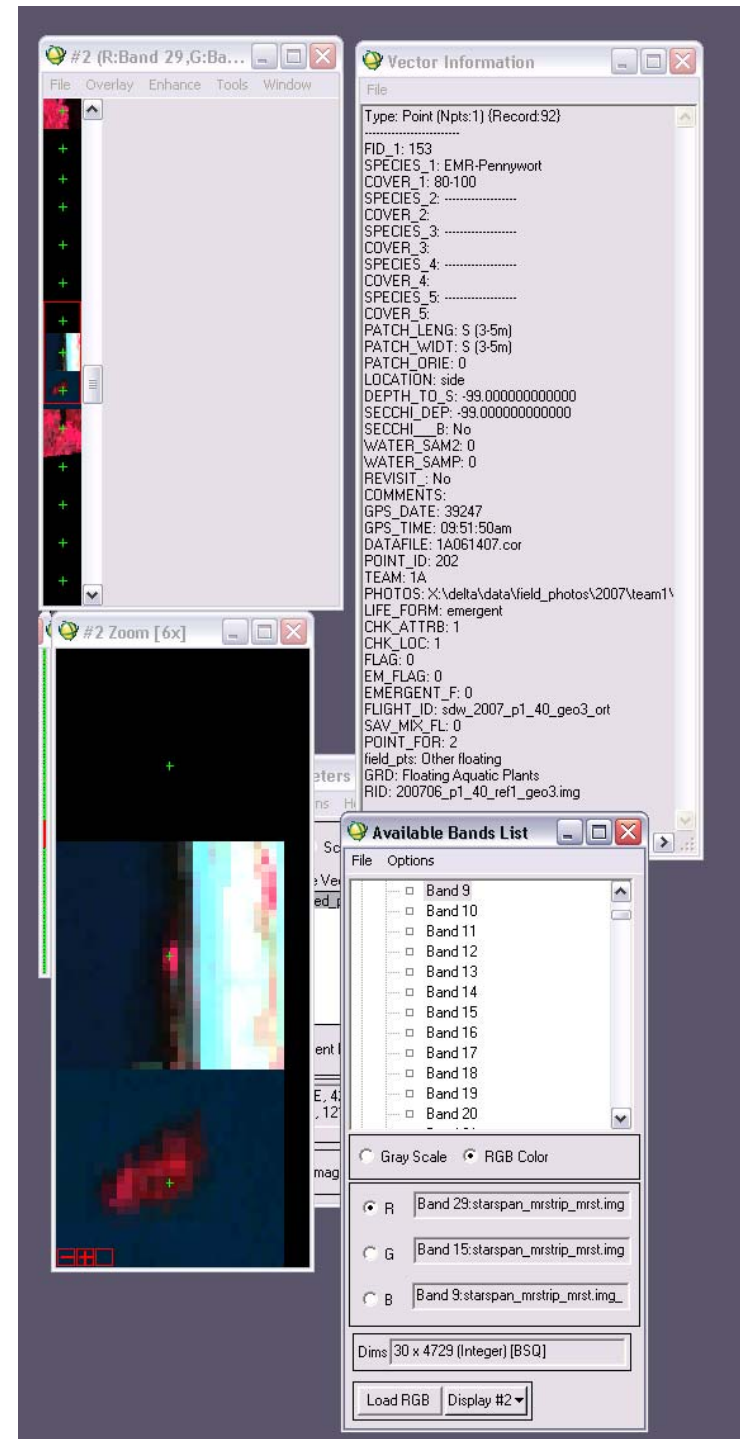
# Advanced Options (Table)

- (Polygons): Summarize the pixels falling within a polygon
  - Default: extract all pixels, which leads to repeated feature info in the database (e.g. the user might summarize by FID).
  - Summarize the values: mean, mode, stdev, min, max, sum, median, nulls.
  - Count by class: for each poly, the number of post-classified pixels for a given class falling in the poly (good for summarizing AOI polys).
- Note: this feature is a fundamental part of object oriented analysis.

# Minirasterstrip

- Outputs:
  - Raster image of vertically stacked “mini rasters” surrounding each input vector.
  - “Reprojected” vector into minirasterstrip space.
  - Affiliated GLT and FID files.
- Uses: rapid image transforms (spectral or spatial), classification, training/test without having to classify the entire image(s).

- The minirasterstrip is significantly smaller than the input imagery, and can compile multiple flightlines into a single training/test image.
- A shapefile referenced to the miniraster strip is created, and can be used to re-starspan from the minirasterstrip.



# Advanced Options (Global)

- Suppress extracting nodata values or masked values (raster masks).
- (Polygons): control the % of a pixel falling in a polygon to be extracted.
- Buffer vectors before extracting.
- Perform SQL querying on the input vectors (untested).

# Advanced Options (Global)

- (Multiple images): Control how to handle vectors falling in  $> 1$  image (“duplicate feature removal”).
  - Extract all pixels, allowing for duplicates to occur.
  - Use the raster who’s center is closest to the feature’s centroid.
  - Use the raster who’s center is closest to a pre-defined azimuth (e.g. always pick the image to the north).

# Suggested Work Flow (Classification)

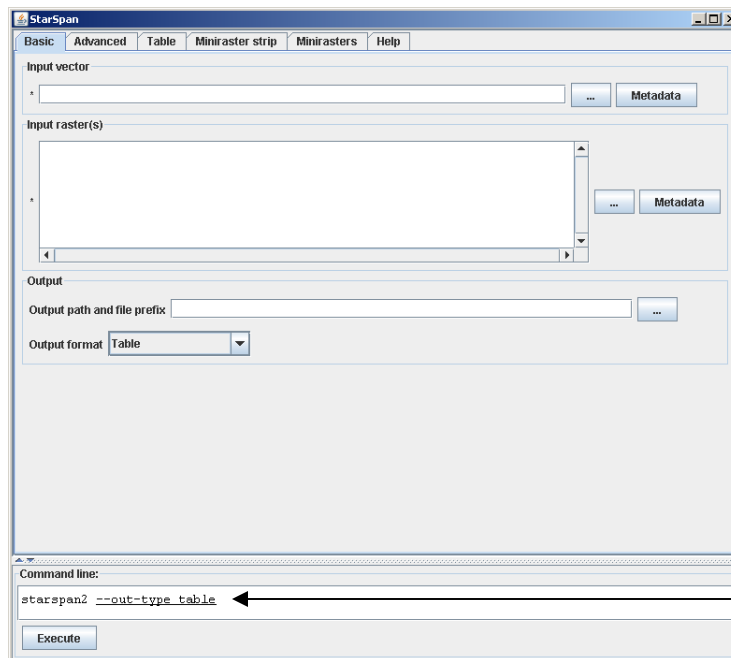
1. Create a minirasterstrip from the images based on the field data with a window big enough to be used for a) texture transforms and b) visual assessment of results.
2. Perform spectral and texture transforms on the minirasterstrip.
3. Split vector into training and test ROIs (ask Jonathan for IDL scripts for this) based on the minirasterstrip-rectified shapefile.
4. Try different classifiers using different combinations of spectral, index and texture.
5. Calculate final accuracy assessments on the minirasterstrip.
6. Once final set of inputs and classifier is chosen, perform relevant transforms and classification on the master dataset.

# Suggested Work Flow (Continuous)

1. Create a table from the images based on the field data.
2. Import table into statistics package.
3. Perform model development and validation using spectral values =  $f(\text{field data})$ .
4. Apply final model to imagery (e.g. band math).

# WAKE UP CODERS

- Starspan is OPEN SOURCE and FULLY COMMAND LINE (if needed) and can be built into a processing chain.



Proof!



# Website

<http://starspan.casil.ucdavis.edu>

**THANK YOU CARLOS!**

**\*sniff\* We will miss you!**

**Good luck at the new job!**