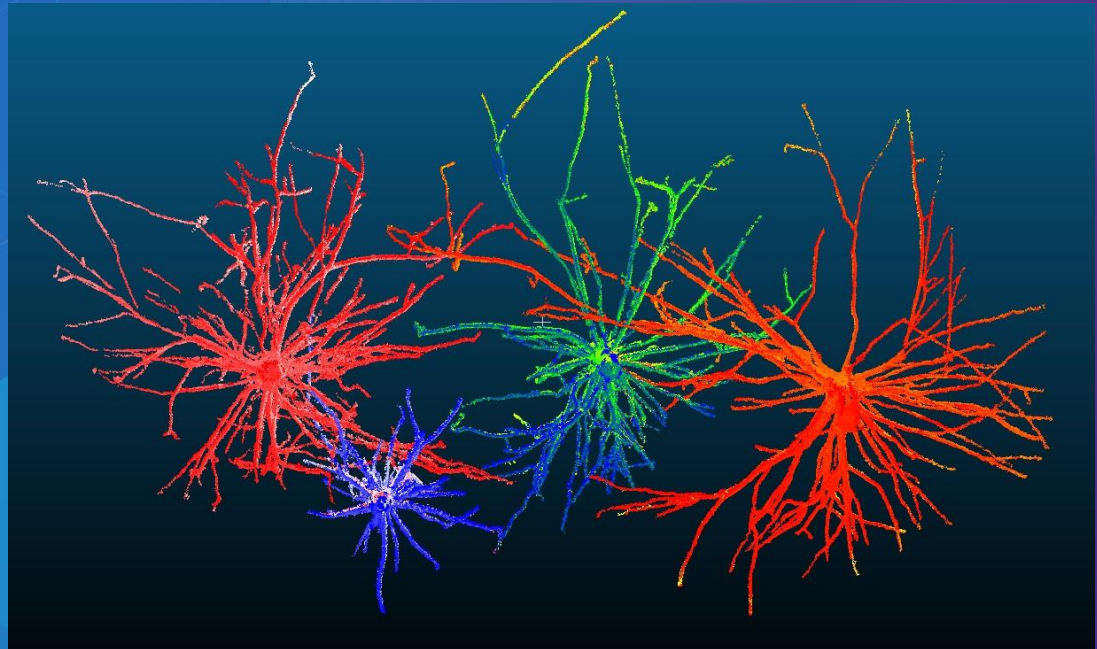


Point Clouds to Vectors: Getting Laser Scan Data into GIS

Keir Keightley

HDR Inc

UC



Introduction

- **Point cloud data in its native format is not “smart”**
- **Deriving information is critical to its utility**
- **ArcGIS has a broad library of tools useful for analyzing data (but requires specific data input formats)**
- **Existing tools get you some of the information you need**
- **Python scripting reveals information not accessible using ESRI toolsets**
- **Goal: a richly attributed feature class providing information useful for expert analysis**



The Challenge

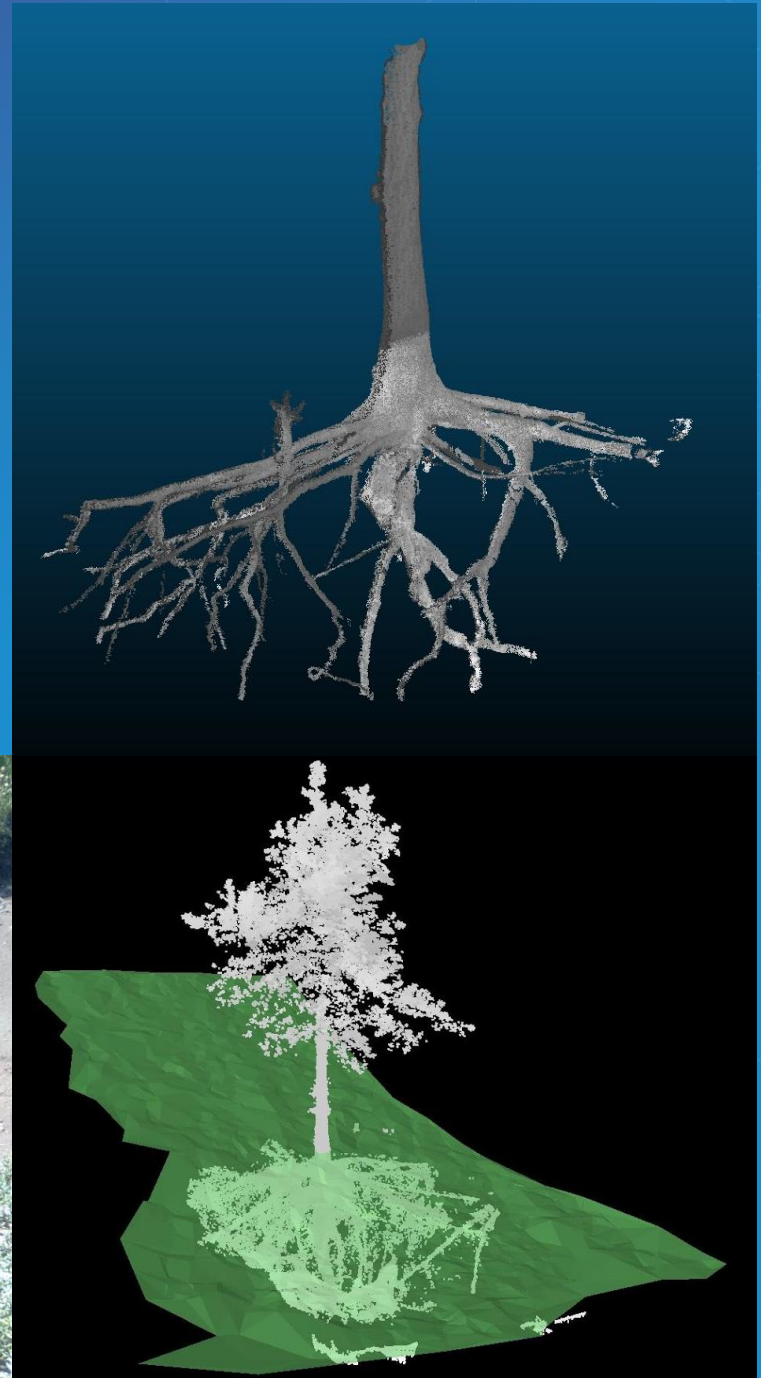
- **Process point cloud of network-friendly structures**
- **Move from point cloud to vector data model**
- **Convert vector model to ArcGIS format**
- **Attribute features with topological information**
- **Design Python code to extract within-feature statistics**
- **Join results to spatial features for analysis and understanding of fundamental 3D interactions**



The Details

Input Data

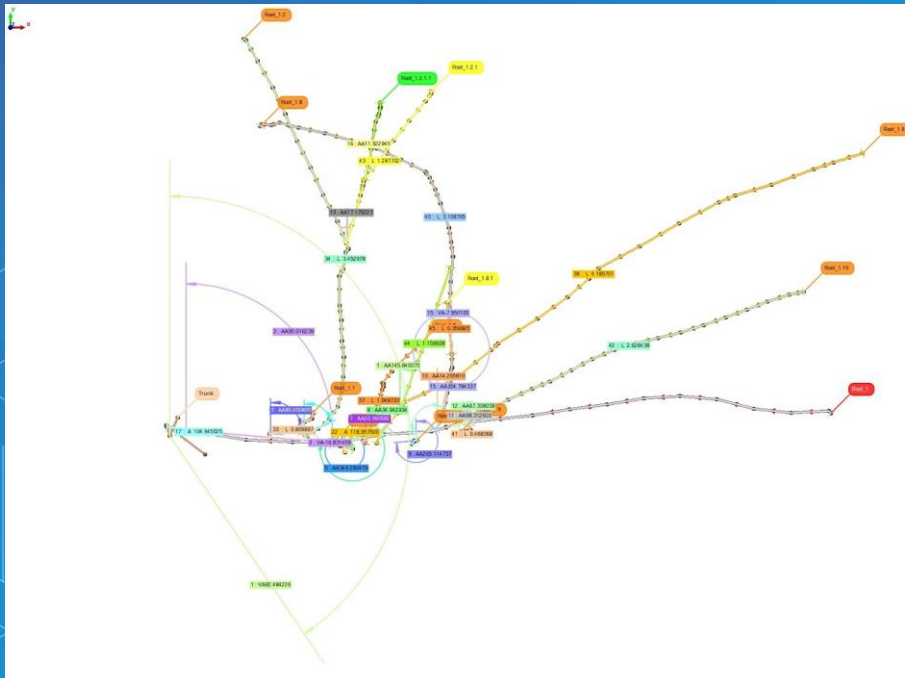
- Multiple laser scans of roots
- Aligned, filtered
- Unified with ancillary data



The Details

Converting to Vector Format

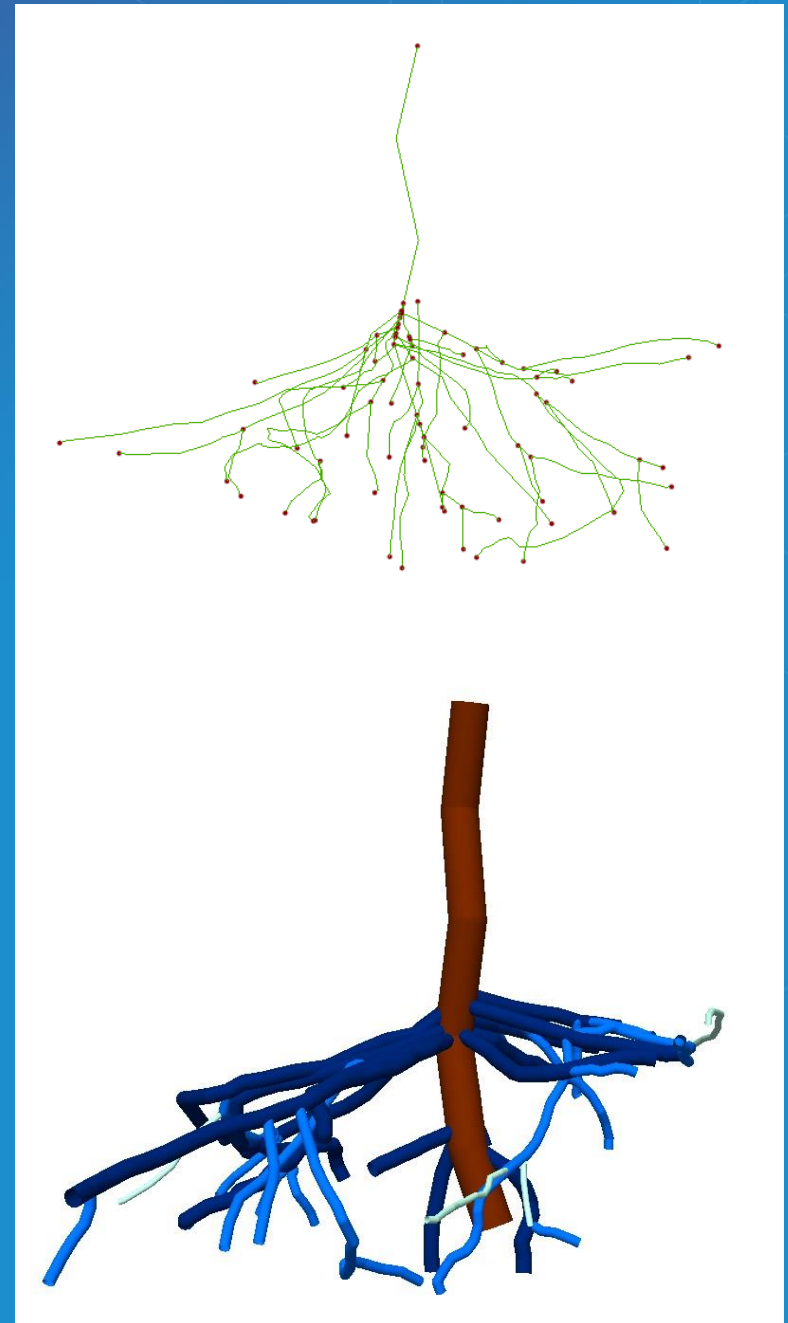
- Semi-automated root tracing
- Creation of individual root geometry
- Not ideal but very effective



The Details

Importing to ArcGIS

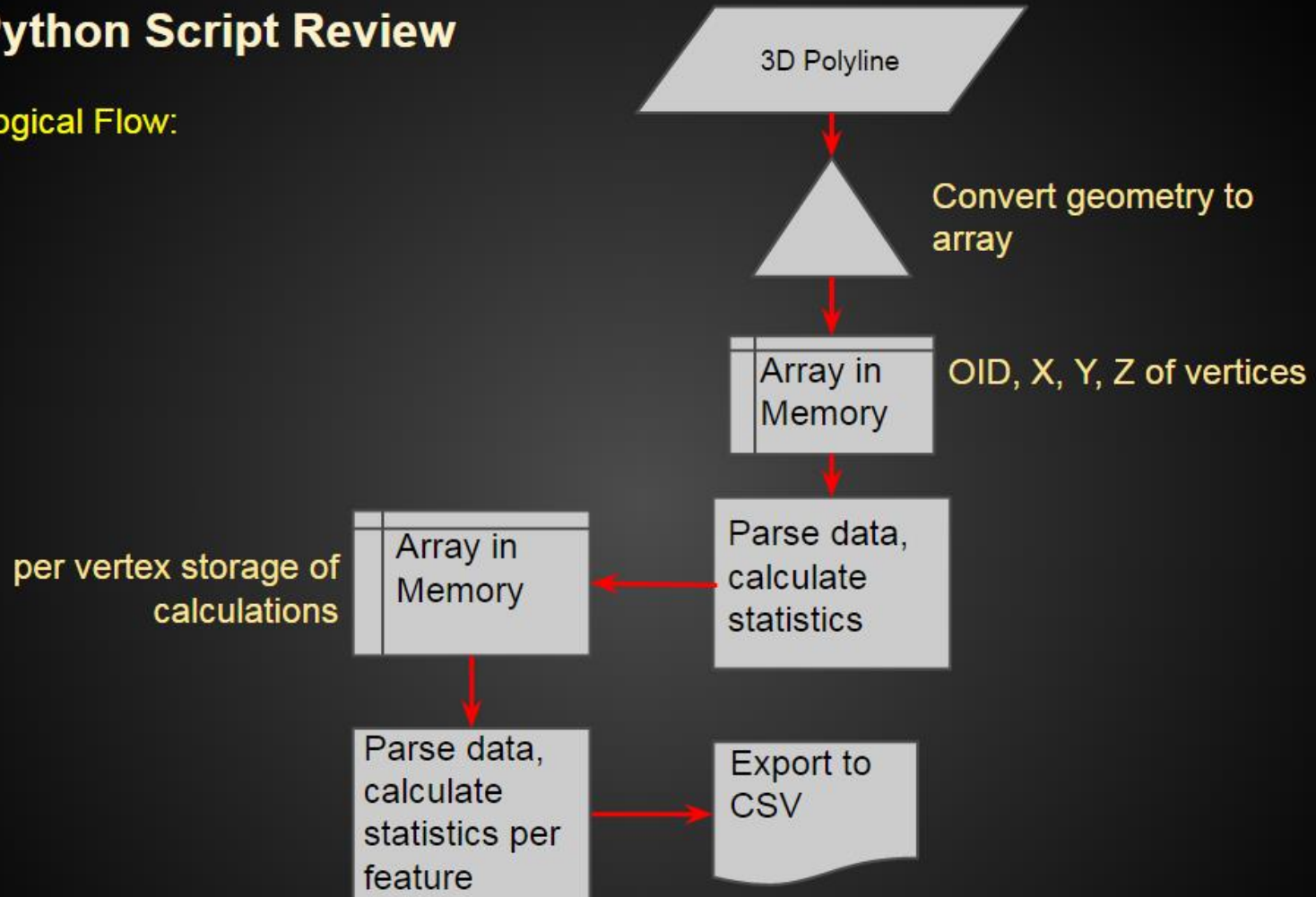
- Vector data imported in IGES format
- Root connectivity established by editing features
- Basic feature information can be viewed or calculated (e.g. 3D length, connectivity, root order/heirarchy)
- But...root systems with different architecture might have very similar information (root lengths, basic topology)



Logical Steps in Analysis

Python Script Review

Logical Flow:



The Details

Calculations used

- **Vertex to vertex analysis**

```
array = arcpy.da.FeatureClassToNumPyArray(fc,["OID@", "SHAPE@X", "SHAPE@Y", "SHAPE@Z"],explode_to_points=True)
```

- **Vector segment dot product**

```
dotprod = np.dot(V1,V2)
```

```
thetarad = math.acos(dotprod/(V1mag*V2mag))
```

- **Summation and division**

```
V1s = math.degrees(math.acos(np.dot(V1,V3)/(V1mag*V3mag)))
```

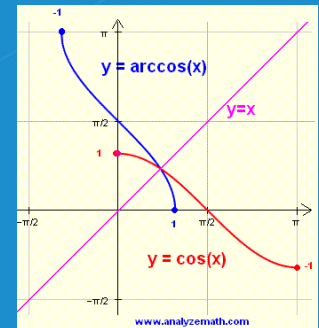
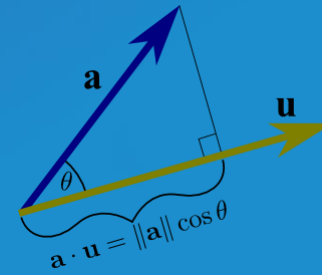
```
if V1[2]>0:
```

```
    V1s = -1*V1s
```

```
    segment[0][5] = V1s
```

```
statsrow[0] = featurarray[0][2] + featurarray[:,3].sum()
```

- **Etc...**

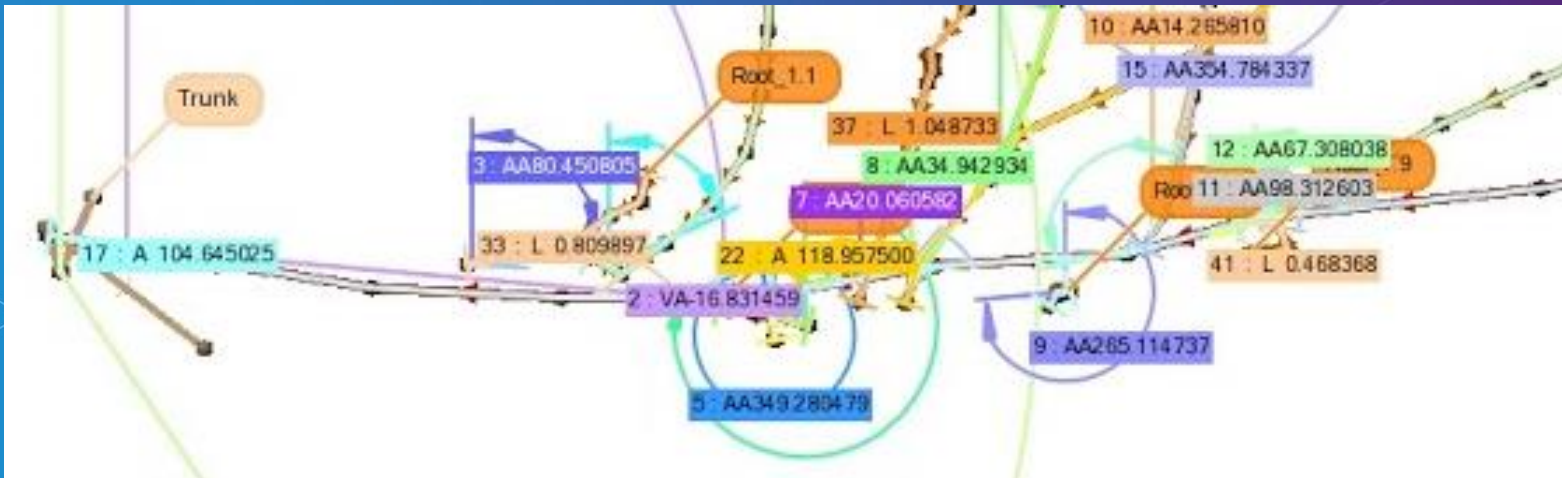


Python Output

```
3D_stats_out.txt - Notepad
File Edit Format View Help
OID,3D_Len,AvgDef,AngTort,MaxDef,Maxslope,Tau,Res
0.0,4105.607,79.481,0.883,103.896,29.679,1.608,0.000,0.000
1.0,4840.124,84.988,1.180,105.684,33.652,1.662,0.000,0.000
```

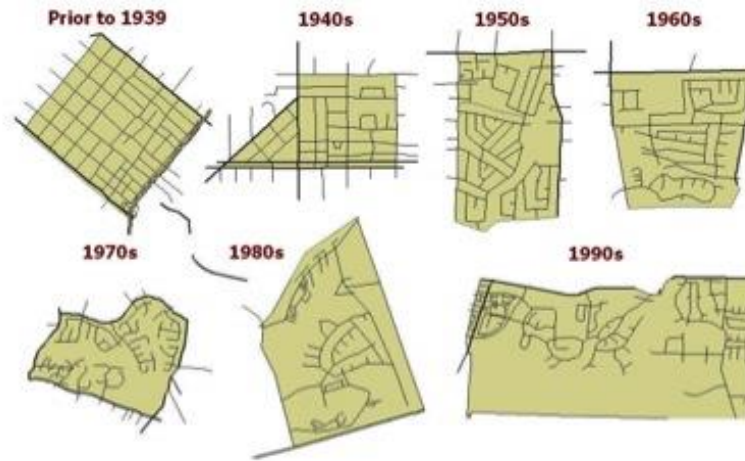
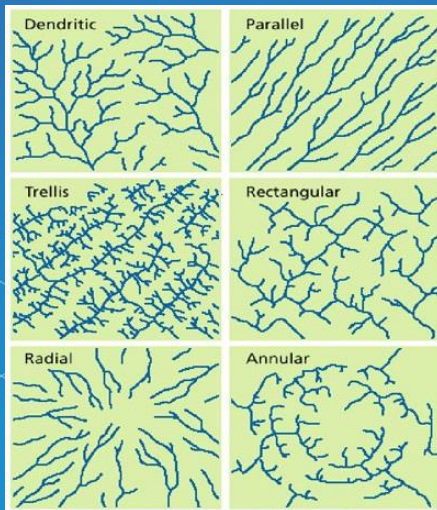
3D_out.txt - Microsoft Excel

	A	B	C	D	E	F	G	H	I
1	Index	OID	V1mag	V2mag	Deflection(deg)	V1slope	V2slope	V1hormag	V2hormag
2	0	0	753.005	1290.971	65.621	1.357	2.896	752.794	1289.323
3	0	0	1290.971	433.301	76.705	2.896	-19.892	1289.323	407.448
4	0	0	433.301	967.691	71.699	-19.892	-8.093	407.448	958.054
5	0	0	967.691	660.639	103.896	-8.093	29.679	958.054	573.972
6	1	1	684.362	899.068	75.517	-27.474	19.541	607.182	847.285
7	1	1	899.068	833.38	95.74	19.541	-24.618	847.285	757.629
8	1	1	833.38	746.709	105.684	-24.618	33.652	757.629	621.575
9	1	1	746.709	790.611	48.565	33.652	-31.244	621.575	675.943
10	1	1	790.611	885.994	99.432	-31.244	28.594	675.943	777.931
11									



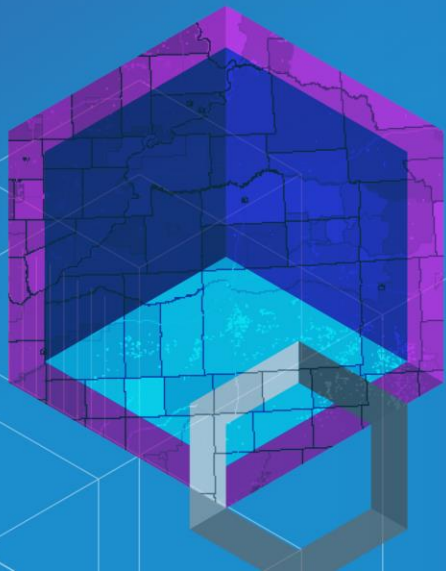
Other Applications

- Hydrology: stream network characterization
- Transportation: efficiency and traffic flow analysis
- MEP (CAD/BIM)



Acknowledgments

- New Mexico Interstate Stream Commission
- Gerald Bawden USGS Western Remote Sensing and Visualization Center
- Allison Berry UC Davis Dept. of Plant Sciences



Calvin: "Did you finish your map of our neighborhood?"

Hobbes: "Not yet. How many bricks does the front walk have?"

Bill Watterson