Using Automated Processing of Aerial Imagery for Municipal Mapping Updates

Stephen Perkins – Senior Project Photogrammetrist, City of Ottawa

Brad Schmidt – Business Development – Intergraph Canada

Monday June 9th, 2014
Agenda

• Part 1: Overview of Automated Processing Techniques (Brad Schmidt)
  • Workflow Authoring Tools
  • Workflow Publishing
  • Accessing and Running Automated Processes
  • Benefits
Agenda

• Part 2: Municipal Mapping Updates from Aerial Photography (Stephen Perkins)

Some examples:

• Ortho-photos
• Vector Mapping
• Road Network
• Property Mapping
• Elevation Models
Session Theme: Transforming the Management and Governance of the Client Experience - “Doing More with Less”

• Making better use of workflow processing tools in support of the client experience
Workflow Authoring (movie)
Workflow Authoring

**Scripting:**

Python is a cross platform scripting language with huge popularity

Easy to learn and use
Workflow Authoring

Modify the model by using Python script, add other operators from ERDAS IMAGINE, GeoMedia, or ArcGIS

Creating Python Scripts:

Execute the model in Python
Publish to Enterprise or Cloud as a Web Processing Service (WPS)

Server-Side Geoprocessing

Process added to the server in the specified category
What is Web Processing Service (WPS)?

Web Processing Service (WPS) is a web service standard providing on-demand, dynamic geoprocessing

- [http://www.opengeospatial.org/standards/wps](http://www.opengeospatial.org/standards/wps)

According to the OGC WPS interface specification, a geoprocess is defined as any calculation operating on spatially referenced data.
Processing Properties
(End User Access)

Processing Properties
Stored in model

**Name**: Unique name for model used internally

**Display Name**: Name seen in web client

**Display Category**: Group in which the geoprocess will be placed

**Description**: Describes the model to help user choose the right geoprocess for their needs
Running the Workflow

Expert Creates Spatial Model

Publishes Model to APOLLO Catalog

Server Runs Spatial Model

Web Clients Query APOLLO

Consumer Downloads Product
Benefits

- User works with data from a collective data pool
- Data stays on the server and no need to download onto the local system.
  - Saves bandwidth, time and disk space
  - Reduces duplication so backups are quicker and the data is maintained centrally with its metadata
- Model runs on the server where it has optimized access to the data and an optimized hardware environment.
- When the model has been run, the results are added back into the data pool so others can visualize and even use again in other models
Benefits

• No need to learn new software for the end users, just use the modeler like any other web page
• No need to load bulky software on the local system (maintenance, disk space etc.)
  – All end users use server side modeler
• Workflows can be easily edited and changed if required
• Brings workflows closer to the end user
• Ortho-Photo’s
• Vector Mapping
• Road Network
• Property Mapping
• Elevation Models
Ortho-photos

- 15cm 1958
- 20cm 1965
- 50cm 1976
- 20cm 1991
- 25cm 1999
- 25cm 2002
- 20cm 2005
- 6cm 2007
- 20cm 2008
- 20cm 2011
- 6cm 2011
Vector Mapping

- 1:1000 scale
- 1:2000 scale
- 1:10000 scale
- Road Network
3D Mapping
Compiled Products

Ortho w/ property
Security for
State Visits
Development Cycles

a) Plane table photogrammetry, from about 1850 to 1900
b) Analog photogrammetry, from about 1900 to 1960
c) Analytical photogrammetry, from about 1960 to 2000
d) Digital photogrammetry
Photogrammetry

Edouard Deville (February 26, 1849 - September 21, 1924), the Surveyor General of the Dominion of Canada
The First Aerial Photographs ...
a hint ...
Dr Julius Neubronner’s Miniature Pigeon Camera ...
Stereo-plotting Machines
Softcopy Photogrammetry
(geoOttawa)
3D Mapping
Scripting ...
# Create a spatial model that mosaics all the images.
# Start from a model with the window rule set to Union:
m = modeler.Solution.Load(dataPath + "union-template.gmdx")
lastImage = None
for filename in images:
    # Add the Raster Input operator:
rasterInput = m.RasterInput(filename)
    # Add the Set To NoData operator:
maskZero = m.SetToNoData(rasterInput, 0)
    # Add the Replace NoData With operator - tricky!:
if lastImage != None:
    replaceNoData = m.ReplaceNoDataWith(lastImage, maskZero)
    # This will be the input to the next Replace operator:
lastImage = replaceNoData
else:
    lastImage = maskZero
Questions?