Workflows for Aerial Frame Cameras

Cody Benkelman
Objectives

- Manage and share collections of imagery from aerial frame cameras

  - Professional digital cameras
    - Metric lens, precise positioning with GPS & IMU
    - Applicable to professional oblique cameras – multiple sensors
  
  - Scanned historical aerial photos include this workflow, w/ more preprocessing steps

- Uncalibrated frame cameras on unmanned aerial systems (UAS) or drones

- Intent is to manage & process single images from a frame sensor
  - Satellite imagery – different sensor model (use proper Raster Type)
  - Aerial line scanners (ADS 40/60/80) – unique geometry, hardware-specific workflow
  - Preprocessed Orthophotos require a different workflow
Image Management Workflow Using Mosaic Datasets
Highly Scalable, From Small to Massive Volumes of Imagery

Create Catalog of Imagery
- Reference Sources
- Ingest & Define Metadata
- Define Processing to be Applied

Apply:
- On-the-fly Processing
- Dynamic Mosaicking

Access as Image or Catalog
Advantages
On-the-fly orthorectification, dynamic mosaics via Mosaic Dataset

- Can be ‘on demand’ and avoid output of duplicate pixels
- Bring overlapping images to foreground
- Navigate on map, but view in image coordinates (unprojected) for best clarity
- 3D mensuration
- Stereo viewing
Ortho Mapping workflows in the ArcGIS Platform

UAS (Drone) → Drone2Map → Pro Ortho Mapping (Desktop) → Ortho Mapping

Satellite → Image Server → Data Store → Python API

ArcGIS Online Workflows → Imagery Workflows → Ortho Mapping

Upload? → Image Server Ortho Mapping → Publish → Ortho → DTM → Tiles

Tiles → DSM → True Ortho Frame

DTM → DSM → Tiles

I3S → LAS → Tiles

3D Workflow → 2D Workflow → 2D Workflow → 2D Workflow

Pro Ortho Mapping (client to Image Server)
Geometric processing – workflows for aerial cameras

- **Pro Ortho Mapping (Desktop)**
- **Overlay/Ortho Mapping (client to Image Server)**
- **Data Store**
- **Image Server**
- **Ortho Mapping**
- **Python API**
- **Satellite**
- **Aerial**
- **UAS (Drone)**
- **ArcGIS Online**

**Workflow Diagram**:
- **DTM**, **DSM**, **Tiles**
- **2D Workflow**
- **Enterprise**
- **ArcGIS Online**

**Key Terms**:
- Geometric processing
- Workflows for aerial cameras
- DTM (Digital Terrain Model)
- DSM (Digital Surface Model)
- Tiles
- Ortho Mapped Images
Frame Camera raster type (interior/exterior orientation)
Frame Camera raster type (interior/exterior orientation)

- UAS (Drone)
- Digital or scanned film
- Frame & Camera Tables
- Image Server
- Ortho Mapping
- Frame & Camera Tables
- Drone2Map
- Pro Ortho Mapping (Desktop)
- ArcGIS Online
- 3D Workflow
- 2D Workflow
- Imagery Workflows
- True Ortho
- Frame & Camera Tables
- DTM
- DSM
- Tiles
- 2D Workflow
- Image Server
Support for frame imagery data

- Use Mosaic Dataset to manage both film and digital frame camera data

- A generic solution to support any frame camera

- Required information:
  - Interior orientation (camera parameters)
  - Exterior orientation (unique frame parameters)
Frame Camera geometry for Photogrammetry in ArcGIS

http://esriurl.com/FrameCameraDetailDoc
Basic workflow in ArcGIS

Frame imagery *including* the Frame Camera raster type

- Create Mosaic Dataset
- Use the appropriate Raster Type to ingest imagery based on available orientation data
  - Applanix, Match-AT, ISAT
  - **Frame Camera** *(added at 10.3.1)*
- Populate integrated metadata into Mosaic Dataset
  - Sensor location \((x,y,z)\) and orientation \((o,p,k)\)
  - Other metadata may be added to facilitate management & analysis
- If accurate → proceed to using orthorectified imagery
- If approximate → proceed with **Ortho Mapping workflow**
Two approaches
Specifically for the Frame Camera raster type

• Image collections with complete orientation parameters
  - LeadAir
  - UltraCam
  - etc.

  → Generate Frames and Cameras tables from calibration report, etc.

• Orientation parameters generated by software
  - OrthoMapping in ArcGIS Pro
  - Drone2Map
  - Pix4D Mapper™
  - etc.

  → Generate Frames and/or Cameras table from exported project report.
Prepare inputs for *Frame Camera* Raster Type

- Consolidate exterior/interior orientation parameters
  - GPS file
  - Camera file
  - Frame parameters file (*.txt, *.csv, or *.xml)

- Create Frames table and Cameras table
  - Format the orientation parameters per *Frame Camera* raster type schema
  - Supports radial distortion correction
  - Works for any camera
  - Input format can be csv/txt/feature class/GDB table

See in ArcGIS Help System:
http://esriurl.com/FrameSchema
http://esriurl.com/CameraSchema
Frame Camera Raster Type – Exterior orientation

Frames table

• Required: PerspectiveX/Y/Z and image path (relative or absolute)
• Omega/Phi/Kappa
• Spatial Reference System (SRS)
• Add raster info fields to speed up ingest process
  - NCols, NRows, NBands, PixelType

<table>
<thead>
<tr>
<th>PerspectiveX</th>
<th>PerspectiveY</th>
<th>PerspectiveZ</th>
<th>Omega</th>
<th>Phi</th>
<th>Kappa</th>
</tr>
</thead>
<tbody>
<tr>
<td>451681.6722</td>
<td>3158470.35</td>
<td>809.574638</td>
<td>-0.064682</td>
<td>-0.139704</td>
<td>-179.71770</td>
</tr>
<tr>
<td>461663.087</td>
<td>3156371.445</td>
<td>811.472574</td>
<td>-0.028023</td>
<td>-0.063748</td>
<td>-179.64660</td>
</tr>
<tr>
<td>461682.6709</td>
<td>3158272.916</td>
<td>811.971936</td>
<td>-0.105496</td>
<td>0.46202</td>
<td>-179.51621</td>
</tr>
<tr>
<td>461682.0233</td>
<td>3158173.322</td>
<td>812.724519</td>
<td>-0.080668</td>
<td>0.000266</td>
<td>-179.54476</td>
</tr>
<tr>
<td>451631.5294</td>
<td>3158074.227</td>
<td>813.701214</td>
<td>0.077</td>
<td>0.206639</td>
<td>-179.64853</td>
</tr>
<tr>
<td>451681.6746</td>
<td>3157975.307</td>
<td>814.514642</td>
<td>-0.052909</td>
<td>0.052173</td>
<td>-179.56249</td>
</tr>
<tr>
<td>461692.6923</td>
<td>3157876.594</td>
<td>815.005099</td>
<td>-0.02076</td>
<td>-0.014977</td>
<td>-179.84698</td>
</tr>
<tr>
<td>461684.4121</td>
<td>3157777.858</td>
<td>814.663818</td>
<td>-0.035867</td>
<td>-0.058924</td>
<td>-179.74516</td>
</tr>
<tr>
<td>461685.7635</td>
<td>3157679.019</td>
<td>813.618601</td>
<td>-0.037719</td>
<td>0.139921</td>
<td>-179.61517</td>
</tr>
<tr>
<td>461585.702</td>
<td>3157580.569</td>
<td>811.634429</td>
<td>-0.056909</td>
<td>0.159438</td>
<td>-179.73050</td>
</tr>
</tbody>
</table>
Frame Camera raster type – Interior Orientation in Cameras table

- Focal length & principal point position
- Pixel size
- Affine transformation (Camera to Image)
- Fiducial locations
- Konrady distortion model
  - Radial (Konrady)/Decentering correction

\[
\begin{align*}
\hat{x}_{\text{corrected}} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\
\hat{y}_{\text{corrected}} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)
\end{align*}
\]

<table>
<thead>
<tr>
<th>FocalLength</th>
<th>Principal X</th>
<th>Principal Y</th>
<th>A0</th>
<th>A1</th>
<th>A2</th>
<th>B0</th>
<th>B1</th>
<th>B2</th>
<th>Konrady</th>
<th>DistortionType</th>
</tr>
</thead>
<tbody>
<tr>
<td>79887.2</td>
<td>-13.3</td>
<td>-54.6</td>
<td>-26852.8</td>
<td>5.2</td>
<td>0</td>
<td>20176</td>
<td>0</td>
<td>-5.2</td>
<td>0;0.000006203849275567615;0.00000000150765316147402;0.00000000000003426861243162828;0</td>
<td>Konrady</td>
</tr>
</tbody>
</table>

Cameras table (10.5 or earlier, MUST be named simply “Cameras”)
How to compute $A_0$, $A_1$, $A_2$, $B_0$, $B_1$, $B_2$

- Calculate affine transformation matrix
  - Inputs:
    - Number of Columns and Rows (for the image file on disk),
    - Pixel size (on the sensor)
  - Calculate $[A_0, A_1, A_2; B_0, B_1, B_2]$  
  - Used to transform coordinates from *image space* (column, row) to *camera space*.

<table>
<thead>
<tr>
<th>A0</th>
<th>A1</th>
<th>A2</th>
<th>B0</th>
<th>B1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>-26852.8</td>
<td>5.2</td>
<td>0</td>
<td>20176</td>
<td>0</td>
<td>-5.2</td>
</tr>
<tr>
<td>-26852.8</td>
<td>5.2</td>
<td>0</td>
<td>20176</td>
<td>0</td>
<td>-5.2</td>
</tr>
<tr>
<td>-26852.8</td>
<td>5.2</td>
<td>0</td>
<td>20176</td>
<td>0</td>
<td>-5.2</td>
</tr>
<tr>
<td>-26852.8</td>
<td>5.2</td>
<td>0</td>
<td>20176</td>
<td>0</td>
<td>-5.2</td>
</tr>
</tbody>
</table>

$C = ((\text{cols}/2)-0.5) \times \text{PS}$

$R = ((\text{rows}/2)-0.5) \times \text{PS}$

$A_0 = -C$

$A_1 = \text{PS}$

$A_2 = 0$

$B_0 = R$

$B_1 = 0$

$B_2 = \text{PS}$

where $\text{PS}$ is camera’s film pixel size in microns.

Spreadsheet available from Esri
Mosaic Dataset workflow Demo

http://esriurl.com/FrameCameraSample
Frame Camera input to **Ortho Mapping** in ArcGIS Pro

- Frame Camera raster type generates orthorectified images on-the-fly
- Accuracy dependent on camera parameters & DEM
- To improve accuracy:
  - Aerial Triangulation & Bundle Block Adjustment → OrthoMapping in ArcGIS Pro
Oblique Image support

- **Frame Camera** raster type is oblique aware
  - Sensor Azimuth - indicate camera horizontal direction (0 ~ 360)
  - Sensor Elevation – indicate camera vertical direction (0 ~ 90)

- Query these values to determine
  - Whether a image is oblique
  - The image’s look angle

- **Web Appbuilder “Oblique Viewer” widget**
  - Create comprehensive web app to view oblique images from frame camera image service
  - Perform mensuration

<table>
<thead>
<tr>
<th>Sensor Azimuth</th>
<th>Sensor Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>358.63081</td>
<td>89.932621</td>
</tr>
<tr>
<td>11.414397</td>
<td>89.929006</td>
</tr>
<tr>
<td>357.567056</td>
<td>89.924382</td>
</tr>
<tr>
<td>10.65224</td>
<td>89.933026</td>
</tr>
<tr>
<td>349.311434</td>
<td>89.933089</td>
</tr>
<tr>
<td>358.810121</td>
<td>89.904169</td>
</tr>
<tr>
<td>12.167622</td>
<td>89.917067</td>
</tr>
<tr>
<td>357.734182</td>
<td>89.917796</td>
</tr>
<tr>
<td>8.229158</td>
<td>89.915742</td>
</tr>
<tr>
<td>1.35762</td>
<td>89.911365</td>
</tr>
<tr>
<td>359.515553</td>
<td>89.916027</td>
</tr>
</tbody>
</table>
Best Practice Recommendations

• Focus on Frames table first
  - Are your images on the correct continent? State? County?

• Cameras table: F and A0..B2
  - DISTORTION TERMS ZERO
  - Are images 2x, 10x the wrong size?
    - (if close but wrong, it could be Z value in Frames table)
  - Are images rotated?
    - A0..B2

• Apply DEM
  - Orthometric or ellipsoidal height?

• LAST: add and resolve distortion terms
Oblique Camera Imagery Demo

Imagery courtesy of Aeroptic, LLC, a KeyW Company
What about Drones?

OrthoMapping in ArcGIS Pro
Standalone processing in Drone2Map

Frame Camera model is applicable, but largely hidden from the user
Overview of alternative Frame Camera workflows

- **Historical imagery**
  - Historical tools
  - Frame/Camera tables
  - Ortho Mapping Workflow
  - Orthos out

- **Newer imagery with good metadata**
  - Frame/Camera tables
  - Ortho Mapping Workflow
  - Orthos out

- **Modern imagery with precise metadata**
  - Frame/Camera tables OR appropriate raster type
  - Ortho Mapping Workflow
  - Orthos out

- **Drone imagery with GPS**
  - Frame/Camera tables OR appropriate raster type
  - Ortho Mapping Workflow
  - Orthos OR Orthomosaics out
Overview of alternative Frame Camera workflows

**Historical Imagery**
- Historical tools
- Frame/Camera tables
- Mosaic Dataset
  - Ortho Mapping Workflow
  - Orthos out

**Newer Imagery with Good Metadata**
- Frame/Camera tables
- Mosaic Dataset
  - Ortho Mapping Workflow
  - Orthos out

**Modern Imagery with Precise Metadata**
- Frame/Camera tables OR appropriate raster type
  - Mosaic Dataset
  - Ortho Mapping Workflow
  - Orthos out

**Drone Imagery with GPS**
- Drone2Map Workflow
  - Orthomosaics out

Python script to ingest photogrammetric tables
Manage outputs in Mosaic Dataset
Summary – and links to further information

Best Practice Workflows for Image Management

Our focus was on creating the mosaic dataset for a single data collection using the Frame Camera Raster Type...

For more info re: data management & automation:

- Guidebook in Help System http://esriurl.com/6007
- ArcGIS Online Group http://esriurl.com/6539
  - Downloadable scripts & sample data
- Recorded webinar: http://esriurl.com/LTSImgMgmt