UAS mapping under the hood

Reuben Settergren and Stewart Walker, BAE Systems, San Diego, California
UAS Mapping Technical Demonstration and Symposium, Palm Springs, 12 September 2016

ES-GEO-071116-0111

BAE SYSTEMS

INSPIRED WORK

Outline

Introduction to the photogrammetric workflow for non-experts from the UAS community

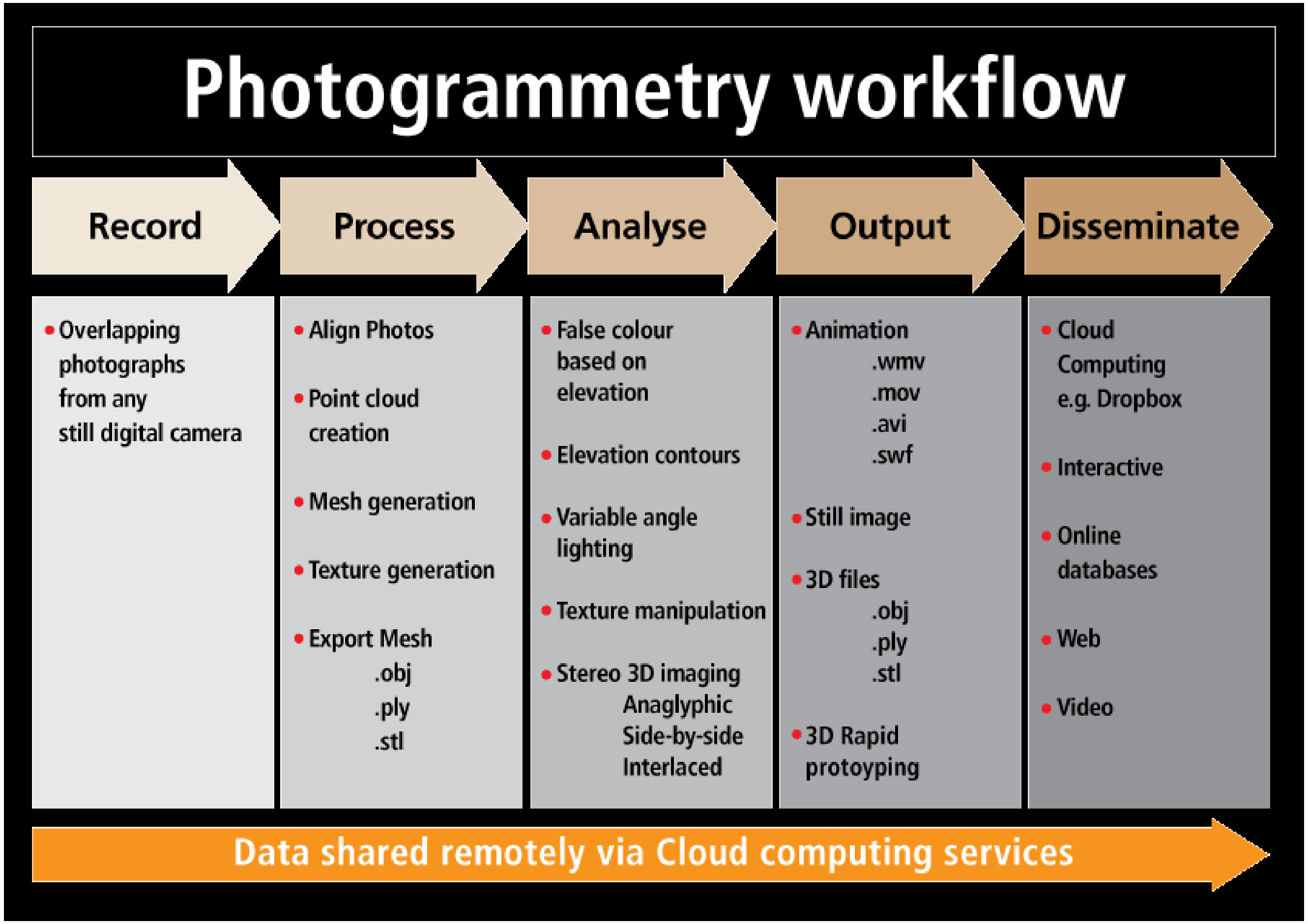
- The photogrammetric workflow
  - Why are UAS projects different?
- Sensor models
  - Computer vision and photogrammetry – two approaches to the same problem
  - Calibration
- Triangulation
  - Image matching
  - Bundle adjustment
  - Quality assessment
- Point clouds – sparse and dense
- Products
  - Terrain
  - Orthorectified images and mosaics

The photogrammetric workflow

The photogrammetric workflow

Walker, A.S., 2010. Application of Thompson's work to modern photogrammetry. E. H. Thompson Centenary Seminar and a Celebration of 60 Years of Geomatics at UCL, London, 17 September 2010.

The photogrammetric workflow …2



http://paleoillustrata.blogspot.com/2013/05/a-photogrammetry-primer-for.html

Why are UAS projects different?

Project characteristic	Manned aircraft	UAS
Typical flying height	500-10,000 m	≤500′
Platform	Stable; small yaw, pitch, roll	Less stable, larger yaw, pitch, roll
Endurance	Long (hours)	Short (20-30 minutes with battery)
Camera	High-performance, metric, calibrated, high optical quality	Prosumer, non-metric, often uncalibrated, moderate optical quality
INS	High performance GNSS[1]/IMU[2]	Usually low-cost GNSS; sometimes low-cost IMU
Area coverage	Large; modest number of images	Small; large number of images
Resolution	Medium-high: ~0.05-1.00 m	High: ~0.01-0.10 m
Flight pattern	Usually parallel strips, carefully planned forward and side overlaps	Ideally parallel strips, sometimes more variable → large overlaps planned to protect against gaps or tilts

1 Global Navigation Satellite System (= GPS + GLONASS + BEIDOU + GALILEO)
2 Inertial Measurement Unit

Sensor models

Photogrammetric sensor models

- Photogrammetry = measuring from photos*
- A photogrammetric sensor model is a geometric/physical model of:
  - how light
  - gets from the ground
  - through the air
  - into a camera
  - to make a pixel

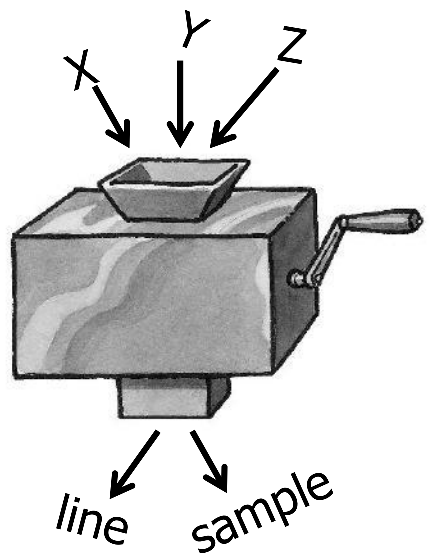
* ASPRS *Manual of Photogrammetry*: "… the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring and interpreting photographic images and patterns of recorded radiant electromagnetic energy and other phenomena."

Photogrammetric sensor models ...2

- Photogrammetry = measuring from photos
- A photogrammetric sensor model is a geometric/physical model of:
  - how light — or laser, or radar pulses
  - gets from the ground — or any location/feature in a scene
  - through the air — or vacuum? vertical? horizontal? oblique?
  - into a camera — or satellite, or SAR
  - to make a pixel — film/digital?
- With a photogrammetric sensor model, you can measure ground (scene) things from images
  - How big?
  - How distant?
  - What angle?
  - **Where**?
  - How fast?

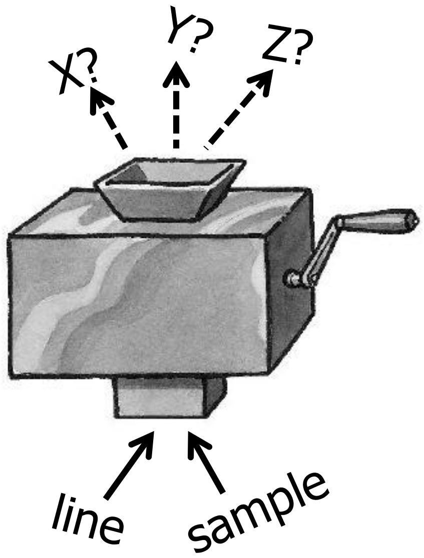
Photogrammetric sensor models …3

- A sensor model is a function/transformation/map: GroundToImage (g2i)
  - From 3-D ground space (i.e., latitude/longitude/height)
  - To 2-D image space (i.e., line/sample)
  - Multiple 3-D ground points for each 2-D image point



Photogrammetric sensor models …4

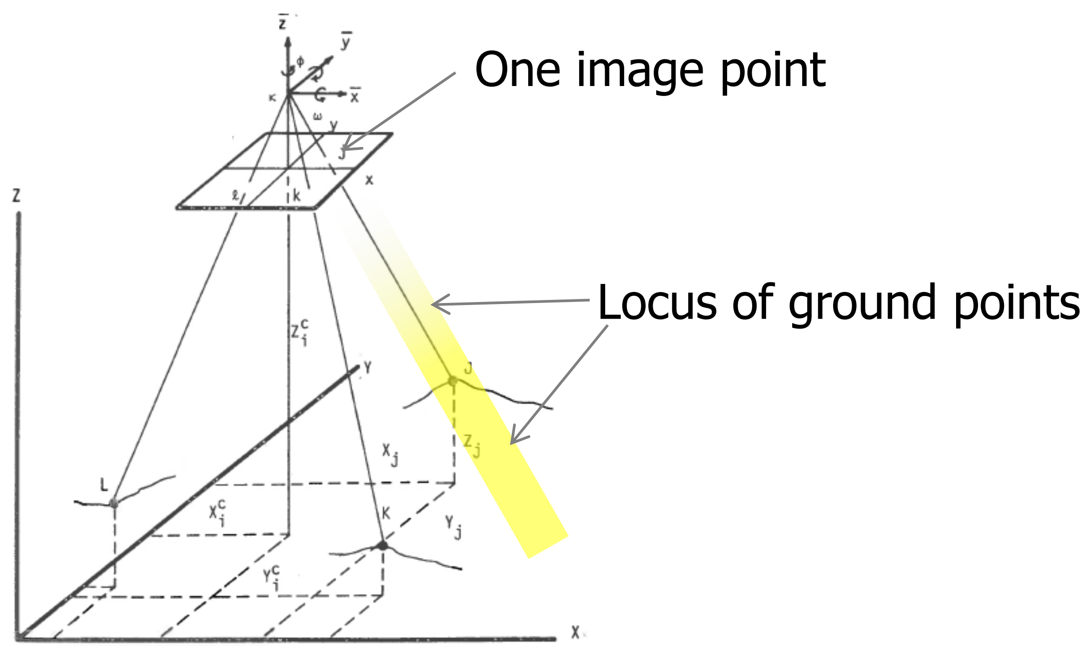
- A sensor model is a function/transformation/map: GroundToImage (g2i)
  - From 3-D ground space (i.e., latitude/longitude/height)
  - To 2-D image space (i.e., line/sample)
  - Multiple 3-D ground points for each 2-D image point
- A sensor model has an inverse: ImageToGround(i2g)
  - From 2-D image space
  - To 3-D ground space
  - Each 2-D image point corresponds to multiple 3-D points
    - (Not a function, mathematically speaking)

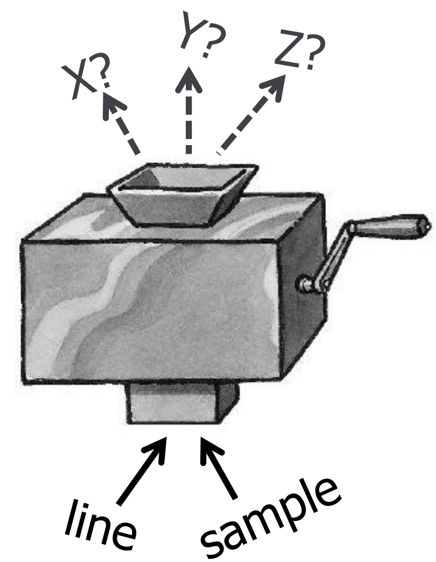


"Locus" = imaging ray

ImageToGround requires elevation as input

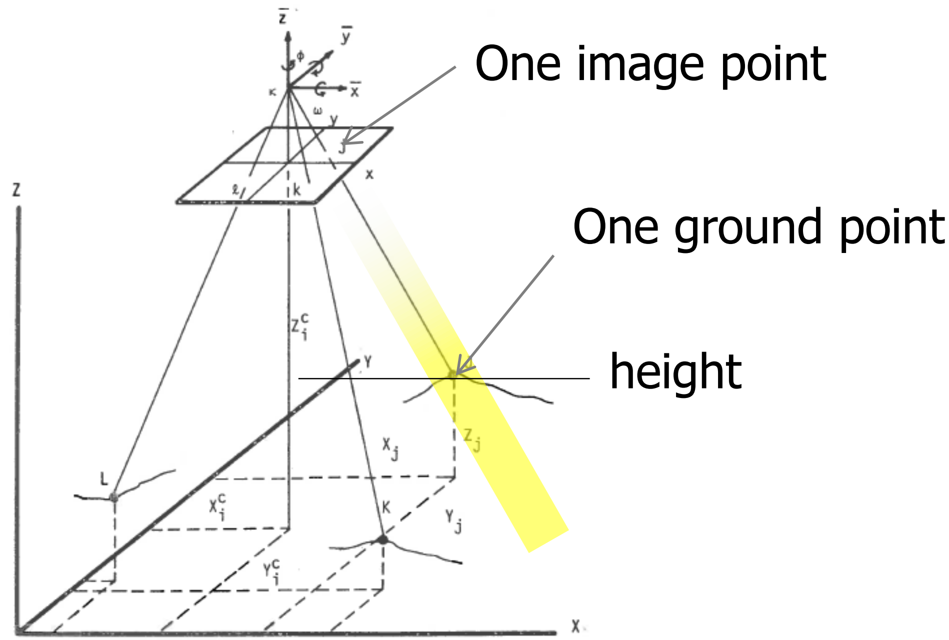
- Every 2-D image point corresponds to an entire locus of 3-D ground points
  - Thus, i2g(l,s) is not a well-determined question
  - To determine a specific ground X,Y, we need to also know the desired ground Z

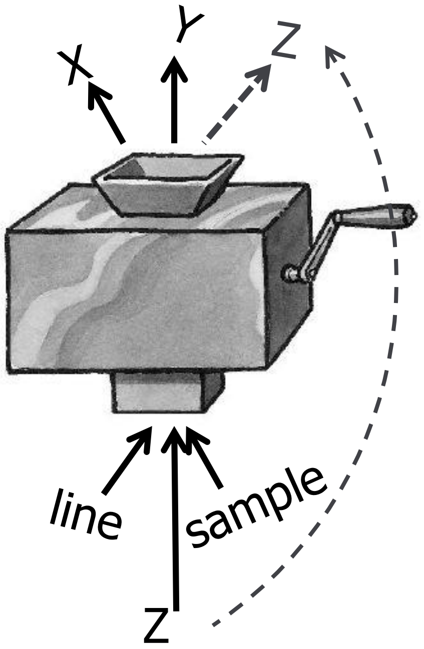




ImageToGround requires elevation as input …2

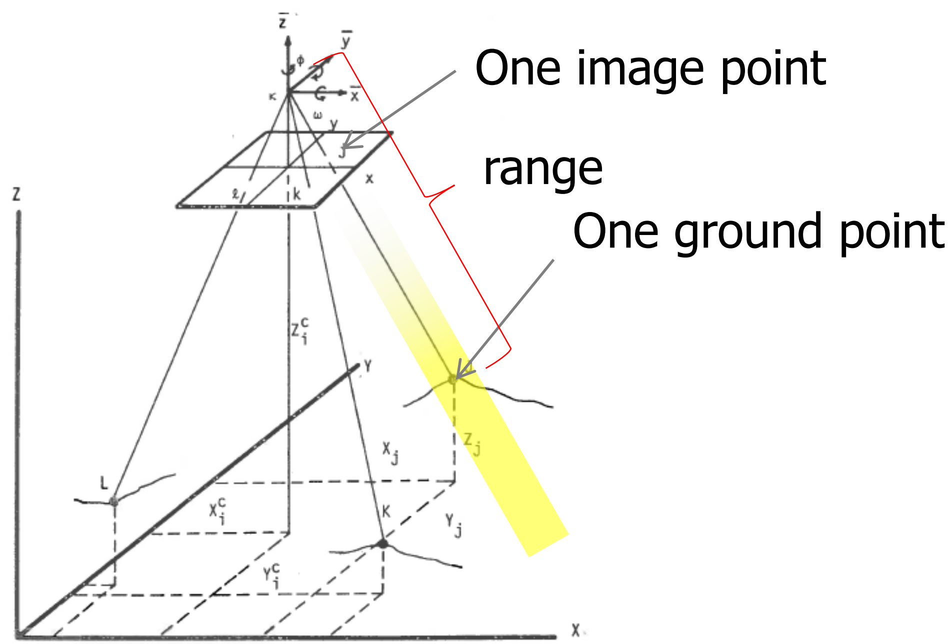
- Every 2-D image point corresponds to an entire locus of 3-D ground points
  - Thus, i2g(l,s) is not a well-determined question
  - To determine a specific ground X,Y, we need to also know the desired ground Z

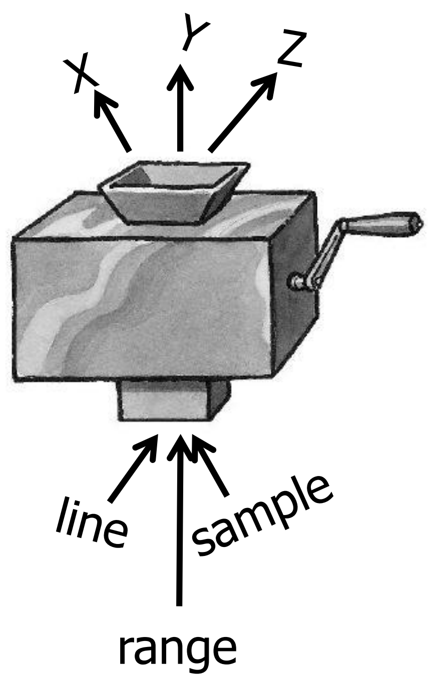




ImageToGround requires elevation as input …7

- Every 2-D image point corresponds to an entire locus of 3-D ground points
  - Thus, i2g(l,s) is not a well-determined question
  - To determine a specific ground X,Y, we need to also know the desired ground Z





Pinhole sensor model

- Easiest physical sensor model to visualize
  - Collinearity condition all about similar triangles

$$\frac{dX_g}{dx_i} = \frac{dY_g}{dy_i} = \frac{h}{f}$$

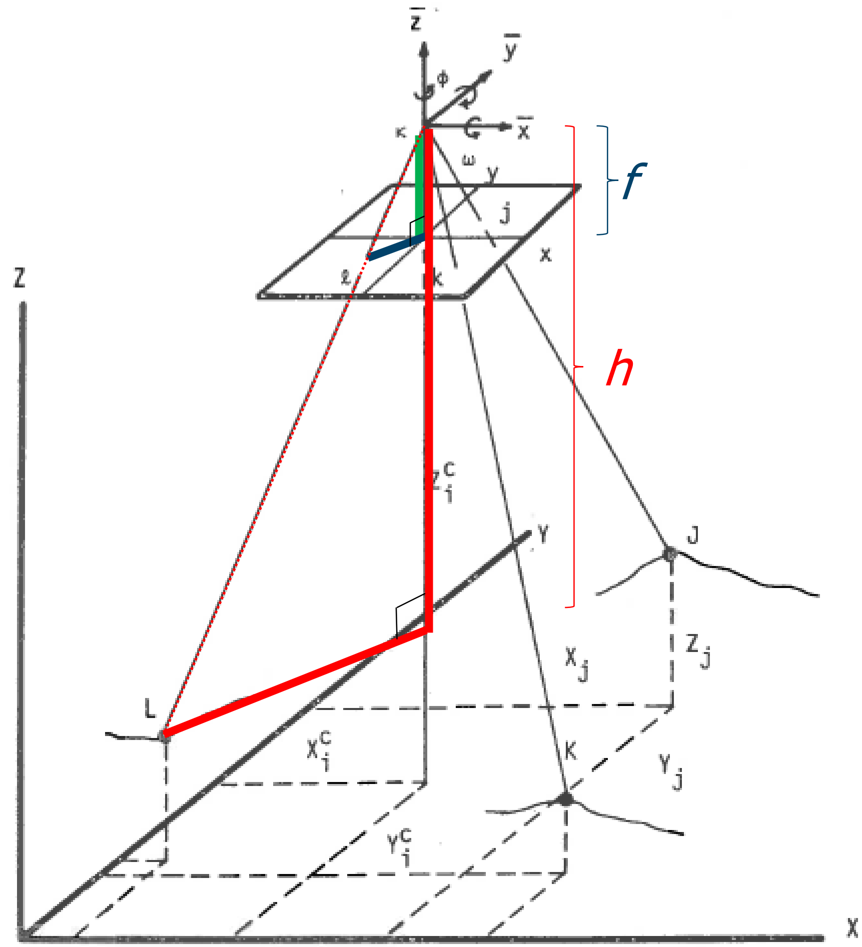
  - Equivalent to 3x4 camera matrix

$$\begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} X_g \\ Y_g \\ Z_g \\ 1 \end{bmatrix}$$

  - Equivalent to Direct Linear Transform

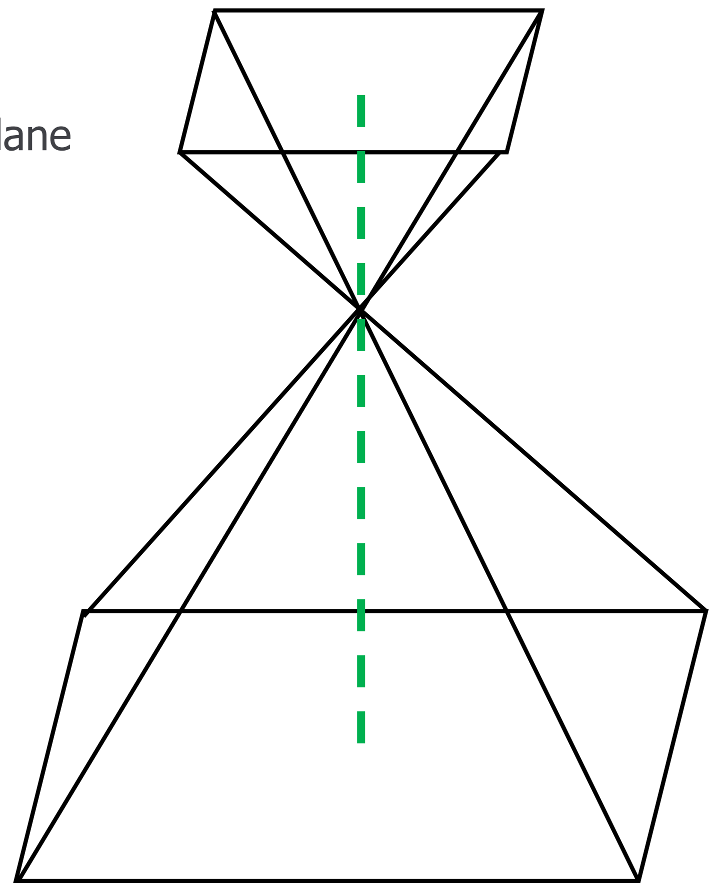
$$l = \frac{a0 + a1X + a2Y + a3Z}{1 + c1X + c2Y + c3Z}$$

$$s = \frac{b0 + b1X + b2Y + b3Z}{1 + c1X + c2Y + c3Z}$$



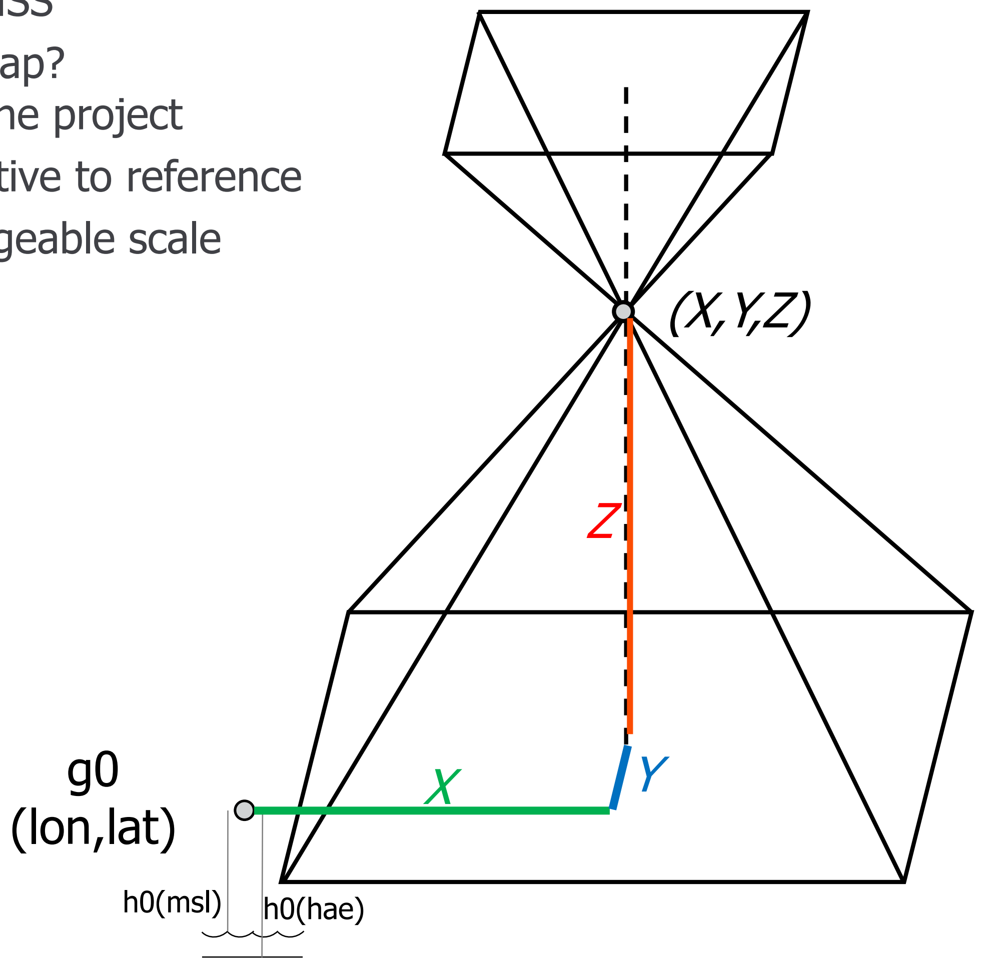
The pinhole frustum

- The behavior of a pinhole is represented by a *frustum*
  - Frustum = rectangular cone of image rays
  - Ground scene inverted through lens onto image plane
- The pinhole camera is specified by:
  - Exterior orientation (EO)
    - Position
    - Rotation
  - Interior orientation (IO)
    - Focal length
    - Principal point
    - Lens distortion



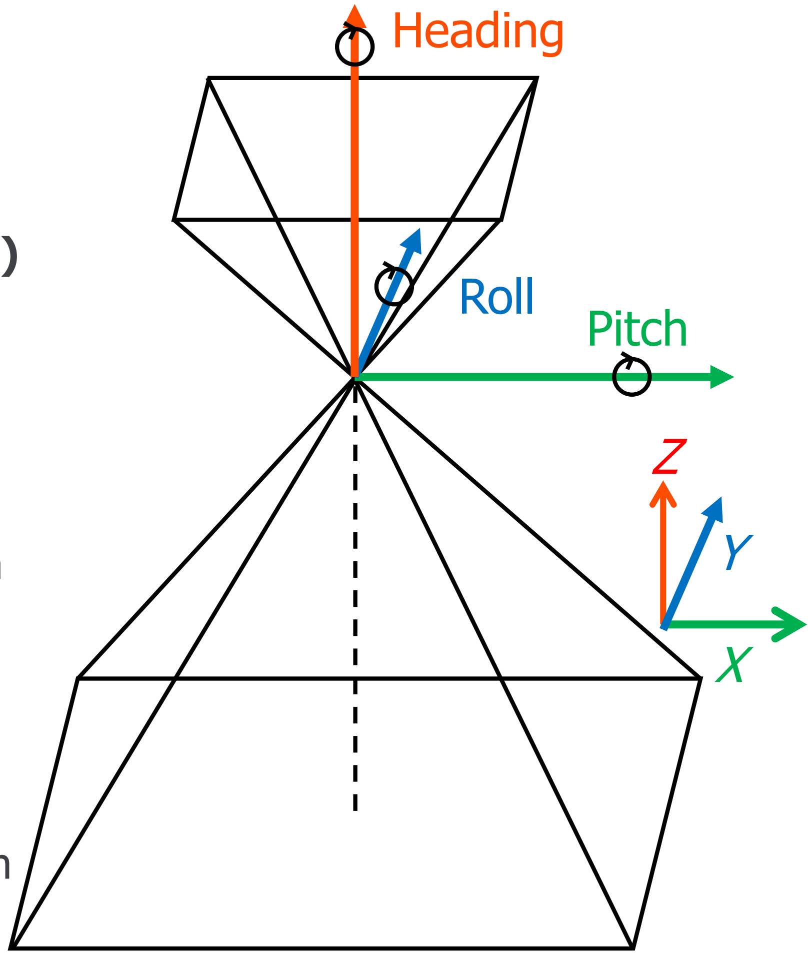
Camera position

- Estimate typically available from GNSS
  - May have to estimate from map?
- Choose reference ground-zero for the project
  - Every image gets a X,Y,Z relative to reference
  - Relative/local X,Y,Z are manageable scale



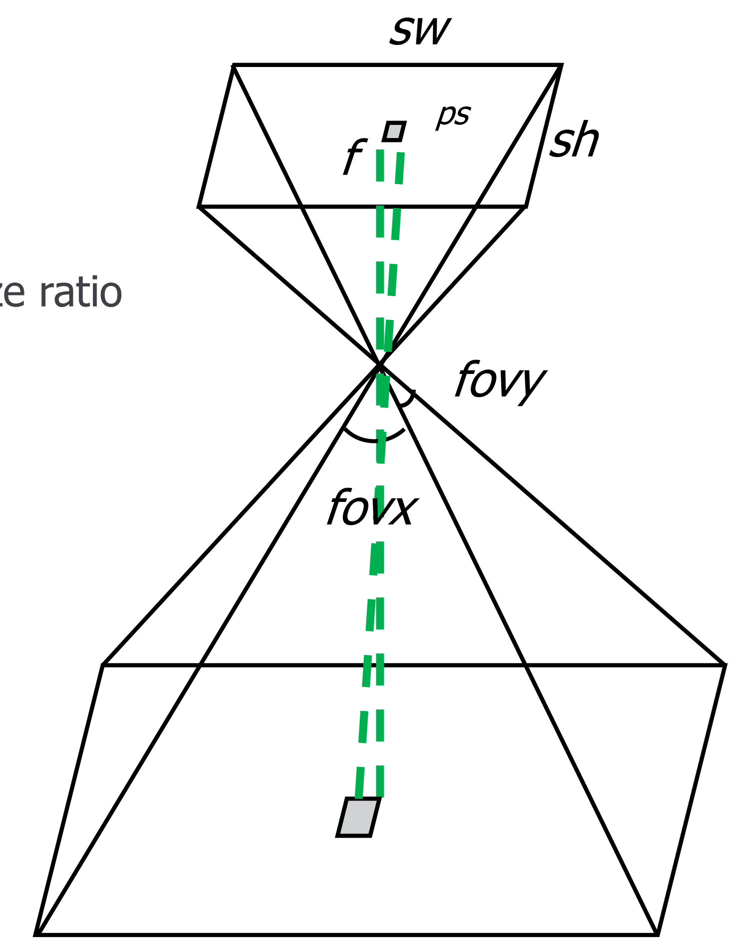
Camera orientation

- Three angles around three ordered axes
  - **ZXY:**
    - **Heading (clockwise from north)**
    - **Pitch (downward from horizontal)**
    - **Roll (clockwise from level)**
  - XYZ: $\omega\ \phi\ \kappa$
  - Axis/angle
    - Direction of 3D vector is rotation axis
    - Length of vector is amount of rotation
  - Quaternions
    - 4D unit vector, similar to axis/angle
    - Easy algebra for no-trig manipulation
- IMU often not available for initial estimate
  - Estimate heading/kappa from flight direction
  - Initially assume other two angles are zero



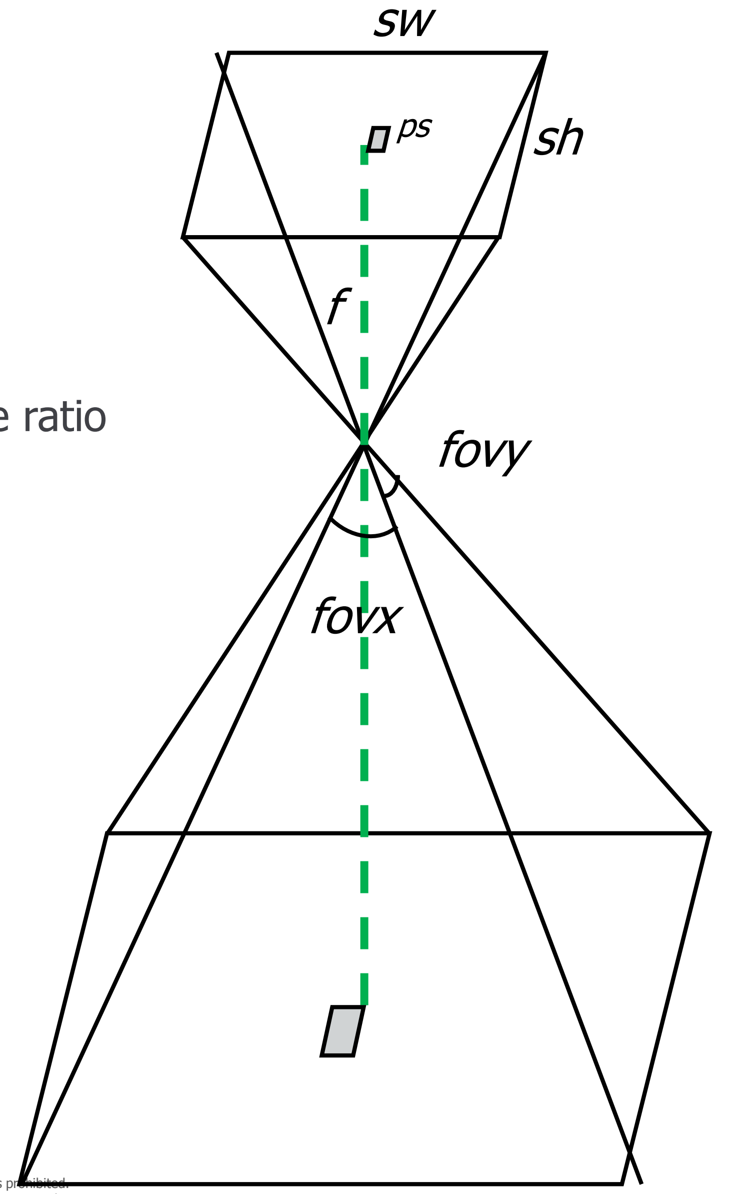
Focal length/field of view angles

- Frustum field of view is measured with angles
  - Horizontal, vertical, diagonal, per pixel
  - Wide field of view = “zoom out”
  - Narrow field of view = “zoom in”
- FOV width is determined by focal length/pixel size ratio
  - Focal length in mm alone says nothing
  - Also need sensor/pixel size



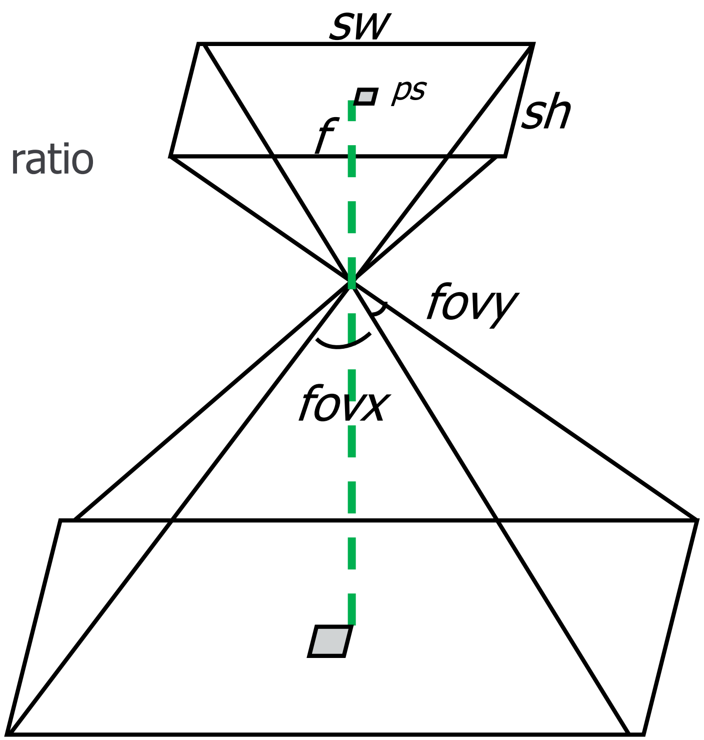
Focal length/field of view angles

- Frustum field of view is measured with angles
  - Horizontal, vertical, diagonal, per pixel
  - Wide field of view = "zoom out"
  - Narrow field of view = "zoom in"
- FOV width is determined by focal length/pixel size ratio
  - Focal length in mm alone says nothing
  - Also need sensor/pixel size
- Longer focal length/smaller pixels = narrower



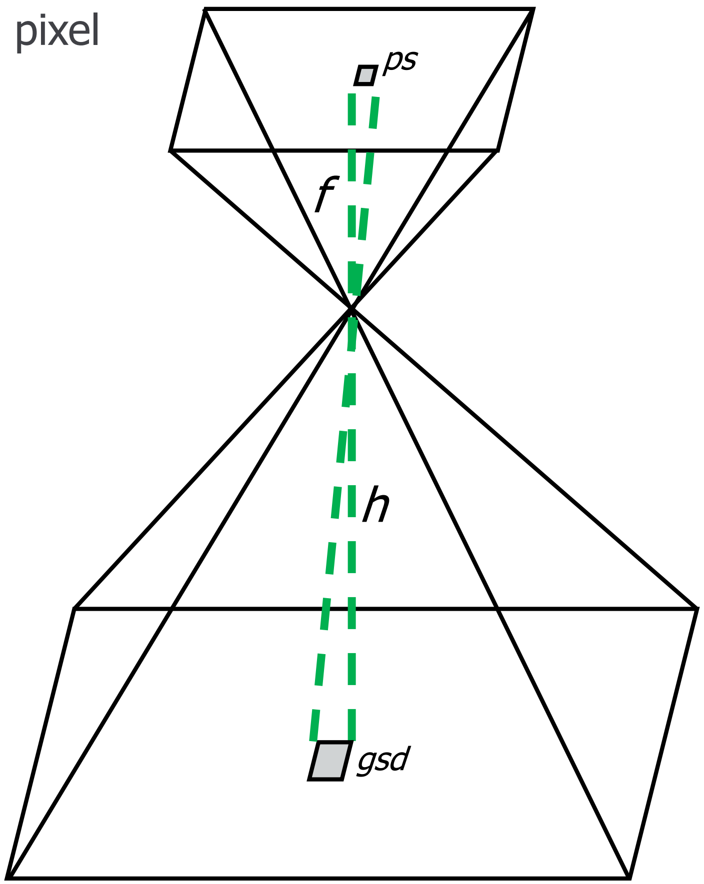
Focal length/field of view angles

- Frustum field of view is measured with angles
  - Horizontal, vertical, diagonal, per pixel
  - Wide field of view = "zoom out"
  - Narrow field of view = "zoom in"
- FOV width is determined by focal length/pixel size ratio
  - Focal length in mm alone says nothing
  - Also need sensor/pixel size
- Longer focal length/smaller pixels = narrower
- Shorter focal length/larger pixels = wider



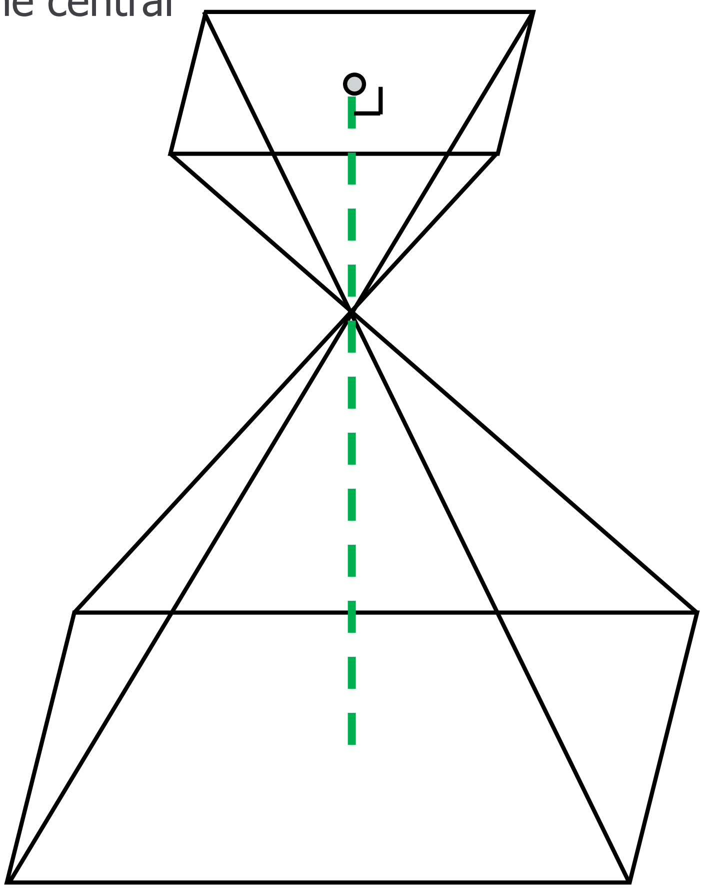
Resolution/ground sample distance

- Ground sample distance (GSD) is the size of one pixel projected on to the ground
- Common ratios (similar triangles):
  - f/ps = h/gsd
  - f/h = ps/gsd
  - gsd = h/f*ps
  - ps = f/h*gsd
- Varies with height=range
  - Ground terrain/features rise/fall
  - Camera heights vary
  - Oblique perspectives, near/far



Principal point

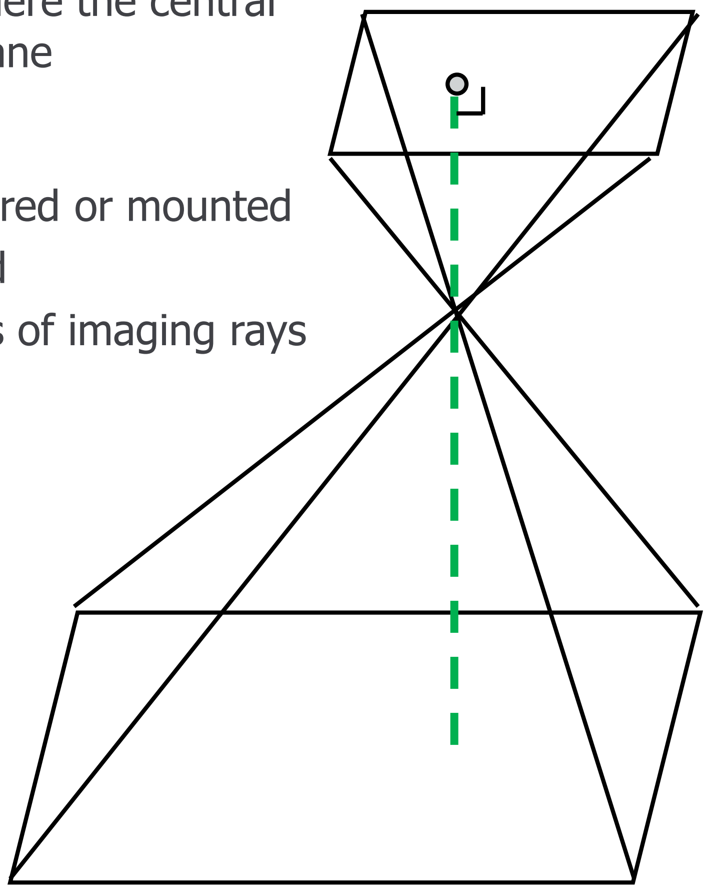
- The principal point (of autocollimation) is where the central (perpendicular) imaging ray hits the focal plane
- Ideally in the precise center of the image



Principal point

- The principal point (of autocollimation) is where the central (perpendicular) imaging ray hits the focal plane
- Ideally in the precise center of the image

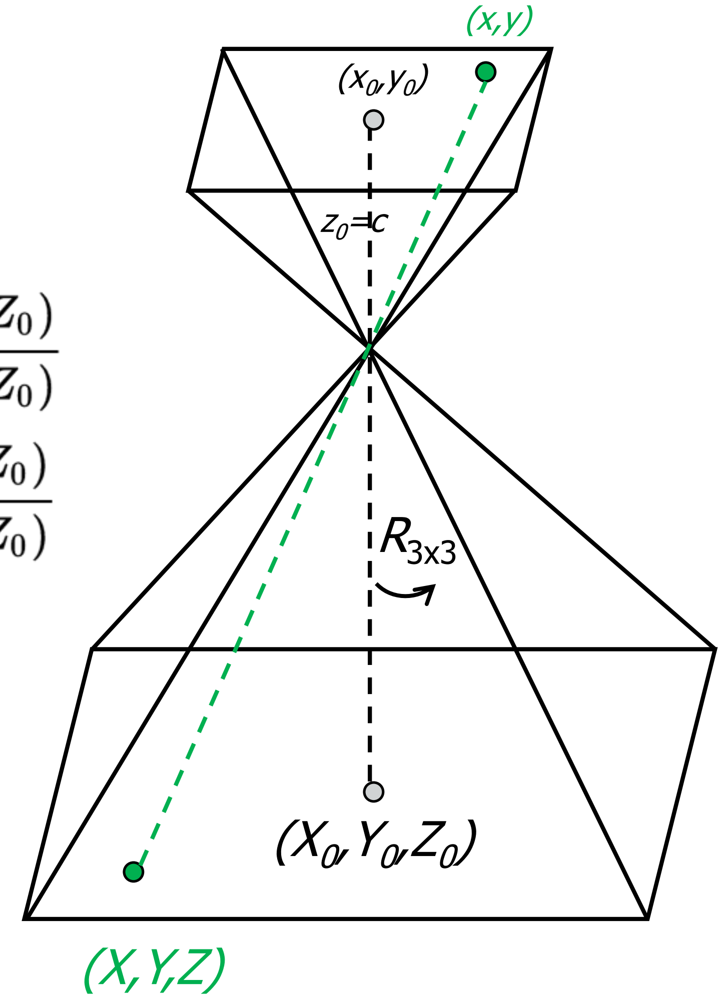
- But in reality, lens is not perfectly manufactured or mounted
  - Principal point is not perfectly centered
  - Offset causes subtle changes to angles of imaging rays



Collinearity equations

$$x - x_0 = -c\,\frac{R_{11}(X - X_0) + R_{21}(Y - Y_0) + R_{31}(Z - Z_0)}{R_{13}(X - X_0) + R_{23}(Y - Y_0) + R_{33}(Z - Z_0)}$$

$$y - y_0 = -c\,\frac{R_{12}(X - X_0) + R_{22}(Y - Y_0) + R_{32}(Z - Z_0)}{R_{13}(X - X_0) + R_{23}(Y - Y_0) + R_{33}(Z - Z_0)}$$



Equation images from:
https://en.wikipedia.org/wiki/Collinearity_equation

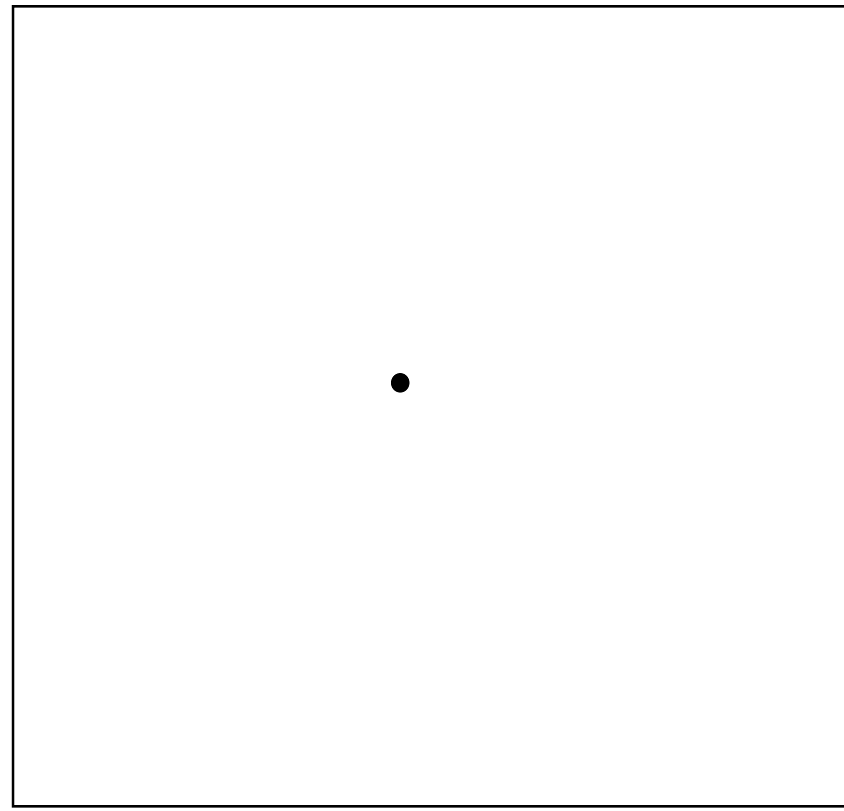
Lens distortion

before

after

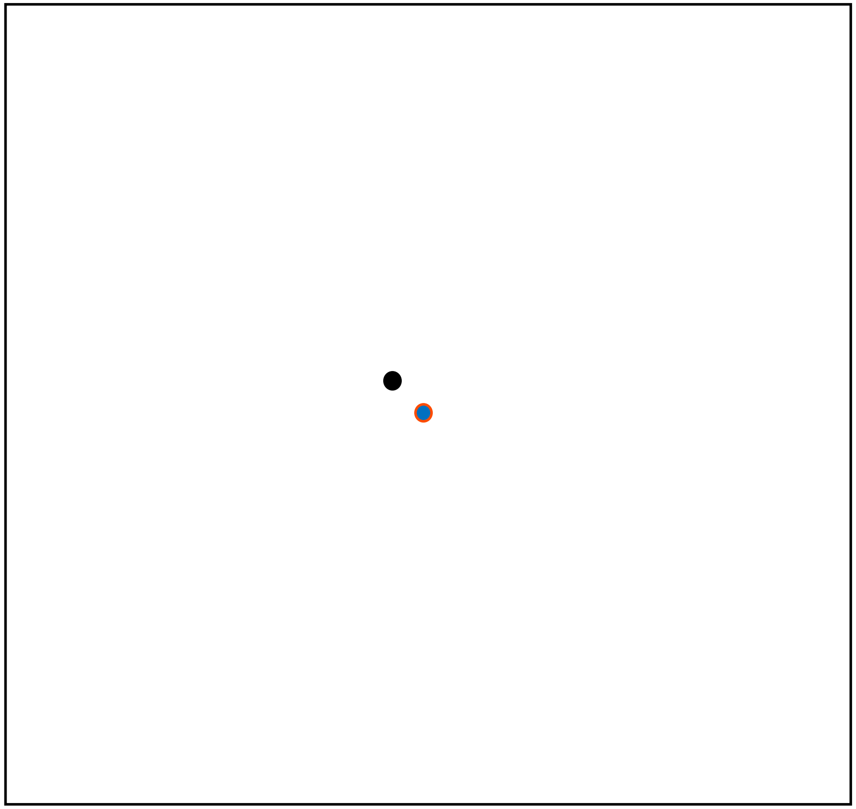
Radial lens distortion

- Center of symmetry



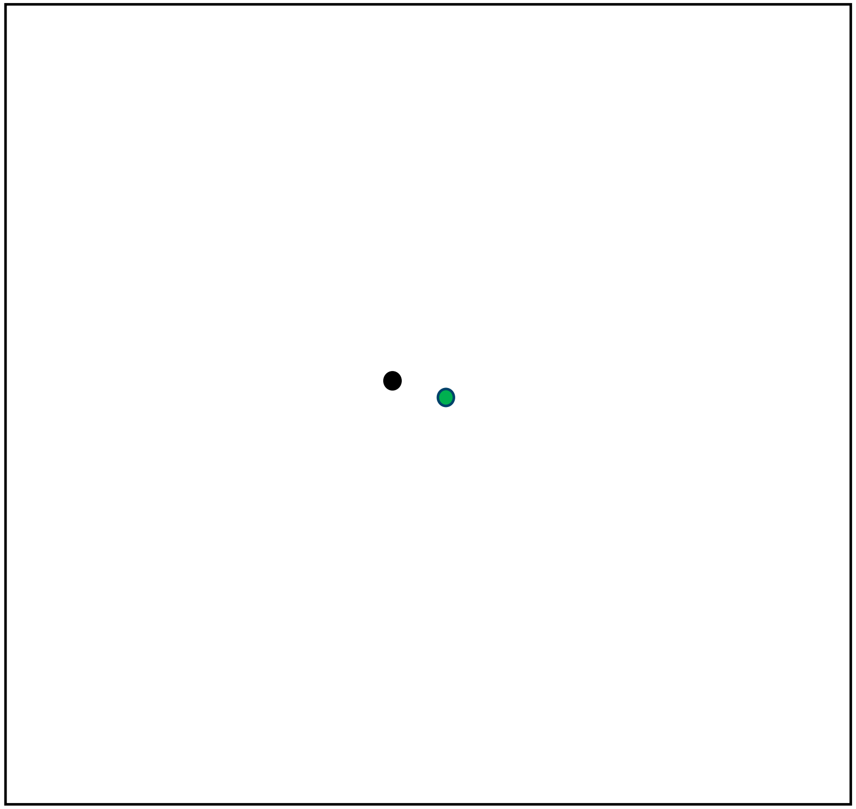
Radial lens distortion ...2

- Center of symmetry
  - Different than center of image



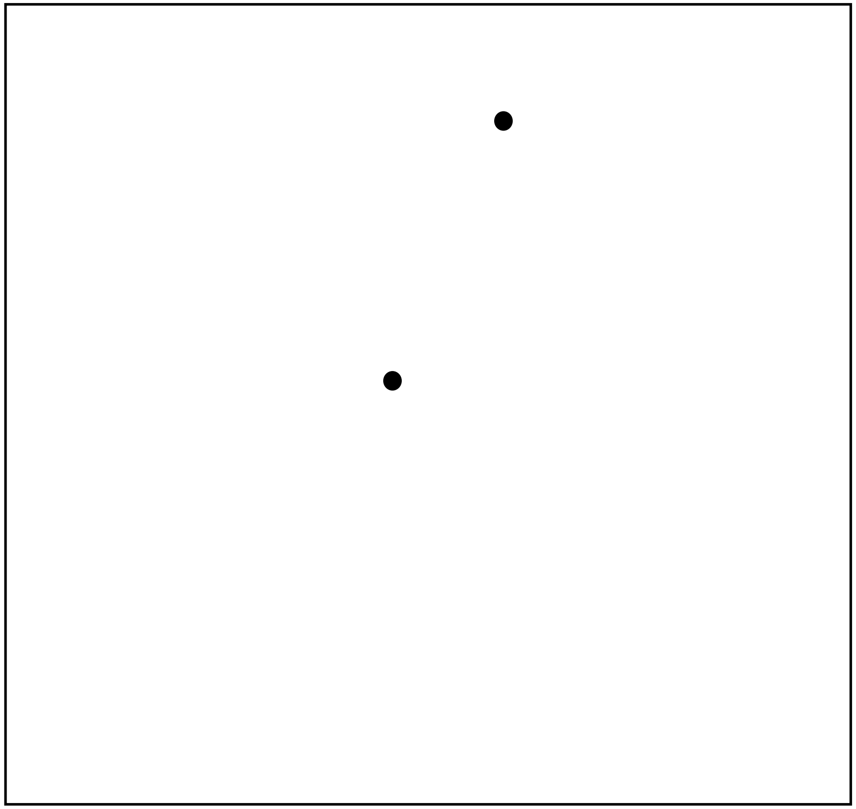
Radial lens distortion …3

- Center of symmetry
  - Different than center of image
  - Different than focal center



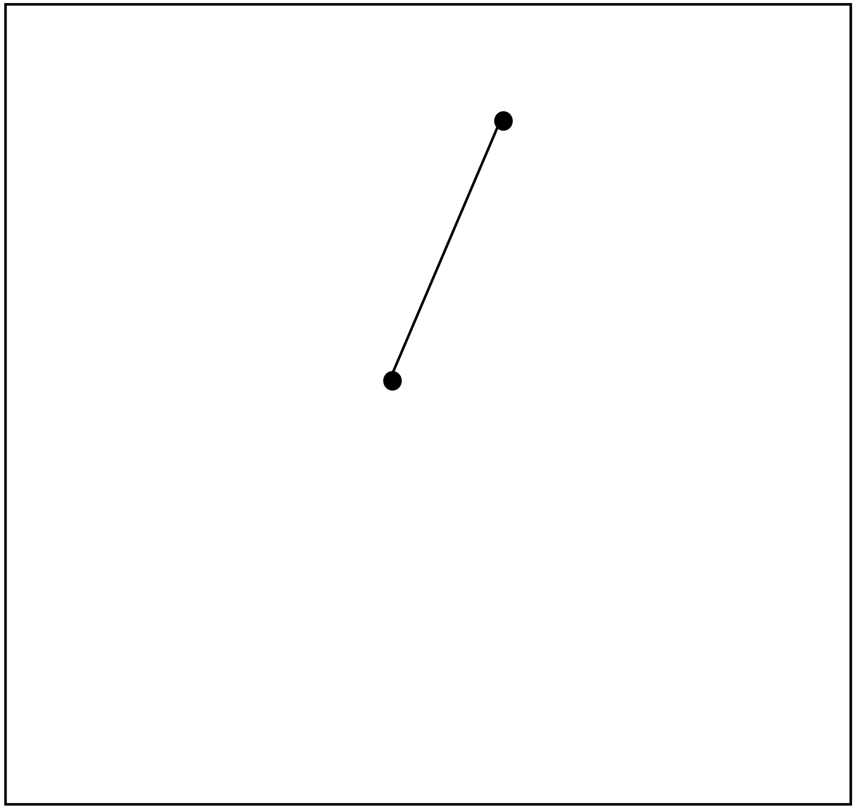
Radial lens distortion …4

- Center of symmetry
  - Different than center of image
  - Different than focal center
- Nominal image point
  - Result of pinhole g2i



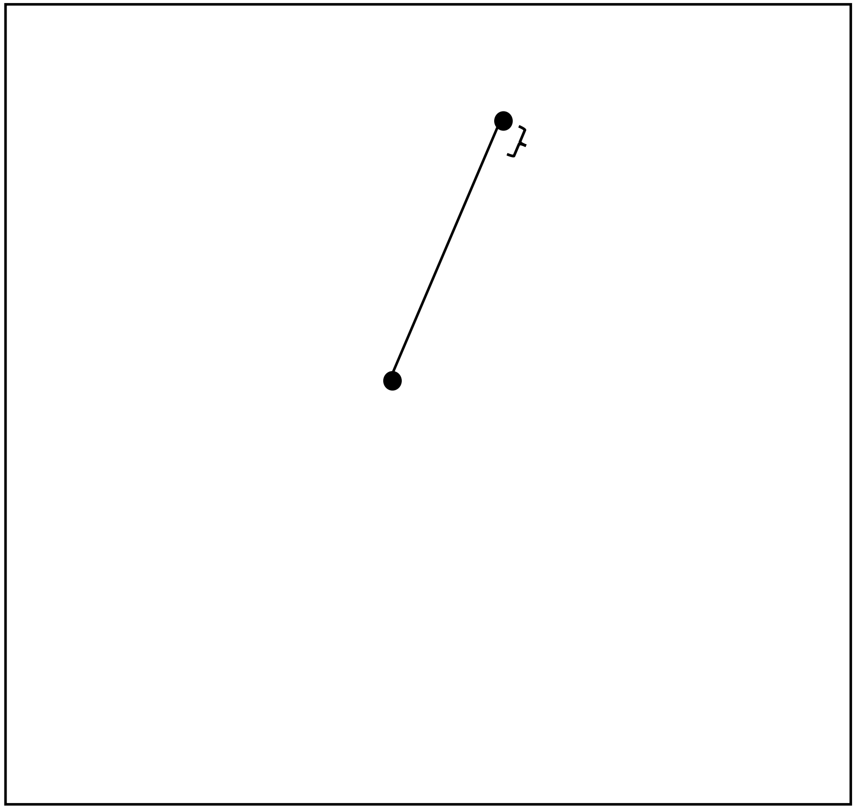
Radial lens distortion …5

- Center of symmetry
  - Different than center of image
  - Different than focal center
- Nominal image point
  - Result of pinhole g2i
- Compute length of radius



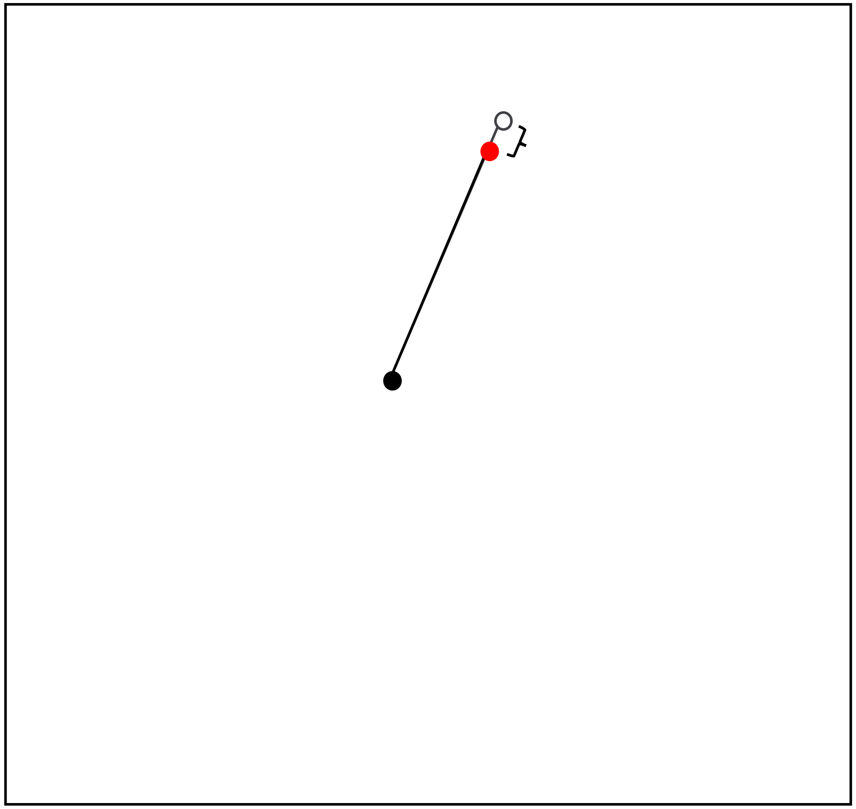
Radial lens distortion …6

- Center of symmetry
  - Different than center of image
  - Different than focal center
- Nominal image point
  - Result of pinhole g2i
- Compute length of radius
- Compute correction to radius



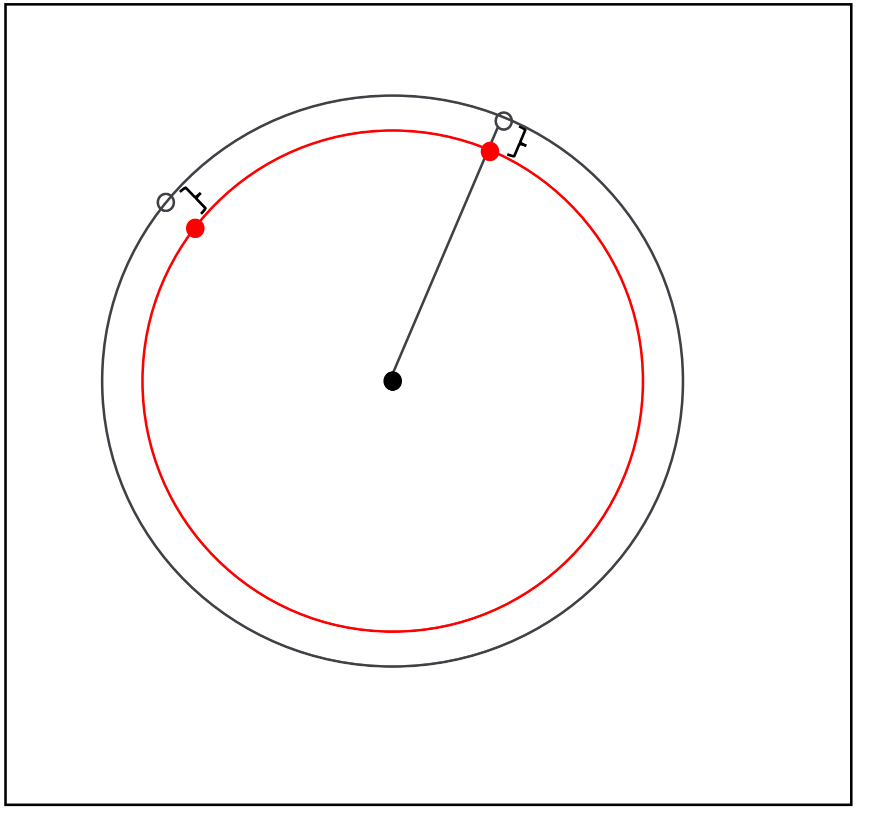
Radial lens distortion ...7

- Center of symmetry
  - Different than center of image
  - Different than focal center
- Nominal image point
  - Result of pinhole g2i
- Compute length of radius
- Compute correction to radius
- Move image point to corrected radius



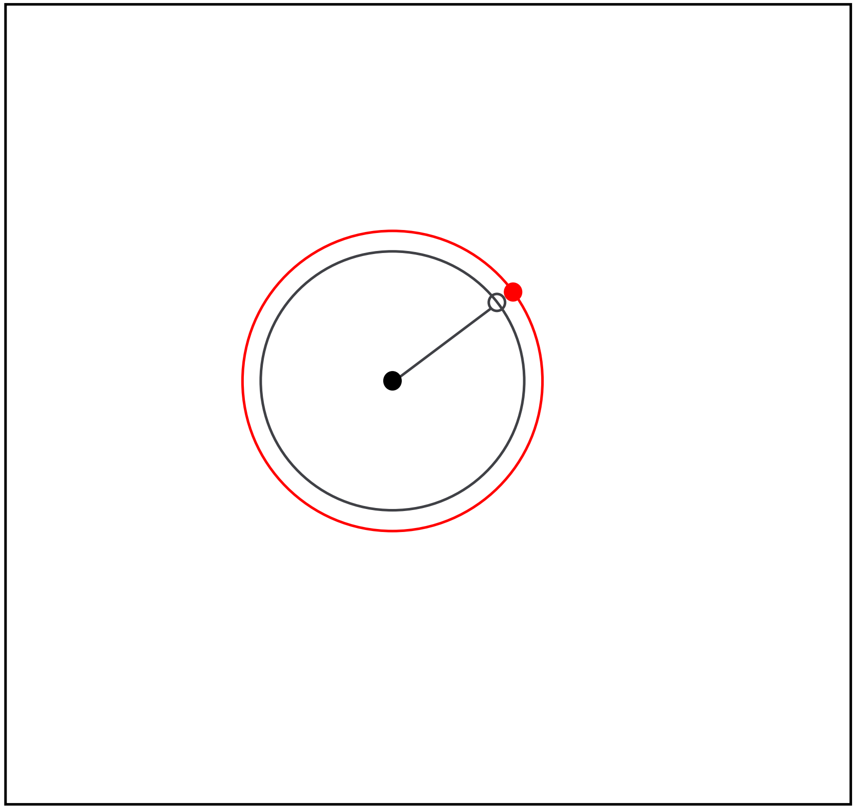
Radial lens distortion …8

- Center of symmetry
    - Different than center of image
    - Different than focal center
- Nominal image point
    - Result of pinhole g2i
- Compute length of radius
- Compute correction to radius
- Move image point to corrected radius
- Same-radius points, same correction
    - Same-radius = circle



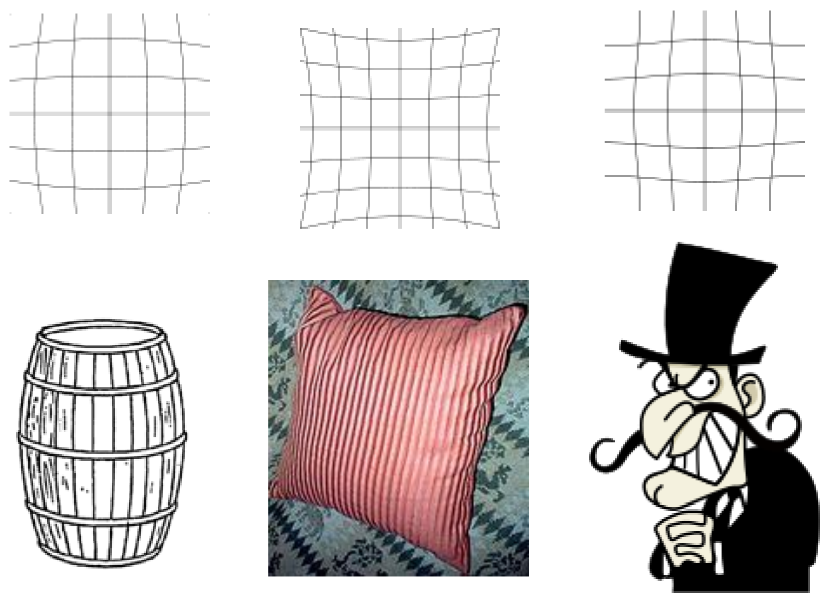
Radial lens distortion …9

- Center of symmetry
  - Different than center of image
  - Different than focal center
- Nominal image point
  - Result of pinhole g2i
- Compute length of radius
- Compute correction to radius
- Move image point to corrected radius
- Same-radius points, same correction
  - Same-radius = circle
- Different radii are corrected differently



Different types of radial distortion

- Barrel: everything pushed out from the center
- Pincushion: everything pulled in towards center
- Mustache: barrel transitioning into pincushion
- Actual radial distortion is a continuously changing amount of radial distortion
  - Usually well modeled with only 2 coefficients (*k1, k2*)
  - Can be refined with optional higher order terms (*k3*, ...)



Images from:
http://en.wikipedia.org/wiki/Distortion_(optics)

BAE SYSTEMS
INSPIRED WORK

Tangential lens distortion

- Tangential lens distortion is the result of the lens/sensor not being parallel
- Well modeled with 2 coefficients (*p1, p2*)

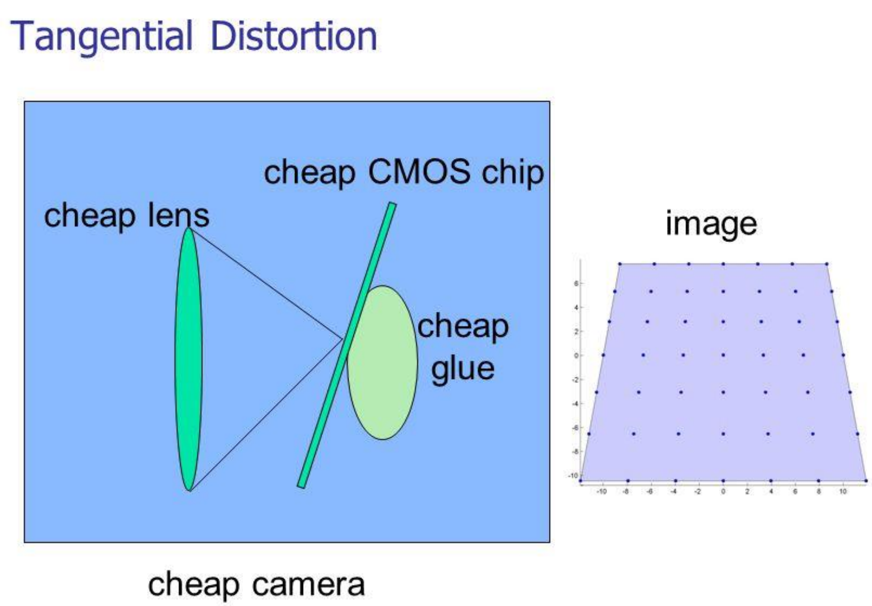


Image from Robert Pless, WUSTL: http://slideplayer.com/slide/3249143/

Projection model with lens distortion

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$$

$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$

$$\text{where} \quad r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

Equation from OpenCV documentation:
http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Projection model with lens distortion

rotation

translation

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

perspective divide

$$x' = x/z$$
$$y' = y/z$$

tangential correction

radial correction

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$$
$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$
$$\text{where} \quad r^2 = x'^2 + y'^2$$

radius of distortion

$$u = f_x * x'' + c_x$$
$$v = f_y * y'' + c_y$$

principal point

focal length

Equation from OpenCV documentation:
http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

Brown-Conrady model for lens distortion

$$x_\mathrm{d} = x_\mathrm{u}(1 + K_1 r^2 + K_2 r^4 + \cdots) + (P_2(r^2 + 2x_\mathrm{u}^2) + 2P_1 x_\mathrm{u} y_\mathrm{u})(1 + P_3 r^2 + P_4 r^4 + \cdots)$$

$$y_\mathrm{d} = y_\mathrm{u}(1 + K_1 r^2 + K_2 r^4 + \cdots) + (P_1(r^2 + 2y_\mathrm{u}^2) + 2P_2 x_\mathrm{u} y_\mathrm{u})(1 + P_3 r^2 + P_4 r^4 + \cdots)$$

where:

$(x_\mathrm{d},\ y_\mathrm{d})$ = distorted image point as projected on image plane using specified lens,

$(x_\mathrm{u},\ y_\mathrm{u})$ = undistorted image point as projected by an ideal pin-hole camera,

$(x_\mathrm{c},\ y_\mathrm{c})$ = distortion center (assumed to be the principal point),

K_n = n^th radial distortion coefficient,

P_n = n^th tangential distortion coefficient [note that Brown's original definition has P_1 and P_2 interchanged],

r = $\sqrt{(x_\mathrm{u} - x_\mathrm{c})^2 + (y_\mathrm{u} - y_\mathrm{c})^2}$, and

$\ldots$ = an infinite series.

Decentering distortion = a form of tangential distortion caused by centers of lens elements not lying on a straight line

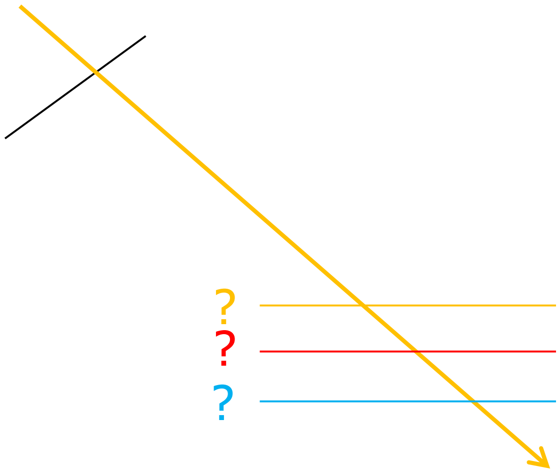
https://en.wikipedia.org/wiki/Distortion_(optics)

What's the point?

- What's the point of all this precise modeling of camera behavior?
- With an accurate understanding of EO/IO, it is possible to accurately ray trace between image and ground
  - From any ground point to a precise (sub)pixel
  - From any (sub)pixel to an accurate image **ray** in ground space
- Rays from images with different perspectives can be intersected
  - Yielding an accurate **point** in ground space

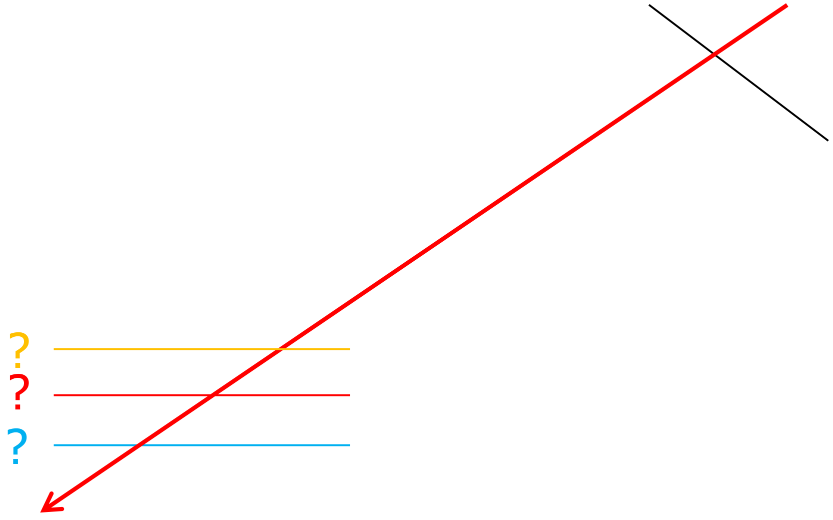
Ray intersection

- A feature located in an image can be anywhere along the locus



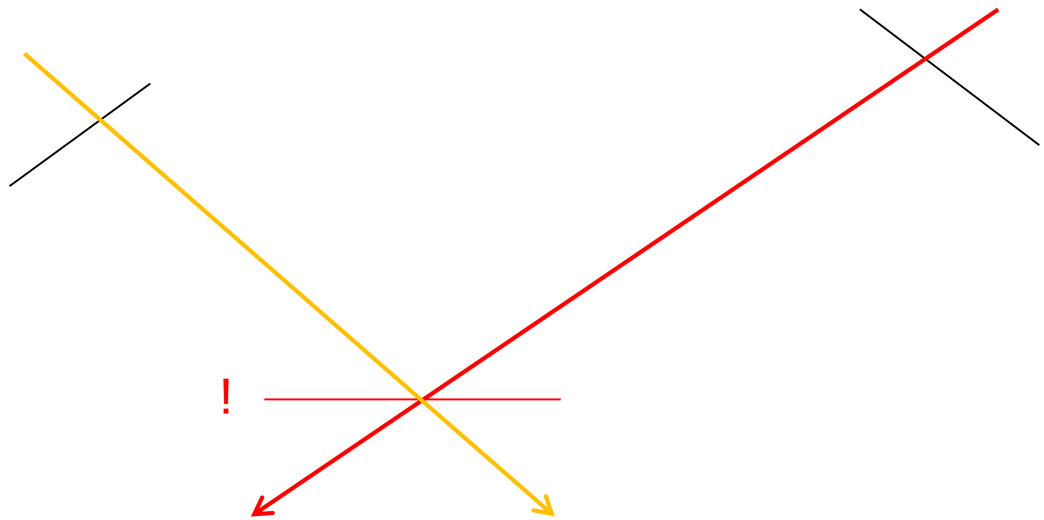
Ray intersection

- A feature located in an image can be anywhere along the locus
- Same is true for any other image



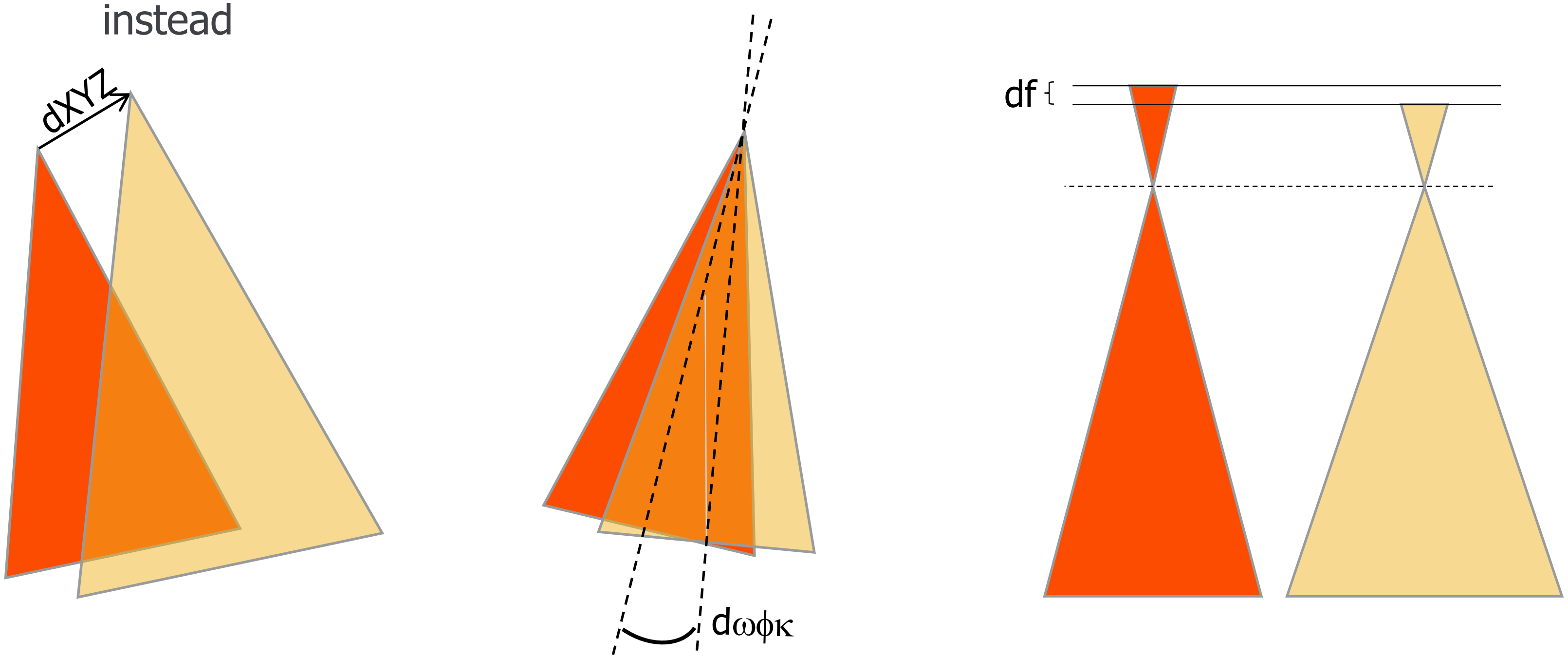
Ray intersection

- A feature located in an image can be anywhere along the locus
- Same is true for any other image
- But two (or more) image rays can be intersected in ground space
  - Requires rays to (nearly) intersect
  - Requires very accurate understanding of EO/IO

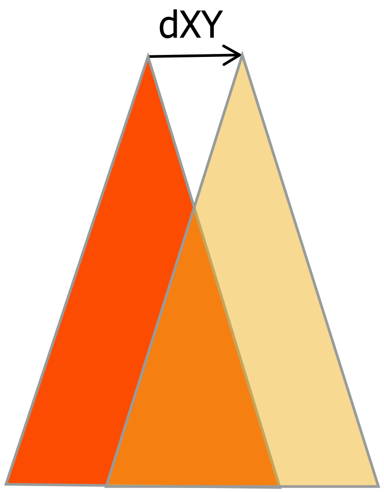


Adjustable parameters

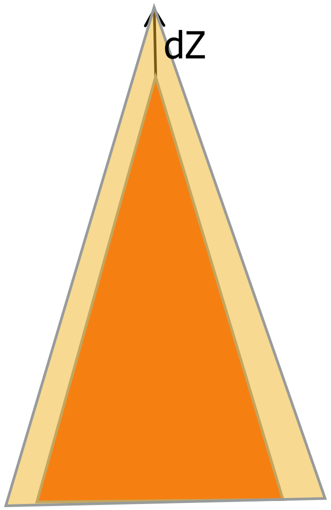
- g2i/i2g computations need metadata
  - Where was the sensor? Where was it pointed? How many pixels are in the Charge Coupled Device (CCD)? What size? How is the CCD situated relative to the lens?
- Adjustable parameters give the sensor model user the ability to make small corrections without reconstructing/reinitializing the whole sensor model
  - Anywhere in the implementation of g2i/i2g a parameter X is needed, use X+dx instead



Impacts of metadata inaccuracy: position

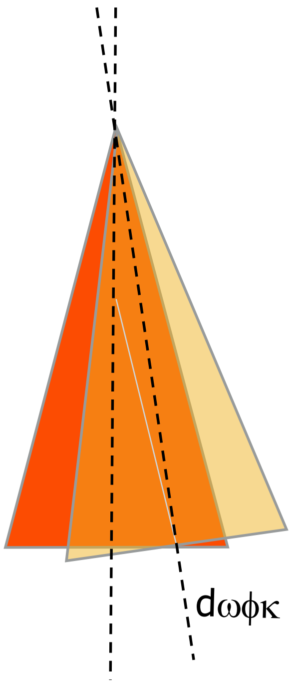


- Position X, Y
  - Ground error is the same X,Y
  - For stereo
    - Correlated (coincident) error ➔ same error
    - Anticorrelated (opposite) error ➔ no X,Y error, but Z error 'scissors' up or down
- Position Z
  - Ground error is scale
  - For stereo
    - Correlated➔same error
    - Anticorrelated➔X,Y and Z error 'scissors' all around



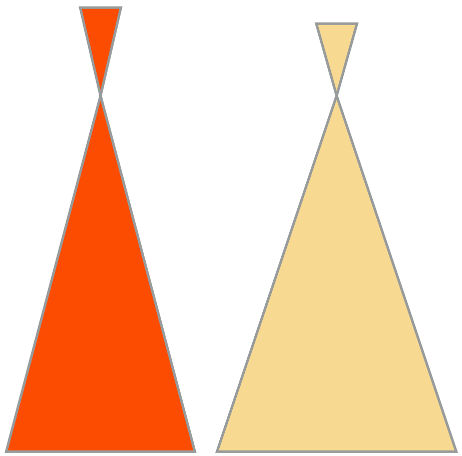
Impacts of metadata inaccuracy: orientation

- Pointing angles heading, pitch/omega, roll/phi
  - Ground error scales with height/range
  - For small angles (few degrees) it scales with angle
  - 1° @ 100m/ft →1.745m/ft
    - 1° @ 200m/ft→3.491m/ft
    - 2° @ 100m/ft→3.492m/ft
  - Rule of thumb? 1° @ 400ft → 7ft
- Rotation angle (roll, kappa)
  - Ground error is 0 at center, larger away from center
- Stereo: like position, scissoring makes bigger errors

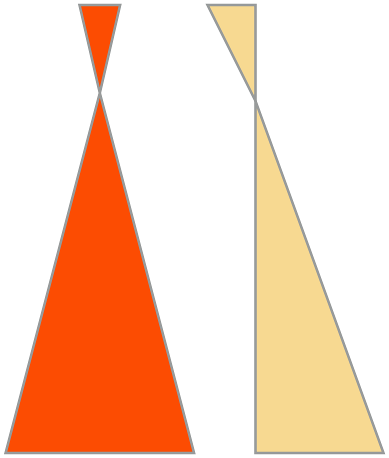


Impacts of metadata inaccuracy: IO

- Focal length
  - Error causes narrowing/widening of frustum→ground scale
  - Very similar to/correlated with Z



- Principal Point
  - Error causes X,Y shift, and subtle change in perspective



Uncertainty

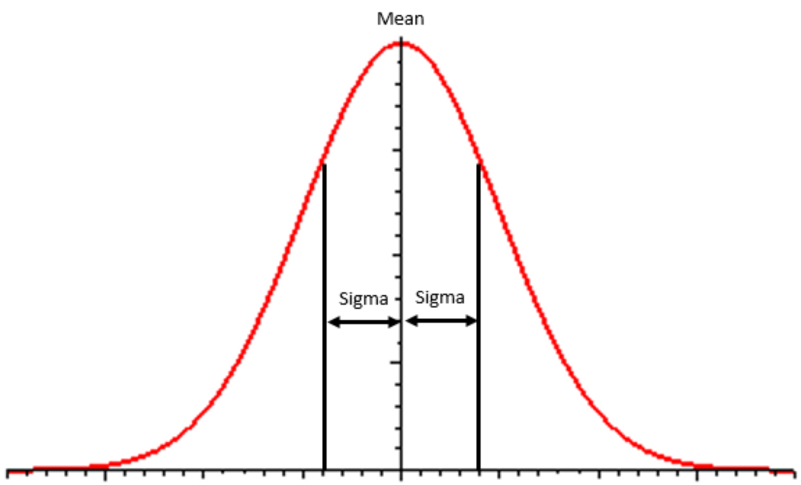
- Adjustable = correctable = may need to be corrected = **may be wrong**
- How wrong is it?

Uncertainty …2

- Adjustable = correctable = may need to be corrected = **may be wrong**
- How wrong ~~is it~~?

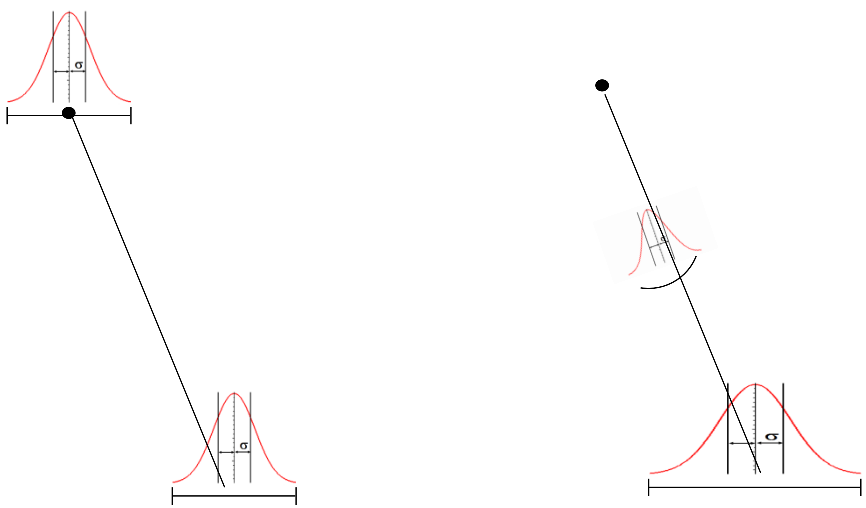
Uncertainty …3

- Adjustable = correctable = may need to be corrected = **may be wrong**
- How wrong could it be?
  - Ideally, a sensor model with adjustable parameters has an idea how wrong each adjustable parameter may be
  - Usually expressed as σ=standard deviation or σ^2=variance

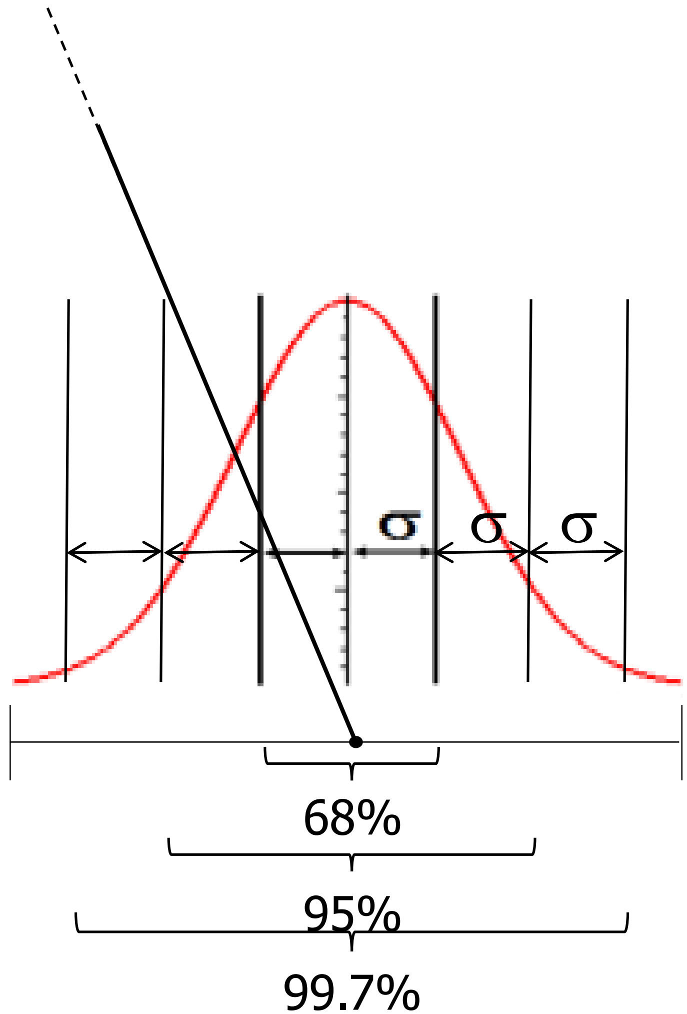


Uncertainty …4

- Adjustable = correctable = may need to be corrected = **may be wrong**
- How wrong could it be?
  - Ideally, a sensor model with adjustable parameters has an idea how wrong each adjustable parameter may be
  - Usually expressed as σ=standard deviation or σ^2=variance
- Error propagation: Uncertainty in sensor model parameters translates to uncertainty on the ground



Uncertainty: the 68/95/99.7 rule



Sensor calibration

Sensor calibration

- Sensor calibration is the determination of the fixed parameters of a camera
  - IO: focal length, principal point, lens distortion
  - Can be determined once and reused for all collections
  - Block adjustment can focus on only EO (position, rotation)
- Multiple methods of calibration:
  - Traditionally, calibration is measured in the factory/lab (now "type calibration")
  - Calibration can be solved for using pictures of a calibration rig/surface
  - Self-calibration (*in situ*)
    - Solved along with EO in bundle adjustment of operational imagery
    - Can be from scratch (*de novo*), or tweak of previously solved values
    - Slight changes to previous calibration are possible
      - Non-fixed focus (zoom)
      - Temperature
      - Vibration

Sample calibration report for aerial film camera

USGS Report No. OSL/3324

United States Department of the Interior

U.S. GEOLOGICAL SURVEY
Reston, Virginia 20192

REPORT OF CALIBRATION
of Aerial Mapping Camera

May 29, 2007

Camera type: Wild RC30*
Lens type: Wild Universal Aviogon /4-S
Nominal focal Length: 153 mm

Camera serial no.: 5324
Lens serial no.: 13365
Maximum aperture: f/4
Test aperture: f/4

Submitted by: Richard Crouse & Associates, Inc.
Frederick, Maryland

Reference: Richard Crouse & Associates, Inc. purchase order No. 07-21, dated May 29, 2007.

These measurements were made on Agfa glass plates, 0.19 inch thick, with spectroscopic emulsion type APX Panchromatic, developed in D-19 at 68° F for 3 minutes with continuous agitation. These photographic plates were exposed on a multicollimator camera calibrator using a white light source rated at approximately 5200K.

I. Calibrated Focal Length: 153.252 mm

II. Lens Distortion

Field angle:	7.5°	15°	22.7°	30°	35°	40°
Symmetric radial (µm)	-2	-3	-4	-4	-2	4
Decentering tangential (µm)	0	0	1	1	2	3

Symmetric radial distortion

K_0 = 0.9327E-04
K_1 = -0.5342E-08
K_2 = -0.1349E-12
K_3 = 0.0000
K_4 = 0.0000

Decentering distortion

P_1 = 0.1755E-06
P_2 = -0.5308E-07
P_3 = 0.0000
P_4 = 0.0000

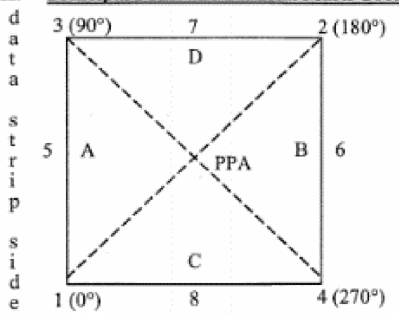
Calibrated principal point

x_p = -0.003 mm
y_p = 0.004 mm

Sample calibration report for aerial film camera ...2

USGS Report No. OSL/3324

VII. Principal Point and Fiducial Mark Coordinates



Positions of all points are referenced to the principal point of autocollimation (PPA) as origin. The diagram indicates the orientation of the reference points when the camera is viewed from the back, or a contact positive with the emulsion up. The data strip is to the left.

	X coordinate (mm)	Y coordinate (mm)
Indicated principal point, corner fiducials	.018	.007
Indicated principal point, midside fiducials	.014	.008
Principal point of autocollimation (PPA)	.000	.000
Calibrated principal point (point of symmetry)	-.003	.004
Fiducial Marks		
1	-105.979	-105.992
2	106.012	106.005
3	-105.989	106.007
4	106.025	-105.992
5	-111.987	.010
6	112.021	.006
7	.011	112.002
8	.017	-111.993

VIII. Distances Between Fiducial marks

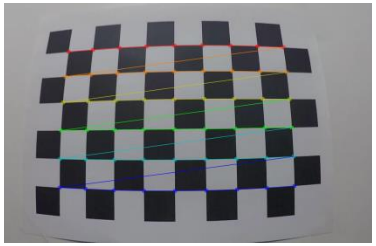
Corner fiducials (diagonals)	1-2:	299.805 mm	3-4:	299.823 mm
Lines joining these markers intersect at an angle o 90° 00' 04"				
Midside fiducials	5-6:	224.008 mm	7-8:	223.995 mm
Lines joining these markers intersect at an angle o 90° 00' 09"				
Corner fiducials (perimeter)	1-3:	211.999 mm	2-3:	212.002 mm
	1-4:	212.004 mm	2-4:	211.998 mm

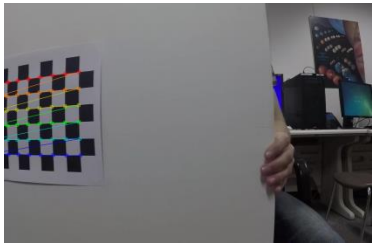
The Method of measuring these distances is considered accurate within 0.003 mm

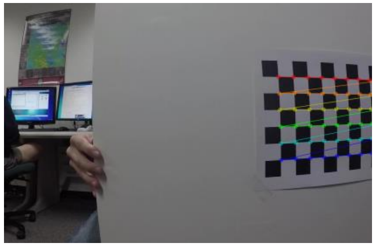
Note: For GPS applications, the nominal entrance pupil distance from the focal plane is 277 mm.

Chessboard calibration

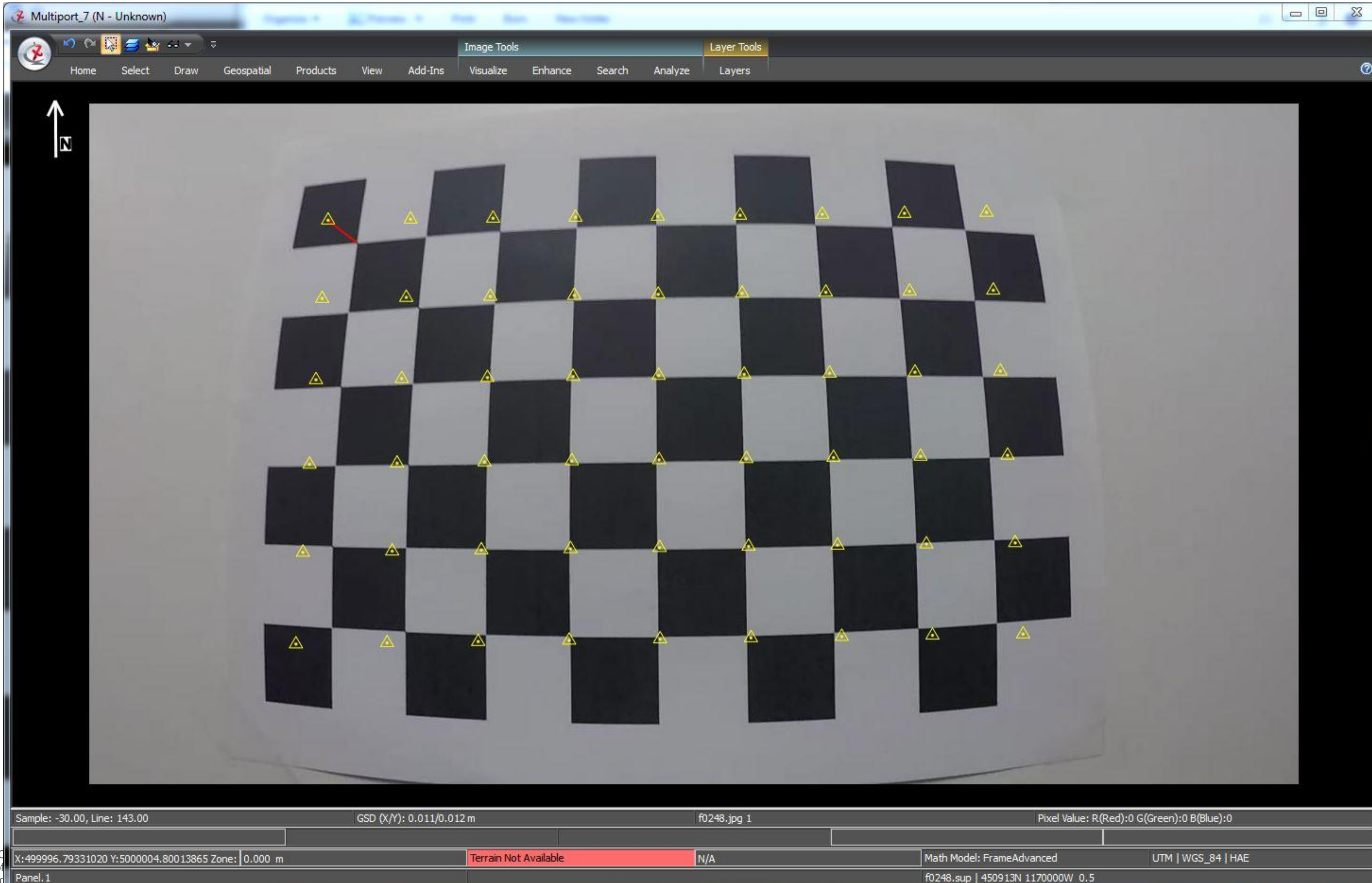
- GoPro HERO3+ Black Edition (U. Alaska, Fairbanks)
  - Paper taped onto posterboard
  - Frames from 1280x720 video
- OpenCV finds internal corners
  - 9x6=54 corners
  - 8 images→432 image points
- OpenCV solves for nominal camera poses
  - ~40pix lens distortion at 1280x720
  - >100pix distortion at full-res 4000x3000
- Output as SOCET GXP triangulation project, solve
  - 9x6=54 XYZ control points, Z=0, regular X/Y spacing
  - Refine EO, solve for focal, principal point, distortion
  - Image RMS: 0.65 pixels
  - Ground RMS: X/Y~0.05mm, Z~0.5mm
    - Note paper not taped on flat



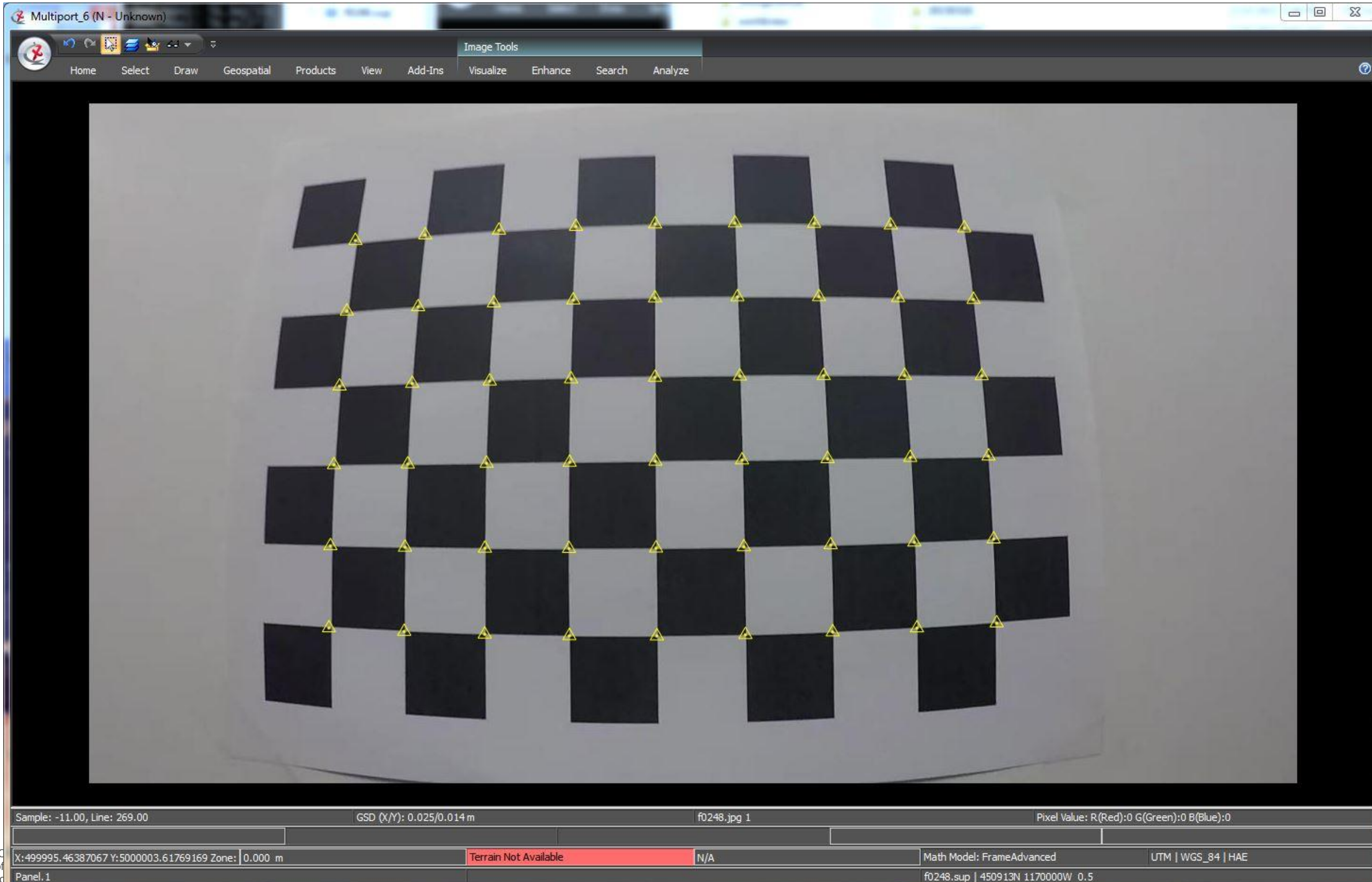




GoPro: Native lens distortion



GoPro: Lens distortion solved



Triangulation and bundle adjustment

Stages of triangulation

Triangulation is a process of fitting the images together optimally to reflect reality

- Ingest images
- Select sensor model(s)
- Initialize sensor model(s)
- Ingest ground control points (if any)
- Use *a priori* metadata to guide tie point matching
- Measurement of tie points by image matching
  - Area correlation
  - Interest operators, e.g. line intersections, corners, Förstner operator
  - Descriptors of key points, e.g. SIFT, SURF, ORB
- Measurement of ground control points
  - Measure in one image manually
  - Measure in other images by image matching

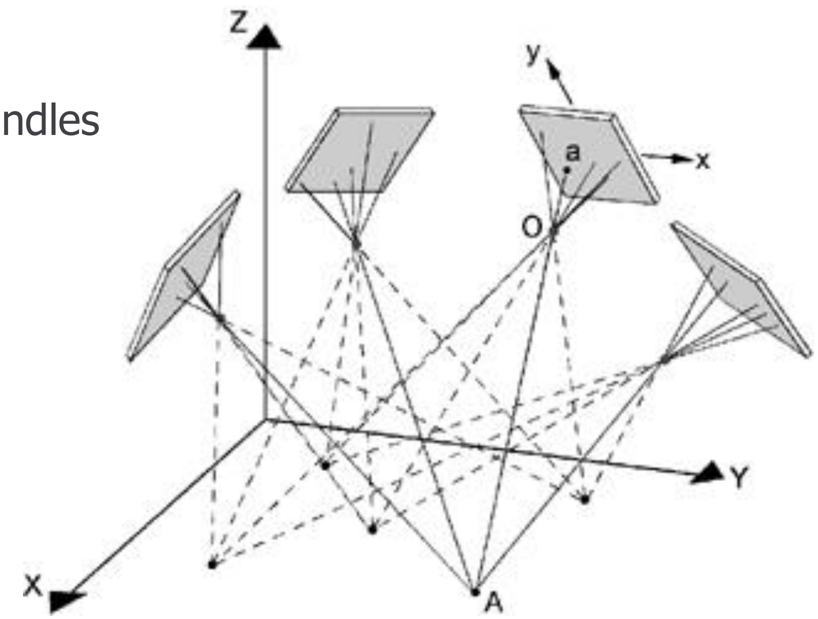
Stages of triangulation … 2

Triangulation is a process of fitting the images together optimally to reflect reality

- Bundle adjustment
  - Using least squares estimation to find optimum values for:
    - Parameters of exterior orientation
    - Parameters of interior orientation
    - Coordinates of tie points in ground system
- Assessment of results
- The above is sometimes called "structure from motion" in the computer vision world
- Exploitation of triangulated data
  - Multi-image geopositioning, i.e. measuring points
  - Sparse point cloud
  - Basis for further photogrammetric operations

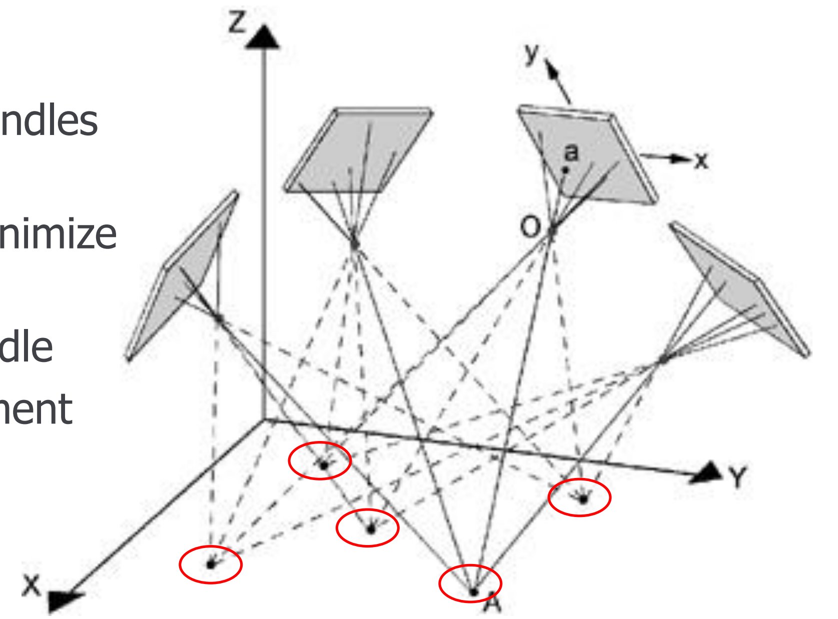
Triangulation – aka bundle adjustment

- Multi-image alignment is never perfect
- Tie points: measure (l,s) of common features
  - Bundle of image rays for each tie point
  - Ground location is estimated between bundles
  - g2i into each image (l',s')-(l,s)→residual



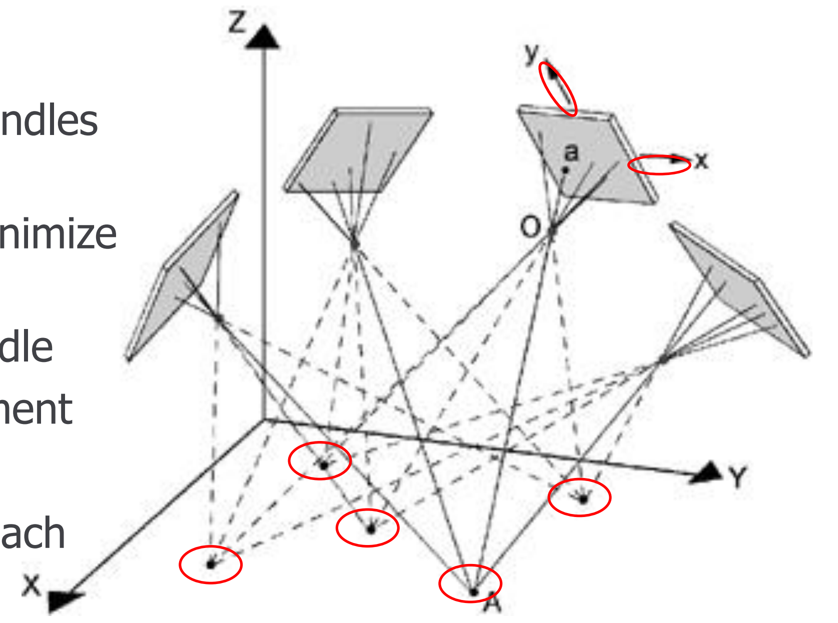
Triangulation – aka bundle adjustment …2

- Multi-image alignment is never perfect
- Tie points: measure (l,s) of common features
  - Bundle of image rays for each tie point
  - Ground location is estimated between bundles
  - g2i into each image (l',s')-(l,s)→residual
- Triangulation uses adjustable parameters to minimize overall residual
  - Imagine rubber bands cinching each bundle
    - Tighter = more accurate measurement
    - Looser = larger residual allowed



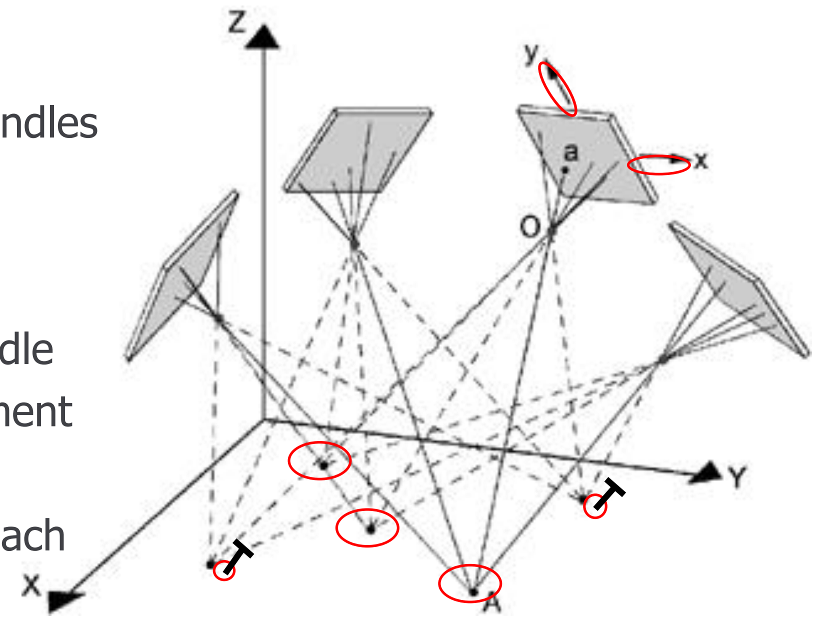
Triangulation – aka bundle adjustment …3

- Multi-image alignment is never perfect
- Tie points: measure (l,s) of common features
  - Bundle of image rays for each tie point
  - Ground location is estimated between bundles
  - g2i into each image (l',s')-(l,s)→residual
- Triangulation uses adjustable parameters to minimize overall residual
  - Imagine rubber bands cinching each bundle
    - Tighter = more accurate measurement
    - Looser = larger residual allowed
  - Also rubber bands allowing flexibility of each adjustable parameter
    - Tighter = smaller sigma



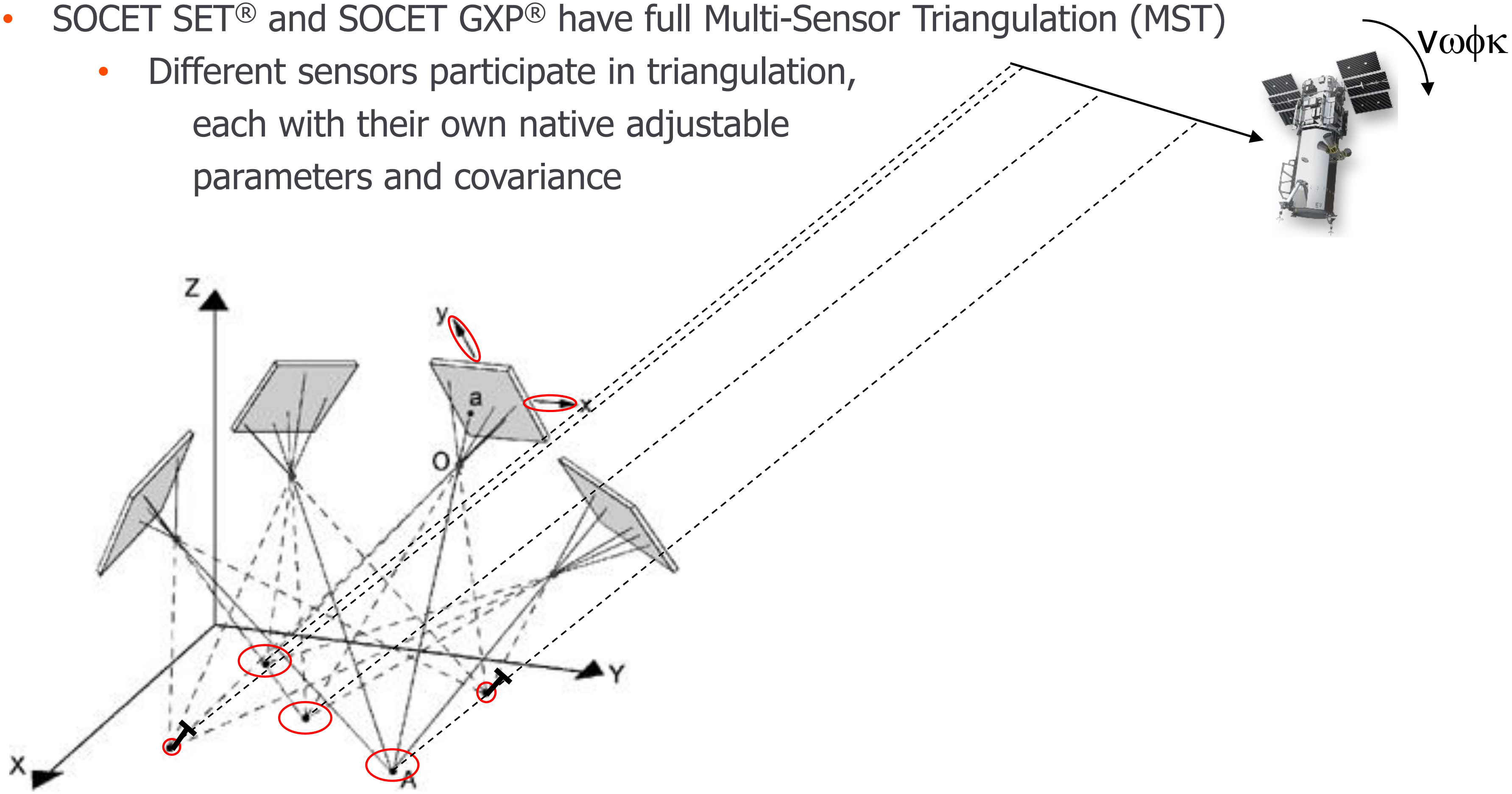
Triangulation – aka bundle adjustment ...4

- Multi-image alignment is never perfect
- Tie points: measure (l,s) of common features
  - Bundle of image rays for each tie point
  - Ground location is estimated between bundles
  - g2i into each image (l',s')-(l,s)→residual
- Triangulation uses adjustable parameters to minimize overall residual
  - Imagine rubber bands cinching each bundle
    - Tighter = more accurate measurement
    - Looser = larger residual allowed
  - Also rubber bands allowing flexibility of each adjustable parameter
    - Tighter = smaller sigma
  - Control point = bundled band attached to ground
    - Tighter = more accurate ground truth
- Triangulation is balancing all those forces



Triangulation – aka bundle adjustment …5

- SOCET SET® and SOCET GXP® have full Multi-Sensor Triangulation (MST)
    - Different sensors participate in triangulation, each with their own native adjustable parameters and covariance



Bundle adjustment in a nutshell

- What goes in
  - Sensor information and model
  - Image coordinates
  - Camera calibration information
  - Ground control points
  - *A priori* standard deviations (and, theoretically, covariances) of the image coordinates, parameters of interior orientation and ground control points
  - Initial values of interior orientation, exterior orientation and ground points
- And what comes out: estimated values of
  - Interior orientation
  - Exterior orientation
  - Coordinates of ground points = "the sparse point cloud"
  - Residuals
  - And associated standard error of unit weight and covariance matrices

Traditional bundle adjustment (metric camera etc.)

- *A priori* EO/IO pretty good
- Sensor models come first, then tie points
    - Small number of manual/semi-automated tie points
    - Bare minimum 'von Gruber Points'
- Small adjustments to EO
    - IO already lab-calibrated
- Achieve subpixel RMS→sub-GSD ground accuracy
- Continue processing:
    - Terrain generation, orthorectification, (A)FE, etc.

- Imagine if you had to do this with your UAS datasets!!

UAS bundle adjustment problems

- *A priori* EO/IO not good, may not even know which images overlap!
- More images (lower flying height→less ground coverage per image)
- Need to ‘self-calibrate’: solve for IO/distortion as well as EO
  - Note: ‘solve for’ >> ‘adjust’ or ‘estimate’
  - Note: can calibrate with images of planar calibration pattern
- Need to fully automate!
  - Thus need higher overlap (which also makes sure we don’t have gaps!):
    - 80% forward/side recommended vs 60/30 traditional
  - Makes for even more images

Ground control

- The need for ground control
- Acquisition of ground control – do it yourself, employ a land surveyor, or use a firm such as CompassData
- Quantity and quality of ground control points
    - How many?
    - How good?
- What do ground control points do to bundle adjustment?
    - Non-singularity – same effect as minimum constraints
    - Effect on accuracy

Arithmetic of least squares adjustment

- Bundle adjustment is typically solved by the method of least squares
- Say we have a simple mathematical model:

  $Ax = b$

  Where A is a design matrix with n rows, x is a vector of u unknowns and b is a vector of n observations
- Suppose the observations, b, are subject to errors: then we can write

  $Ax = b + v$, where v is a vector of residuals

- The least squares solution is to minimize v^TWv, where $W = \Sigma_b^{-1}$, which gives

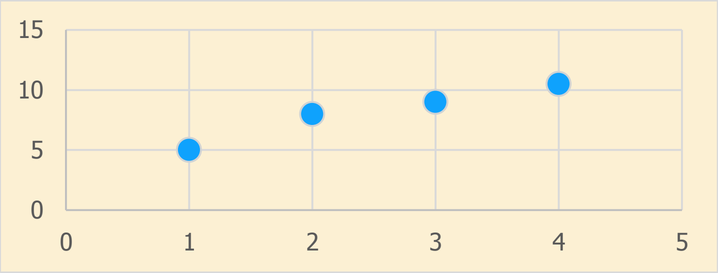
  $\hat{x} = (A^TWA)^{-1}A^Tb$; $\hat{v} = A\hat{x} - b$; $\hat{\sigma}_0^2 = v^TWv/(n-u)$; $\hat{\Sigma}_{\hat{x}} = \hat{\sigma}_0^2\,(A^TWA)^{-1}$

  [The equations $A^TWA\hat{x} = A^tb$ are known as the *normal equations*]

Example of simple least squares

- Mathematical model: straight line, i.e. y = mx + c
- Data

x	1	2	3	4
y	5	8	9	10.5

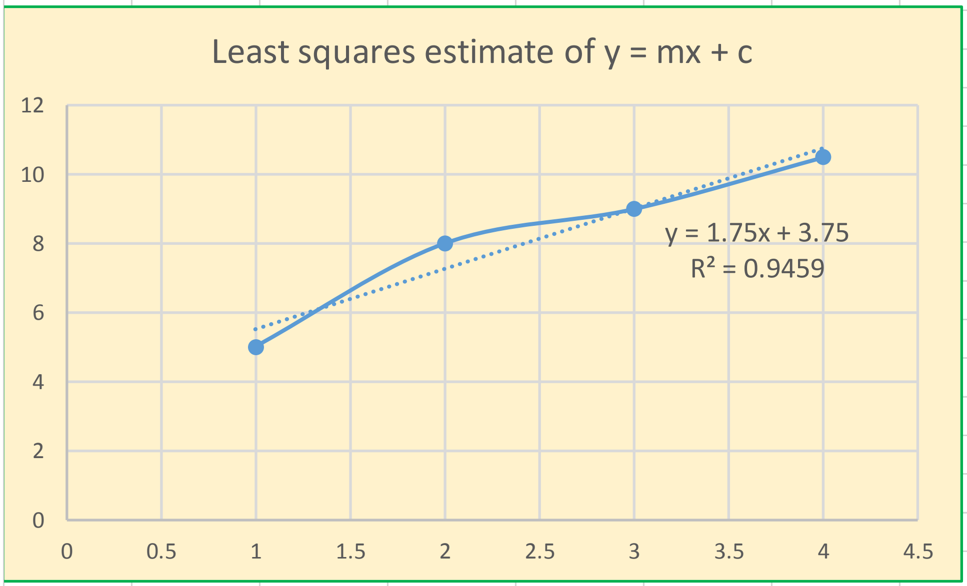


- Matrices and vectors
  - $A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix}^T$ $\qquad b = [5\ \ 8\ \ 9\ \ 10.5]^T \qquad W = I$
  - $A^TWA = \begin{bmatrix} 30 & 10 \\ 10 & 4 \end{bmatrix} \qquad A^TWb = \begin{bmatrix} 90 \\ 32.5 \end{bmatrix} \qquad (A^TWA)^{-1} = \begin{bmatrix} 0.2 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}$
  - $\hat{x} = [1.75\ \ 3.75]^T \qquad \hat{v} = [0.50\ \ -0.75\ \ 0.00\ \ 0.25]^T$
  - $\hat{\sigma}_0^2 = 7/16 \qquad \hat{\sigma}_m = \sqrt{[(7/16) * 0.2]} = 0.30 \qquad \hat{\sigma}_c = \sqrt{[(7/16) * 1.5]} = 0.81$

Example of simple least squares ...2

- Excel gets the same answer!

x	1	2	3	4
y	5	8	9	10.5



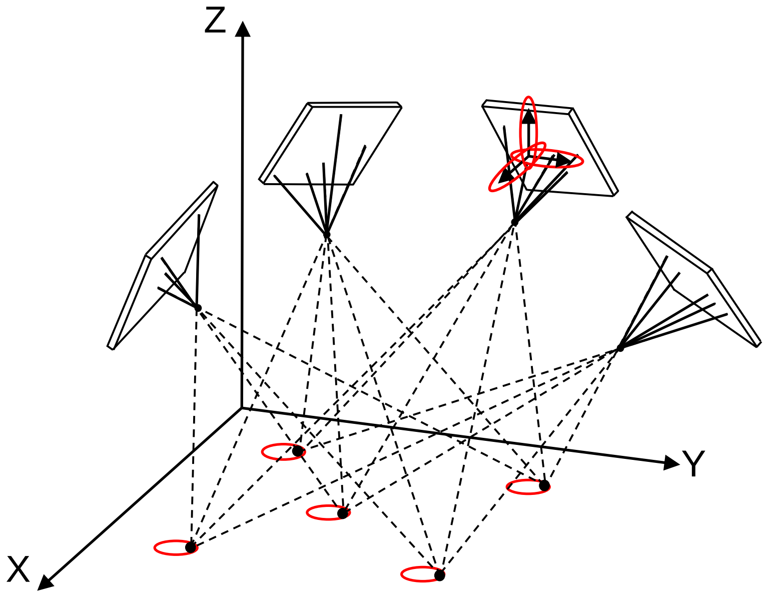
Bundle adjustment in reality

- Interested students should refer to the *Manual of Photogrammetry*, 6th edition, section 10.2, pp. 860 *et seq*., to show mathematical model, normal equations and solution
- Large number of unknowns, observations
  - Unknowns = $6p + 3q + r$, where
    - p = number of images
    - q = number of ground points
    - r = number of interior orientation parameters
  - Knowns = 3 * number of control points

    + Σ_i(2 * number of ground points observed in image i)
    - The second term is absolutely enormous
- This enormous system of equations is easier to solve than it looks, because the normal equations have a characteristic structure, called “sparse”, with lots of zeroes, thus reducing the number of arithmetic operations (at the expense of more complex algorithms!)
  - Again, see the *Manual of Photogrammetry* to understand this pattern

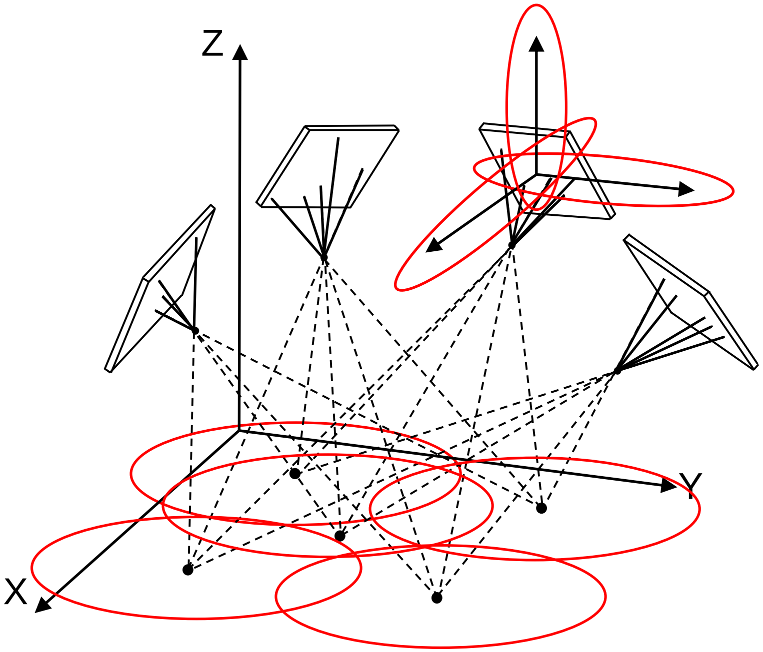
Traditional data collections

- Reasonably accurate GNSS and IMU data available
- Metric cameras with calibration reports: interior orientation known
- Minimal number of images collected to achieve sufficient spatial coverage
- Minimal number of tie points, less than 100 per image



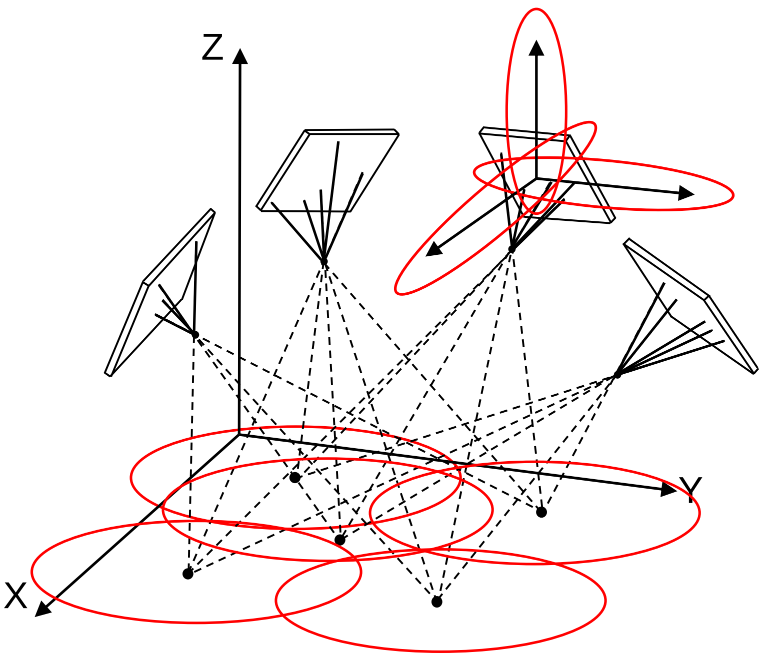
UAS data collections

- Poor GNSS and no IMU
- Consumer-grade cameras: interior orientation not well known
- Low flying heights and close-to-the-ground images – significantly more images for sufficient spatial coverage
- Feature-based automatic tie point extraction returns thousands of tie points per image



UAS data collections …2

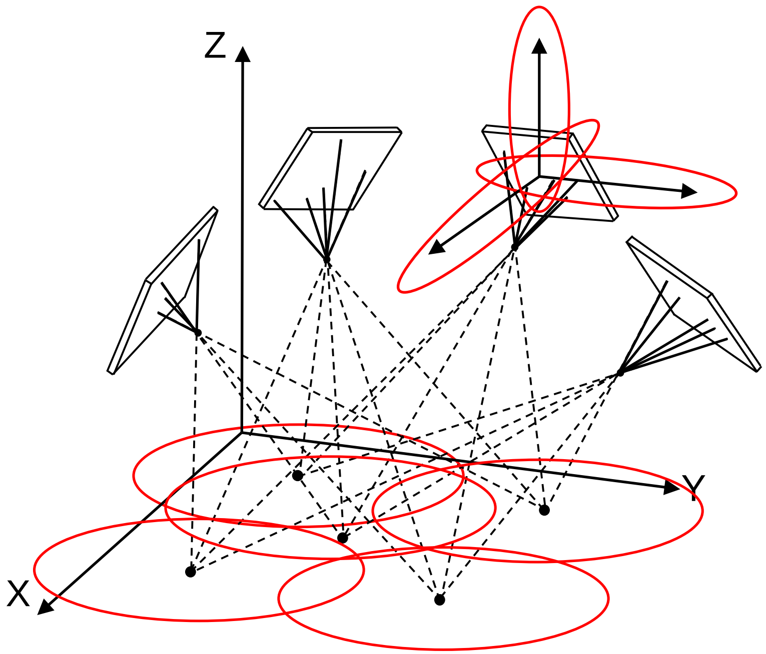
- Poor GNSS and no IMU
- Consumer grade cameras, interior orientation is not well known
- Low flying heights and close range images – significantly more images for sufficient spatial coverage
- Feature-based automatic tie point extraction returns thousands of tie points per image



- Many more images
- Many more tie points
- Little or no information about the sensor

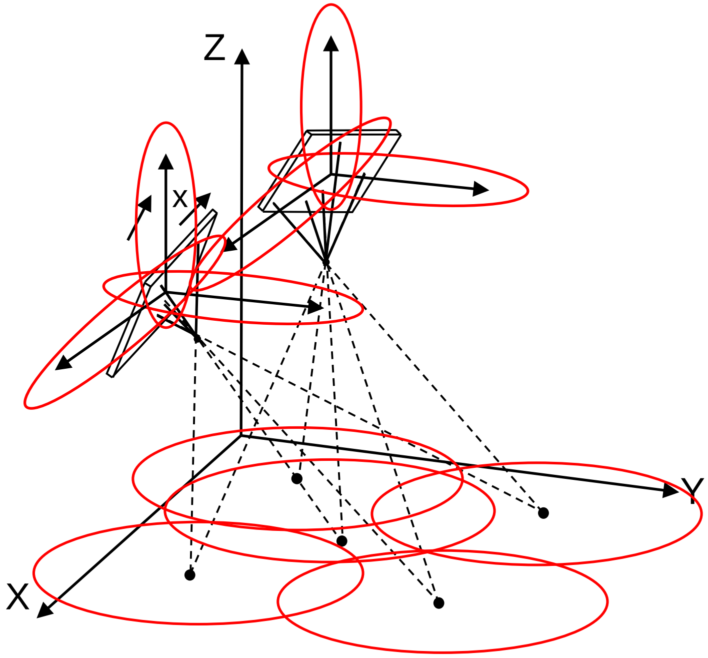
Why traditional triangulation fails

- Traditional triangulation is a one-step solution
- Too many variables – system of equations is too large to solve in one step
- Poor *a priori* values for adjustable parameters: one-step solution numerically unstable



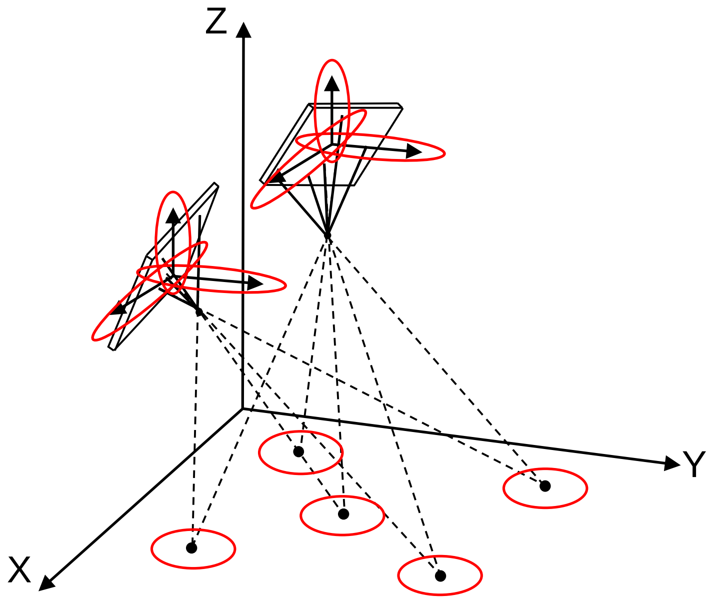
An incremental bundle approach

- Start with a small subset of the images



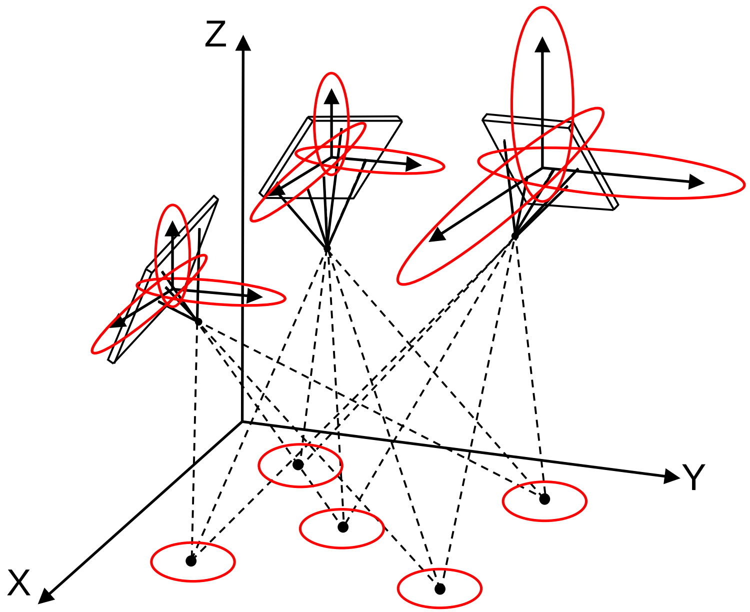
An incremental bundle approach …2

- Triangulate



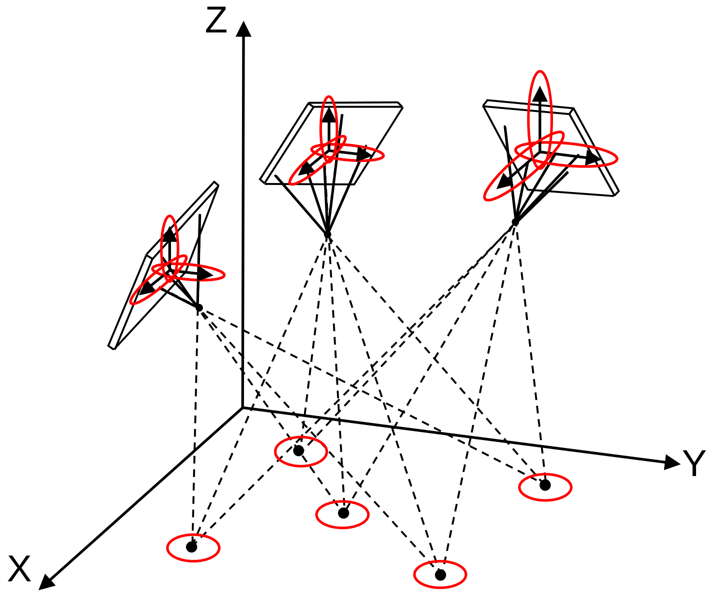
An incremental bundle approach …3

- Add the next image to the block



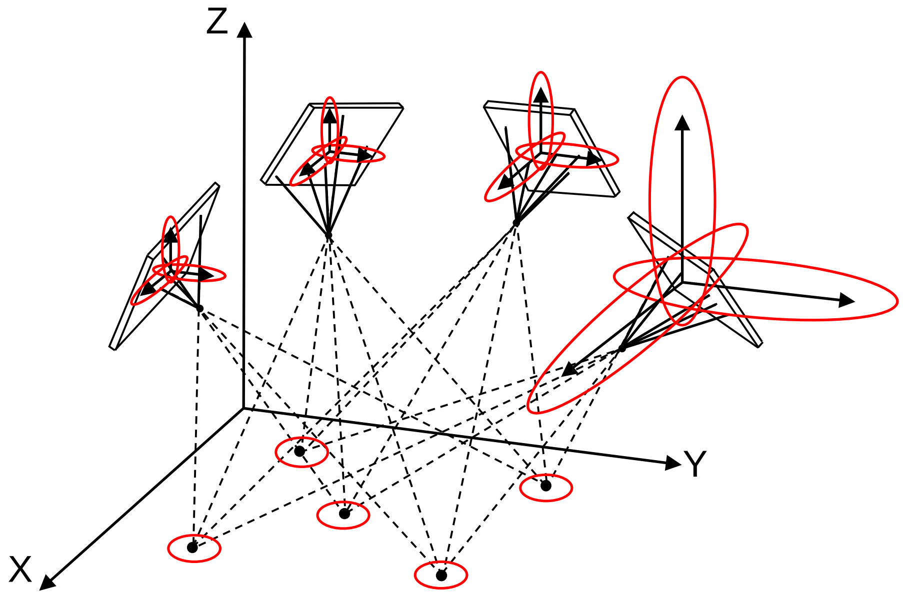
An incremental bundle approach …4

- Triangulate



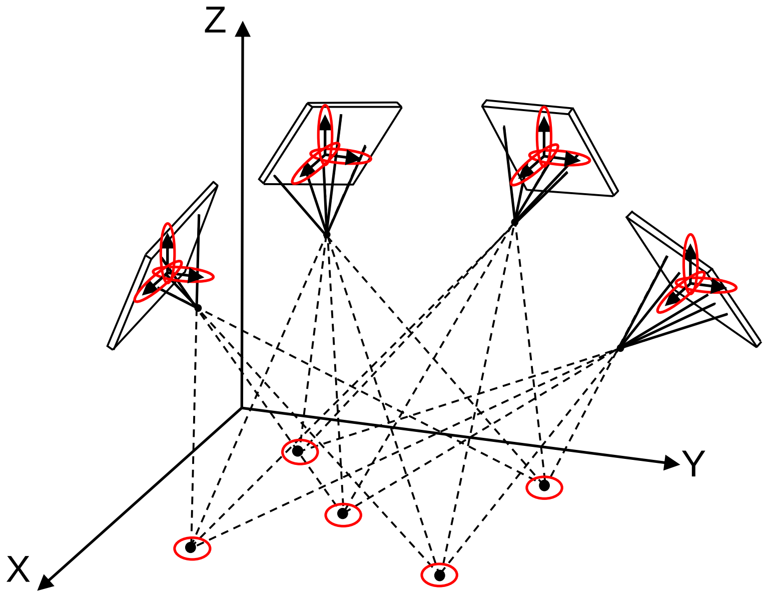
An incremental bundle approach …5

- Repeat process for all images



An incremental bundle approach …6

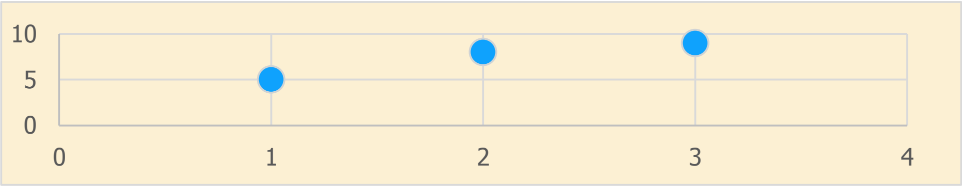
- Repeat process for all images



Example of simple sequential least squares

- Let's begin with our previous example, but use only the first three data points
- Mathematical model: straight line – y = mx + c
- Data

x	1	2	3
y	5	8	9

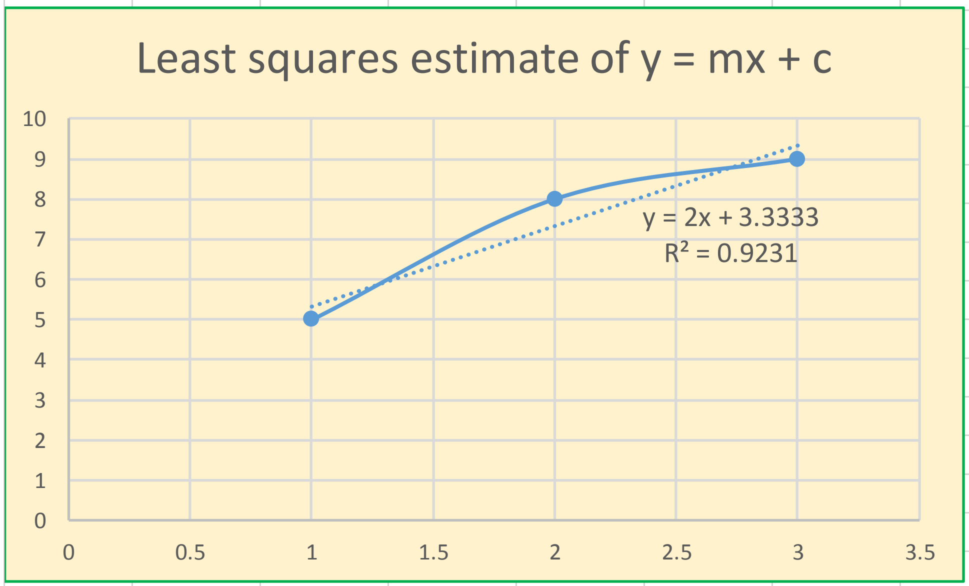


- Matrices and vectors
  - $A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \end{bmatrix}^T$ $b = [5 \ 8 \ 9]^T$ $W = I$
  - $A^TWA = \begin{bmatrix} 14 & 6 \\ 6 & 3 \end{bmatrix}$ $A^TWb = \begin{bmatrix} 48 \\ 22 \end{bmatrix}$ $(A^TWA)^{-1} = \begin{bmatrix} 0.5 & -1 \\ -1 & 14/6 \end{bmatrix}$
  - $\hat{x} = [2.00 \ \ 3.33]^T$ $\hat{v} = [0.33 \ -0.67 \ \ 0.33]^T$
  - $\hat{\sigma}_0^2 = 6/9$ $\hat{\sigma}_m = \sqrt{[(6/9) * 0.5]} = 0.58$ $\hat{\sigma}_c = \sqrt{[(6/9) * (14/6)]} = 1.25$

Example of simple least squares …2

- Excel gets the same answer!

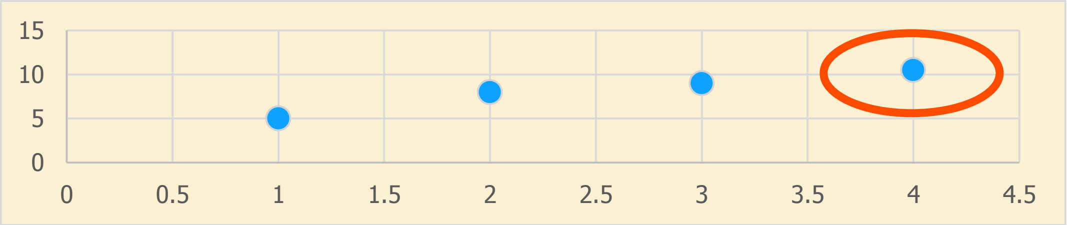
x	1	2	3
y	5	8	9



Example of simple sequential least squares ...3

- Using sequential least squares and lots of matrix equations, we can add the fourth point
- Mathematical model: straight line – y = mx + c
- Data

x	4
y	10.5



- Matrices and vectors
  - $A_2^T = \begin{bmatrix} 4 \\ 1 \end{bmatrix}^T$ $\quad b_2 = [10.5]^T \quad W_2 = I$
  - $\hat{x}_2 = \hat{x}_1 - N_1^{-1} A_2^T [(W_2^{-1} + A_2 N_1^{-1} A_2^T)^{-1} (A_2 \hat{x}_1 - b_2)] = [1.75 \;\; 3.75]^T$
  - $N_2^{-1} = N_1^{-1} - N_1^{-1} A_2^T [(W_2^{-1} + A_2 N_1^{-1} A_2^T)^{-1} A_2 N_1^{-1} = \begin{bmatrix} 0.2 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}$

Photogrammetric products

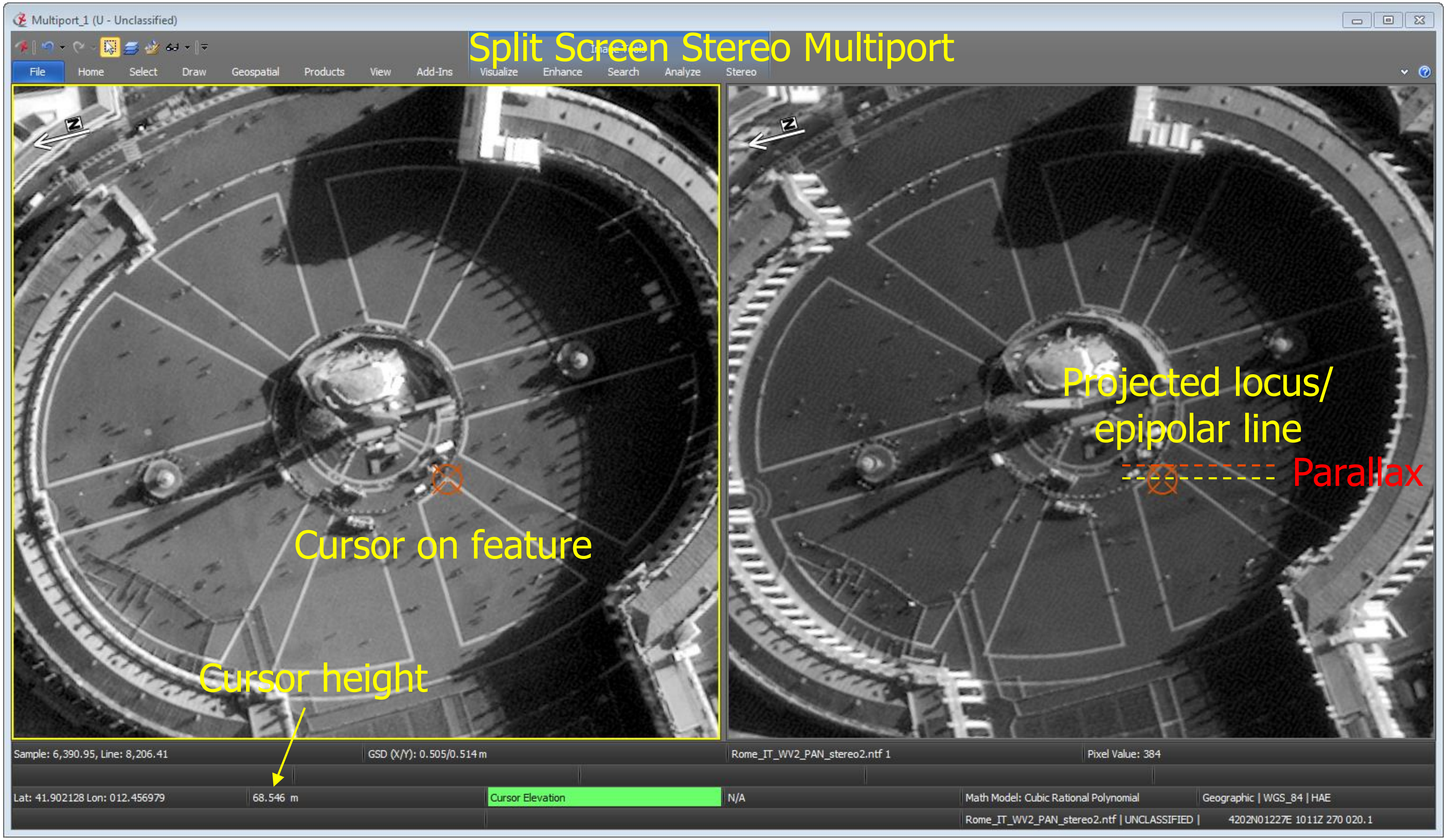
The possibilities are endless – listen to the papers tomorrow!

- Orthorectified imagery (with UAS imagery, this is usually a mosaic)
- Image analysis products, with measurements, annotation, title, margin information etc.)
- Terrain analysis, e.g. terrain shaded relief, contour maps, slope maps, aspect maps, volumetrics (e.g. stock piles and tailings), hydrographic information and flood risk maps
- Building models
- Radiometrically derived information, e.g. NDVI
- Land cover and land use
- Change detection

Terrain generation

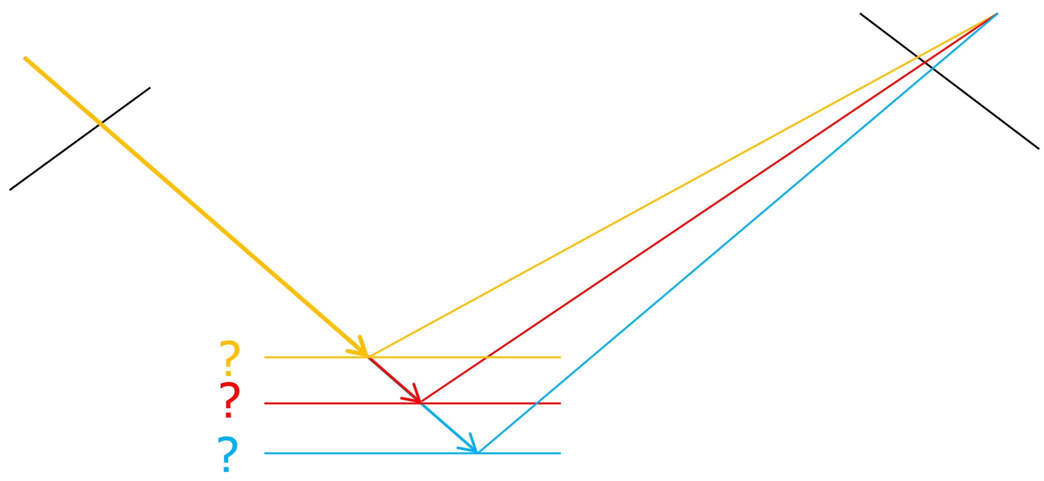
- Distinction between sparse point cloud and dense point cloud
- Ray intersection at as many possible pixels a possible
- Pair (tuple?) selection/merge 3D information from multiple pairs?
- What is semi-global matching (SGM: Hirschmuller)?
- Various formats of terrain products
    - DSM vs DEM
    - Cloud/Grid/TIN
    - Colorized from imagery
- Extraction of features such as buildings

Stereo parallax



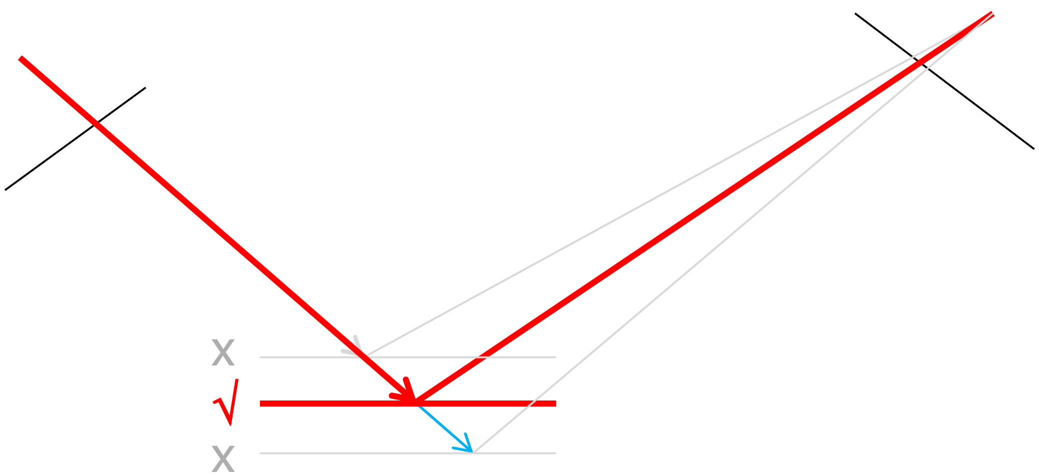
Why parallax is critical

- Terrain generation needs to find a matching (sub)pixel in the other/right image for each pixel in this/left image
- For ideal imagery, i2g at different possible heights projects into a line of possible pixels in the other image, find the match ➔ fix the height



Why parallax is critical …2

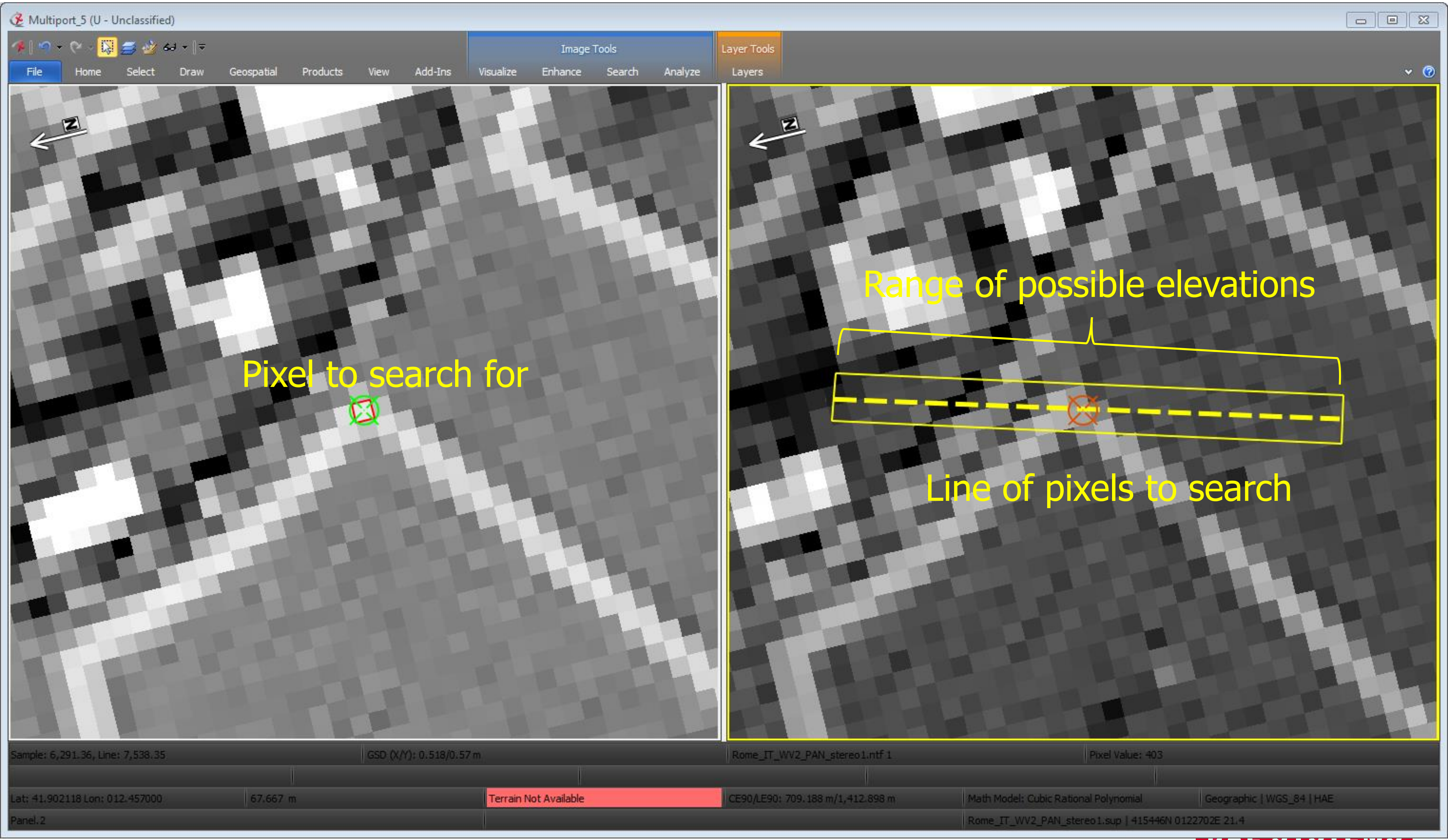
- Terrain generation needs to find a matching (sub)pixel in the other/right image for each pixel in this/left image
- For ideal imagery, i2g at different possible heights projects into a line of possible pixels in the other image, find the match ➔ fix the height



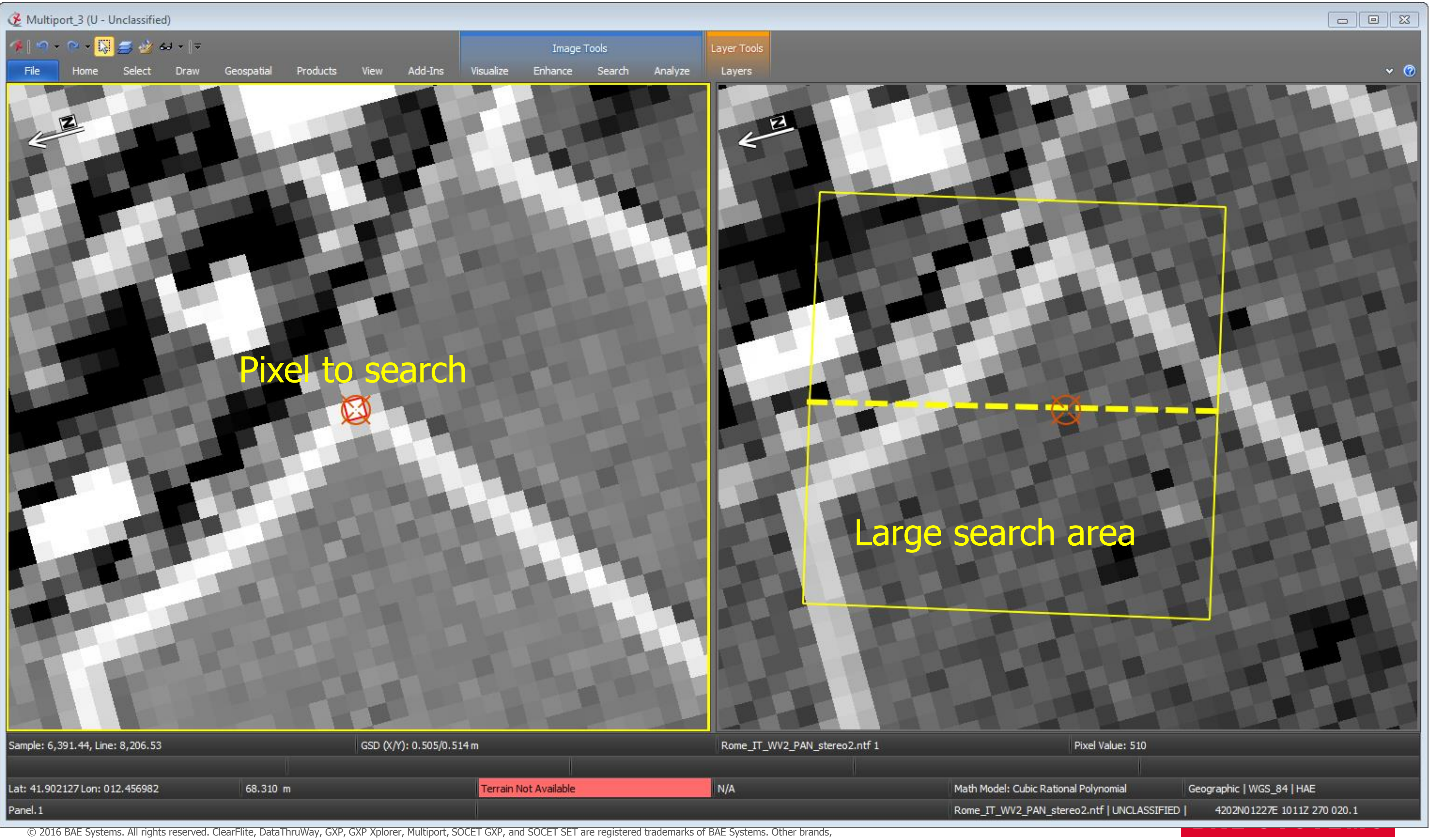
Why parallax is critical …3

- Terrain generation needs to find a matching (sub)pixel in the other/right image for each pixel in this/left image
- For ideal imagery, i2g at different possible heights projects into a line of possible pixels in the other image, find the match → fix the height
- Parallax presents three problems:
  - Matching pixel is away from the projected line, thus to find it requires searching an **area**, not a **line** → much slower
  - Area vs. line also: more potential matches → more potential false positives
  - When the matching pixel is found, rays do not intersect → how to choose ground point in space between them? Which ray should move how far?
- Frame imagery case will demonstrate effect of triangulation solution quality on terrain generation

Matching pixel search with no parallax



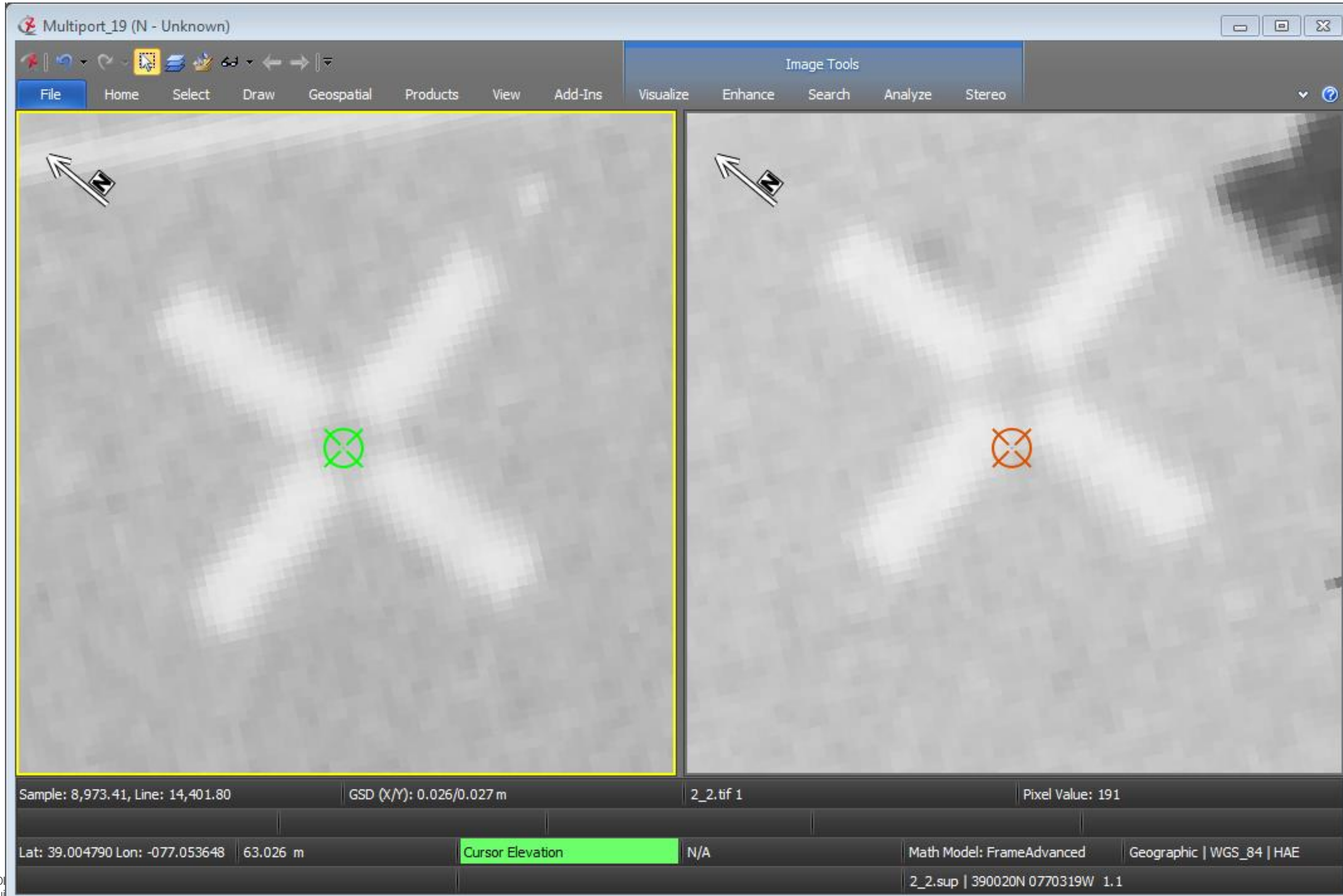
Matching pixel search with parallax



Case 3: High-end frame triangulation

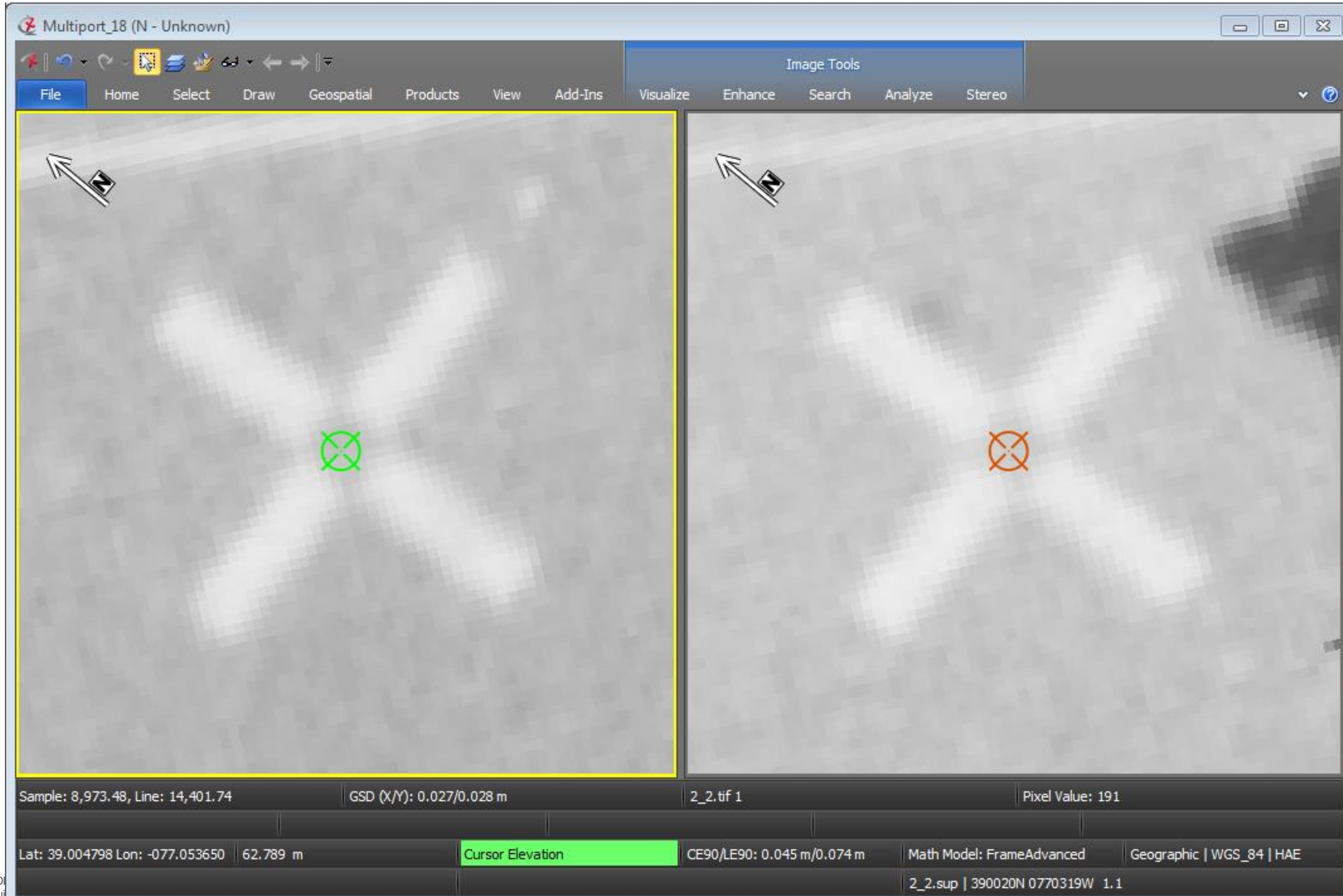
- Dataset: Walter Reed Army Institute of Research (outside DC)
  - High-end film mapping camera (RC30, calibrated), 2.8 cm GSD
  - 150 mm focal length, 23x23 cm film scanned at about 14 μm
  - GPS/IMU metadata
  - Ground control points
- Workflow
  - Frame import (images, metadata, calibration, control)
  - *A priori* parallax is quite small, 5-10 pixels only! But ASM has low success (31%)
  - Automatic workflow is easy, 1.8pix RMS, ASM success rises to 59%
  - ~1 hour of manual editing brings RMS down to 0.45pix, ASM success 64%

A priori parallax

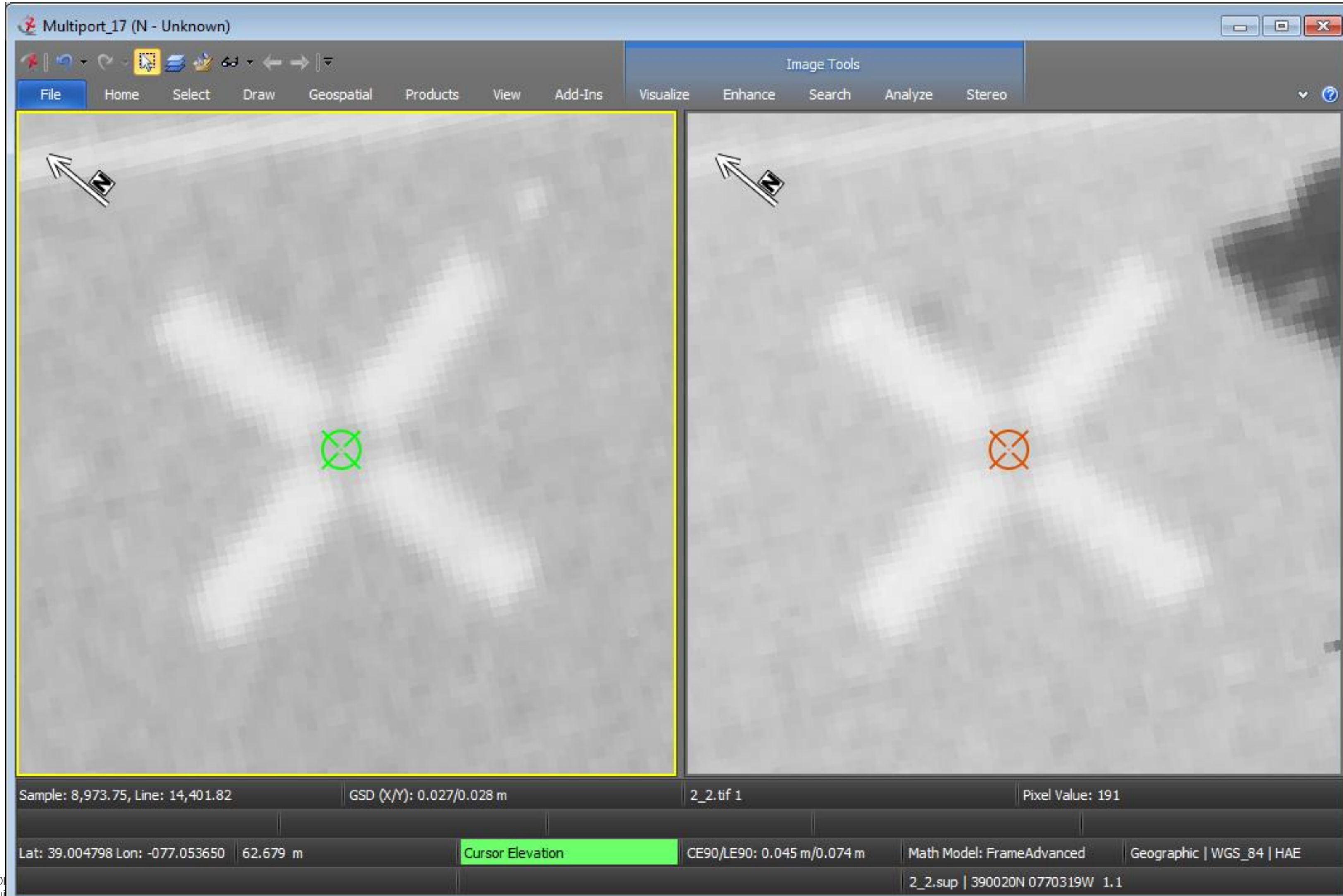


BAE SYSTEMS
INSPIRED WORK

Parallax after automatic triangulation

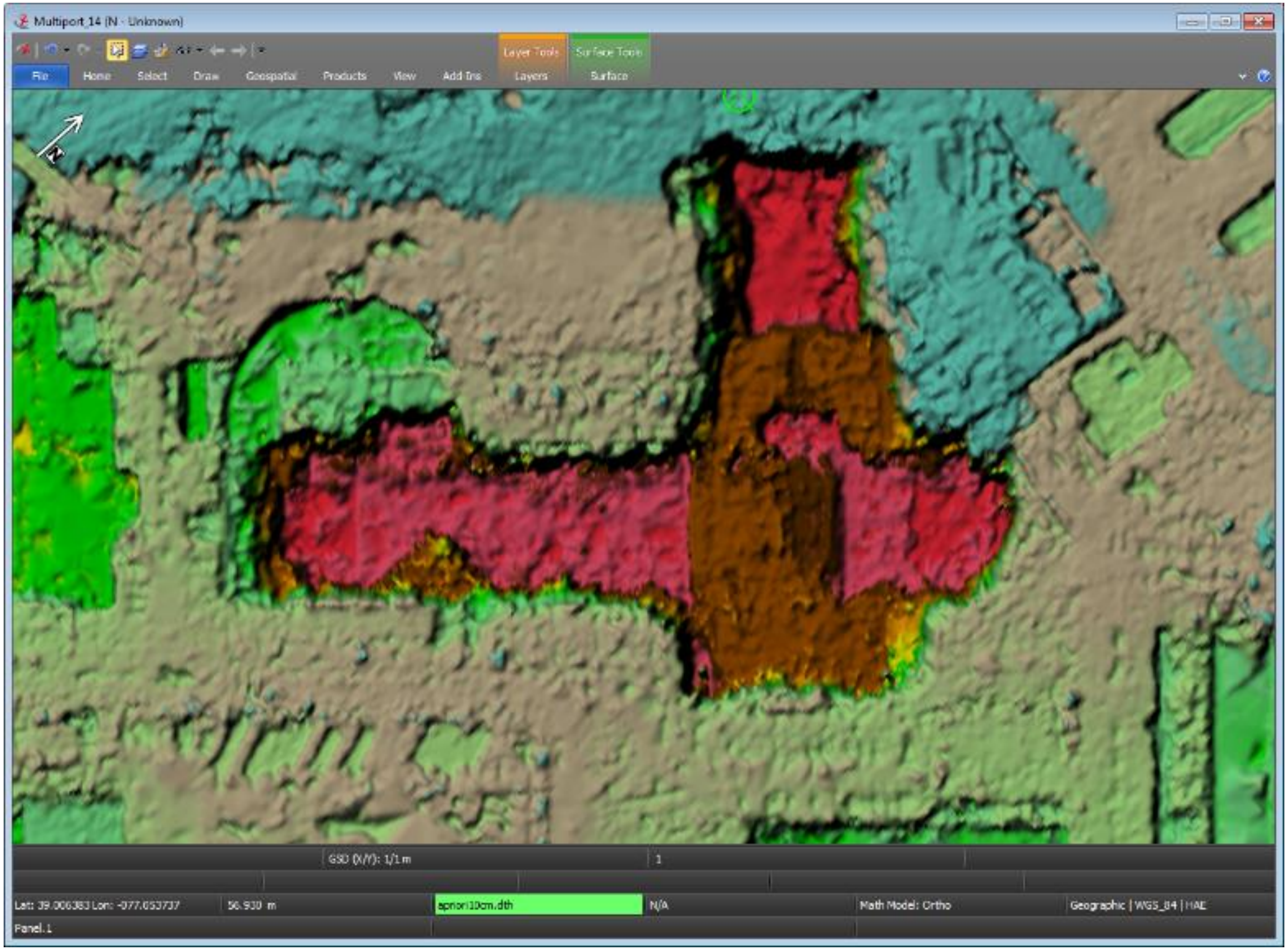


Parallax after manually edited triangulation

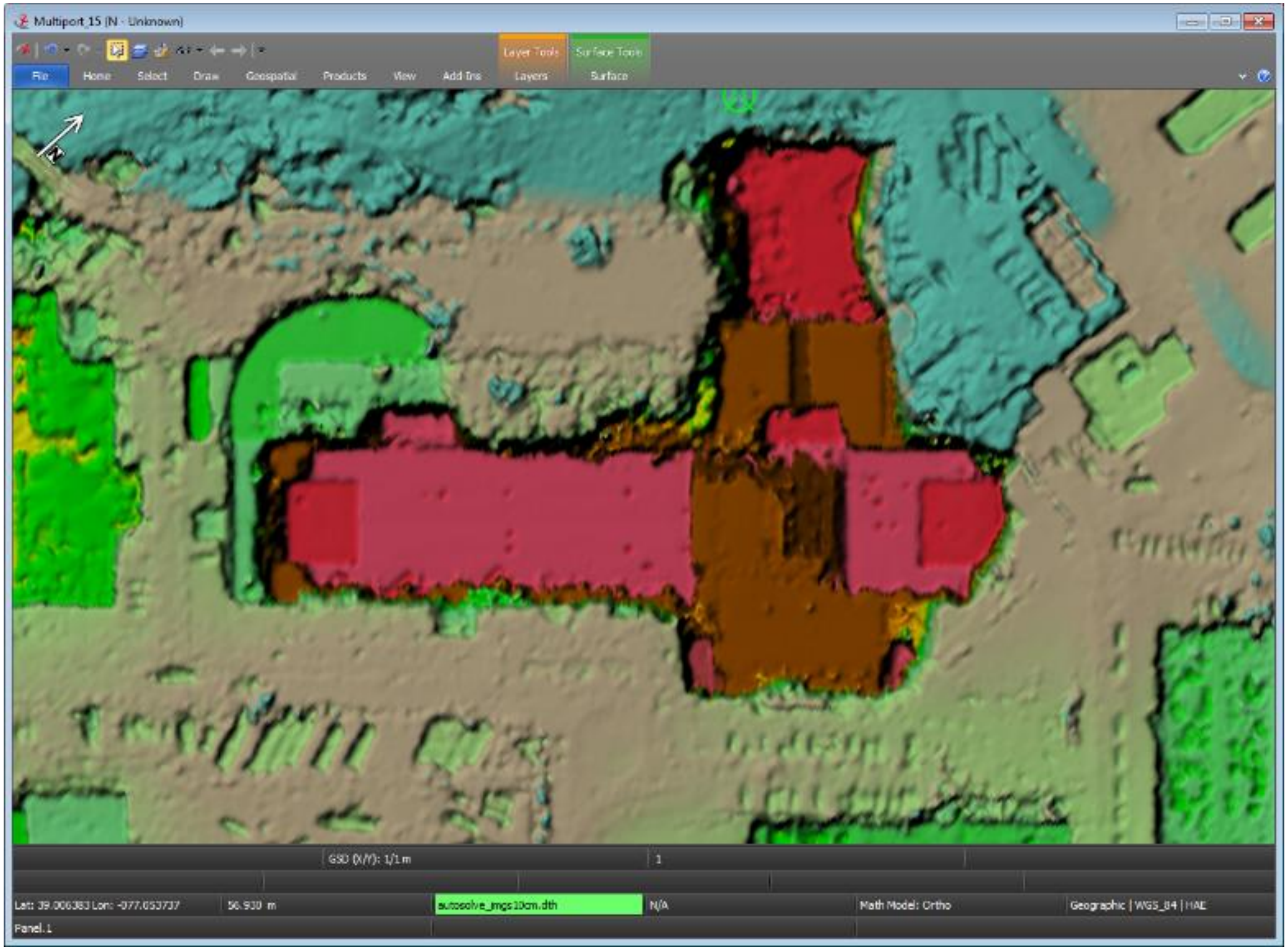


ASM terrain comparison

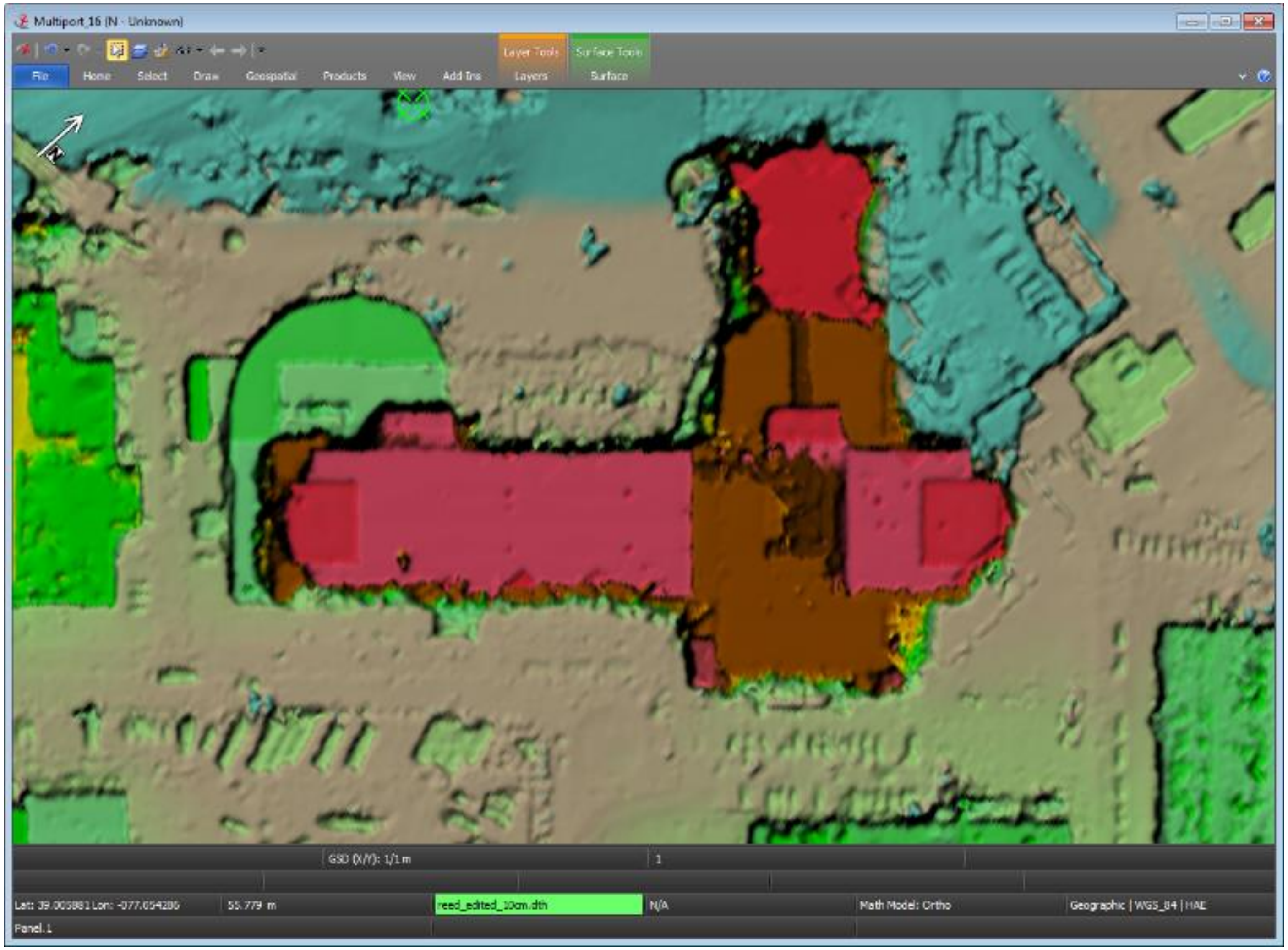
ASM terrain after no triangulation



ASM terrain after automatic triangulation

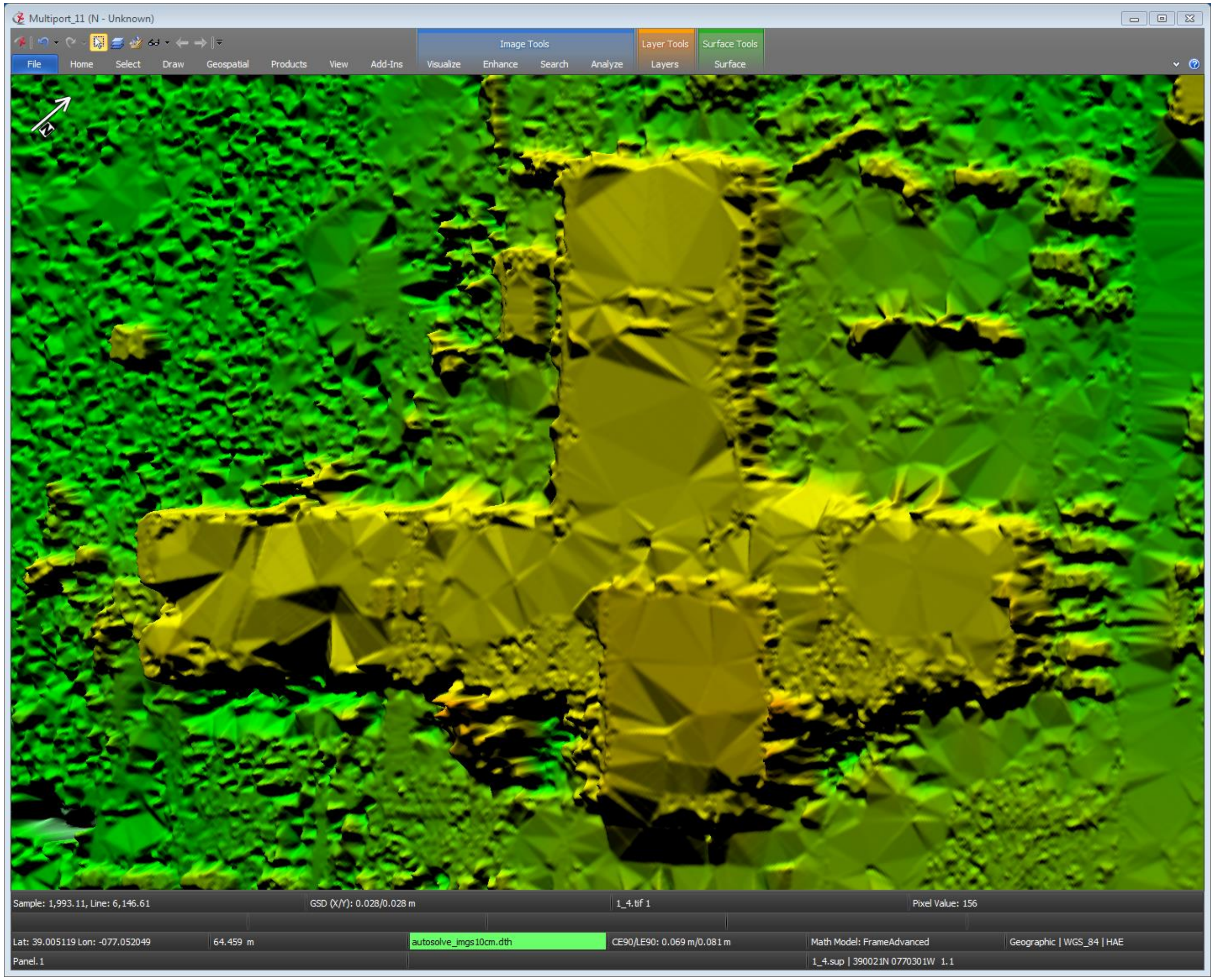


ASM terrain after manually edited triangulation

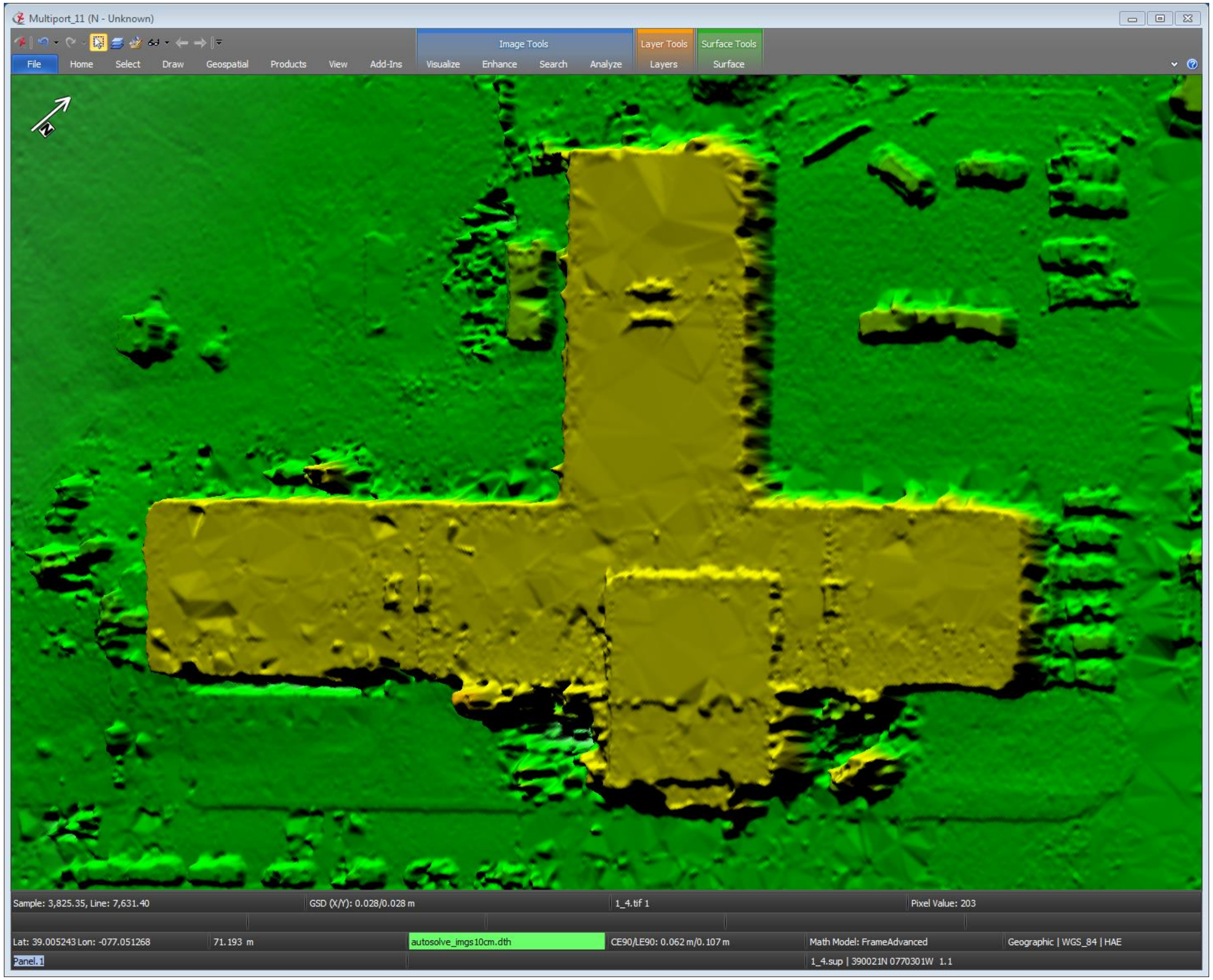


ASM terrain comparision ...2

ASM terrain after no triangulation



ASM terrain after automatic triangulation



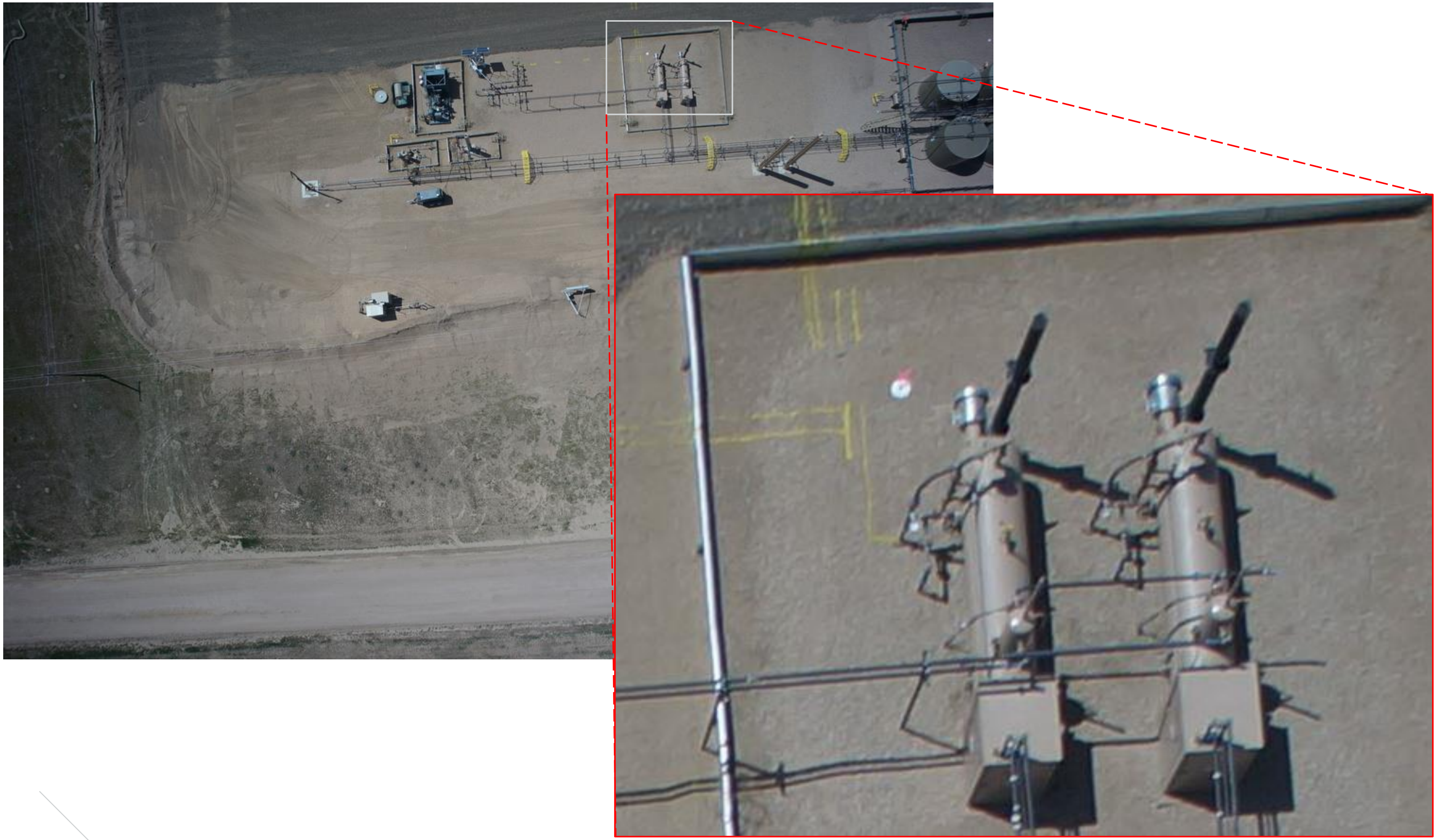
Case 4: UAV triangulation

- UAV imagery of under-construction industrial site from Compass Data, Inc.
  - Sony NEX-5R (off-the-shelf consumer camera, under $700)
    - Flying height 75 m; 2.1 cm GSD
    - Spec: 15.5 mm focal length; 4.77 μm pixel size
    - No calibration; hundreds of pixels of lens distortion
  - Metadata:
    - Quadcopter has GPS/IMU
    - GPS-surveyed ground control points

Case 4: UAV triangulation …2

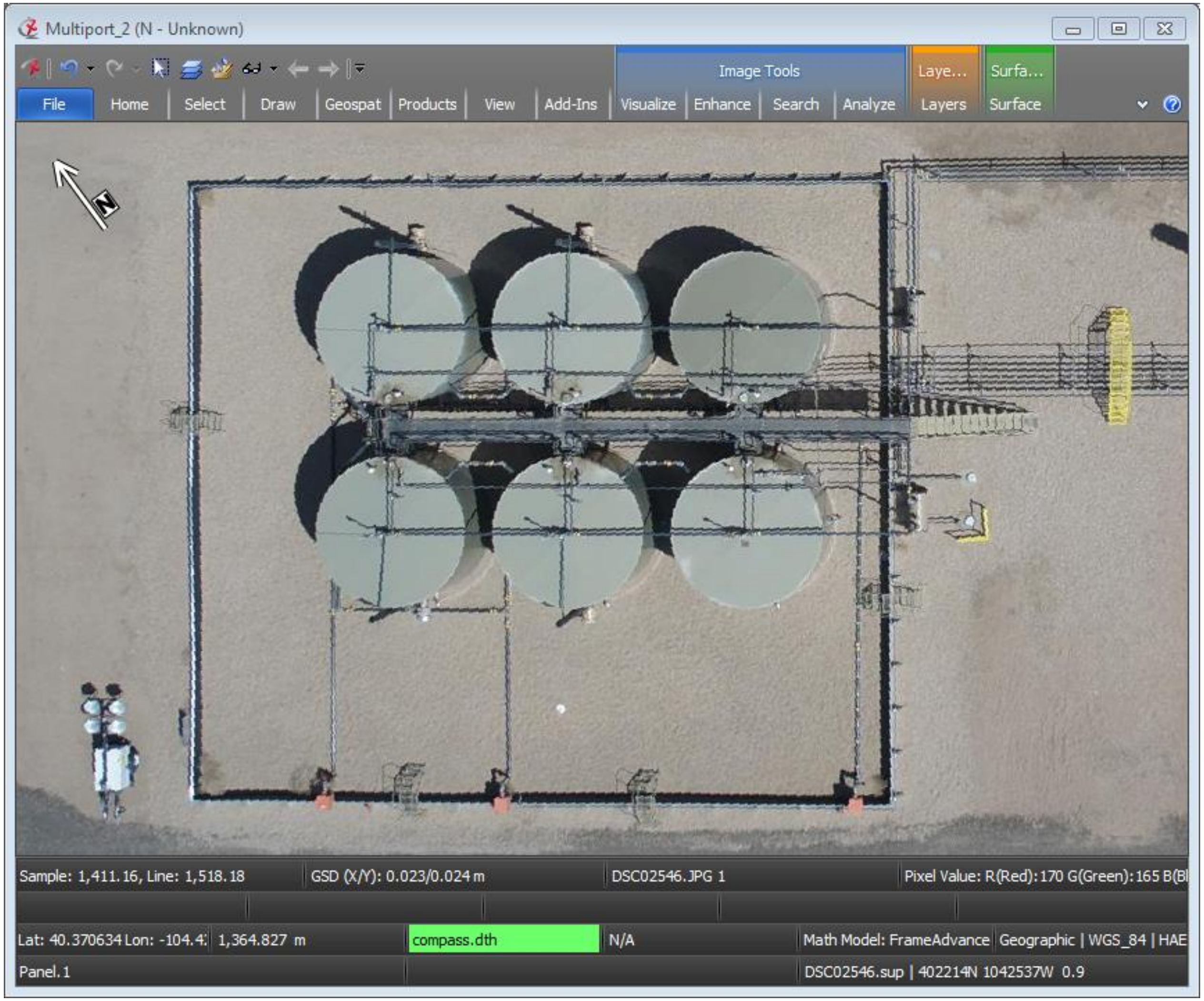
- Workflow:
  - Import imagery with GPS camera positions and heading/pitch/roll data
    - (Peek at GPS-only import with auto-heading)
  - Load measurements of control and tie points
  - Solve Exterior Orientation (EO) for each strip separately
    - RMS~10-20pix, large residuals at image edges, due to lens distortion
  - Solve Interior Orientation (IO) with constraint file (self-calibrate)→RMS<1pix
  - Inspect *a posteriori* parallax

UAV sample imagery

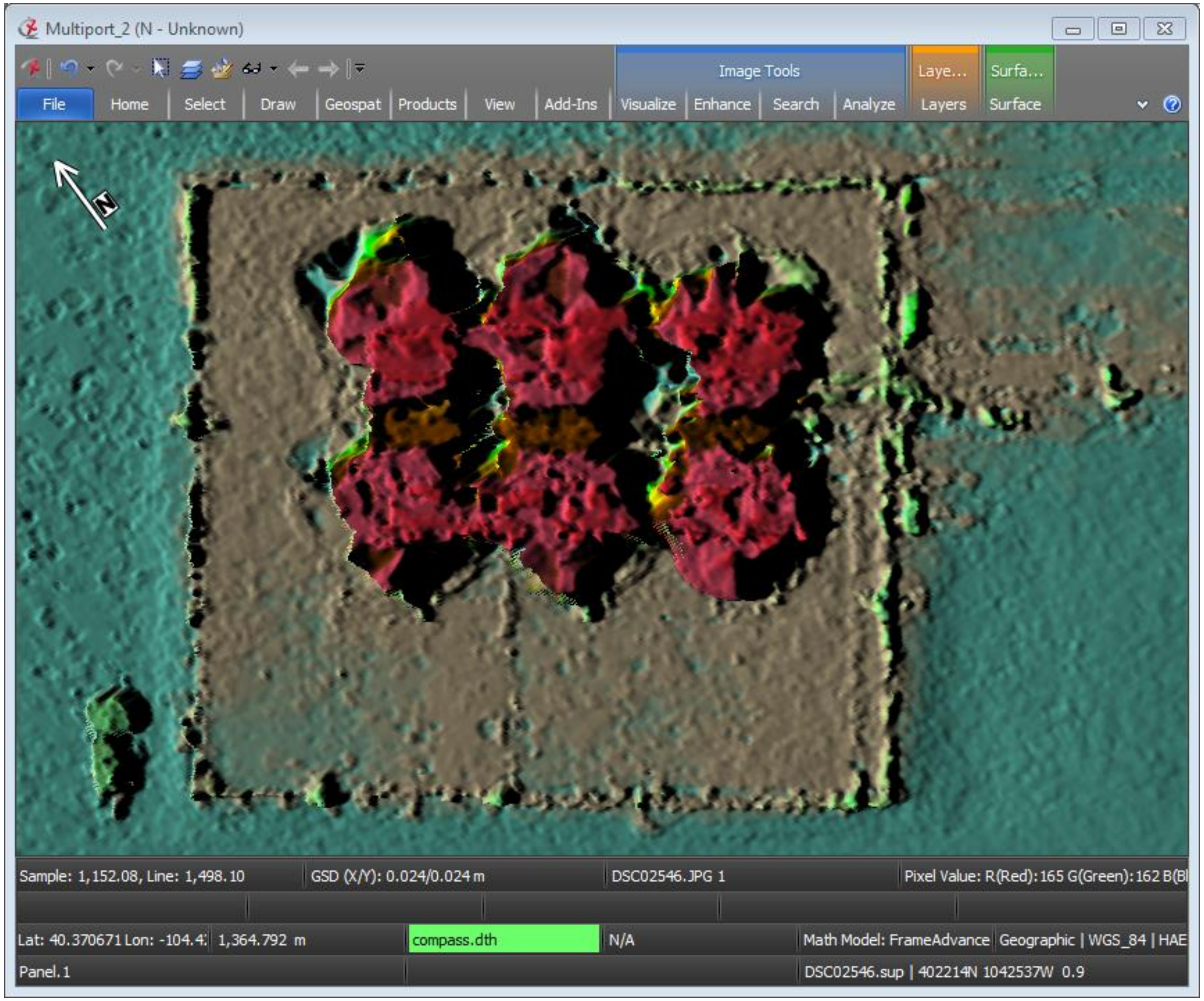


BAE SYSTEMS
INSPIRED WORK

ASM terrain generated from UAV imagery



ASM terrain generated from UAV imagery ...2

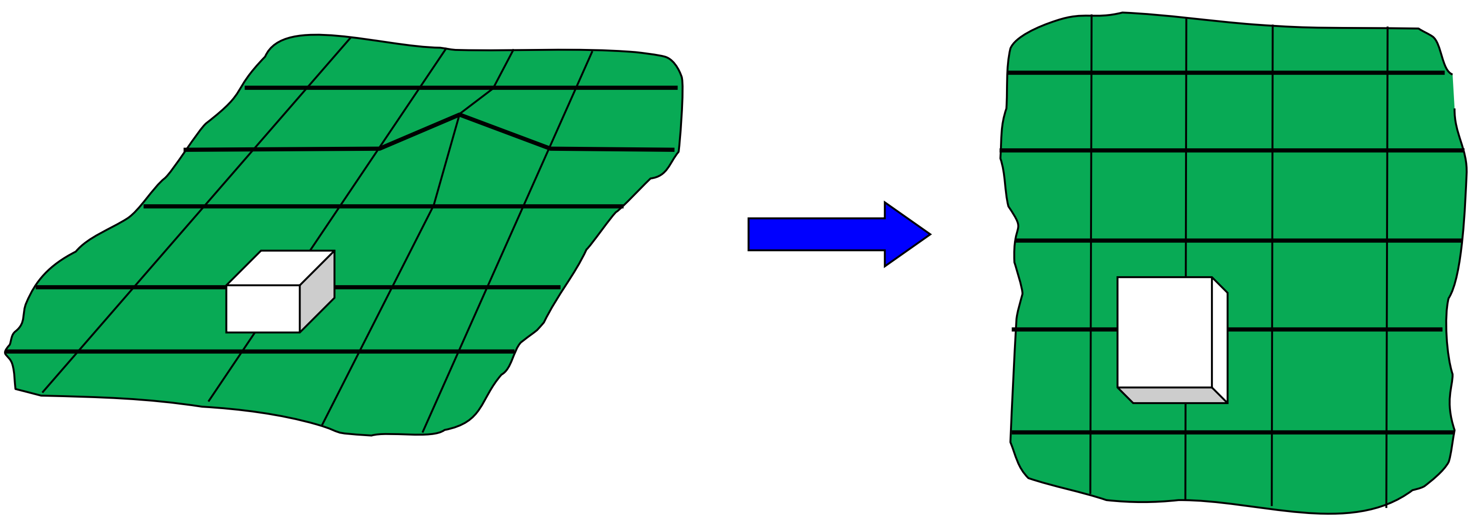


Ortho

- Orthorectified imagery means:
  - Ortho=right: square pixels (oblique perspective gives diamond-shaped pixels)
  - Rectified=straightened: north-up, east-right (arbitrary imagery has rotation)
  - Everywhere purely overhead perspective (no layover from height)
- To remove layover, it is important to understand feature shapes/heights
  - Create grid at desired horizontal resolution
  - For each horizontal grid location (pixel), determine height from 3D model
  - Determine which image(s) see the 3D location
    - g2i with adjusted camera models
    - avoid occlusion using 3D model
  - Apply RGB from (sub)pixel location of chosen image

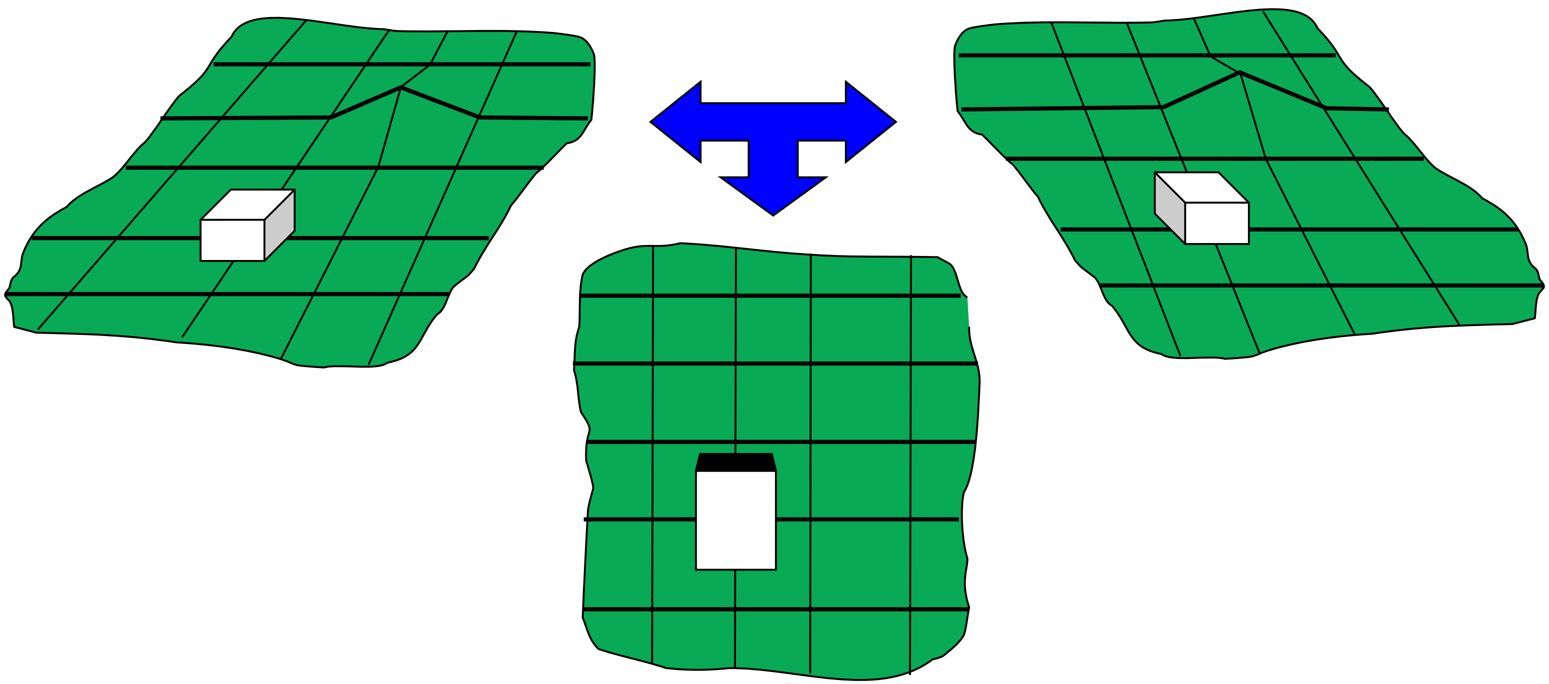
Orthorectification using only terrain

- Running Ortho Manager with a Digital Terrain Model (DTM) removes distortions due to DTM relief, but features such as building, remain tilted



True orthorectification requires features

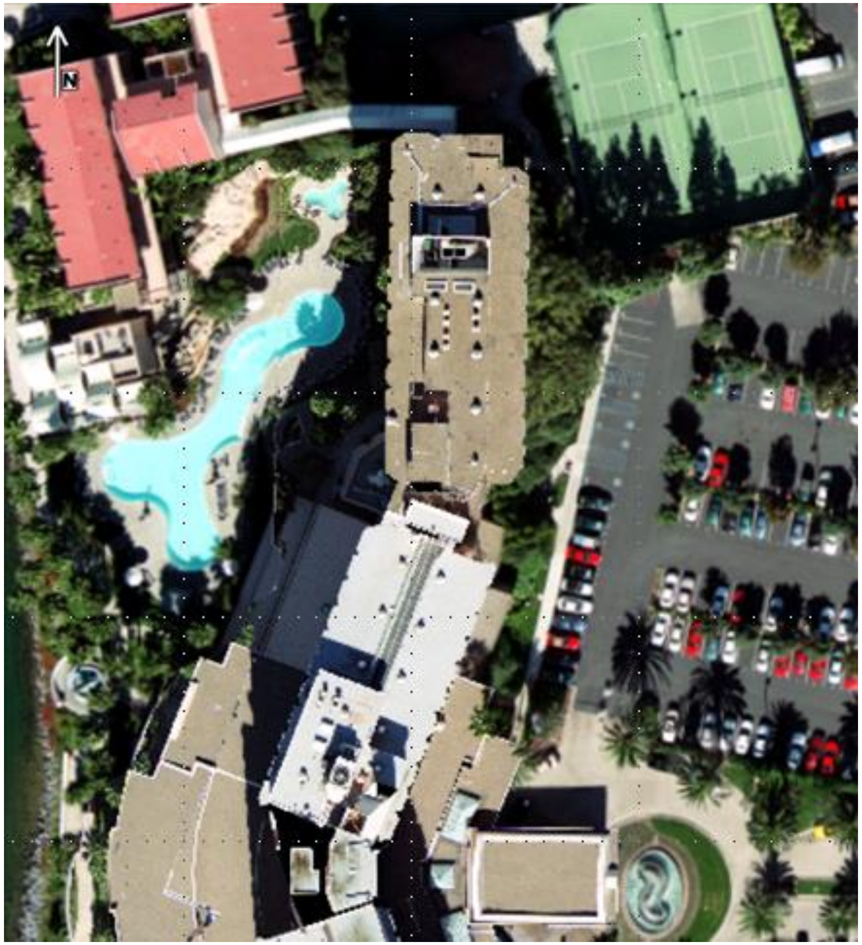
- Running Ortho Manager in **True** Orthophoto mode with a DTM and 3-D feature file pulls buildings upright by putting their roofs over their footprints
  - “Shadows” are present if no supplemental imagery is available
  - In the scenario below, the stereo mates supplement each other and (mostly) cover the shadow area by filling in with valid imagery



True orthophotos

Original image

True orthophoto



True orthophotos ...2

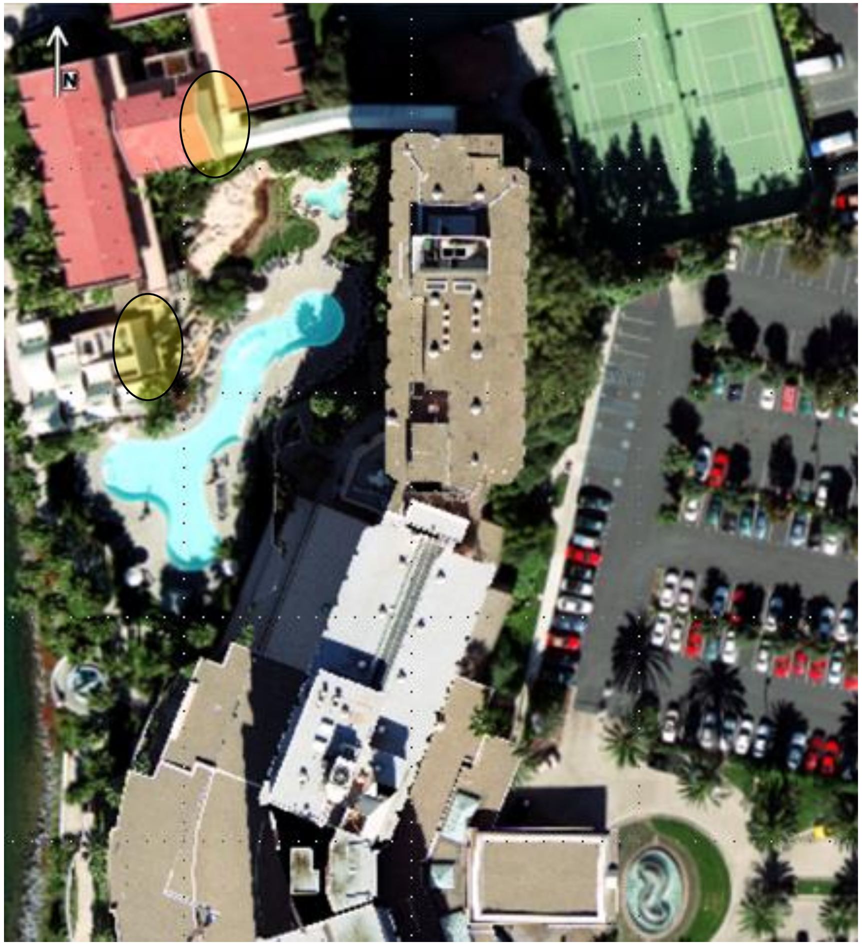
Original image

True orthophoto

True orthophotos …3

Original image

True orthophoto



Final points on products

- The main focus in photogrammetry has always been on complex, time-consuming, detailed, accurate products, i.e
  - Line maps created by painstaking, interactive feature extraction
  - DSMs and DEMs created by automated processes backed up by interactive editing
  - Orthorectified images, usually joined together into a mosaic, with sophisticated software for dodging and balancing the images so that the result is aesthetically pleasing
  - True orthos (see above)
  - Visualization using fast, sophisticated rendering algorithms to make building models, fly-throughs etc.
- But straightforward, fast products are also very important
  - Image-based product with a few annotations, measurements etc.
  - Colored overlay on orthorectified images, e.g. NDVI
- In most cases there is some cartographic finishing, e.g. scale, north point, legend, title, date, source, copyright …

Summary and conclusions

Triangulation changes for the UAS world…

UAS workflow is similar to "traditional workflow" – it just looks a bit different!

- Characteristics of UAS flight mission are the main factor:
  - Poorer metadata, owing to lack of camera calibration, less precise GNSS/IMU, more tilts and variations in flying height
  - Cameras have large distortions
  - Images are acquired with very large overlaps
  - Result is that initial values to start the triangulation may be poor

Triangulation changes for the UAS world …2

UAS workflow is similar to "traditional workflow" – it just looks a bit different!

- Triangulation suppliers have attempted to meet the above challenges, while providing the user with a straightforward, satisfactory experience
- Since UAS triangulation has very large number of tie points, owing to the use of matchers such as SIFT and its congeners, the adjusted ground coordinates of the tie points provide a "sparse point cloud"
- The approaches used draw more heavily from computer vision and pattern recognition than "traditional photogrammetry"
- UAS community expects fully automated workflow
- Further photogrammetric processes then continue…

Approvals/dataset information

- Case 3 dataset is the Walter Reed Army Institute of Research, from SOCET GXP Training Data
- Case 4 dataset is UAV imagery from Compass Data, Inc., same as used in GXP360 2015 R&D workshop and many other external UAV presentations
- We also acknowledge our former colleague Seth Merickel for creating many of the graphics amidst many lively discussions

Thank you

- Dr. Reuben Settergren
  - Geospatial eXploitation Products
  - 10920 Technology Place, San Diego, CA 92127
  - (858) 592-5725 | reuben.settergren@baesystems.com
- Dr. Stewart Walker
  - Geospatial eXploitation Products
  - 10920 Technology Place, San Diego, CA 92127
  - (858) 592-1764 | stewart.walker2@baesystems.com

ES-GEO-071116-0111