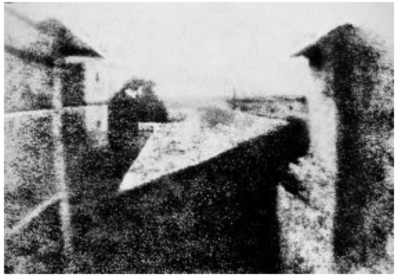
GEO(3|4)460

Photogrammetry in 2h

Luc Girod

Department of Geosciences
University of Oslo

A tiny bit of history



- 1816 : Nicéphore Niépce invent photography
- 1840s-1850s : Aimé Laussedat invent photogrammetry
- 1867 : Albrecht Meydenbauer uses the word ‘photometrographie’ for the first time

Jahrgang I.

Zusendungen bittet man zu richten an die Expedition, Oranienstr. 75. (Buchhdlg. v. C. Beelitz).
Insertionen die gespaltene Petitzeile oder deren Raum 2½ Sgr.

ZEITUNG-STEMPEL

Wochenblatt

herausgegeben von Mitgliedern des

Architekten-Vereins zu Berlin.

№ 14.

Bestellungen übernehmen alle Postanstalten und Buchhandlungen, in Berlin die Expedition Buchhandl. v. C. Beelitz Oranienstrasse 75.

Erscheint jeden Sonnabend. | Berlin, den 6. April 1867. | Preis vierteljährlich 18¾ Sgr.

Die Photometrographie.

Von A. Meydenbauer.

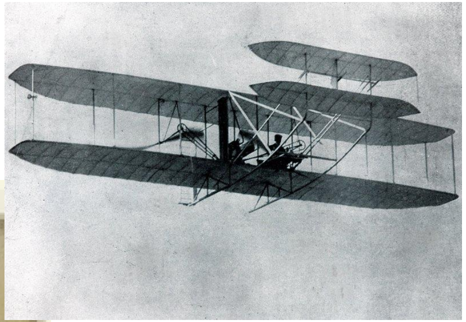
In einem Aufsatze im ersten Heft des laufenden Jahrganges der „Zeitschrift für Bauwesen“ ist die Theorie der Photometrographie, der Anwendung der Photographie zur Architektur- und Terrain-Aufnahme, niedergelegt. Wenngleich in der staunenswerthen Entwickelung der Photographie die Praxis der Theorie immer um einen grossen Schritt voraus war, so verhält es sich doch bei dieser neuen

haupten: die Nähmaschine taugt Nichts, weil sie den ganzen Rock nicht allein fertig macht.

Die Photometrographie wird sich hauptsächlich da nützlich machen, wo die bisherigen Aufnahme-Methoden einen unverhältnissmässigen Aufwand an Zeit, Geld und Arbeitskräften verursachten. Wir behaupten sogar, eine Menge sehr nothwendiger Architektur- und Terrain-Aufnahmen wurde aus

A tiny bit of history

- 1903-12-17 : First flights by the Wright brothers



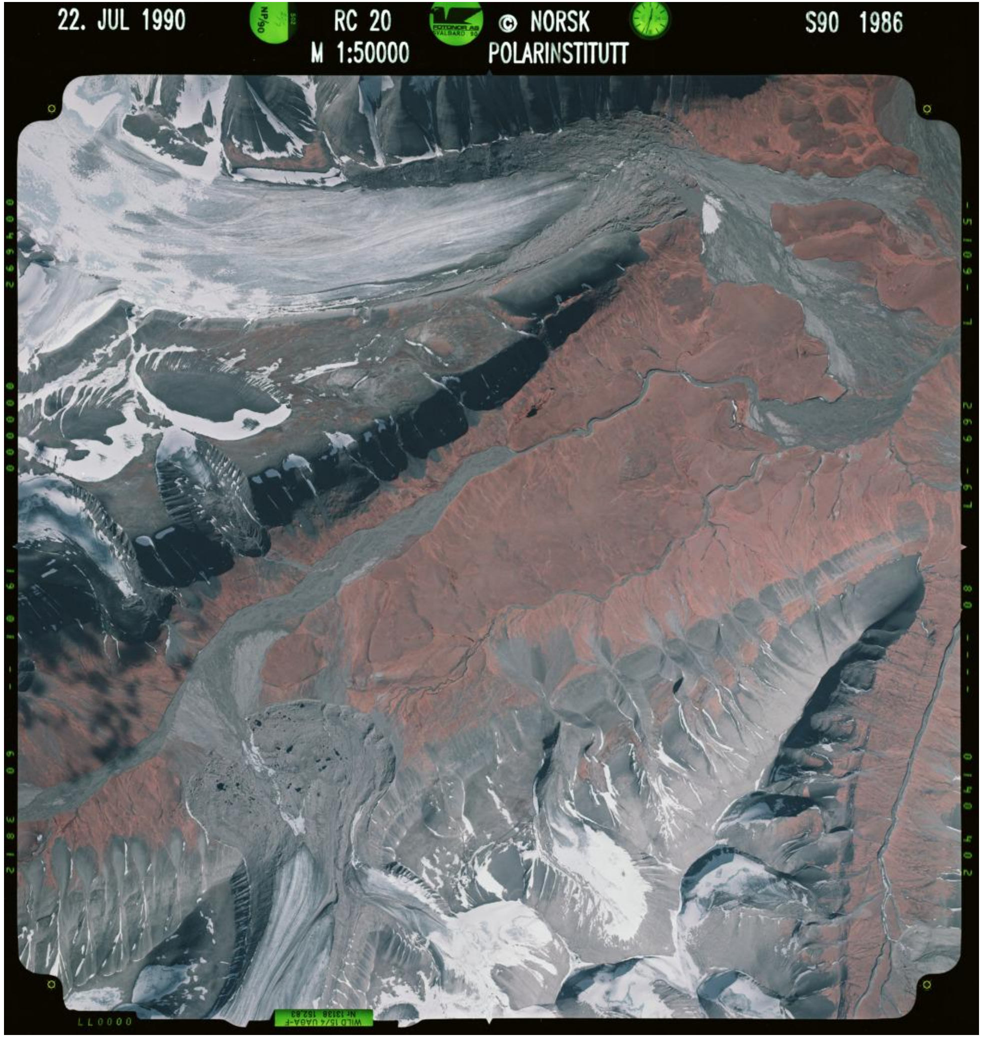
- 1921 : Fairchild's aerial survey, Manhattan Island, New York City
- 1920s-1930s : first photogrammetric aerial surveys

A tiny bit of history

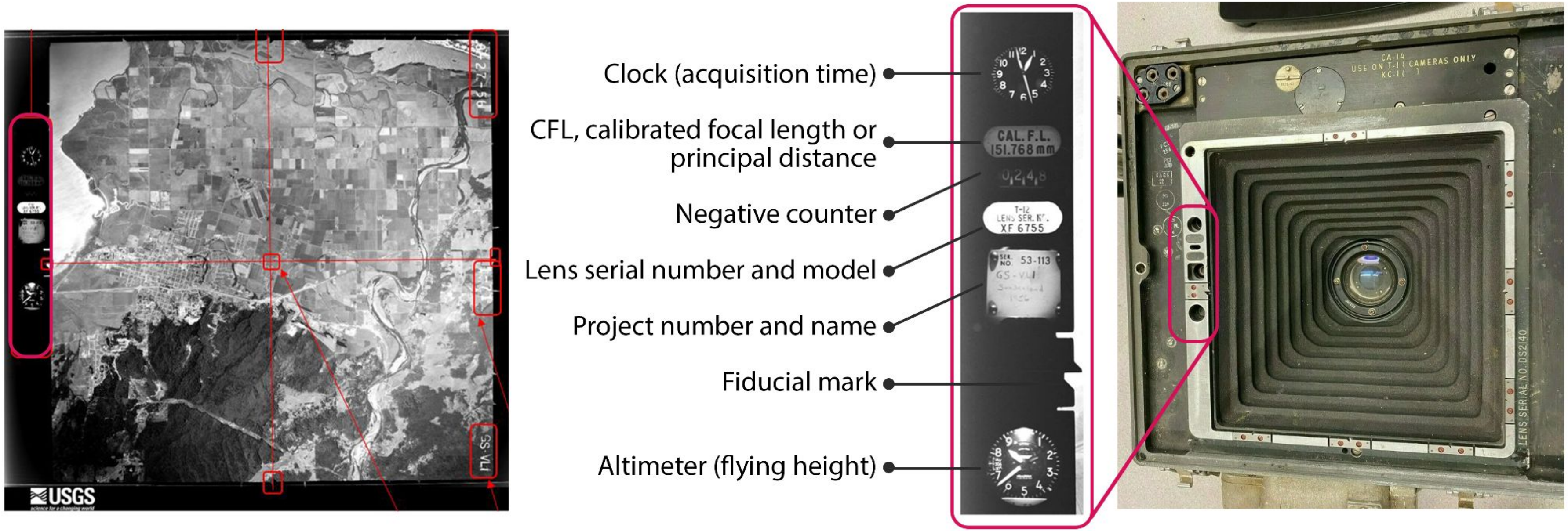
- Plane technology evolves quickly during WW2 (to perform accurate bombing runs on London/Germany)
- A lot of planes are left unused after the war and european geographical institutes place cameras by the bomb shute of planes to use them as surveying planes.

Photogrammetric plane


3218/3518


22. JUL 1990
RC 20
M 1:50000
© NORSK
POLARINSTITUTT
S90 1986

A tiny bit of history



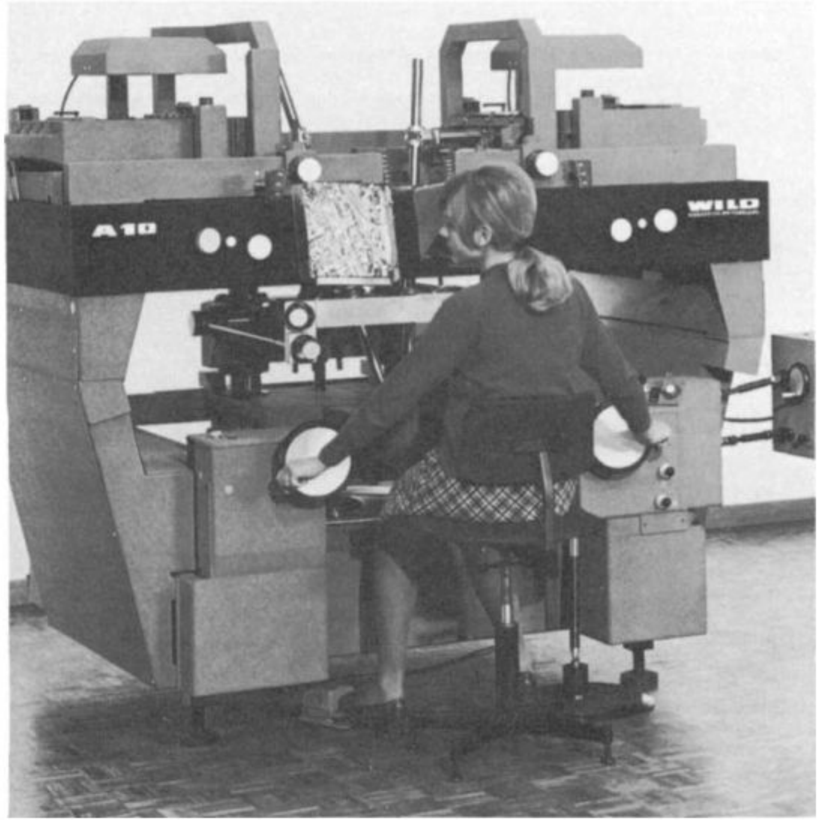
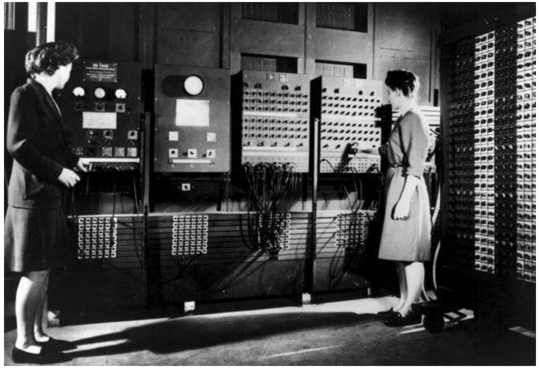


Figure 7.15 Wild A-10 stereoplotter. (*Courtesy*: Wild Heerbrug)

A tiny bit of history



- Computers enter the scene (see ENIAC on the right)
- Slowly, photogrammetry gets computerized
  - First with digitizing analog photographs (1990s-2000s)
  - Then with digital photography

ure from motion

:ed camera relative

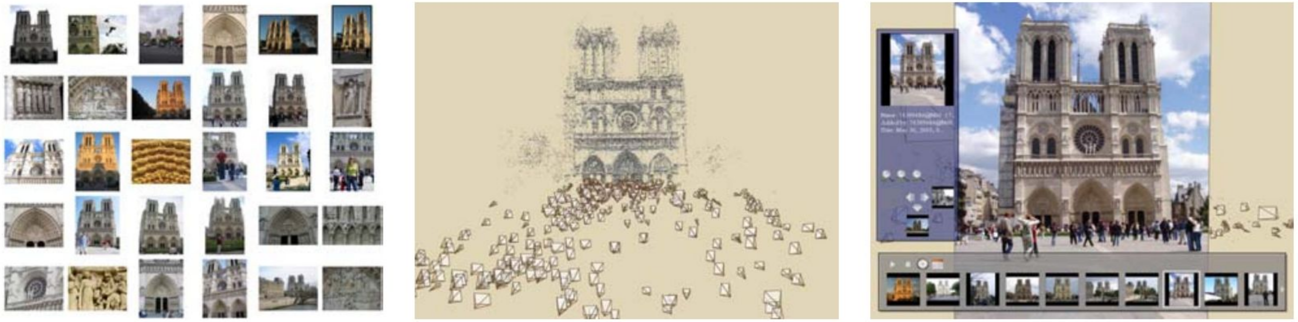
Scale-Invariant

st to provide enough

Figure 13. This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640 × 315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.

Transformation

- 1991 : Koenderink and Van Doorn, *Affine structure from motion (**SfM**)*
  - Introduces the mathematical backbone of automated camera relative orientation
- 2004 : Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*
  - Introduces the **SIFT** feature point algorithm, the first to provide enough high quality tie points for SfM
- 2006 : Snavely, Seitzand and Szeliski, *Photo tourism: exploring photo collections in 3D*
  - Implements SfM at scale, providing an easy to use tool to compute camera calibration and orientation : **Bundler**



(a) (b) (c)

Figure 1: Our system takes unstructured collections of photographs such as those from online image searches (a) and reconstructs 3D points and viewpoints (b) to enable novel ways of browsing the photos (c).

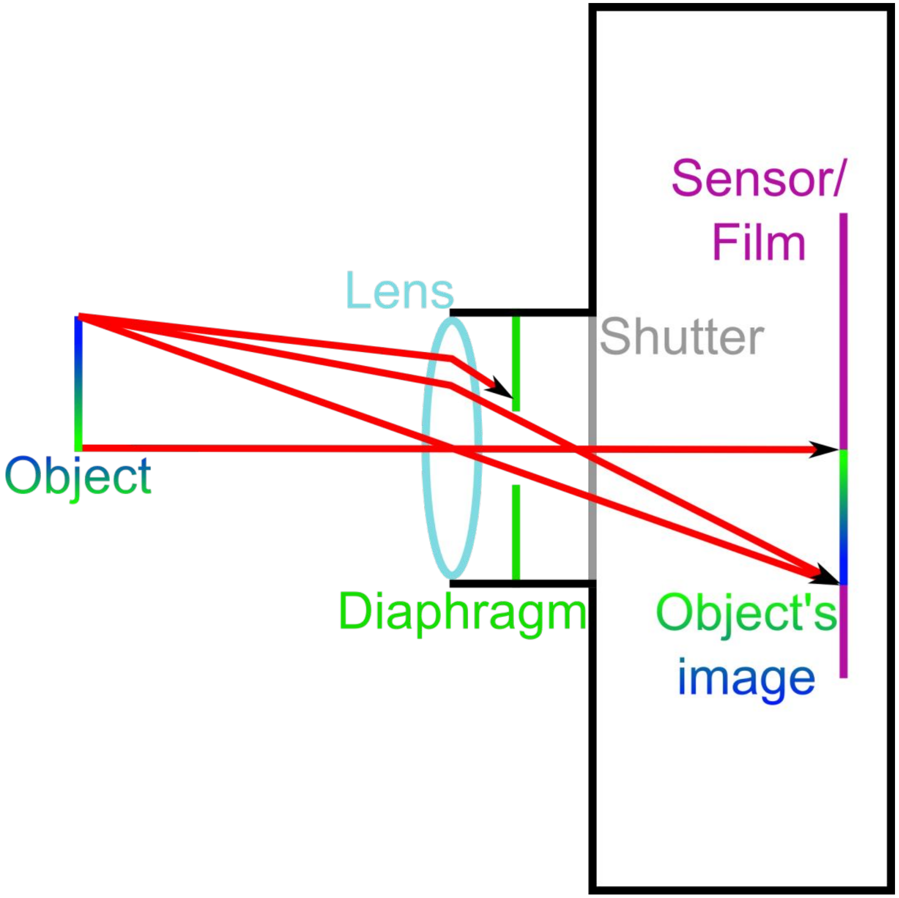
Transformation

- 1991 : Koenderink and Van Doorn, *Affine structure from motion (**SfM**)*
  - Introduces the mathematical backbone of automated camera relative orientation
- 2004 : Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*
  - Introduces the **SIFT** feature point algorithm, the first to provide enough high quality tie points for SfM
- 2006 : Snavely, Seitzand and Szeliski, *Photo tourism: exploring photo collections in 3D*
  - Implements SfM at scale, providing an easy to use tool to compute camera calibration and orientation : **Bundler**
- 2010 : Furukawa and Ponce, *Accurate, dense, and robust multiview stereopsis*
  - Build upon Bundler with tools to create dense point clouds from a set of oriented and calibrated images: **PMVS/CMVS**

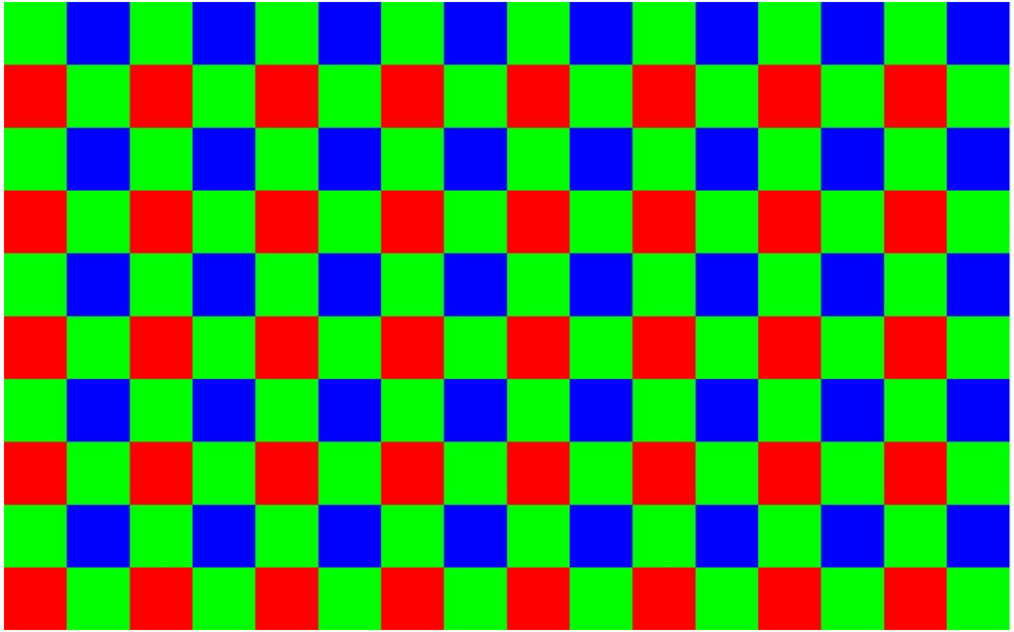
Fig. 1. Overall approach. From left to right: A sample input image, detected features, reconstructed patches after the initial matching, final patches after expansion and filtering, and the mesh model.

A quick intro to cameras

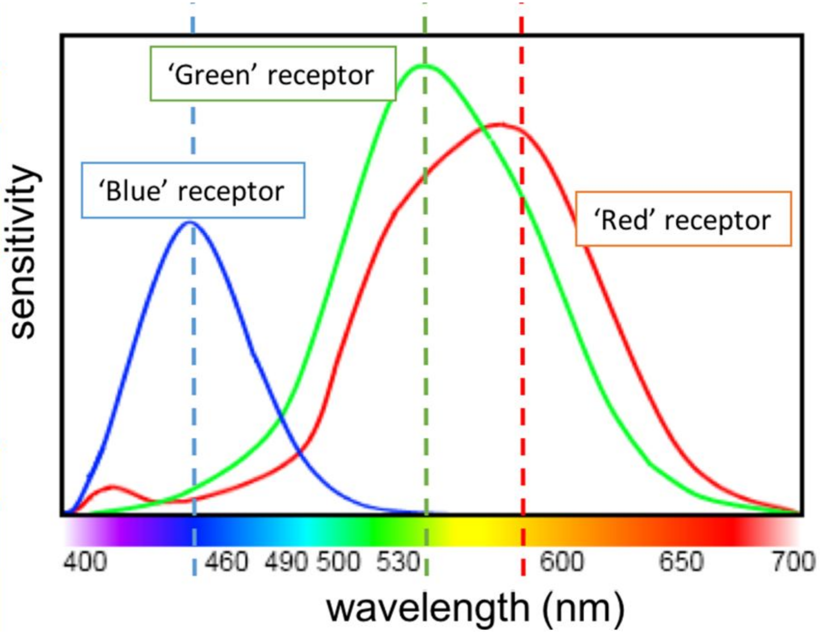
Payloads - RGB



Payloads - RGB



It’s a grid of photosensitive sensors with a bayer matrix (this thing above) filtering the colours.

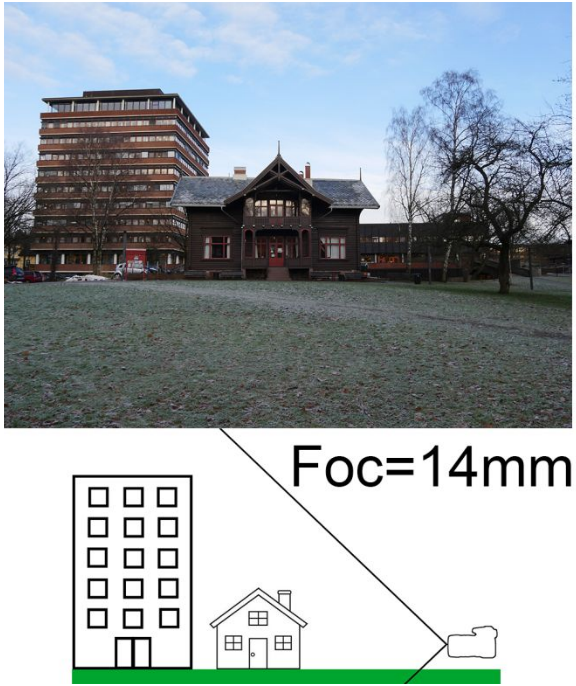


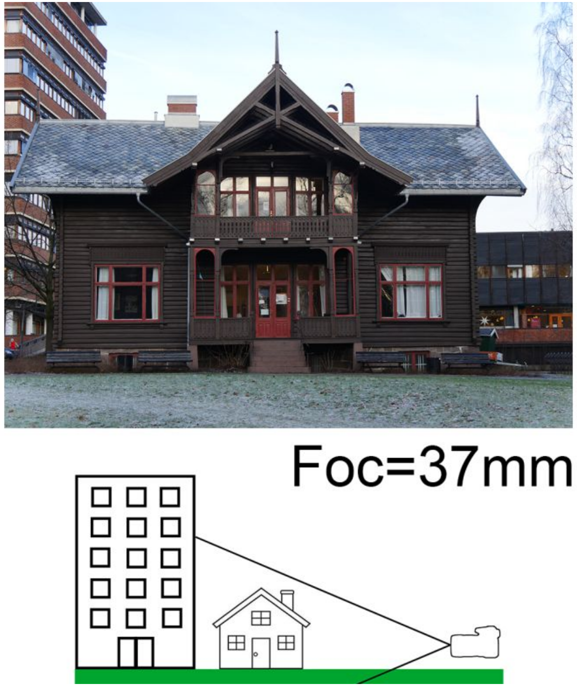
Human eye typical sensitivity

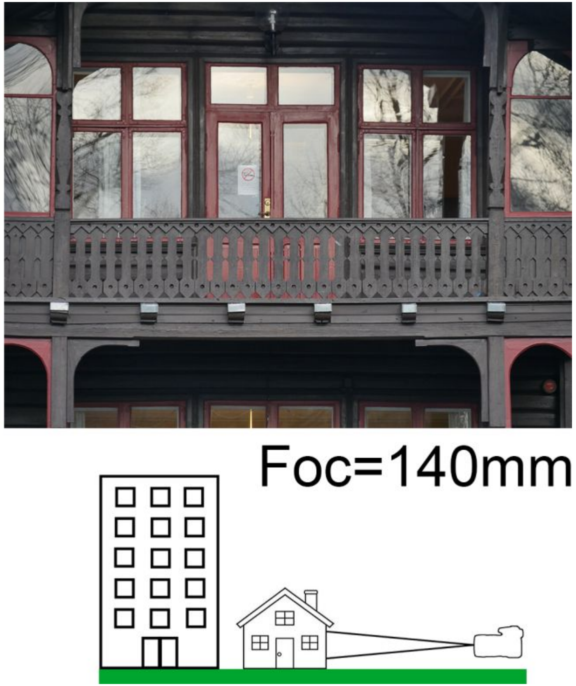
Camera architecture

1. **Focal length** (in mm)
   - Influence the zoom level (a longer focal length will result in a narrower FOV)

Focal length (“zoom”)







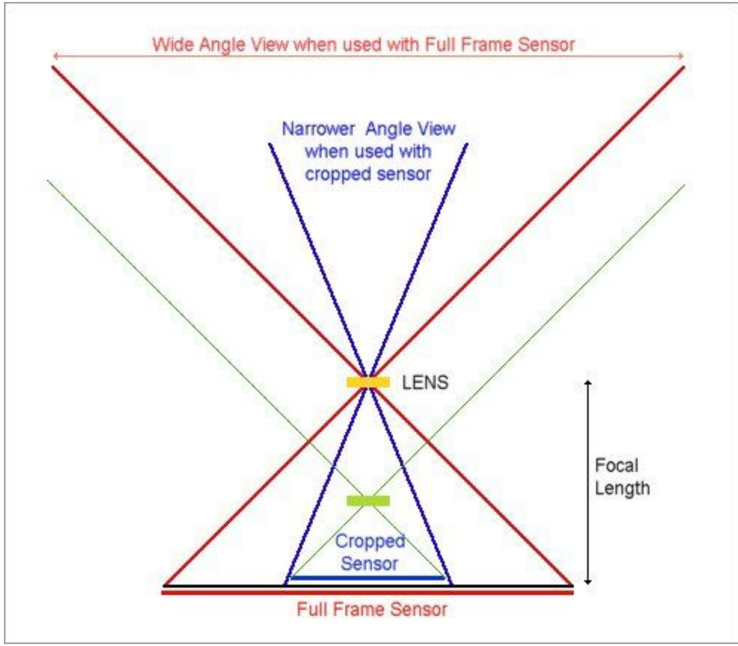
Focal length (“zoom”)

Focal length (“zoom”)

A focal length will give different field-of-view (FoV) with different sized sensor.

“35mm Foc eq” is the normalized Focal length for a given FoV on a *Full frame* sensor.

μ4/3 sensors are ½ the size of *Full frame*, so a 25mm lens would have the same FoV as a 50mm on a *Full frame* sensor. It would be called **50mm focal eq**

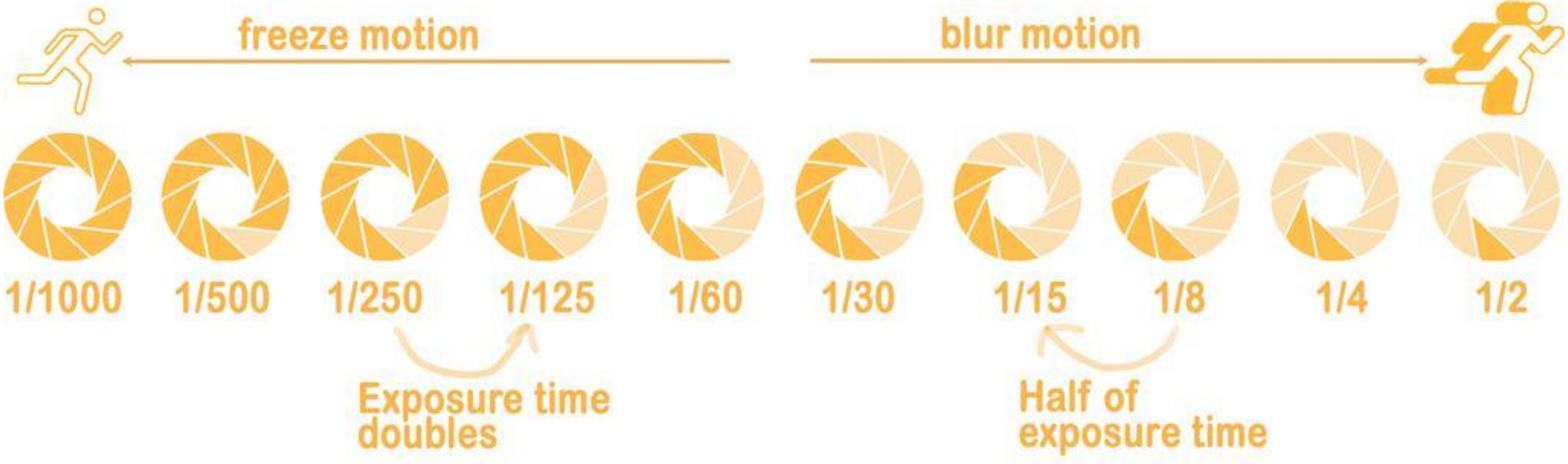


Camera architecture

2. Exposure time

- Duration the light is allowed through the lens to the film/sensor

Exposure time



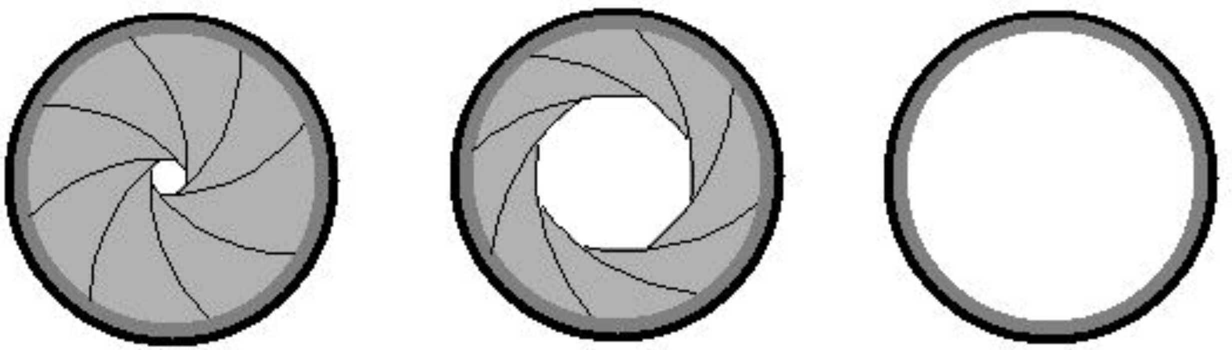
Fast exposure reduces blur from camera and object motion, but allows less light to reach the sensor. One must find the right balance.

Camera architecture

3. Aperture (f-number, f-stop)

- influences amount of light going through the lens and depth of field
- also influences vignetting (distribution of light across the lens)
- aperture = focal length / diameter of diaphragm

Aperture (F number)



Aperture : f/20
Exposure time : 1/13s

Aperture : f/10
Exposure time : 1/50s

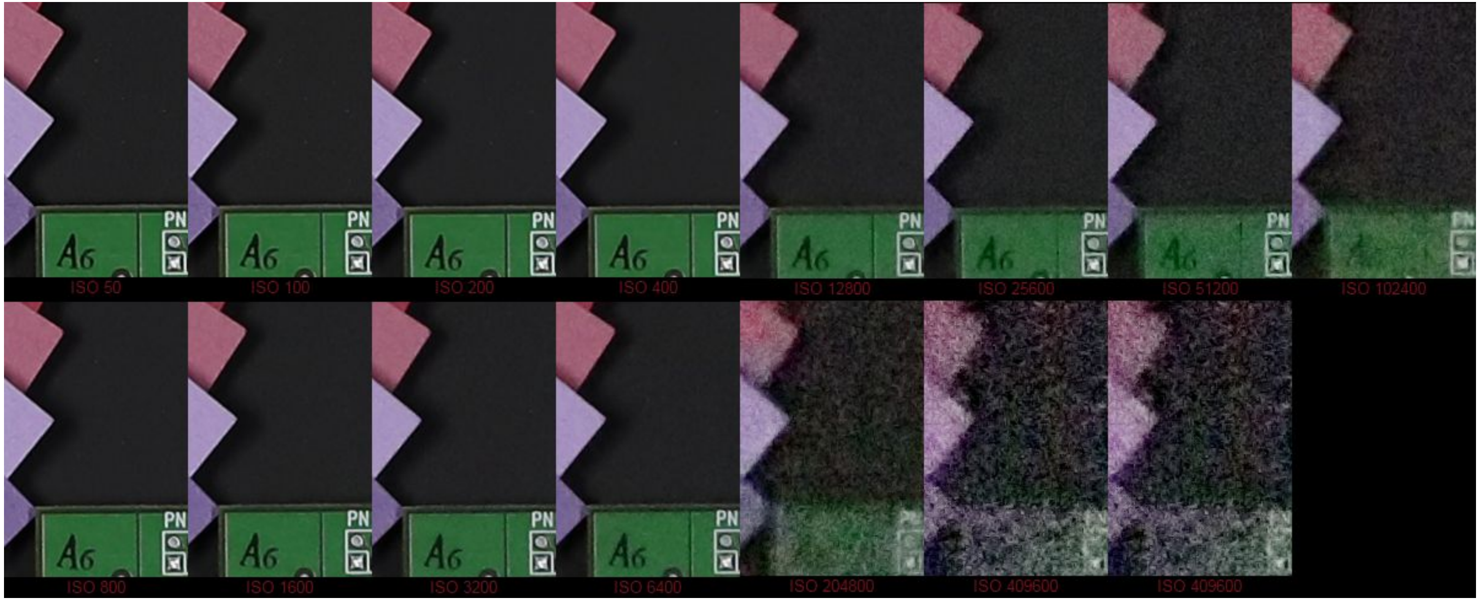
Aperture : f/5
Exposure time : 1/200s

Camera architecture

4. ISO / sensitivity

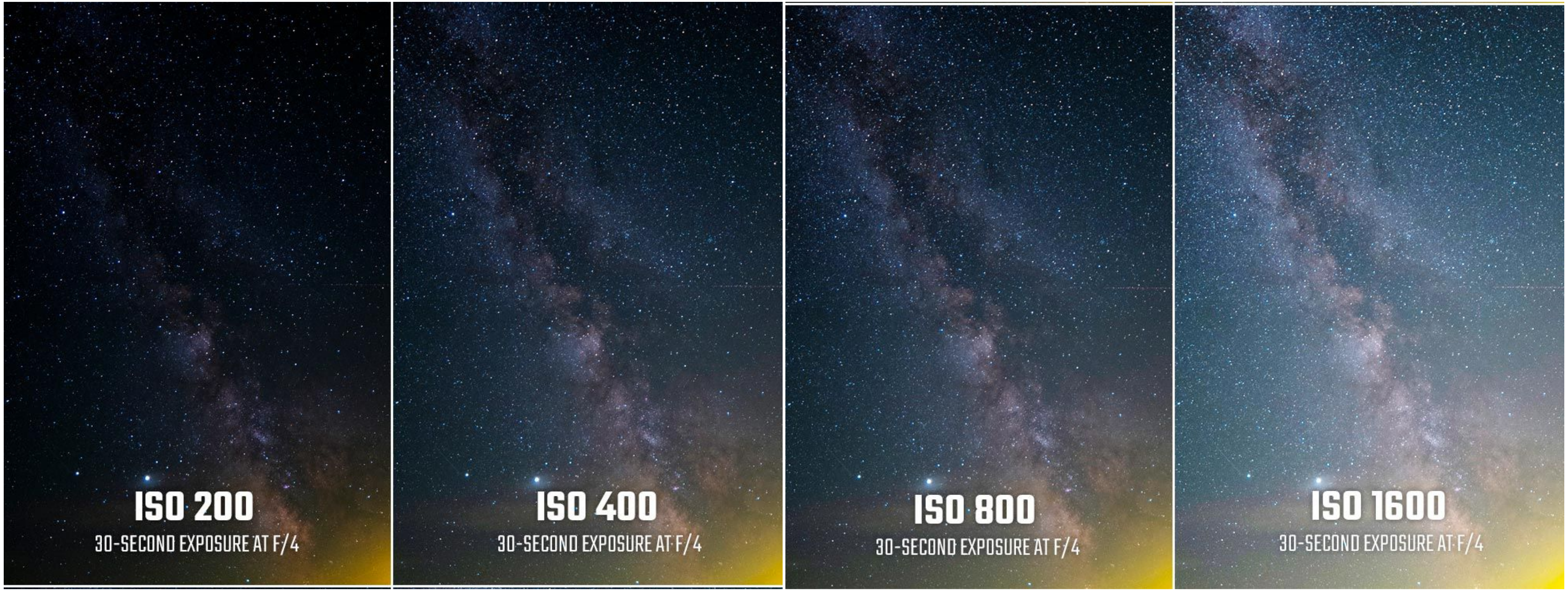
- how reactive/sensitive the film/sensor is to light stimulation
- a higher ISO number indicates higher camera sensitivity, so less light is needed to take a picture
- higher iso numbers mean with more noise (think nighttime photography)

ISO (sensibility)



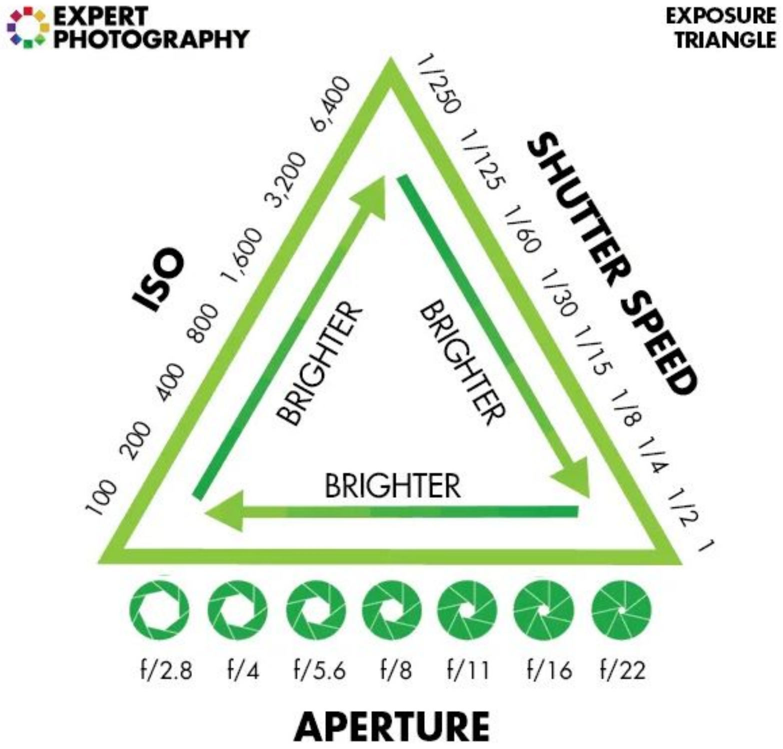
High ISO means a brighter image, but electronic noise comes along.

Payloads - RGB

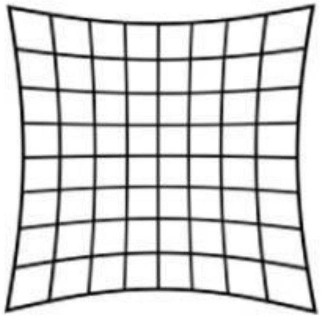


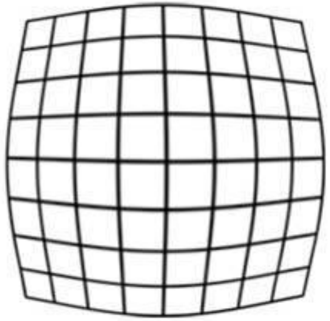
source: https://astrobackyard.com/wp-content/uploads/2022/02/best-ISO-for-astrophotography.jpg

Payloads - RGB



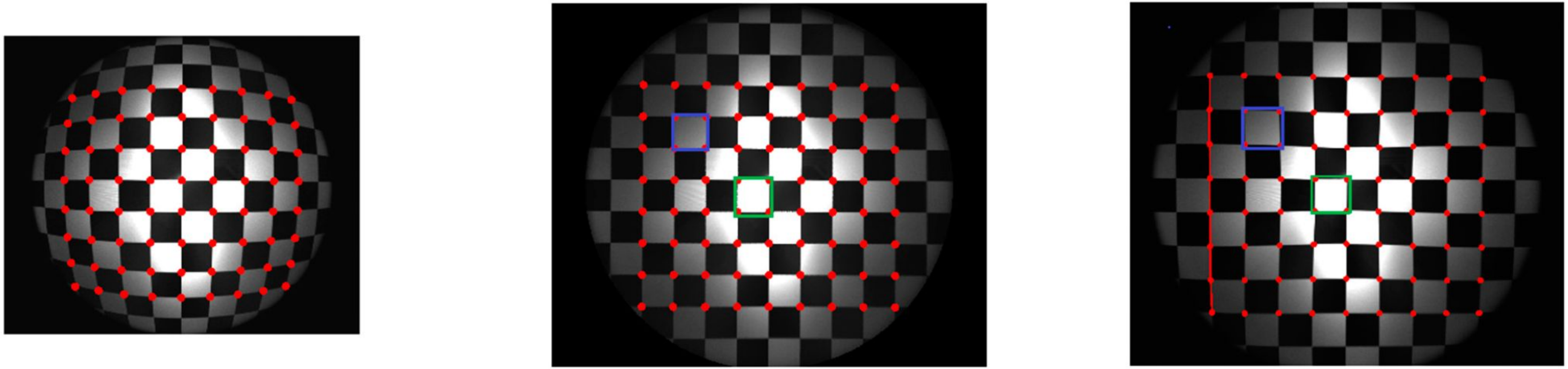
Geometric Distortion





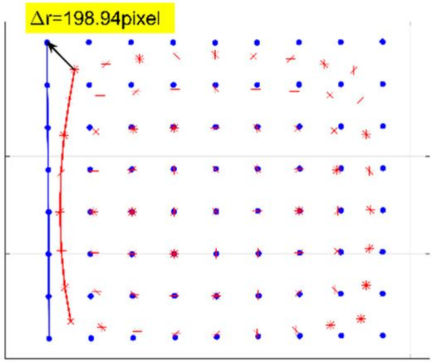
- Results from imperfections of the optical system and flatness of the sensor
- More pronounced in zoom and wide-angle lenses
- Most extreme with fish-eye lenses
- Most noticeable effect is the abnormal curvature of straight lines

Payloads - RGB



(a) (b) (c)

can be corrected by calculating a distortion matrix for the lens



source: https://doi.org/10.3390/s21010209

Payloads - RGB

Payloads - RGB

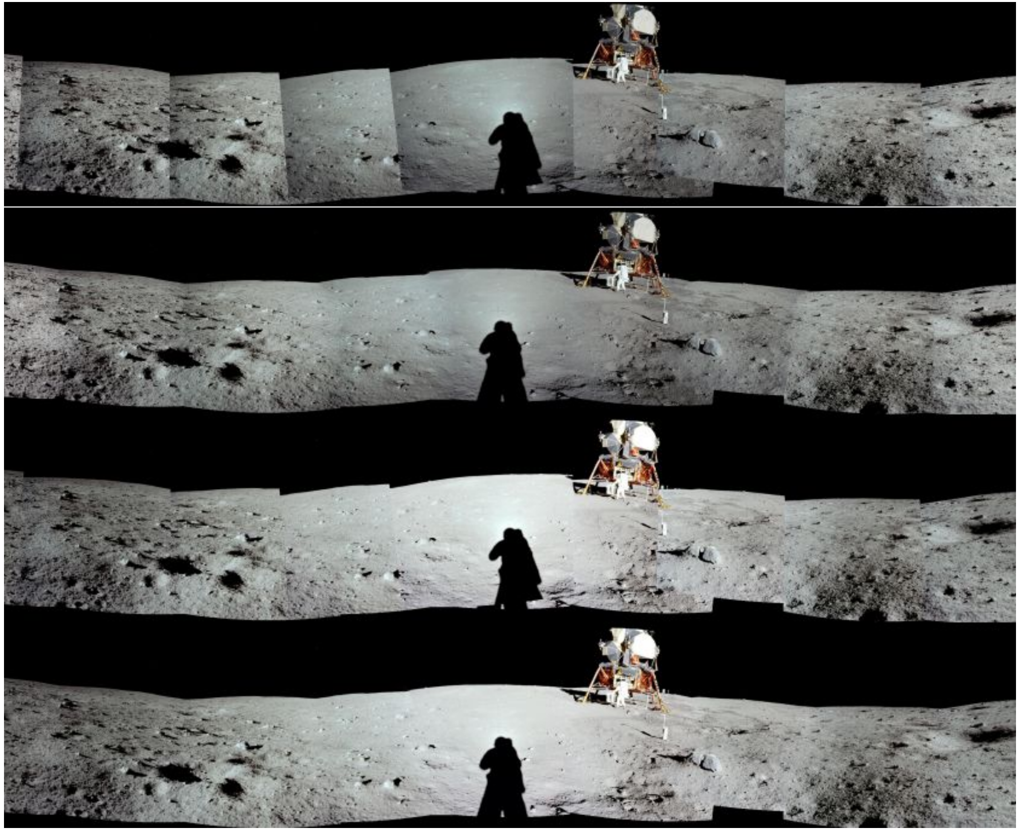
Chromatic Distortion

- Failure of a lens to focus all colours/wavelengths to the same pixel/point
- Caused by the refractive index of the lens varying with the different wavelengths of the light
- Linked with focal length and focusing

Vignetting

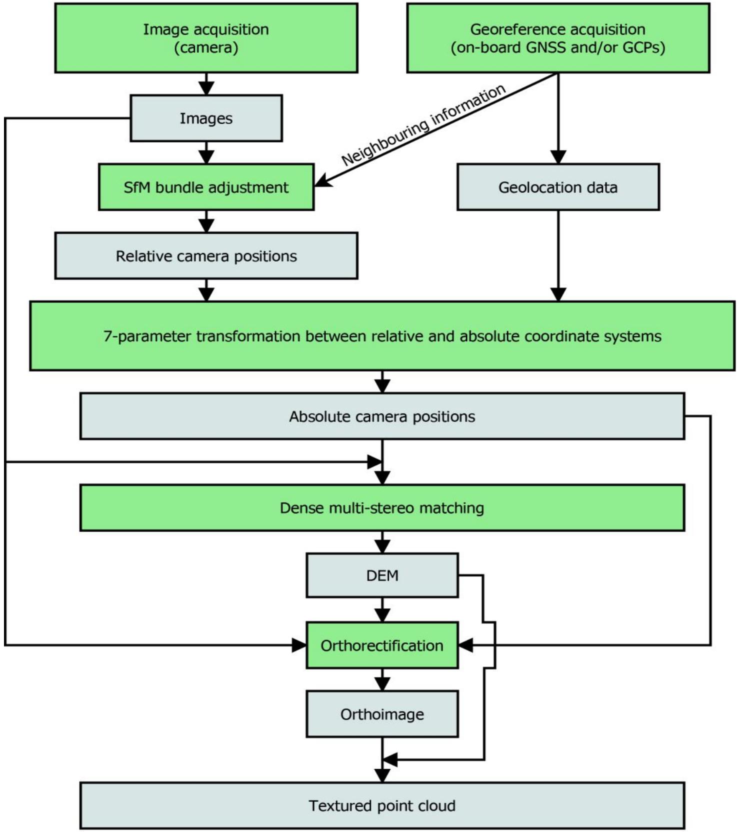
Dark corners are annoying when making mosaics!

Vignetting

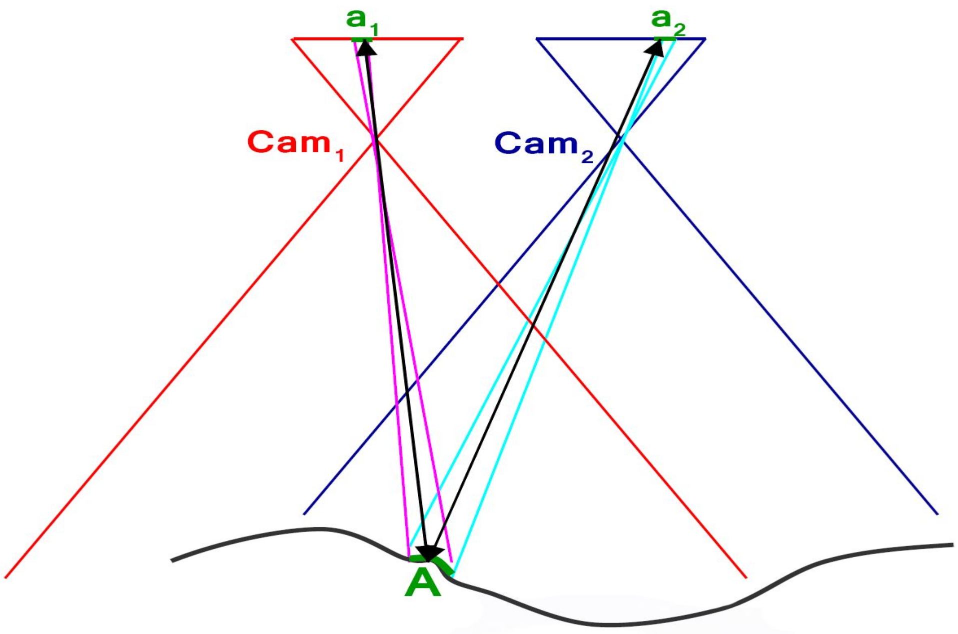


Photogrammetry

Photogrammetry



Photogrammetry



Known information:

1. Internal characteristics of the camera(s)
2. The location of where the pictures were taken (Cam_1, Cam_2)
3. How the camera was oriented in space (e.g. nadir)

Using object *A*, identified in both images, we compute the interaction of the projective rays from where *A* is located in each image.

Doing this on multiple points means we can reconstruct the 3D shape of the photographed object or scene.

Inputs

- Images
- Camera Information
- Location information
  - ➢ Onboard GNSS
  - ➢ Ground Control Points (GCPs)

IMAGE ACQUISITION

Two methods for acquisition geometry

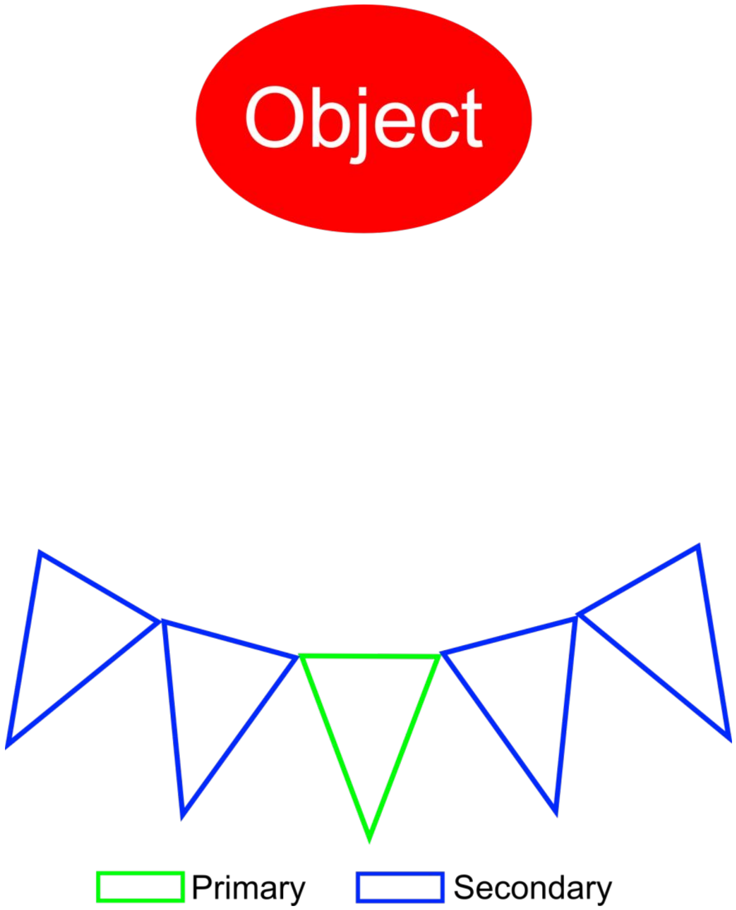
1. Convergent : to be used for creating a model of a 3D object

2. Parallel : to be used to create a model of a planar object

Convergent method

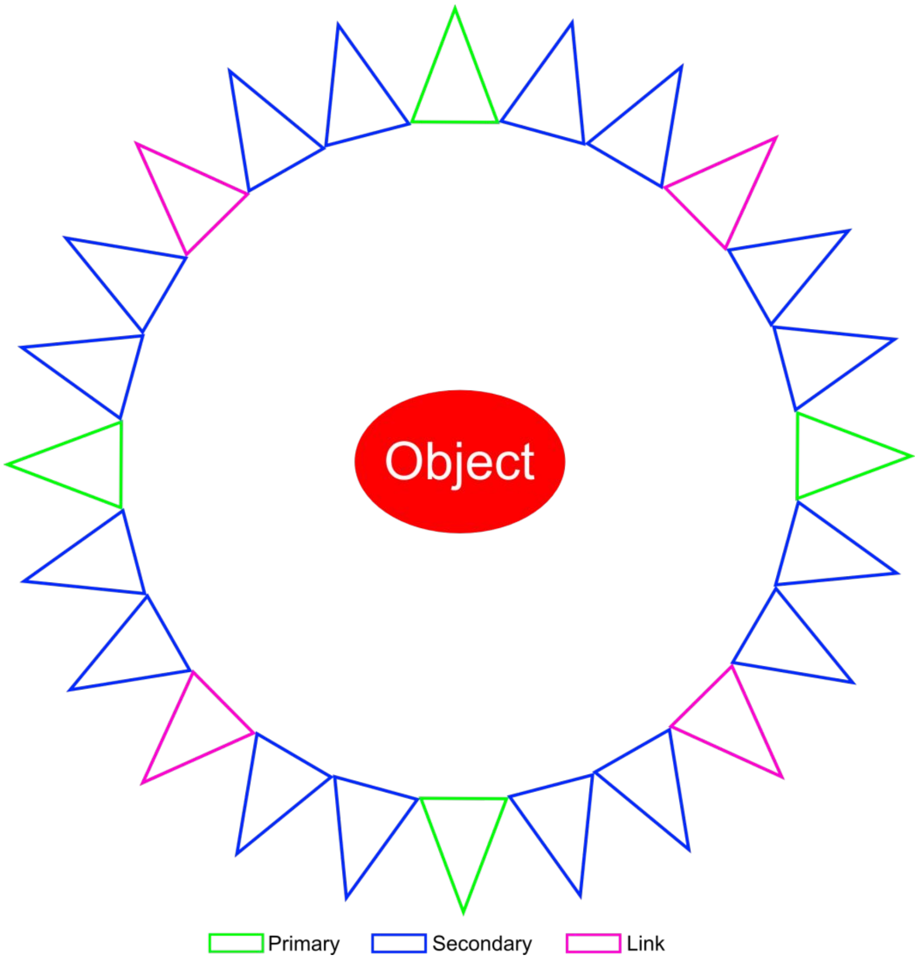
- Pictures are taken aiming at the same point in space / object of interest

- The primary picture in the middle uses the secondary ones to get the 3D information

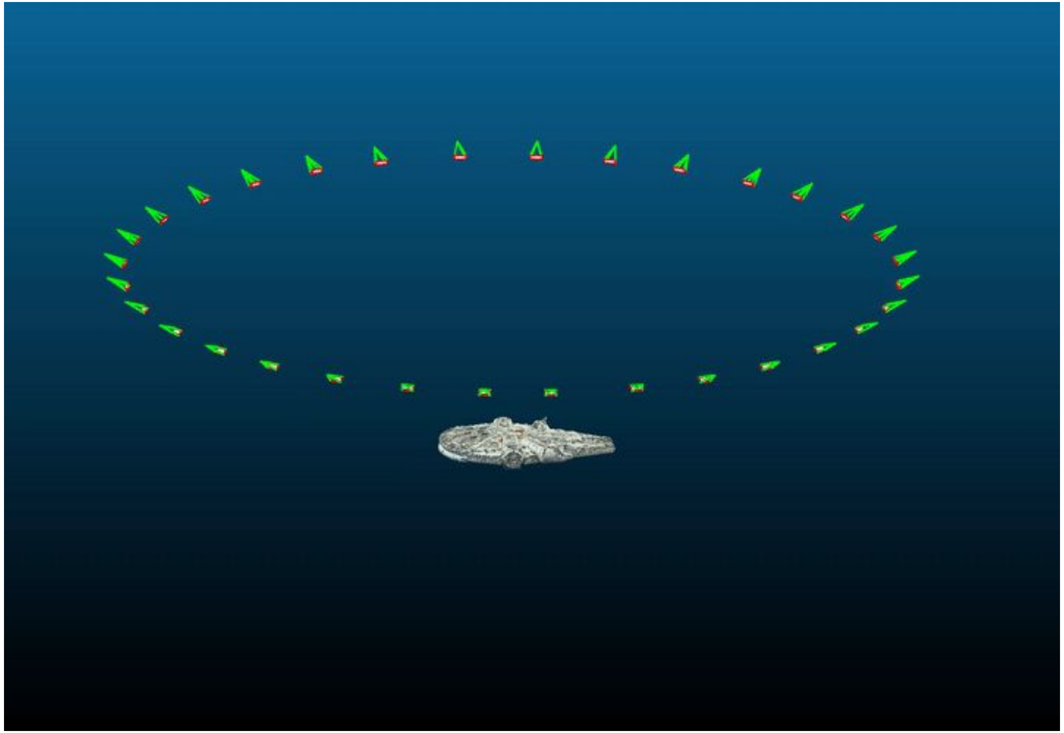


Convergent 360°

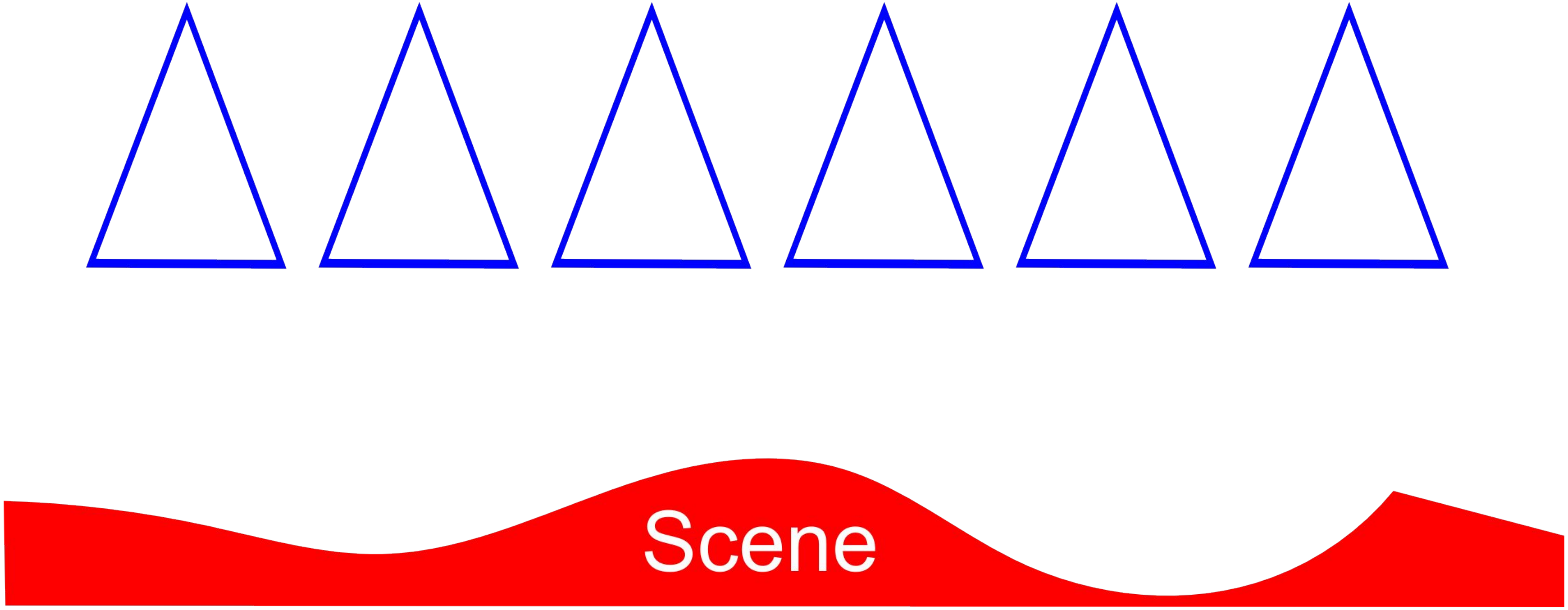
- Having linking images ensures robust geometrical link between the different points of view
- Aim at max 10° between pictures (min 36 pictures to a circle) to ensure both computation of tie points and image correlation work
- A setup with several circles taken from different "altitudes" is a logical next step



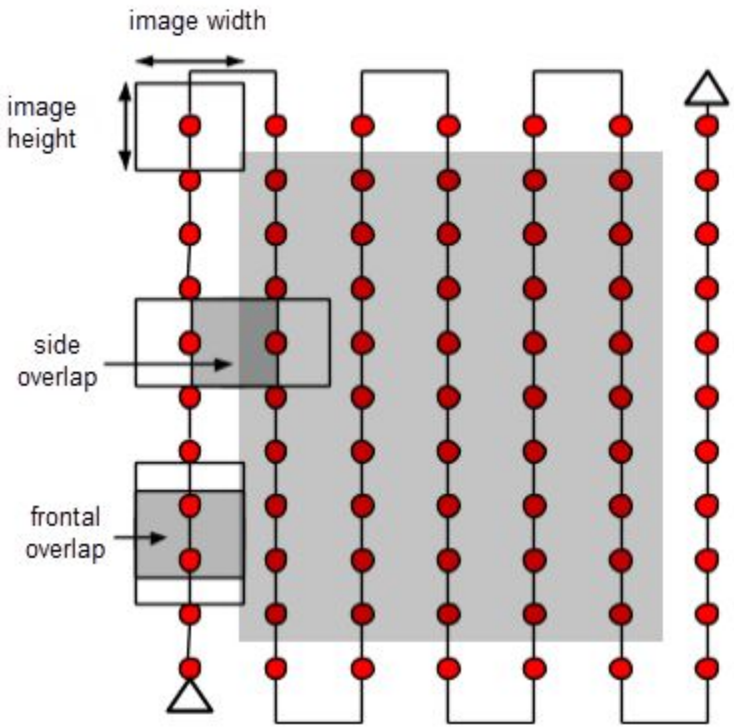
Convergent 360°



Parallel method



Parallel method / Lawnmower pattern



- Used in most aerial surveys
- Most useful when area of interest is planar
- Not one image represents the whole scene
- Images are tiles fo the scene with overlaps so that each point in the scene is in more than one image.

“POINTS” in photogrammetry

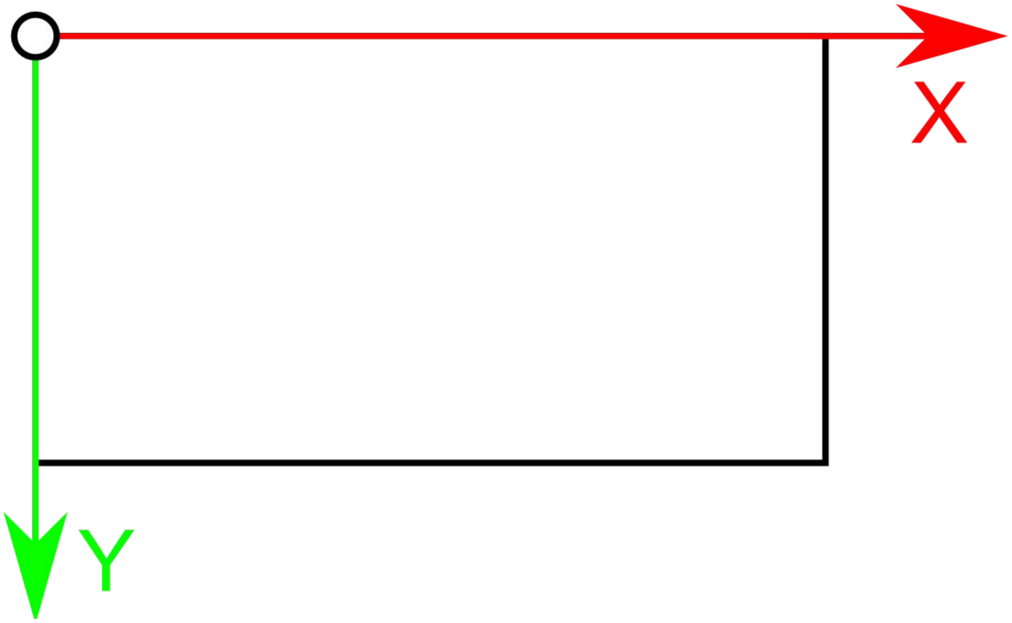
- **Tie points** - used to calculate how images are related to each other
- **Control points** - points visible in images with known ground coordinates
  - **Ground control points** (GCPs) - used in post-processing to reference the model
  - **Check points** - used to validate accuracy of model, camera alignment, optimisation procedures etc. (other subset of GCPs)

Coordinate systems

We deal with several coordinate systems :

1. Image coordinates (2D, pixels)
2. Camera coordinates (3D space, m)
3. 3D coordinates, either relative space or georeferenced coordinates (3D space, m)

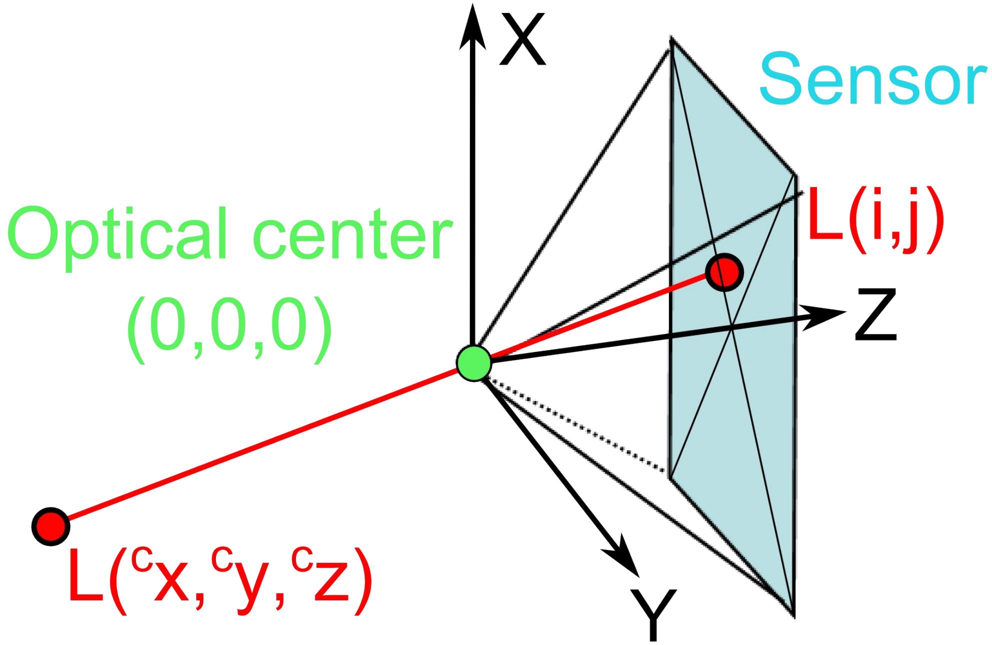
1. Image coordinates (2D, pixels)



Sz(Y) – 3000 pixels

Sz(X) – 4000 pixels

2. Camera coordinates



3. Relative space and world coordinates

- Are metric 3D spaces
- Are defined by the user :
  - Relative space is usually one of the camera spaces
  - World coordinates are in respect of control points
    - World space can be a local coordinate system not linked to the Earth. For instance, a laboratory setup.

CALIBRATION AND ORIENTATION

Calibration and orientation

$$\begin{pmatrix} i_{L,K} \\ j_{L,K} \end{pmatrix} = \Im\left(\pi\left(R_K * \left(\begin{pmatrix} {}^{rs}x_L \\ {}^{rs}y_L \\ {}^{rs}z_L \end{pmatrix} - C_K\right)\right)\right)$$

Calibration and orientation

$$\begin{pmatrix} i_{L,K} \\ j_{L,K} \end{pmatrix} = \Im\left(\pi\left(R_K * \left(\begin{pmatrix} {}^{rs}x_L \\ {}^{rs}y_L \\ {}^{rs}z_L \end{pmatrix} - C_K\right)\right)\right)$$

- L is an object.
- K is an image.
- $({}^{rs}x_L; {}^{rs}y_L; {}^{rs}z_L)$ are the coordinates of the object L in Relative space coordinates.
- C_K is the coordinates of the optical center of the camera K in Relative space coordinates.
- R_K is the rotation matrix from Relative space coordinates to Camera coordinates.
- π is the function projecting points in Camera coordinates to a canonical 2D space.
- $\Im$ is the function of the camera parameters (Focal, distortions, sensor size (in mm and pixels)...) converting canonically projected points into Image coordinates.
- $(i_{L,K}, j_{L,K})$ are the pixel coordinate of the object L in image K.

Calibration and orientation

$$\boxed{\begin{pmatrix} i_{L,K} \\ j_{L,K} \end{pmatrix}} = \boxed{\mathfrak{I}}\left(\boxed{\pi}\left(\boxed{R_K} * \left(\boxed{\begin{pmatrix} {}^{rs}x_L \\ {}^{rs}y_L \\ {}^{rs}z_L \end{pmatrix}} - \boxed{C_K}\right)\right)\right)$$

Position of object in image space

Camera K intrinsic parameters

2D projection from camera space to canonical space

Rotation matrix from relative to camera K space

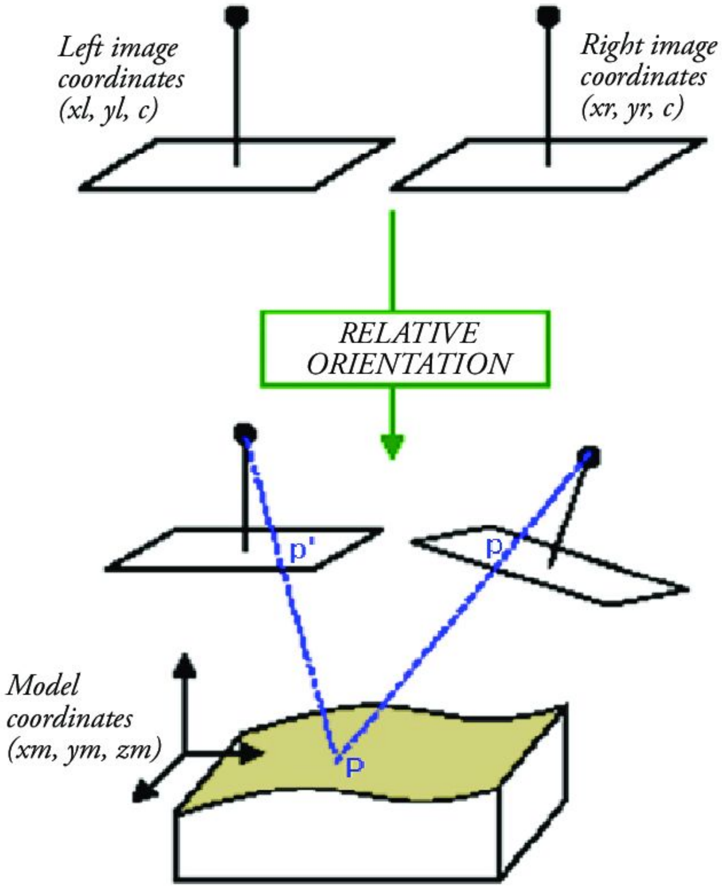
Position of object in relative space

Position of camera K in relative space

Calibration and orientation

1. Relative orientation of cameras to each other / calculation of tie points
2. Camera orientation -> how each camera relates to the real world
3. Camera calibration
4. Absolute orientation

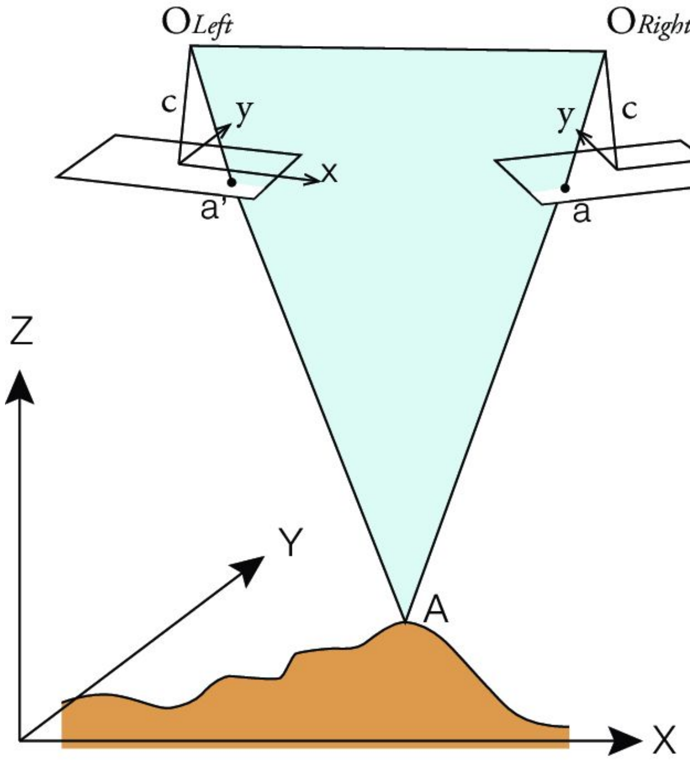
Relative orientation



The step in the process where the images are geometrically related to each other

source: Rosado, E.M.Q., 2018. *Introduction to applied photogrammetry and cartography for civil engineering*.

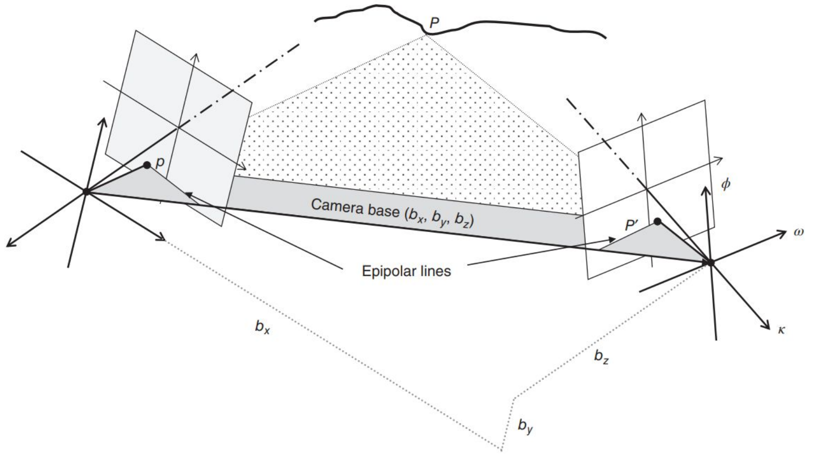
Relative orientation



- Uses tie points to establish the relationship between two images (with no knowledge of object)
- Based on coplanarity condition:
  - two images on a stereopair, the perspective centres of the two images and the object point lie on the same plane
- Assumes orientation and position of left image are both fixed.

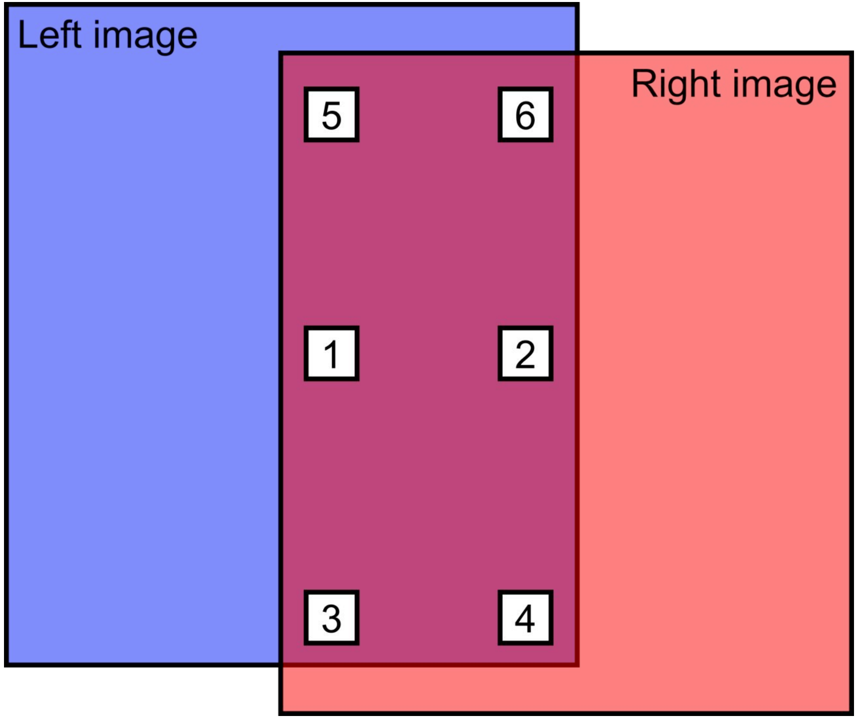
source: Rosado, E.M.Q., 2018. *Introduction to applied photogrammetry and cartography for civil engineering*.

Calibration and orientation



- Assume the image on the left (I_1) is fixed and we want to determine the relative position of the image on the right (I_2)
- b_y and b_z are translations require to correctly position the perspective centre of I_2
- ω, ϕ, and κ are the rotation angles that are needed to make the right image coordinate system parallel to the left image coordinate system

Tie points



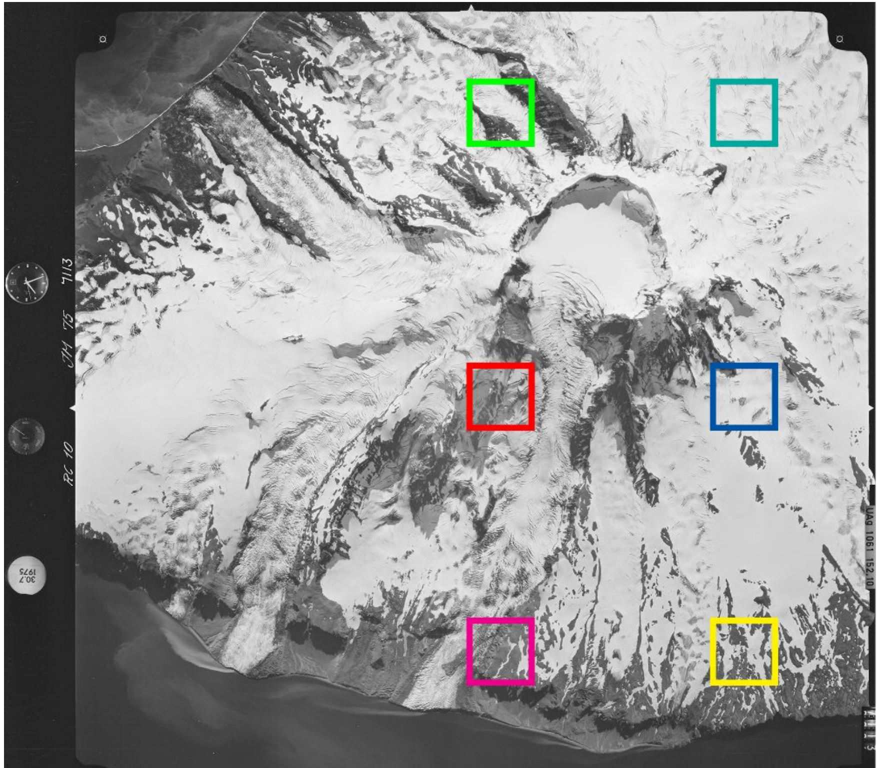
To calculate b_y and b_z, ω, ϕ, and κ:

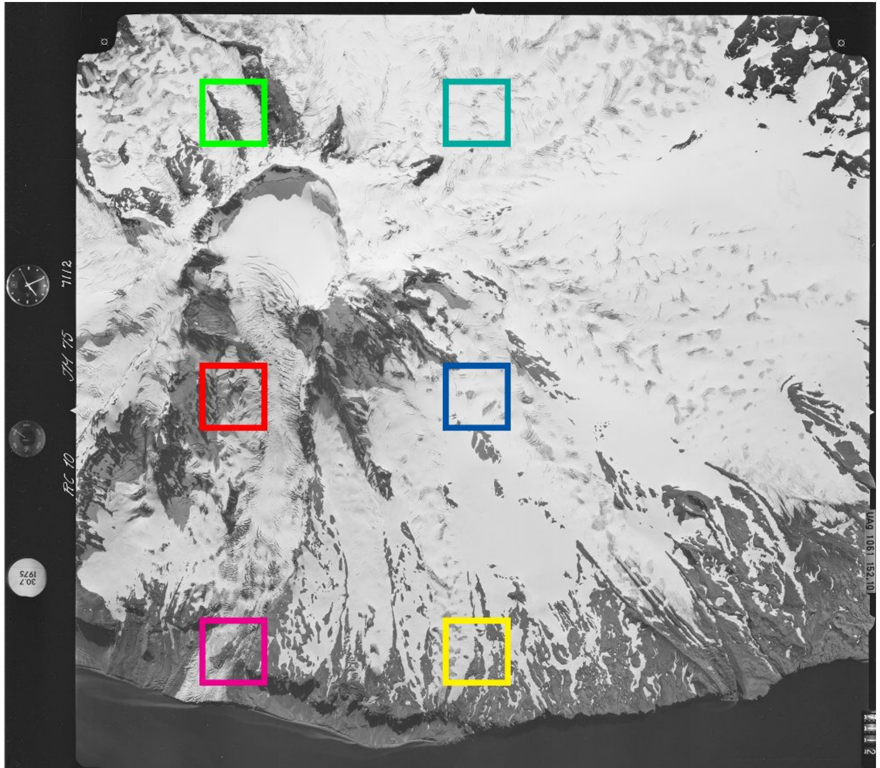
- each tie point pair is one observation
- we need at least 6 pairs to solve for these variables
- these points should be distributed across the six von Gruber locations
- historically, these points were manually selected

Tie points

- Used to determine relative orientation
- Points visible in 2 or more images
- Found manually or (nowadays) with automatic tie point detection algorithms (SIFT/SURF…)
- Usually very particular/distinct (easy to differentiate)

Tie points (historical – Von Gruber)





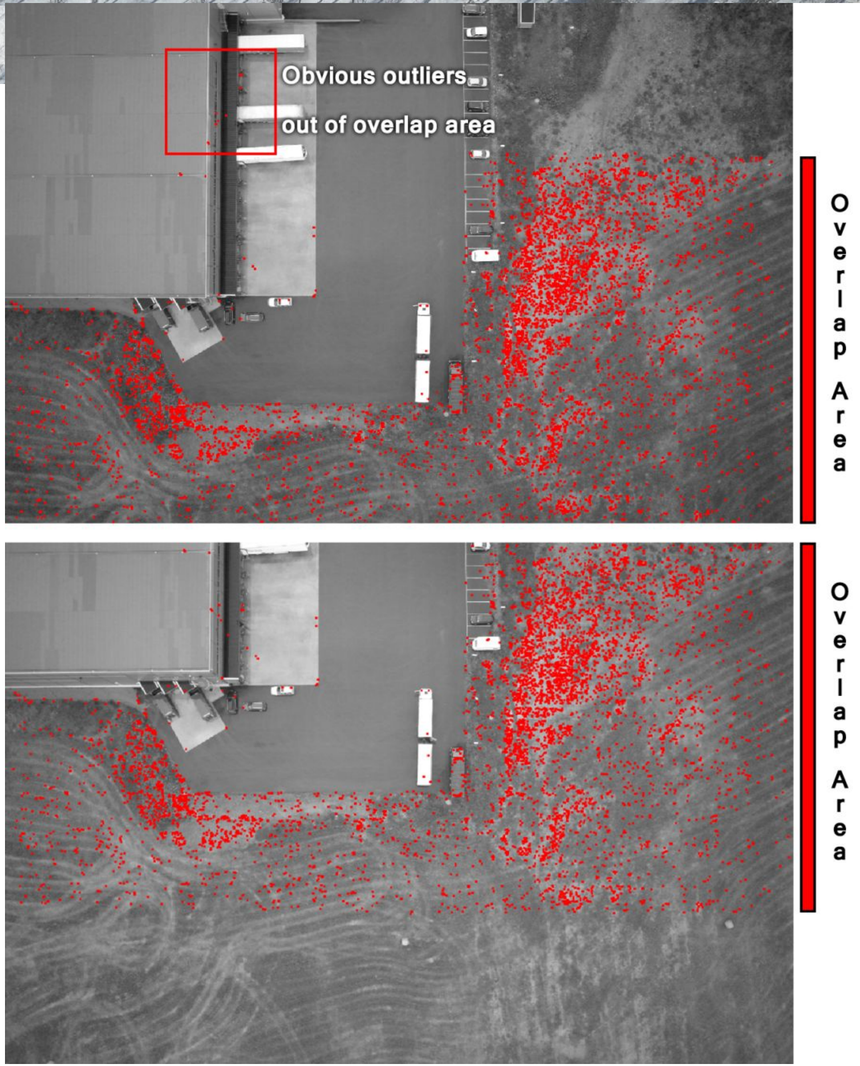
Tie points

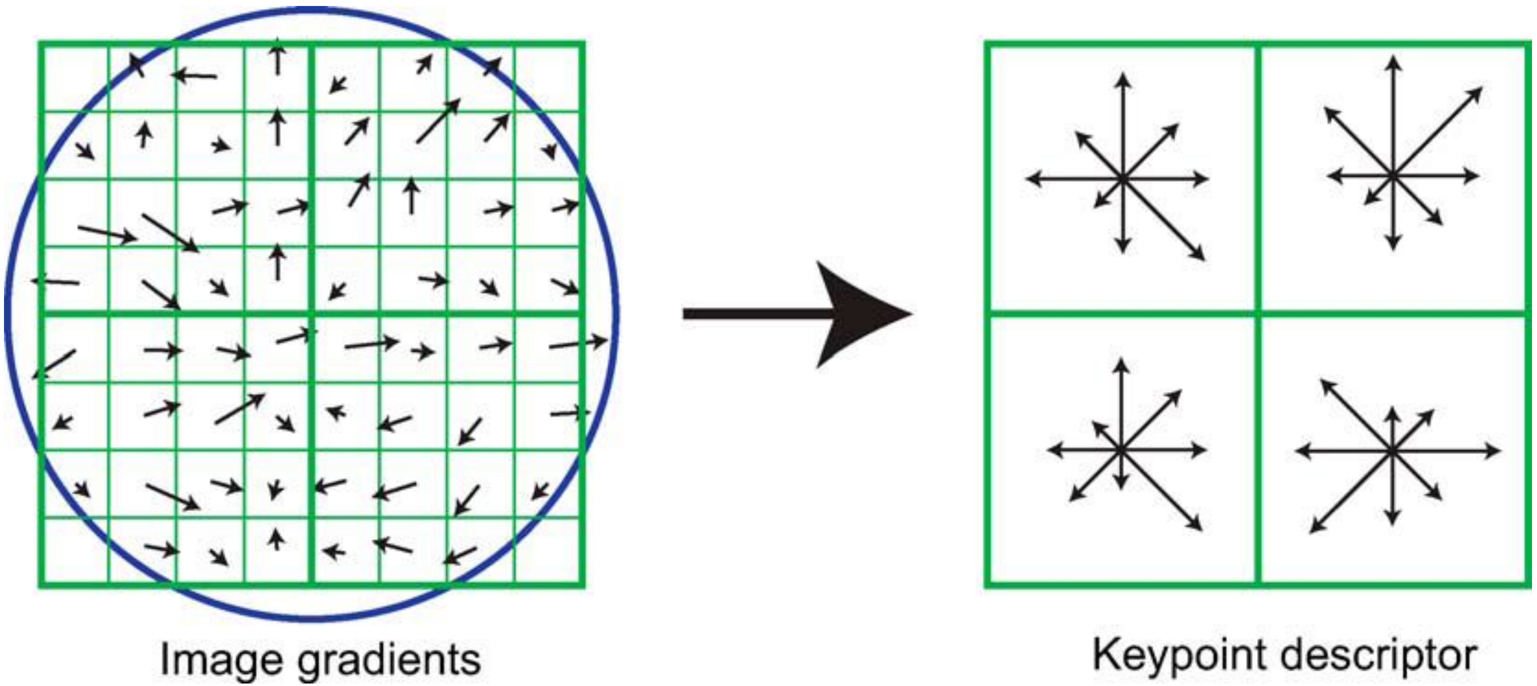
- Used to determine relative orientation
- Points visible in 2 or more images
- Found manually or (nowadays) with automatic tie point detection algorithms (SIFT/SURF…)
- Usually very particular/distinct (easy to differentiate)

Tie points (modern - SIFT)

- Improved computational power mean the number of tie points we can calculate are order of magnitudes higher than older manual methods.

- Why is there an homogeneous distribution?

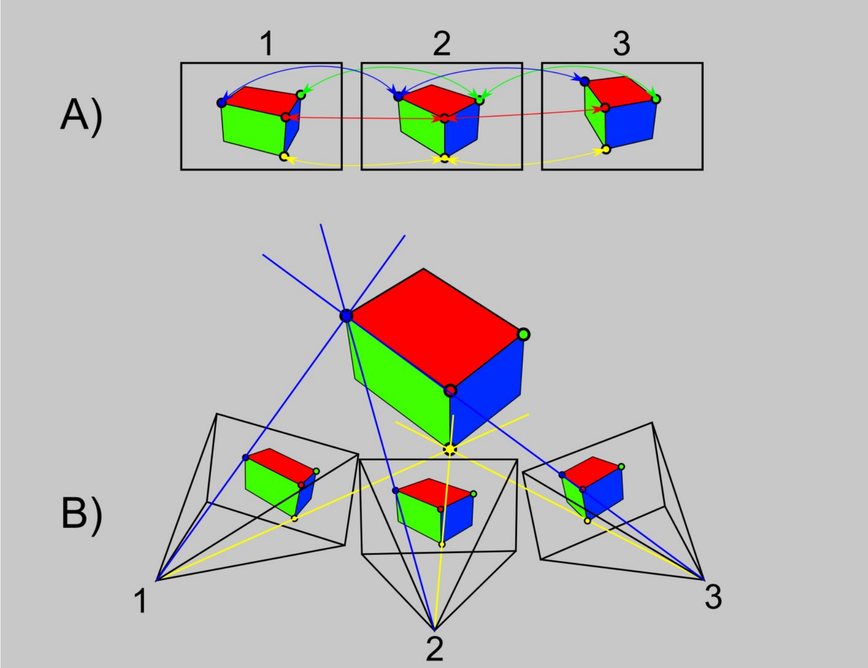




Attribute a Keypoint Descriptor to them

Match the descriptors for each image together to create Tie Points

Tie points to camera positions



We need to compute for each camera :

- The rotation matrix (or the three rotations)
- The position of the center of projection (in X, Y and Z)

$$\begin{pmatrix} i_{L,K} \\ j_{L,K} \end{pmatrix} = \mathfrak{I}\left(\pi\left(\boxed{R_K} * \left(\boxed{\begin{pmatrix} {}^{rs}x_L \\ {}^{rs}y_L \\ {}^{rs}z_L \end{pmatrix}} - C_K\right)\right)\right)$$

rotation matrix from relative to camera coordinates

object in relative space

First camera

We first work in relative space, it is easy to define the first camera such as :

$$ {}^{rs}R_{c_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad {}^{rs}PC_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} $$

Second camera

- One of the translation between the two cameras can be fixed, to fix a scale
- 3 rotation and 2 translations now need to be computed using tie points

More cameras

- Just keep on incrementally adding cameras using the same method
- “Bundle adjustment” is a mathematical method to to global estimates of all unknowns, “spreading out” the errors/noise

Everything is now in *relative camera coordinate space*

Go from camera space to 2D in order to carry out image distortion calculations and subsequently apply the corrections

CAMERA CALIBRATION

Calibration and orientation

$$\begin{pmatrix} i_{L,K} \\ j_{L,K} \end{pmatrix} = \boxed{\mathfrak{I}} \left(\pi \left(R_K * \left(\begin{pmatrix} {}^{rs}x_L \\ {}^{rs}y_L \\ {}^{rs}z_L \end{pmatrix} - C_K \right) \right) \right)$$

camera
parameters

- We need to estimate the camera's optical parameters:
  - Focal
  - Distortion

Ideal camera

Centre of image

$$\begin{pmatrix} {}^{id}i_L \\ {}^{id}j_L \end{pmatrix} = \begin{pmatrix} i_{PP} \\ j_{PP} \end{pmatrix} + Foc * \begin{pmatrix} {}^{s}x_L \\ -{}^{s}y_L \end{pmatrix} / Sz_{Pix}$$

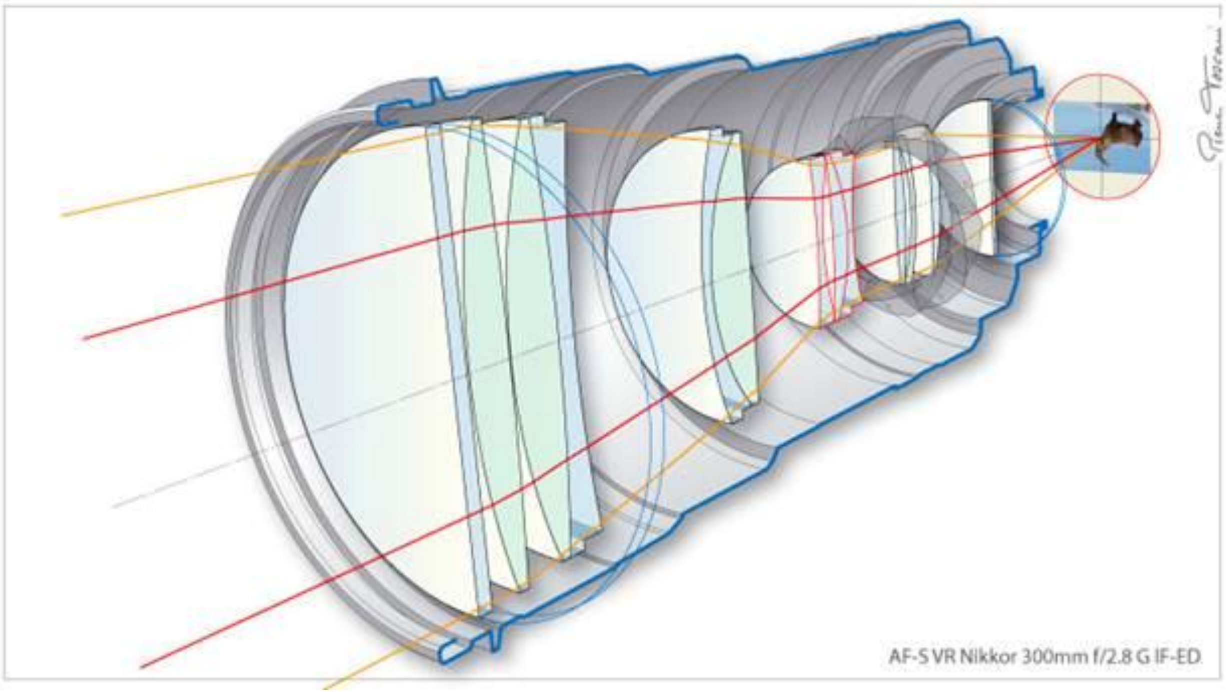
Real camera

Real lenses are optically imperfect, and therefore introduce distortion

D is the distortion model for the lens:

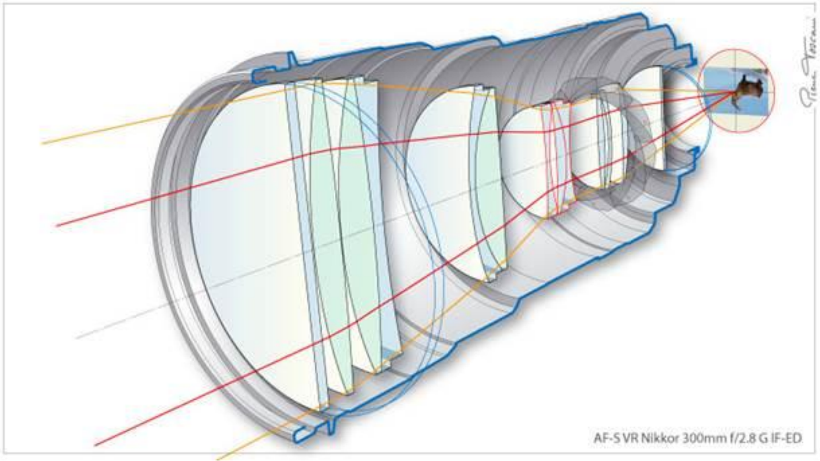
$$\begin{pmatrix} i_L \\ j_L \end{pmatrix} = D\left(\begin{pmatrix} i_{PP} \\ j_{PP} \end{pmatrix} + Foc * \begin{pmatrix} {}^s x_L \\ {}^s y_L \end{pmatrix} / Sz_{Pix} \right)$$

Distortion



Distortion

- If we consider that the distortion is the result of a projection through a cylindrically symmetrical system, we know the distortion is close to radial.
- It can be defined by:
  - A distortion centre **C**
  - A radial distortion function **Dr(**d**)**, with d the distance to **C**



Distortion

Since we have extremely regular surfaces within the camera lens, we can consider that Dr is an odd polynomial function

$$D_r(d) = d + \alpha * d^3 + \beta * d^5 + \ldots$$

$$D(L) = C + \overrightarrow{CL}(1 + \alpha * d^2 + \beta * d^4 + \gamma * d^6)$$

Distortion

Since we have extremely regular surfaces within the camera lens, we can consider that Dr is an odd polynomial function

$$D_r(d) = d + \alpha * d^3 + \beta * d^5 + \ldots$$

$$D(L) = C + \overrightarrow{CL}(1 + \alpha * d^2 + \beta * d^4 + \gamma * d^6)$$

Distortion

More complex models can be used to account for lens misalignment or imperfections

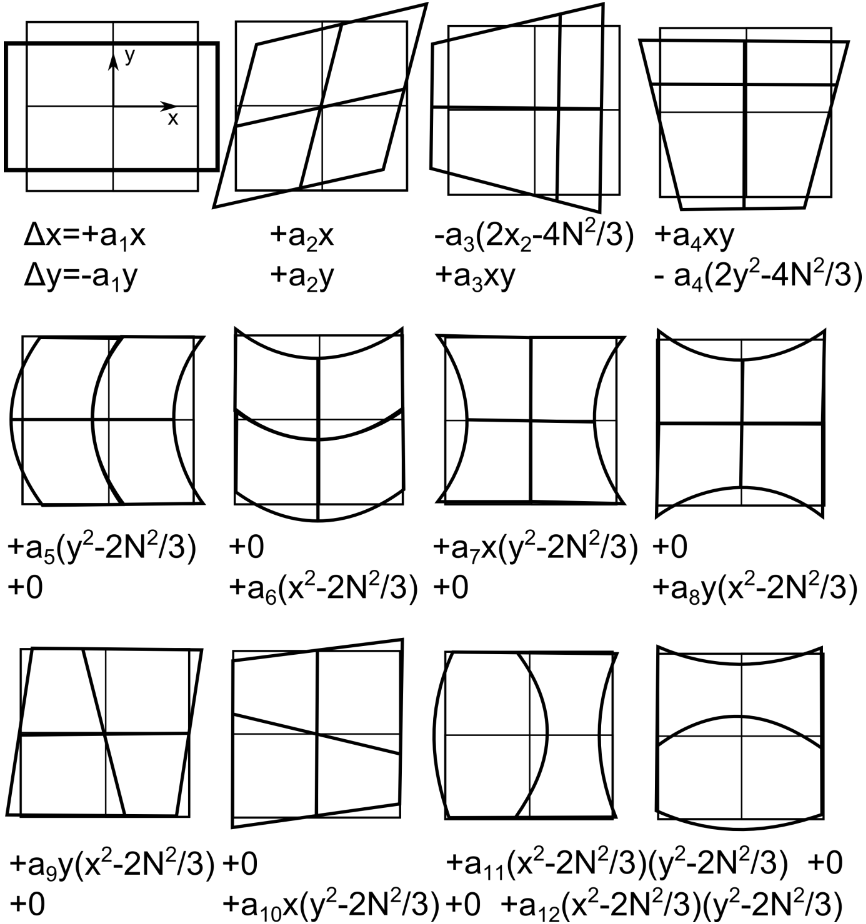
$$\begin{pmatrix} i_L \\ j_L \end{pmatrix} = \begin{pmatrix} {}^{id}i_L * (1 + K_1 * d^2 + K_2 d^4 + ...) + (P_2(d^2 + 2 * {}^{id}i_L^2) + 2P_1 {}^{id}i_L {}^{id}j_L) * (1 + P_3 d^2 + P_4 d^4 + ...) \\ {}^{id}j_L * (1 + K_1 * d^2 + K_2 d^4 + ...) + (P_1(d^2 + 2 * {}^{id}j_L^2) + 2P_2 {}^{id}i_L {}^{id}j_L) * (1 + P_3 d^2 + P_4 d^4 + ...) \end{pmatrix} \tag{30}$$

The parameters are :

- d the distance between ideal point and the distortion center.
- K_n the radial distortion coefficients.
- P_n the tangential distortion coefficients.

Distortion

More complex Ebner distortion models (a_i parameters, N the largest image coordinate)



Estimating distortion parameters

- Initializing with D is the identity (no distortion)
- PP is the image centre
- Foc is the focal given by the user (or the EXIF)
- **Thankfully the computer does the rest**

Georeferencing - absolute orientation

Absolute orientation

- Registering the relative space (R_{RS}) to a given reference space
- Almost always needed, at the very least to provide scale

Absolute orientation

$$\boxed{{}^{w}L = \lambda * {}^{w}R_{rs} * \left({}^{rs}L - {}^{w}C_{rs}\right)}$$

λ = scaling

${}^{w}R_{rs}$ = 3 rotations

${}^{w}C_{rs}$ = 3 translations

GCPs and CPs

1. Ground control points
   - Used to control your 3d model

2. Check points
   - Used to check/validate your model

> Also used to combine two datasets

Control Points

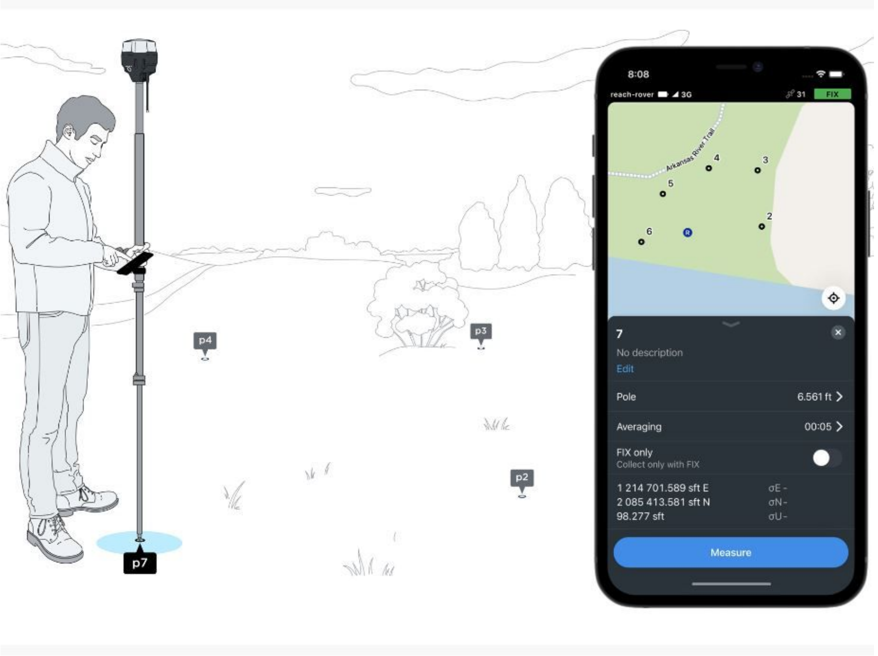
- Points visible in the imagery with known coordinates in the desired coordinate system
- Collected either:
  - through manual surveying with GNSS
  - using other georeferenced products (e.g. previously generated DEMs, orthoimage, map, google maps)
- Accuracy influenced by GSD

map.geo.admin.ch

(b)

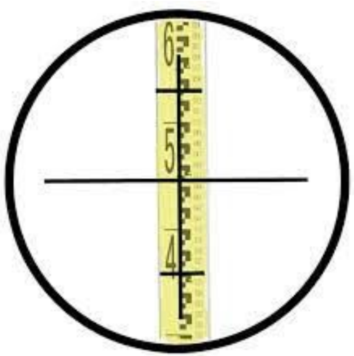
Measuring Control Points

Ideally, use a dGPS



Measuring Control Points

You can also use traditional surveying equipment to create a local reference system specific to your field area



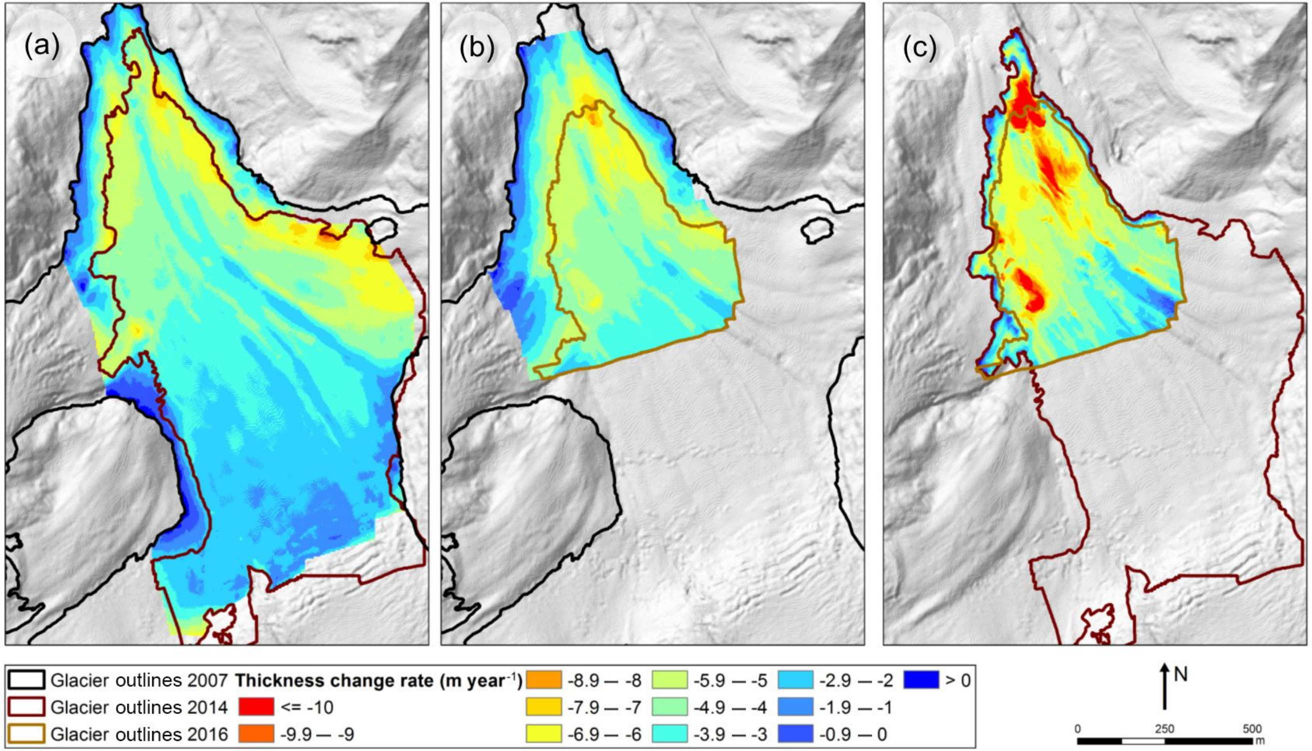
Why do they matter? Do you really need them?

Yes!

Because even in you don’t care about absolute real-world coordinates, distortions happen in image alignment and can affect your scientific conclusions

Dataset fusion

Need high accuracy if you want to look at temporal changes, compare datasets, or fuse datasets

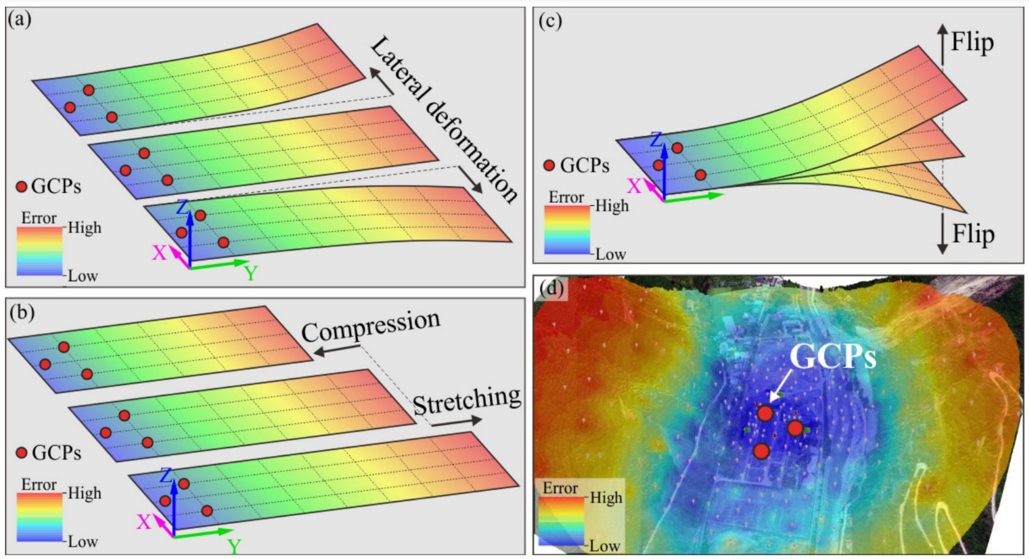


https://doi.org/10.5194/nhess-18-1055-2018

Model accuracy

How accurate you want your model to be will depend on your research question

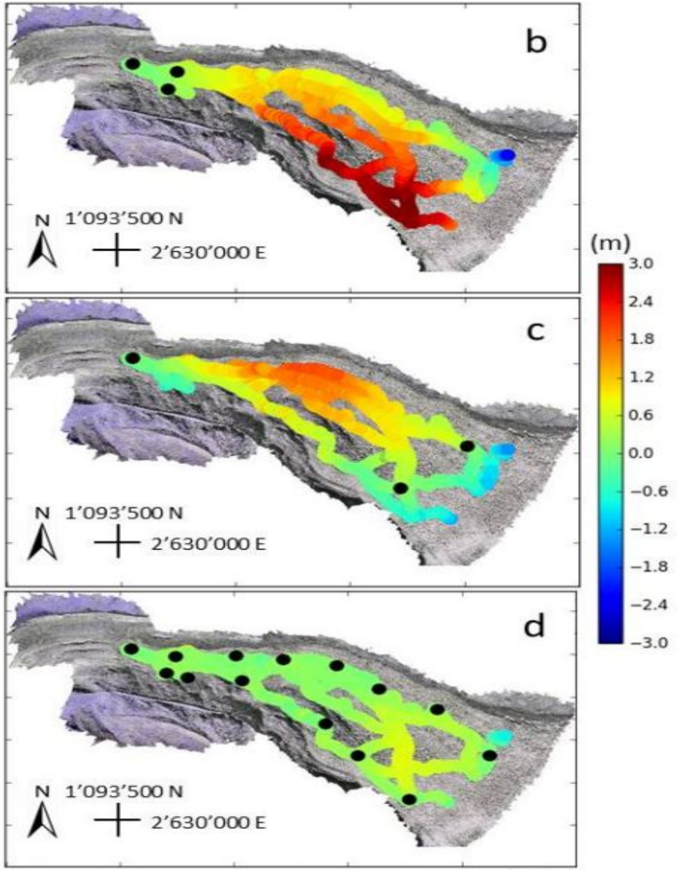
Even if you don't care about geo-referenced accuracy, any measurements from your data product will be affected by the accuracy of the product



Placement

- Should be distributed evenly through the ROI, around the boundaries and across z-distribution

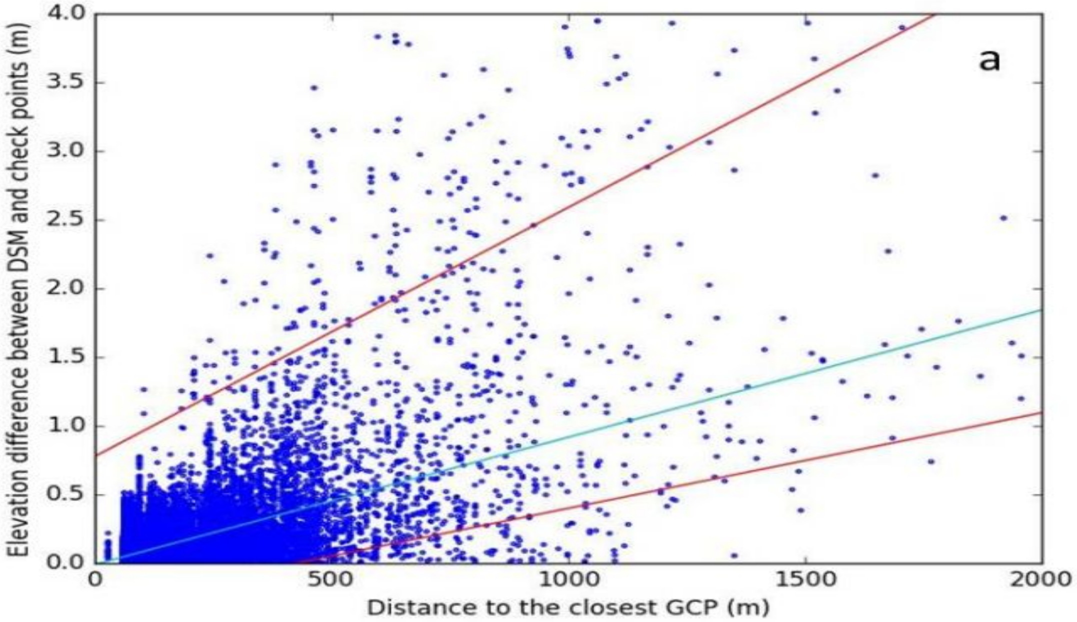
- Error will increase with increasing distance from GCP



https://www.mdpi.com/2072-4292/9/2/186

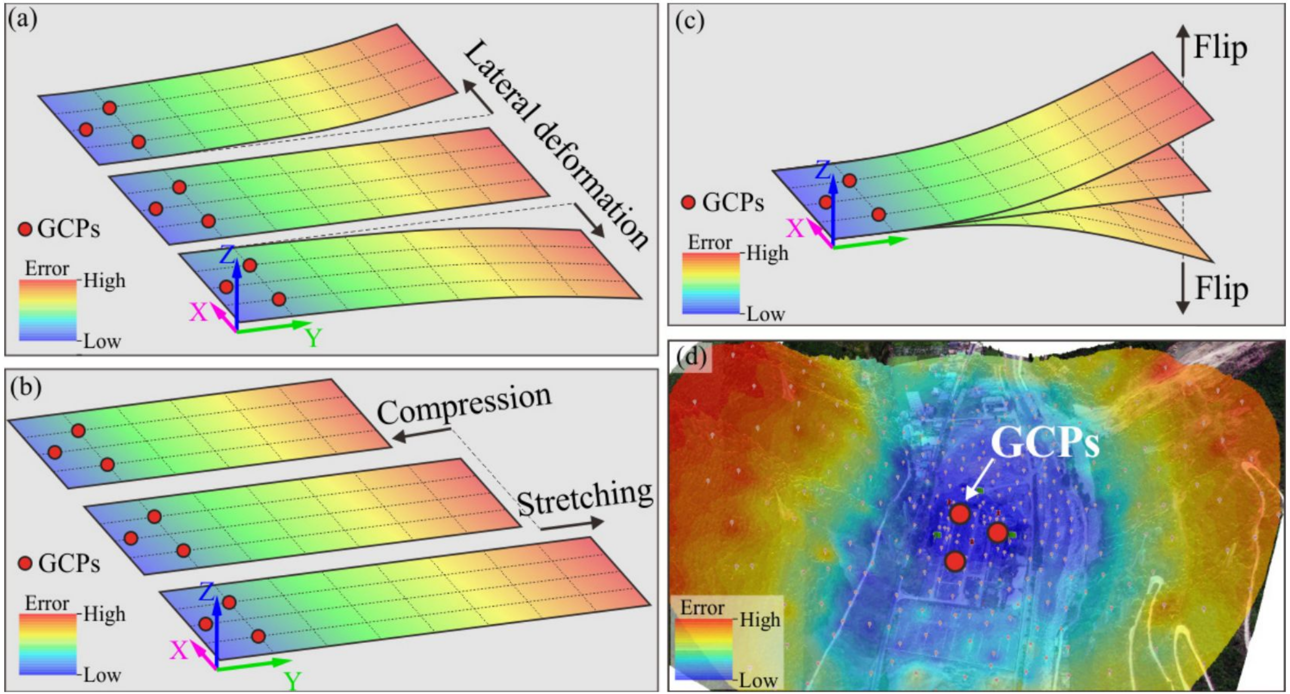
Placement

Z-axis most prone to error



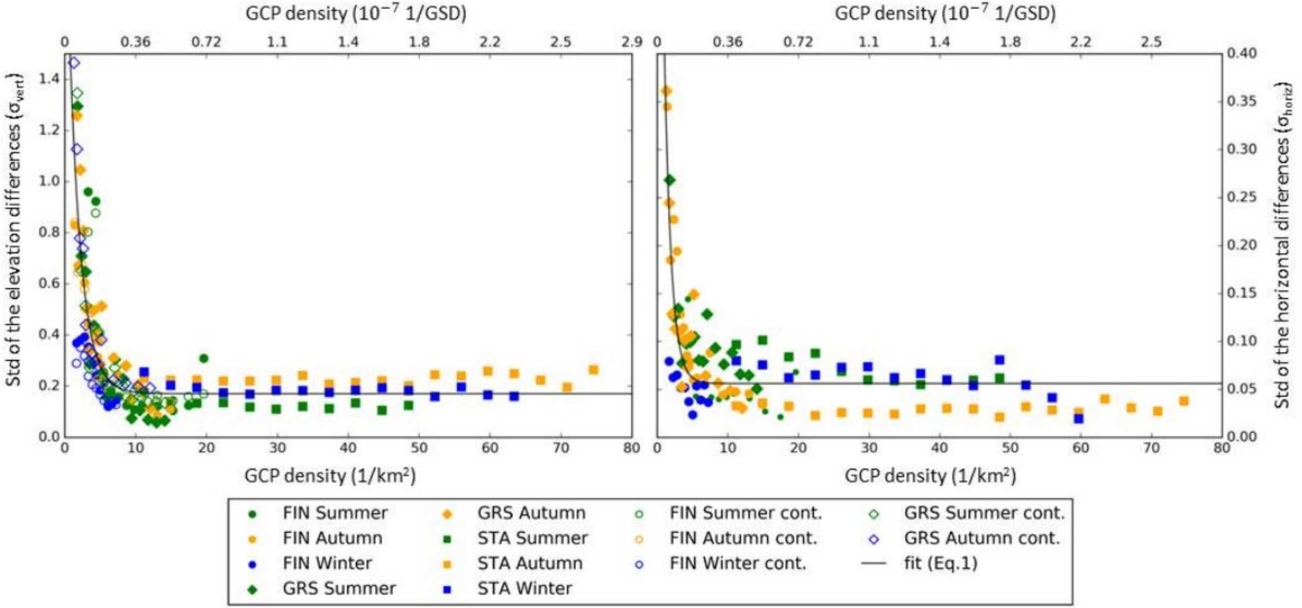
https://www.mdpi.com/2072-4292/9/2/186

Extrapolation and distortion



http://dx.doi.org/10.1109/TGRS.2022.3181258

How many?



https://www.mdpi.com/2072-4292/9/2/186

Flight planning

- Ground resolution **GSD** (in m)
- Pixel size **SzPix** (1->6µm)
- Sensor size **SzX** and **SzY** (in mm)
- Focal length **Foc** (in mm)
- Flight height **H** (in m)
- Along-track base **B** (in m)
  - Associated $\textbf{Overlap}_{\textbf{along-track}}$
- Cross-track base **D** (in m)
  - Associated $\textbf{Overlap}_{\textbf{cross-track}}$
- Aircraft velocity **V** (in m/s)
- Frequency of acquisition **Freq** (in Hz)

$$GSD_{image} = \frac{H * Sz_{Pix}}{Foc}$$

$$GSD_{image} \approx GSD_{orthoimage}$$

$$GSD_{orthoimage} = GSD_{DEM}/2$$

$$Overlap_{along-track} = 1 - \frac{B}{GSD * SzY}$$

$$Overlap_{cross-track} = 1 - \frac{D}{GSD * SzX}$$

$$B = \frac{V}{Freq}$$

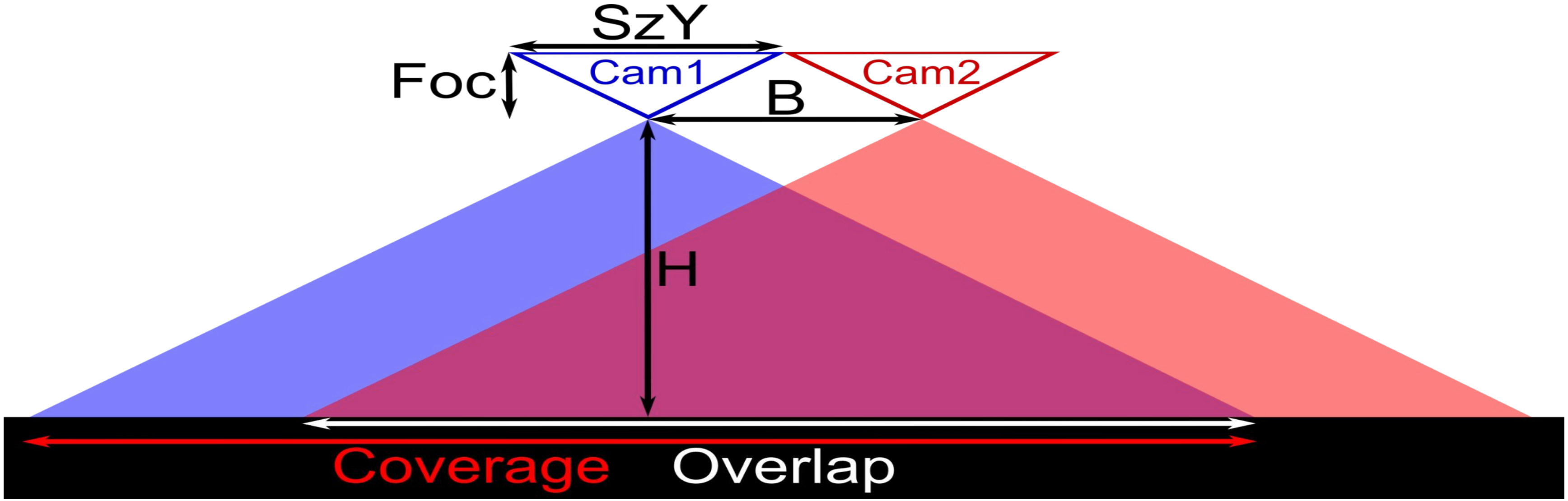
Flight planning

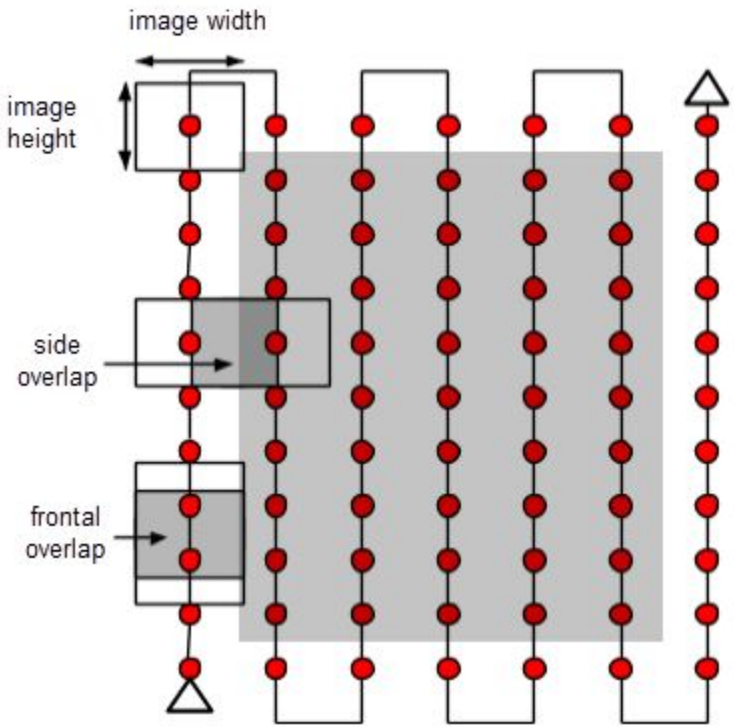
- Ground resolution **GSD** (in m)
- Pixel size **SzPix** (1->6µm)
- Focal length **Foc** (in mm)
- Flight height **H** (in m)

$$GSD_{image} = \frac{H * Sz_{Pix}}{Foc}$$

NOTE: your smallest identifiable object will be 3 x GSD

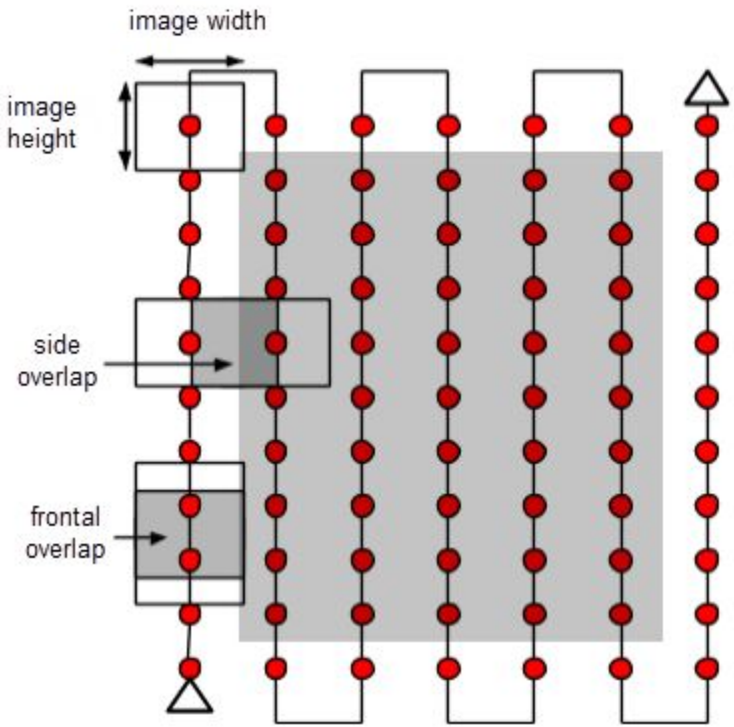
Flight planning





Along-track / forward overlap

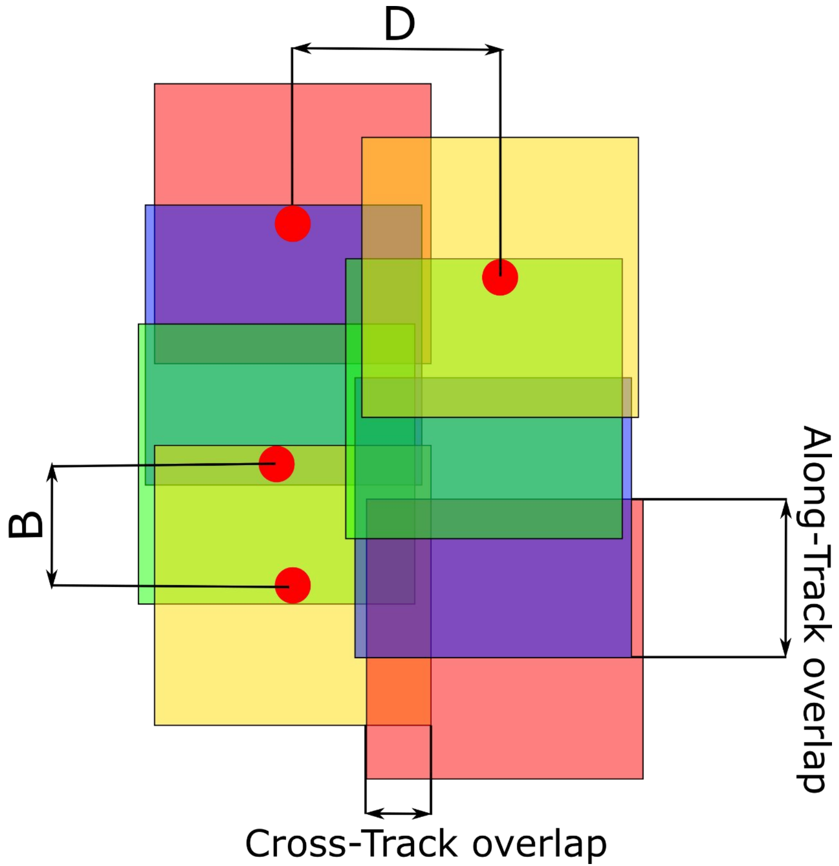
Need a minimum overlap of 67% to ensure a point on the ground is seen in at least three images. At least 80% overlap is preferred, particularly over more complex terrain.



Cross-track / side overlap

Needed to link flight lines together, but less important for linking individual images together for 3d reconstruction

Flight planning



Bases

B = distance between two image centres along track

D = distance between two image centres, cross track

Flight planning

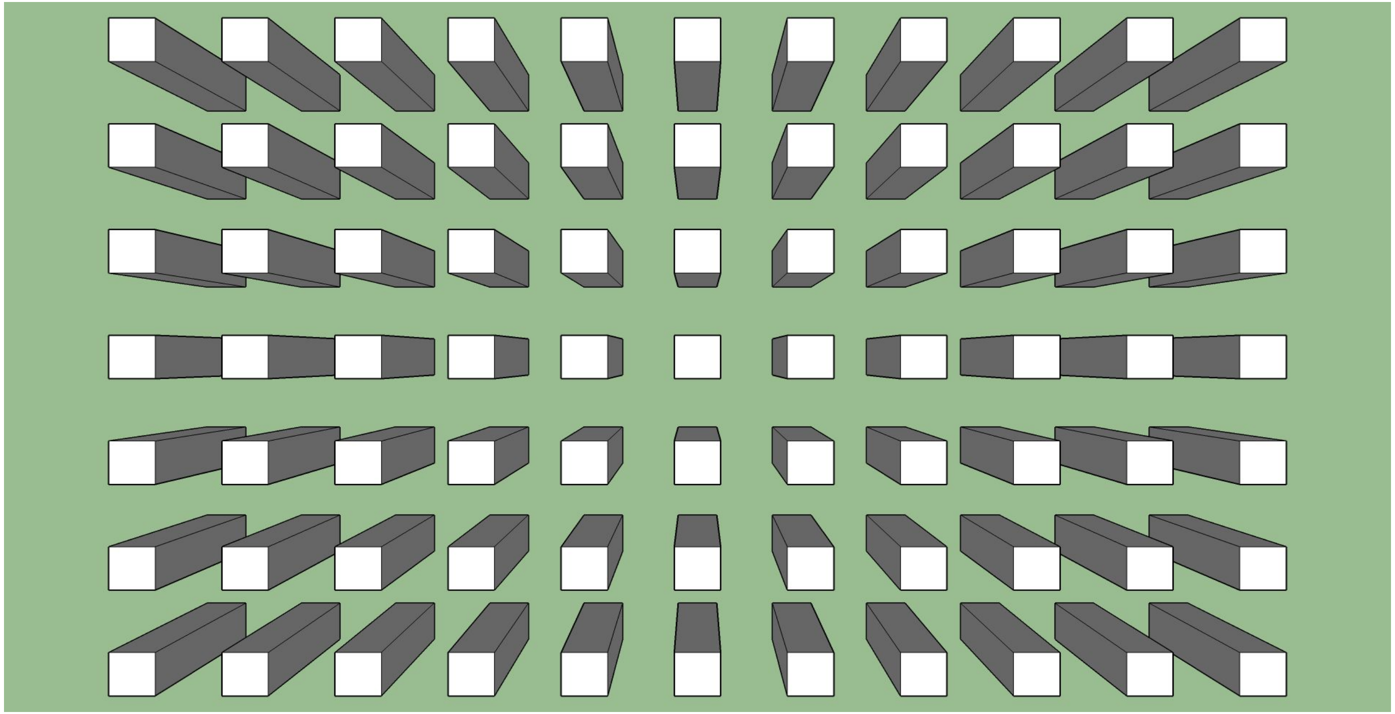
- Along-track base **B** (in m)
  - Associated **Overlap**$_{\text{along-track}}$
- Cross-track base **D** (in m)
  - Associated **Overlap**$_{\text{cross-track}}$
- Sensor size **SzX** and **SzY** (in mm)
- Aircraft velocity **V** (in m/s)
- Frequency of acquisition **Freq** (in Hz)

$$Overlap_{along-track} = 1 - \frac{B}{GSD * SzY}$$

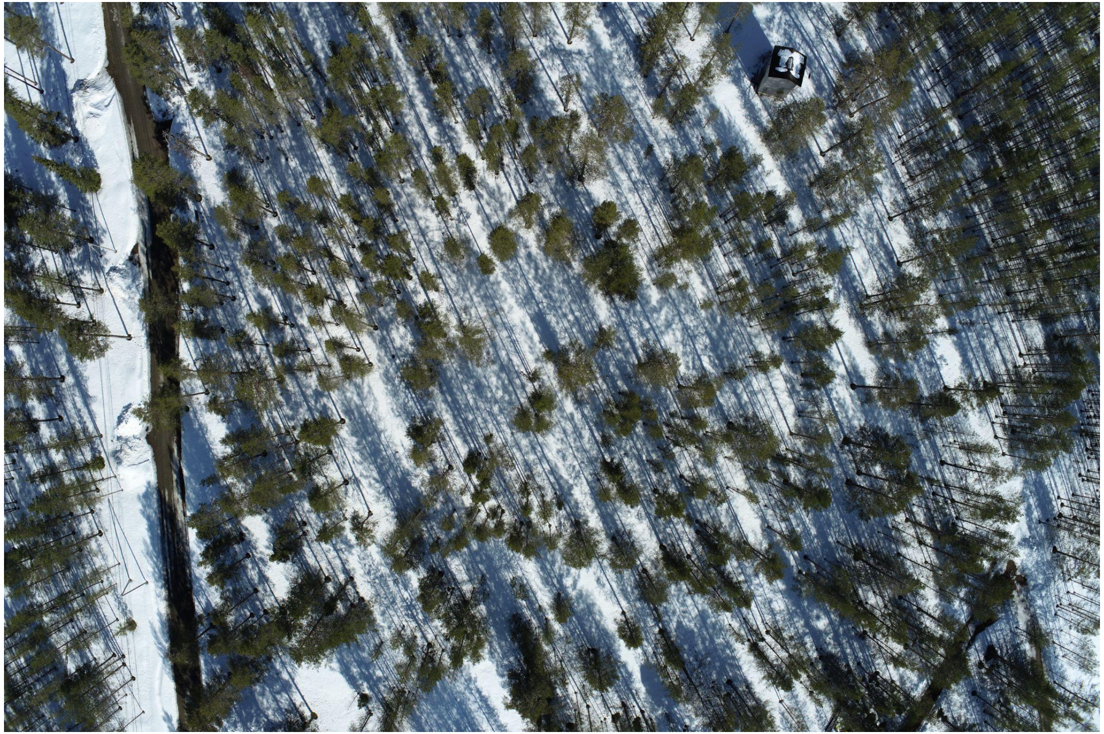
$$Overlap_{cross-track} = 1 - \frac{D}{GSD * SzX}$$

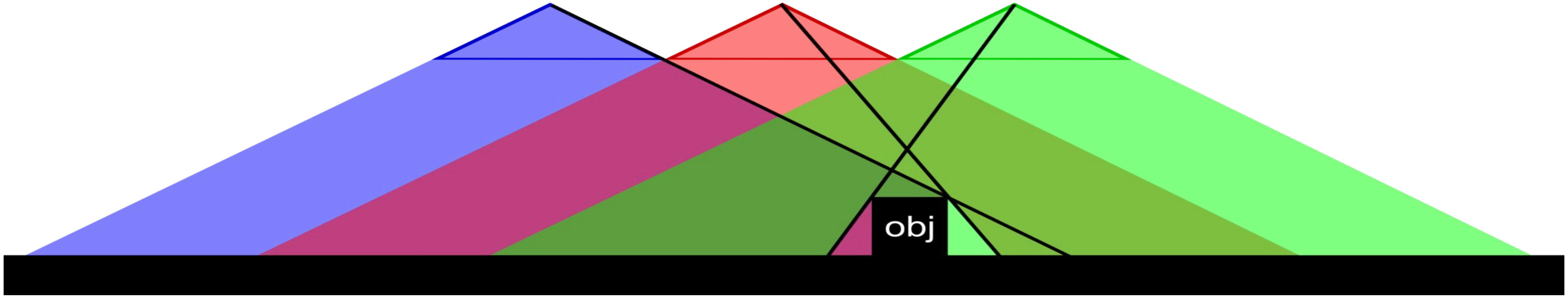
$$B = \frac{V}{Freq}$$

Foreshortening / Perspective distortion



Flight planning

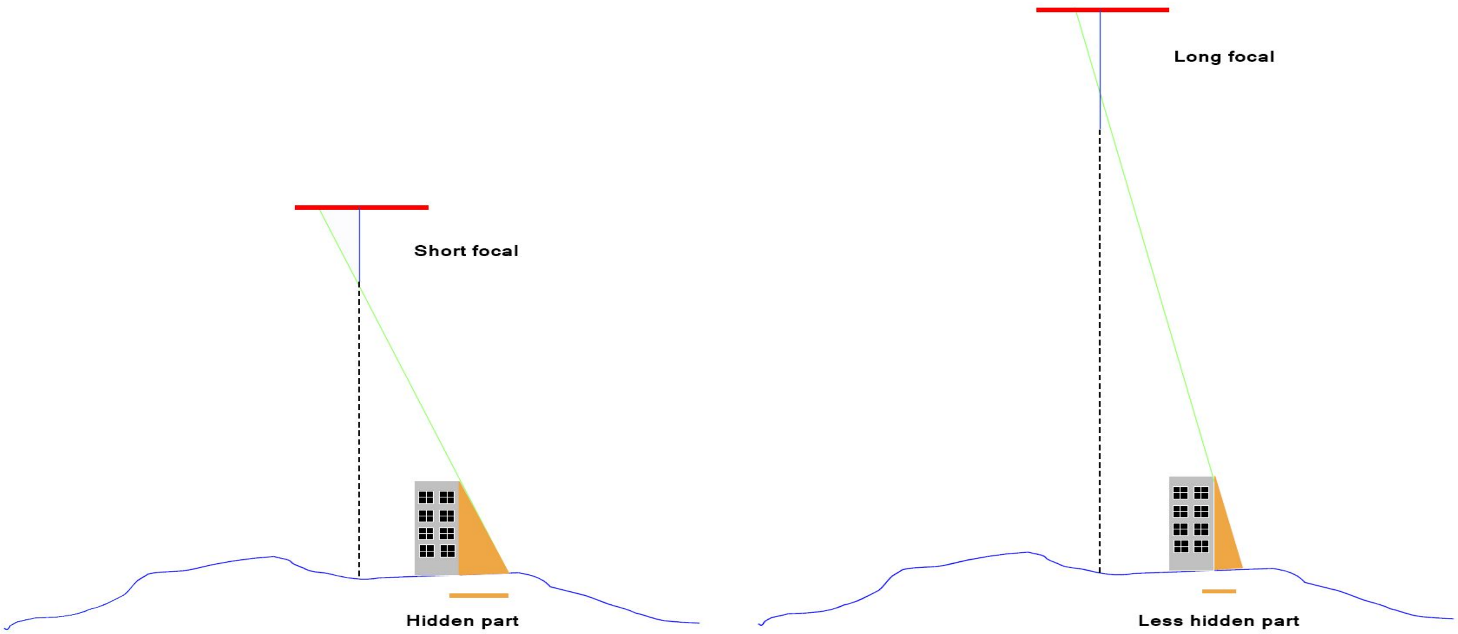




Even with high overlap, there can still be areas that only a single image is covering.

Keep this in mind when deciding on side-overlap

Flight planning

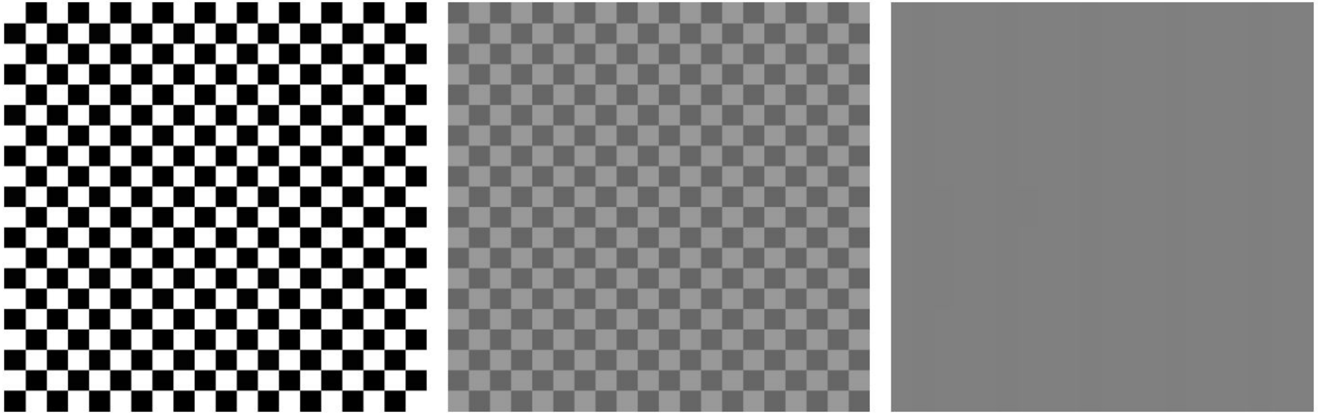


- Using a longer focal length creates less hidden parts.
- Increasing the camera distance from the surface (e.g. higher flying height) also helps.

Motion blur

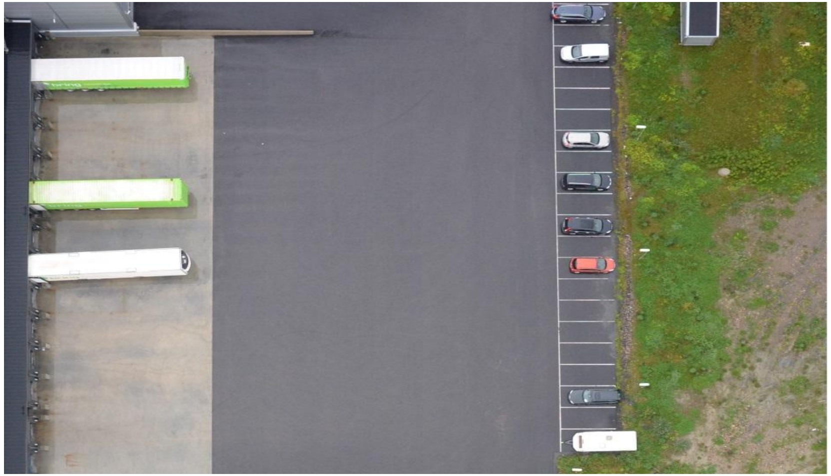
- Caused by long exposure times when the camera is moving.
- Solved by the following, where “Acceptable Blur” is in pixels, usually < 0.5.

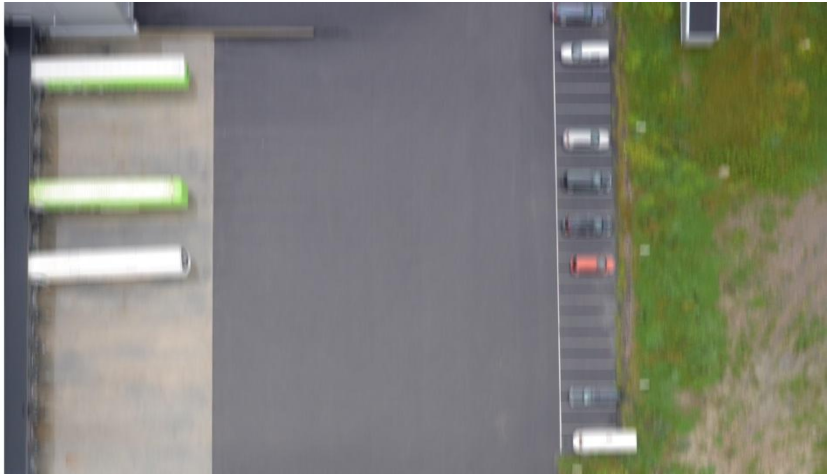
$$V < \frac{Res}{ExpTime} * AcceptableBlur$$



Motion blur on a checker board (0 pixel – 0.5 pixel – 1 pixel)

Motion blur

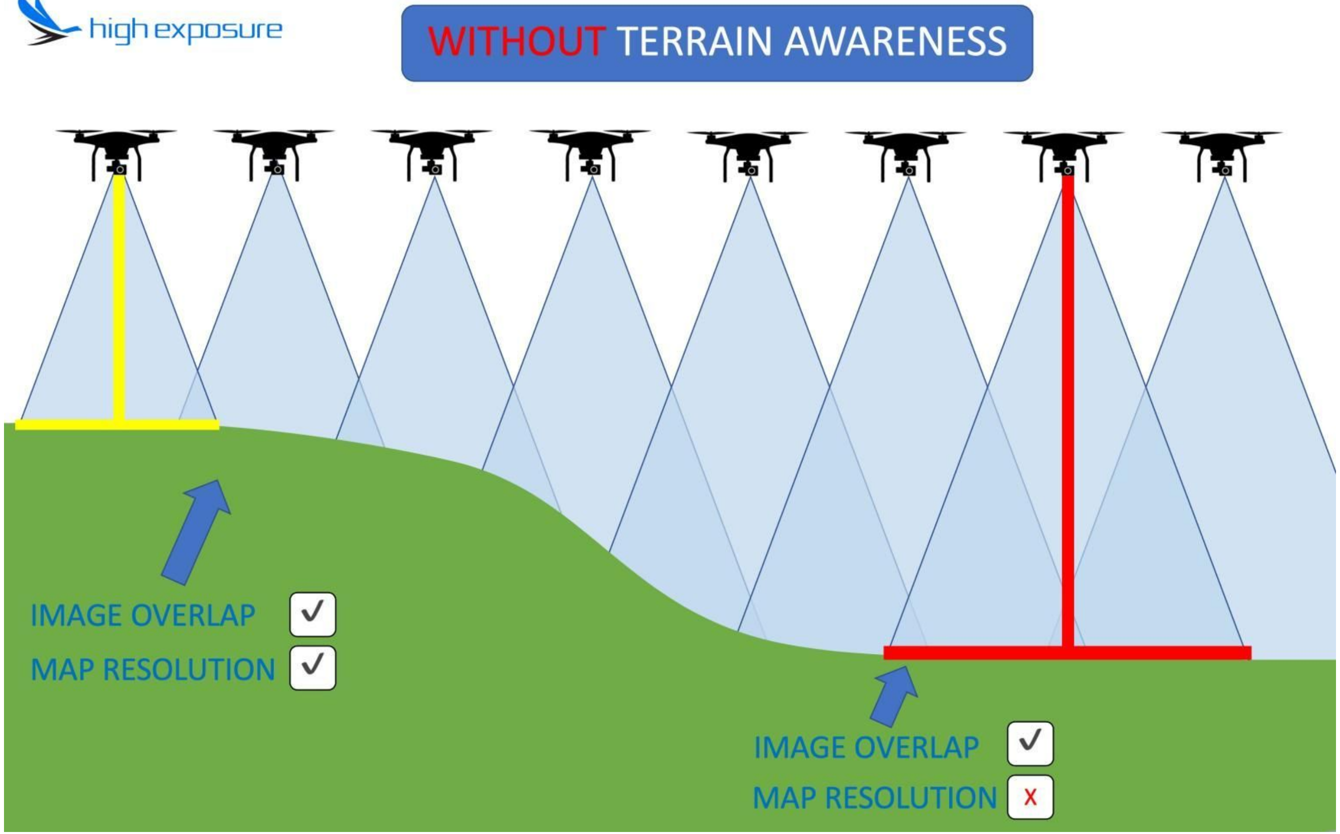




Example of a four-pixel motion blur

Terrain following

high exposure

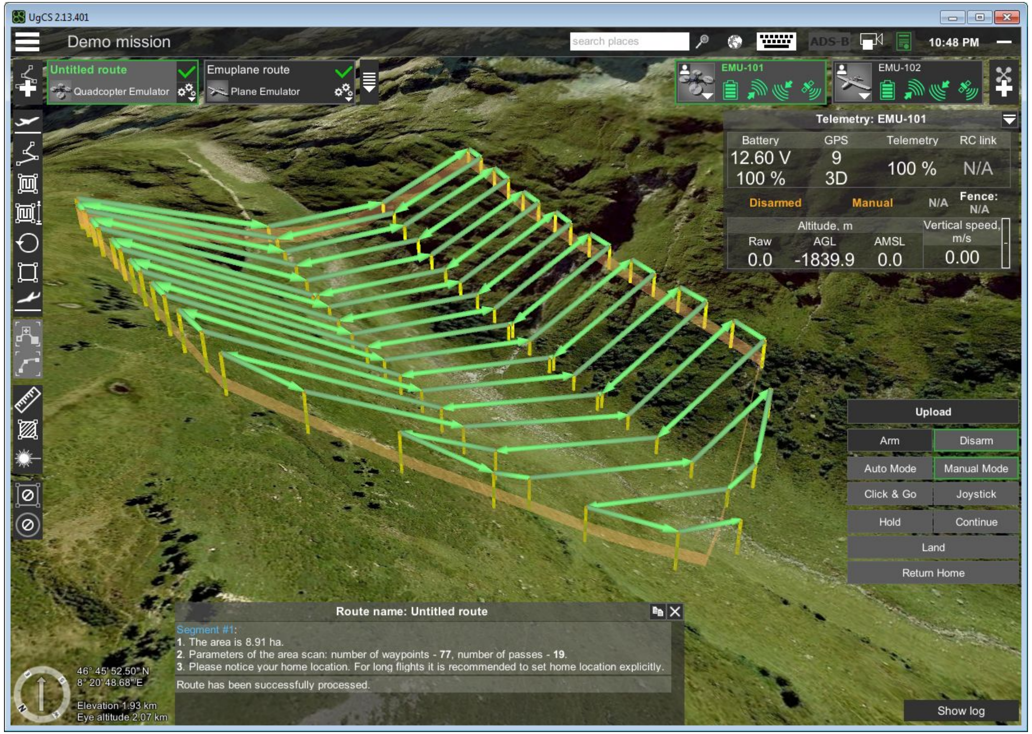


Software - Planning

Most consumer drones will have an associated flight planning app for mobile/tablet, e.g.

There are also external options available, particularly if you want to do something more complicated, e.g.

Software - Planning



source: https://phantompilots.com/attachments/93049/

The gear

Consumer/Prosumer cameras

Blimp

Drone/UAV/UAS

Photogrammetric camera

4 panchromatic cameras at full resolution and R/G/B/NIR cameras at half-resolution

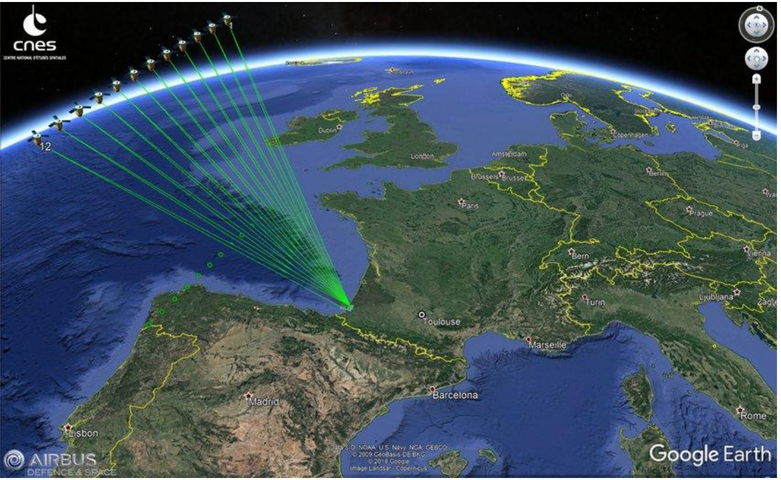
Photogrammetric camera

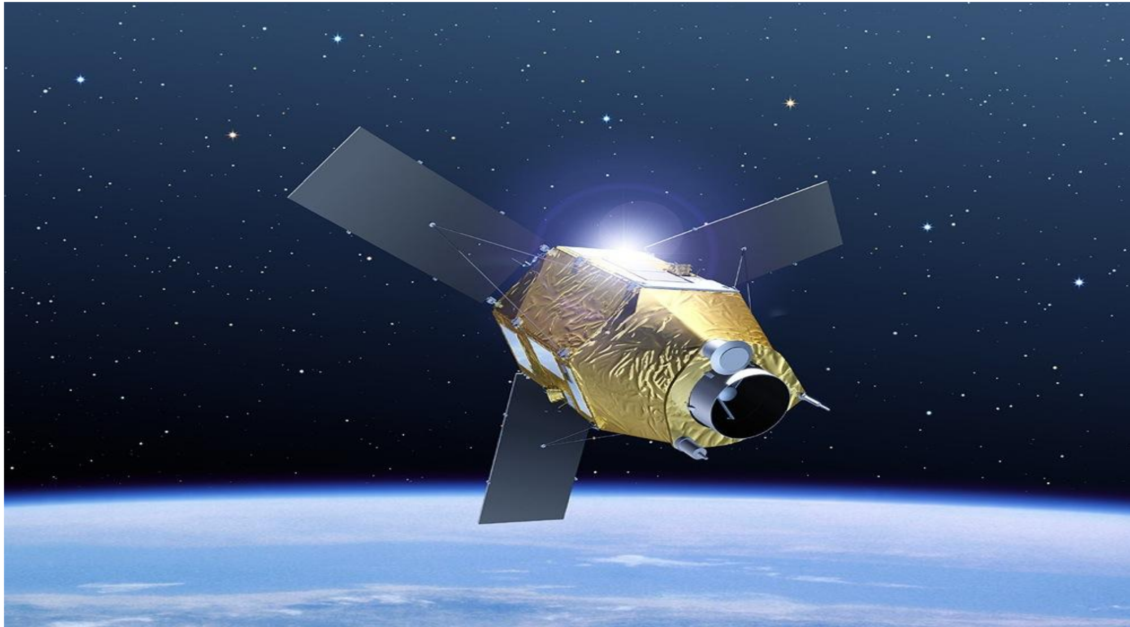
R/G/B/NIR cameras at nadir, and 4 RGB cameras at 45° oblique for urban mapping

Photogrammetric plane

The camera window is covered when close to the ground to avoid getting dirty, and opens to take pictures. This is the plane used by the French Geographgical Institute (IGN)

Optical satellite





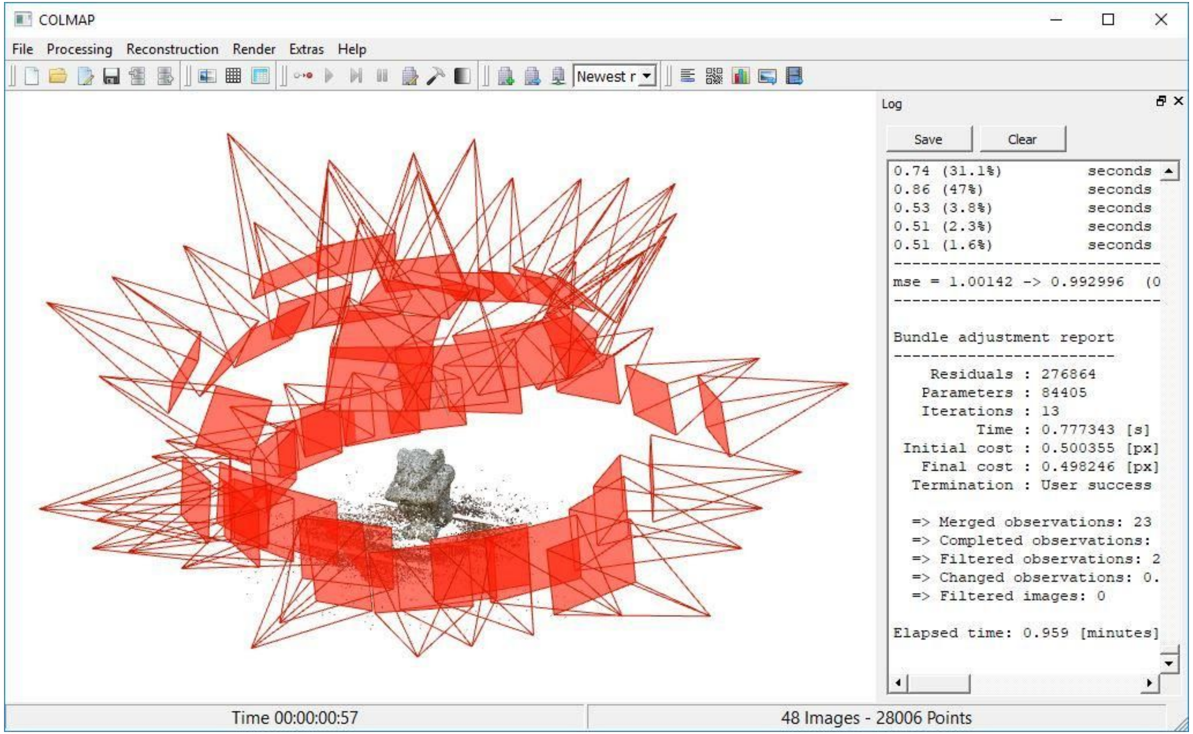
GNSS equipment

Software - Post-processing

Common in industry/enterprise:

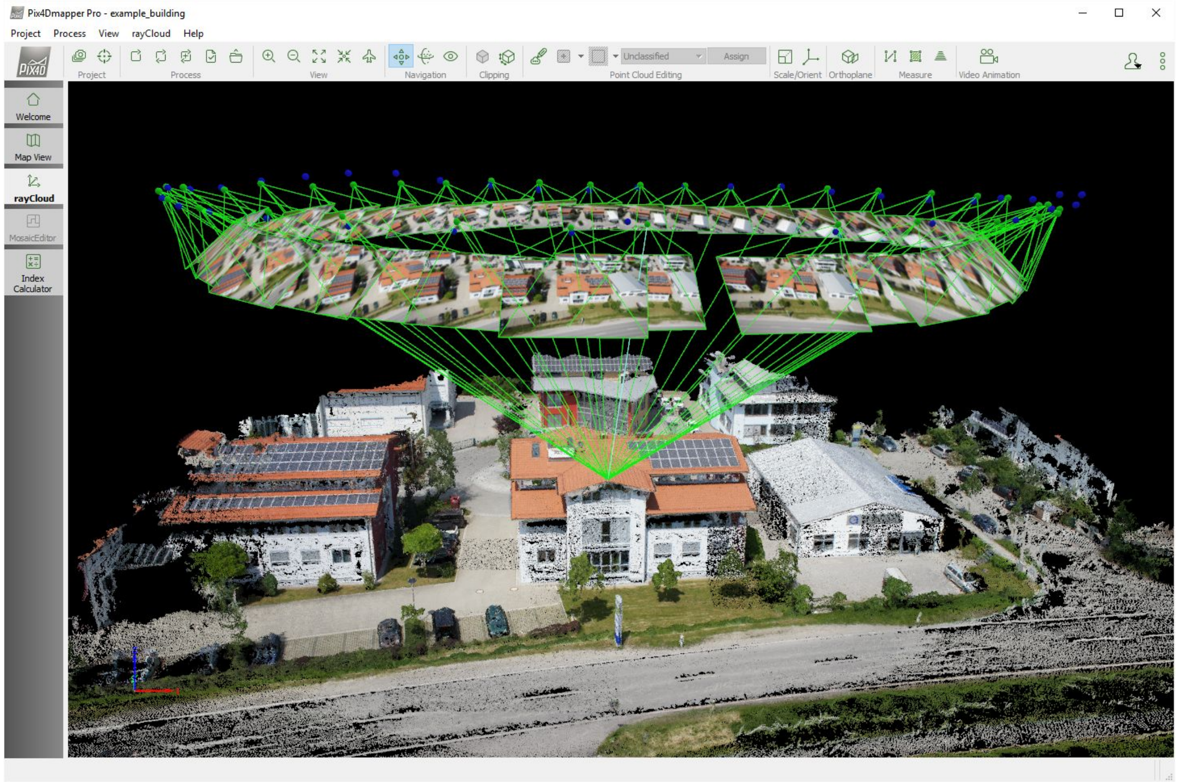
- Agisoft Metashape - probably most well-known
- Pix4D - great for multi-spectral
- DJI Terra - first party DJI processing suite
- OpenDroneMap - open source
- **Micmac** - open source, command-line based
- Reality Capture (free for Edu from May 2024)
- colmap - computer vision, not photogrammetry
- *iTwin Capture Modeler* (Bentley - UiO license)
    - We'll use this one in the lab session

Software - COLMAP



source: https://blog.sketchfab.com/wp-content/uploads/2018/08/image6-6.jpg

Software - Pix4D



source: https://geospatialmedia.s3.amazonaws.com/wp-content/uploads/2016/06/Pix4D.png