Sensor models - 3D detection and reconstruction – Pulsed LiDAR and aerial cameras – geometry

Lecture 2 & LAB 1

Learning goals

- Understanding photogrammetric 3D data capture
- Understanding LiDAR geometry and canopy interactions
- Measuring the geometry of trees and canopies with these data
- Something about the technical implementation

We focus on geometry in this lecture & LAB

<u>Photogrammetry</u> when broken into it's roots is "using light to measure and draw“.

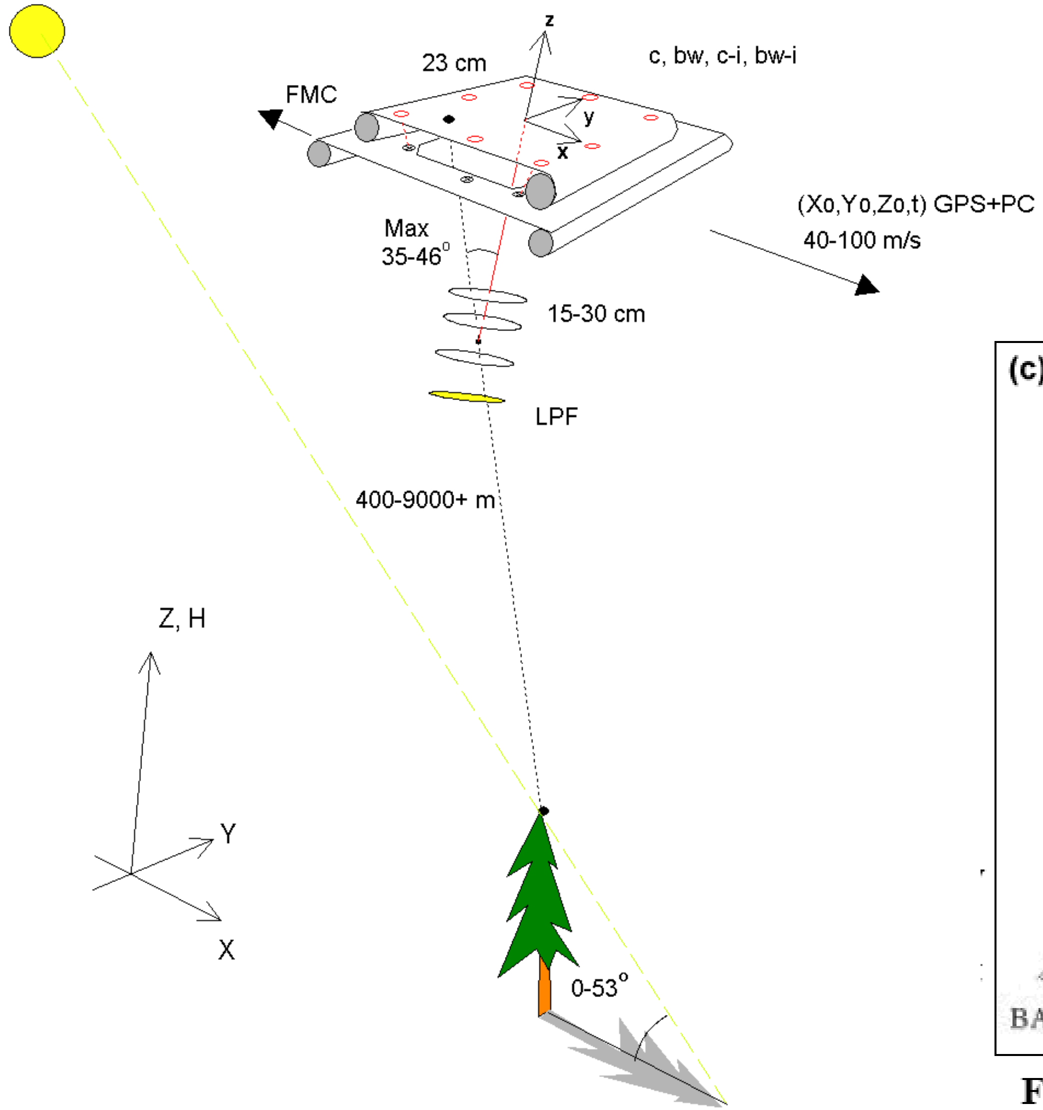
Using image(s), a target is modelled in 2D, 2.5D or 3D (shape, size, position, orientation, qualitative attributes)

Close range photogrammetry – mapping forest plots (also TLS)
Aerial photogrammetry
Space (satellite) photogrammetry

Stereo photogrammetry (Stereoscopic vs. Multiscopic)
~ 3D machine vision

Passive sensors and active sensors

Sensors: from pinhole camera to more complex



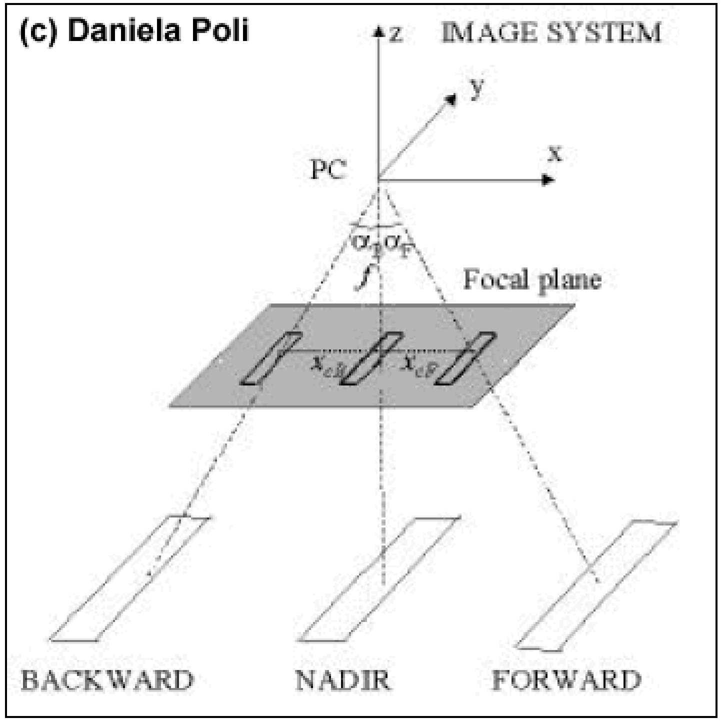
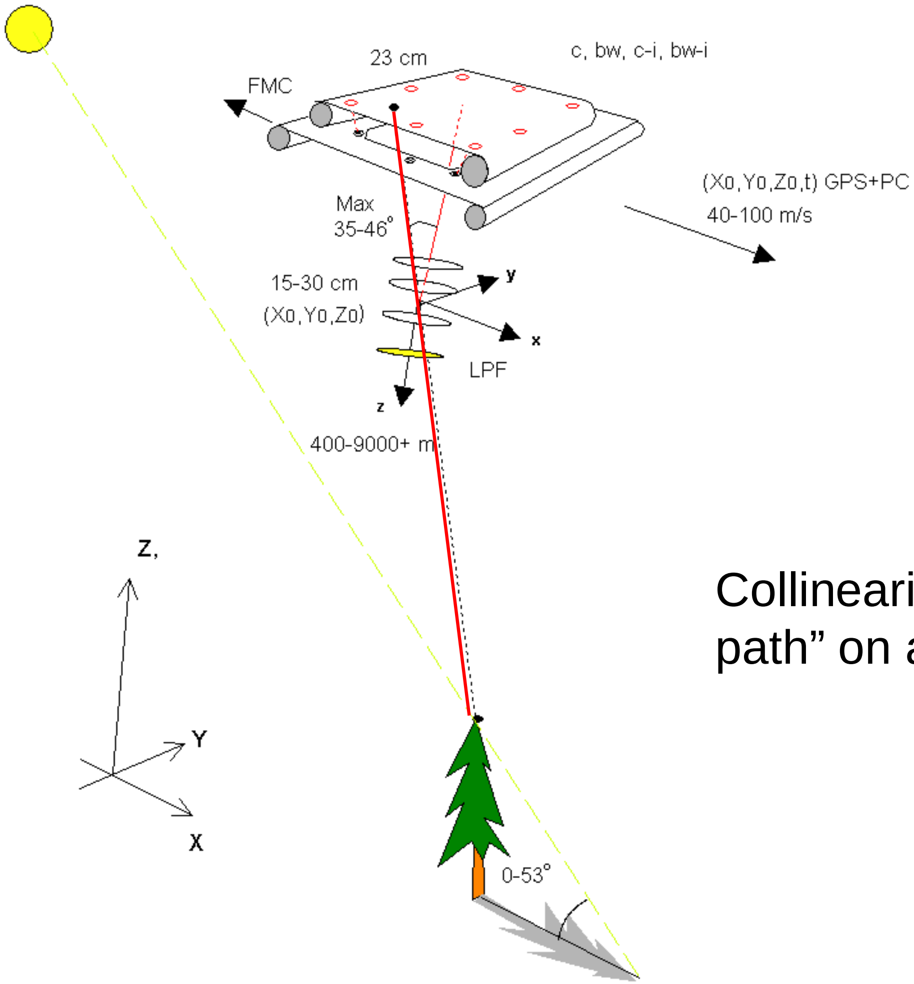


Figure 1. Geometry of CCD linear array sensors with three viewing directions.

Basic principles – pinhole camera model for frame sensors

3D image ray: p(t) = (X,Y,Z) + $t \times$ (i,j,k)



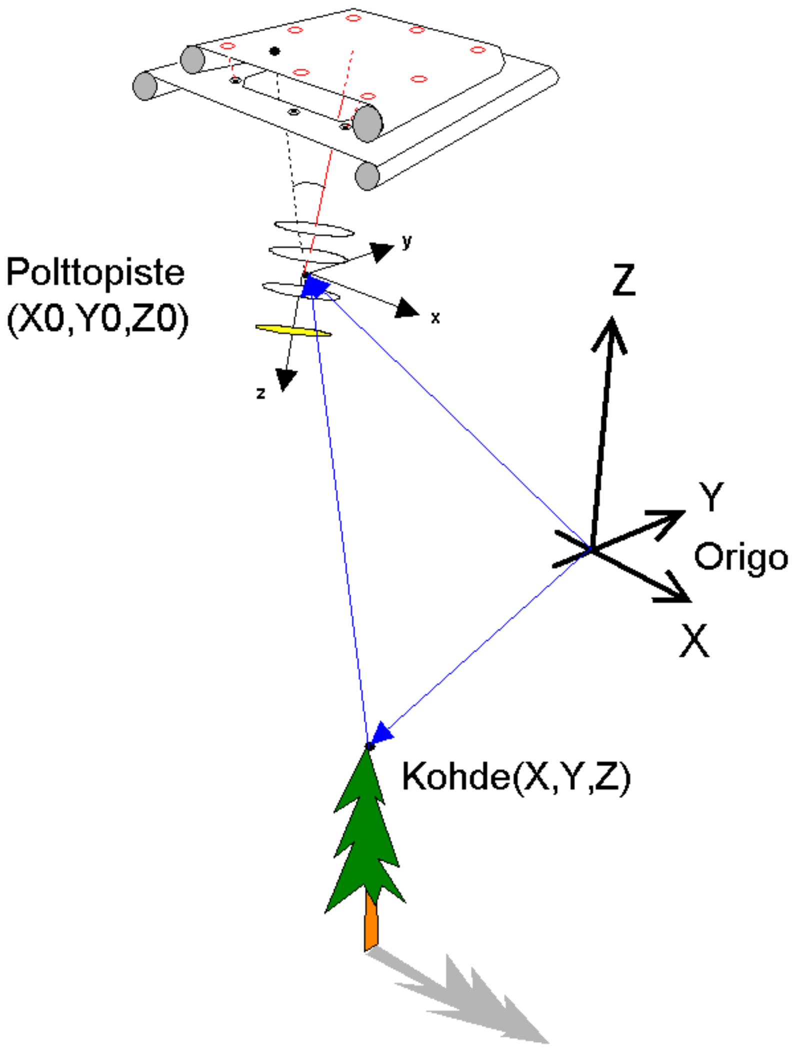
or

$$\begin{cases} a \cdot X + b \cdot Y + c \cdot Z = d \\ e \cdot X + f \cdot Y + g \cdot Z = h \end{cases}$$

$\Rightarrow$ 6 unknowns

Collinearity principle: the ray ”continues its path” on a straight line.

Basic principles – pinhole camera parameters



Camera is rotated about the principal point – 3 attitude angles

Camera is in the 3D space, 3 offsets

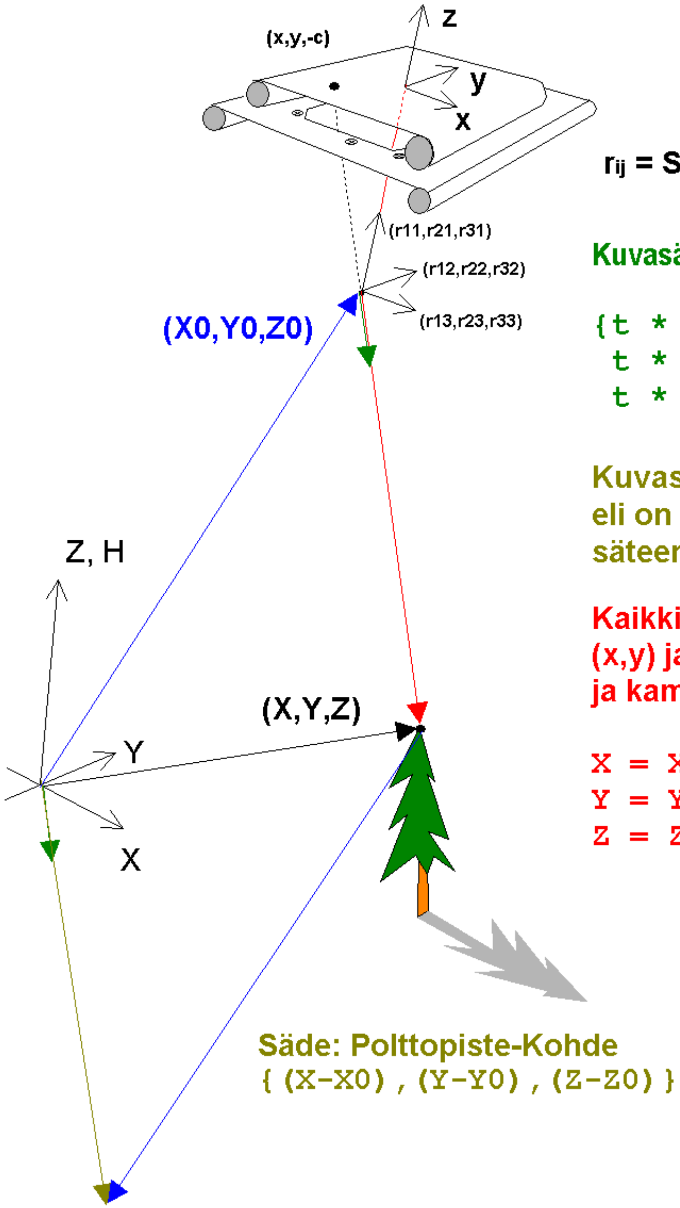
⇒ Exterior orientation

Focal plane sits a distance away from the principal point => focal lenght

Optical axis is not always in the centre of the focal (film, CCD) plane, 2 offsets

=> Interior orientation

Basic principles – Monoplotting



r_{ij} = Suuntakosinit eri akselien välillä

Kuvasäde Kamera->kohde, t antaa pituudeen (skaalaa)

```
{t * (r11*x+r12*y+r13*z),
 t * (r21*x+r22*y+r23*z),
 t * (r31*x+r32*y+r33*z)}
```

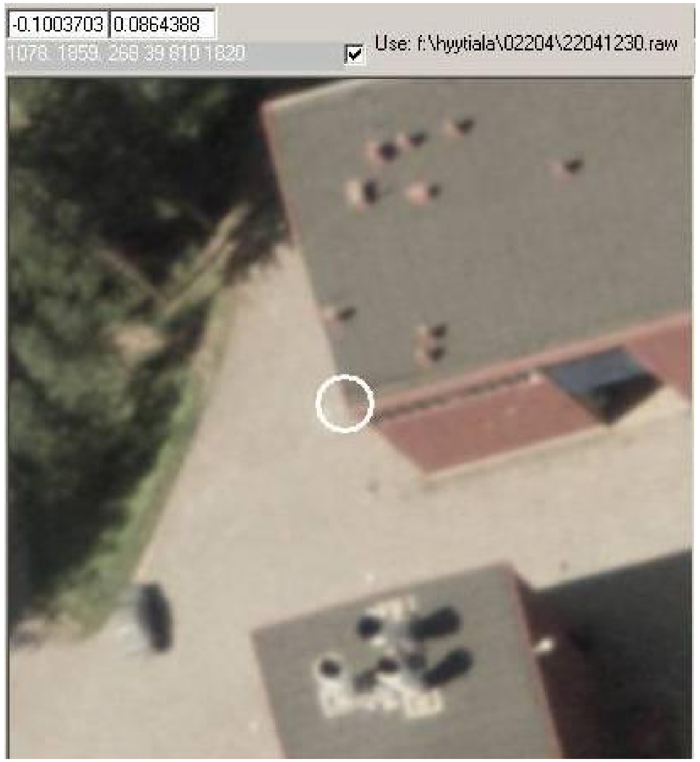
Kuvasäde leikkaa polttopisteen
eli on yhdensuuntainen
säteen { (X-X0) , (Y-Y0) , (Z-Z0) } kanssa

Kaikki pisteet kuvasäteellä voidaan laskea kuvakoordinaateista (x,y) ja kameravakiosta (z), kun kameran kierrot r(ij) tunnetaan ja kameran polttopisteen paikka (X0,Y0,Z0):

```
X = X0 + t * (r11*x+r12*y+r13*z)
Y = Y0 + t * (r21*x+r22*y+r23*z)
Z = Z0 + t * (r31*x+r32*y+r33*z)
```

Monoplotting: acquire xy-image observation, fix Z, solve t, solve XY.

From image (x,y) → to target



Ex. Image FM02204_1230

```
Directional cosines
r11:=-0.26788;r12:=-0.96343;r13:=0.00667;
r21:= 0.96344;r22:=-0.26783;r23:=0.00721;
r31:=-0.00516;r32:= 0.00836;r33:=0.99995;
Principal point (camera):
X0:=2515731.81;Y0:=6860644.14;Z0:=1132.57;
Image observations (c=focal length):
x:=-0.100370;y:=0.086453;c=0.153277;
Assume Z = 165 m.
165-Z0-t*(r31*x+r32*y+r33*(-c))=0
Solution for t = 6364.391249, subs. t
X-X0-t*(r11*x+r12*y+r13*(-c))=0
Y-Y0-t*(r21*x+r22*y+r23*(-c))=0
Solution:
   X = 2515366.32,Y = 6859874.30, Z =165.0
cf.    2515366.58,    6859874.90,    165.8
```

```
X = X0 + t * (r11*x+r12*y+r13*z)
Y = Y0 + t * (r21*x+r22*y+r23*z)
Z = Z0 + t * (r31*x+r32*y+r33*z)
```

From target to image, Example

Image FM02204_1230; Parameters X_0,Y_0,Z_0 and directional cosines as in previous example

Directional cosines:

```
> r11:=-0.26788;r12:=-0.96343;r13:=0.00667;
  r21:= 0.96344;r22:=-0.26783;r23:=0.00721;
  r31:=-0.00516;r32:= 0.00836;r33:=0.99995;
```

Principal point location:

```
> X0:=2515731.81;Y0:=6860644.14;Z0:=1132.57;
```

Target position, focal lenght:

```
X:=2515300;Y:=6859900; Z:=165; c:=0.153277;
```

Solution (collinear equations):

```
> x = -c*(r11*(X-X0)+r21*(Y-Y0)+r31*(Z-Z0))/
  (r13*(X-X0)+r23*(Y-Y0)+r33*(Z-Z0));
-0.093663 m
> y = -c*(r12*(X-X0)+r22*(Y-Y0)+r32*(Z-Z0))/
  (r13*(X-X0)+r23*(Y-Y0)+r33*(Z-Z0));
 0.095386 m
```

Making an orthoimage:

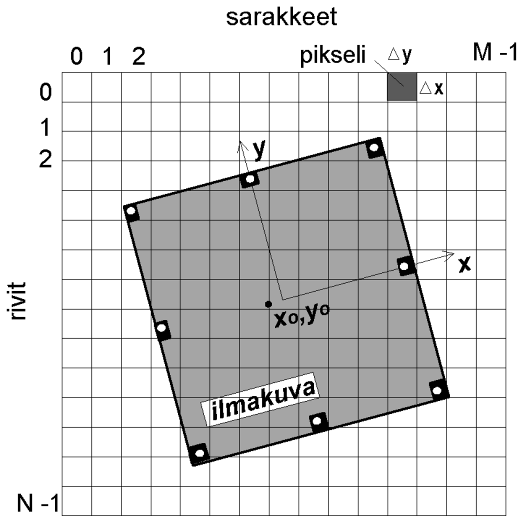
1. Define the XY-area, and give the mapping surface Q

2. Make a grid in XY

3. Q is approximated by a elevation model, $Z_Q = f(X,Y)$.

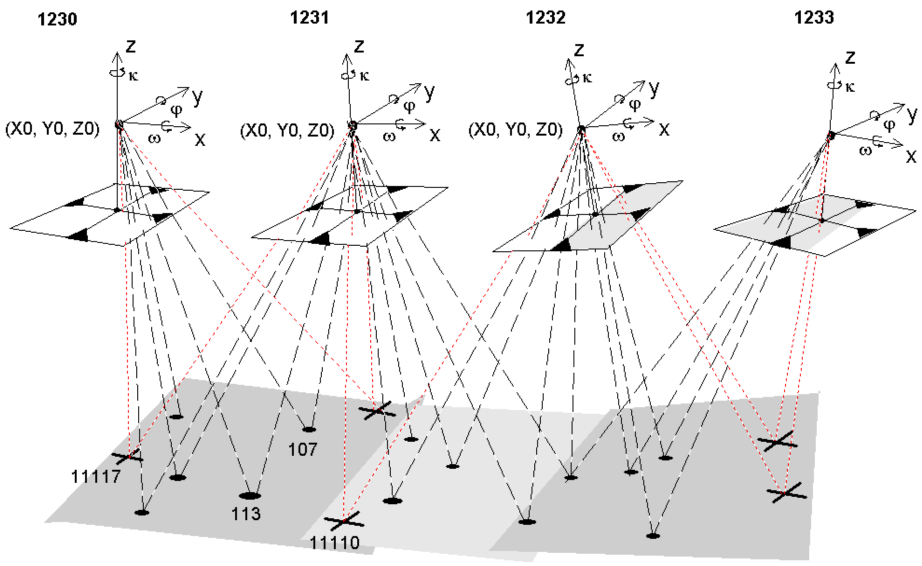
4. Make the image in the XY-grid, point-per-point, using the DEM to get Z for pint XY. Using the collinear equations fetch the pixels from the image and put them in the orthophoto.

Orientation parameters – where to get?



Camera calibration certificate
3 parameters of interior orientation
(lens error parameters, if large)

Direct sensor orientation / automated aerial triangulation; 3 attitude and 3 position parameters



Multilens and pushbroom sensors

ADS40

Multilens (UltraCAM, Z/I DMC) - usually preprocessing leads to ”pinhole image”, where lens errors and mismatches are removed (at the expense of some image interpolation)

Pushbroom images consist of ten of thousands of lines with pinhole geometry.

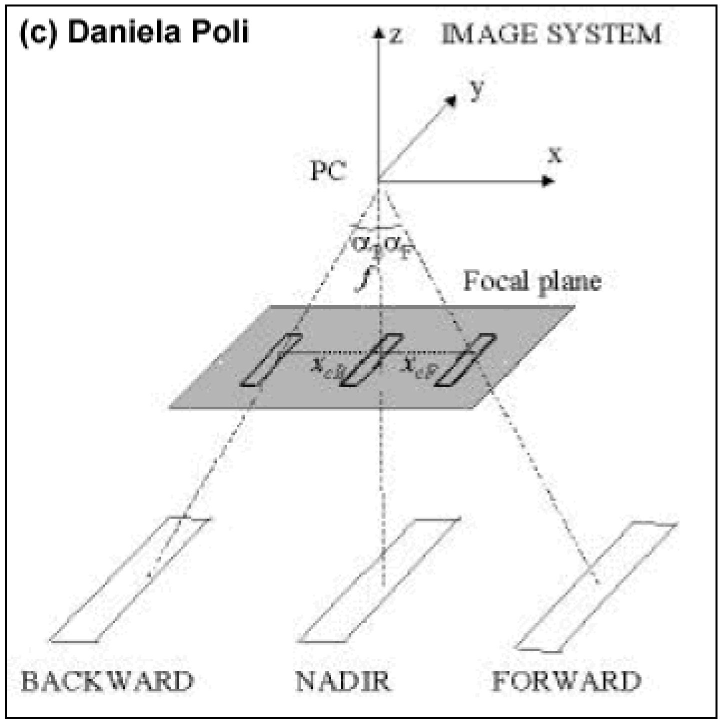
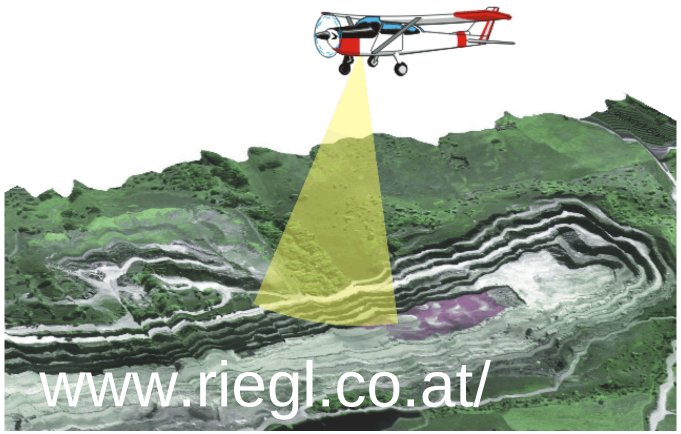
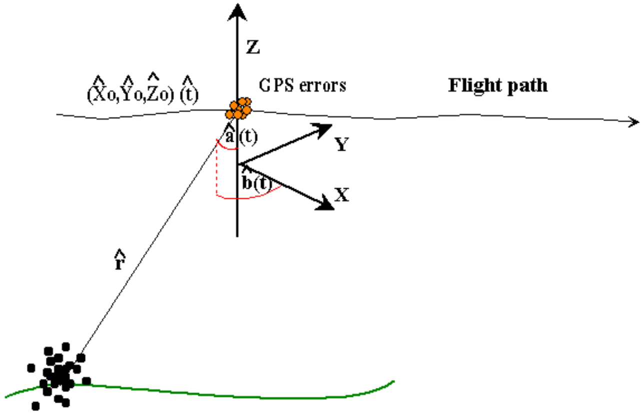
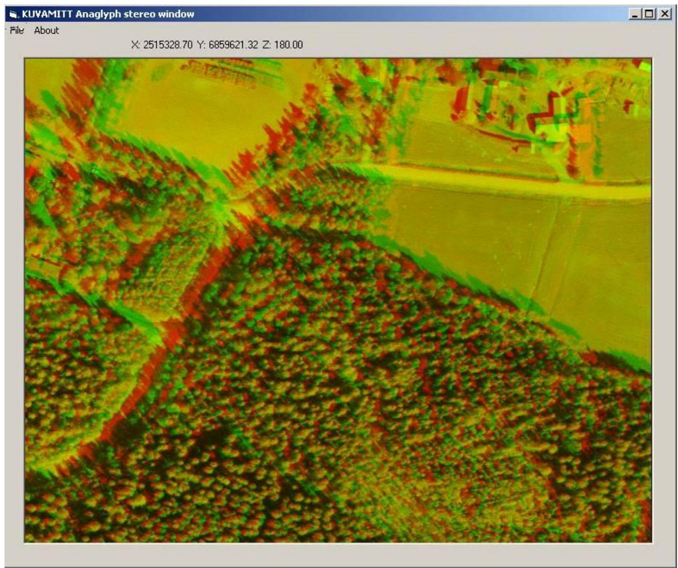


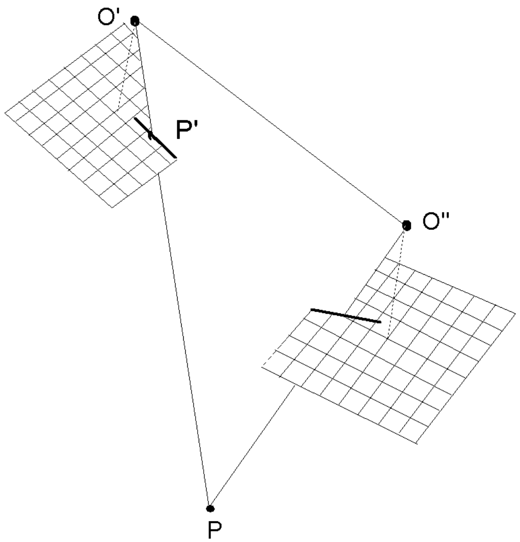
Figure 1. Geometry of CCD linear array sensors with three viewing directions.

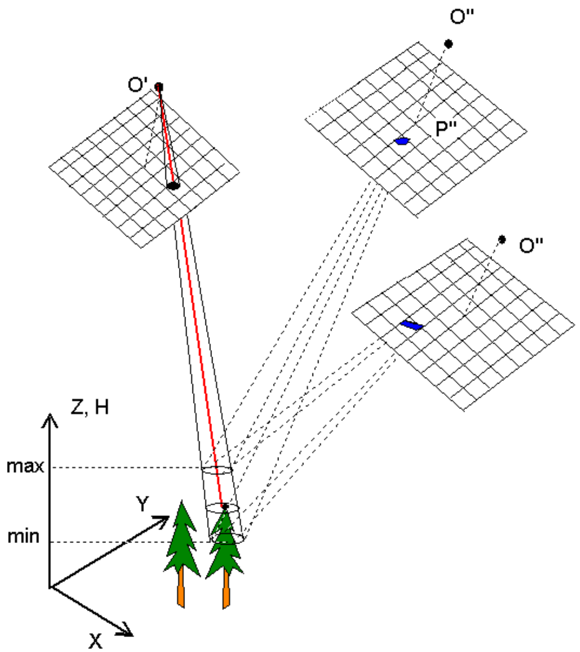
OK let’s measure in 3D using visible light











... Orientation is known; Image pair → 3D target

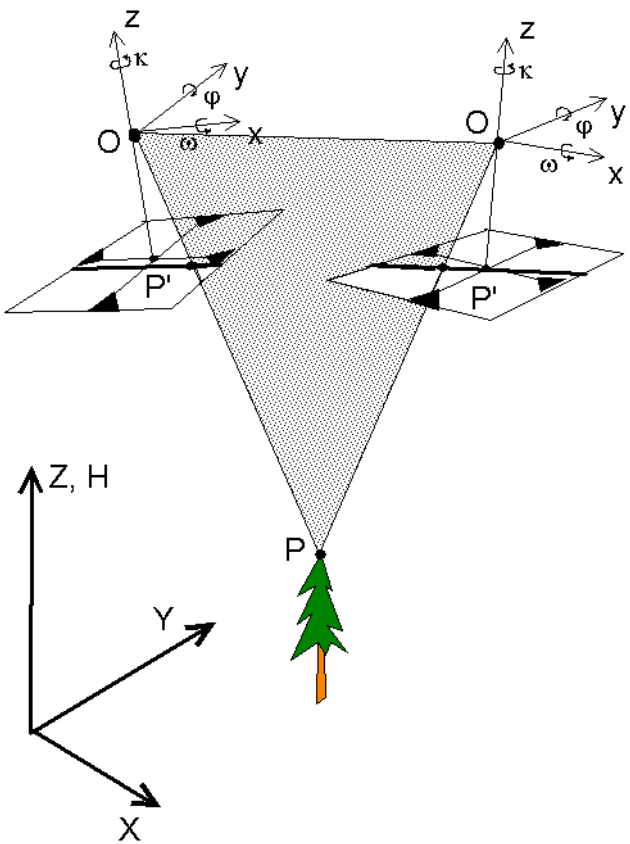


Image to target rays:

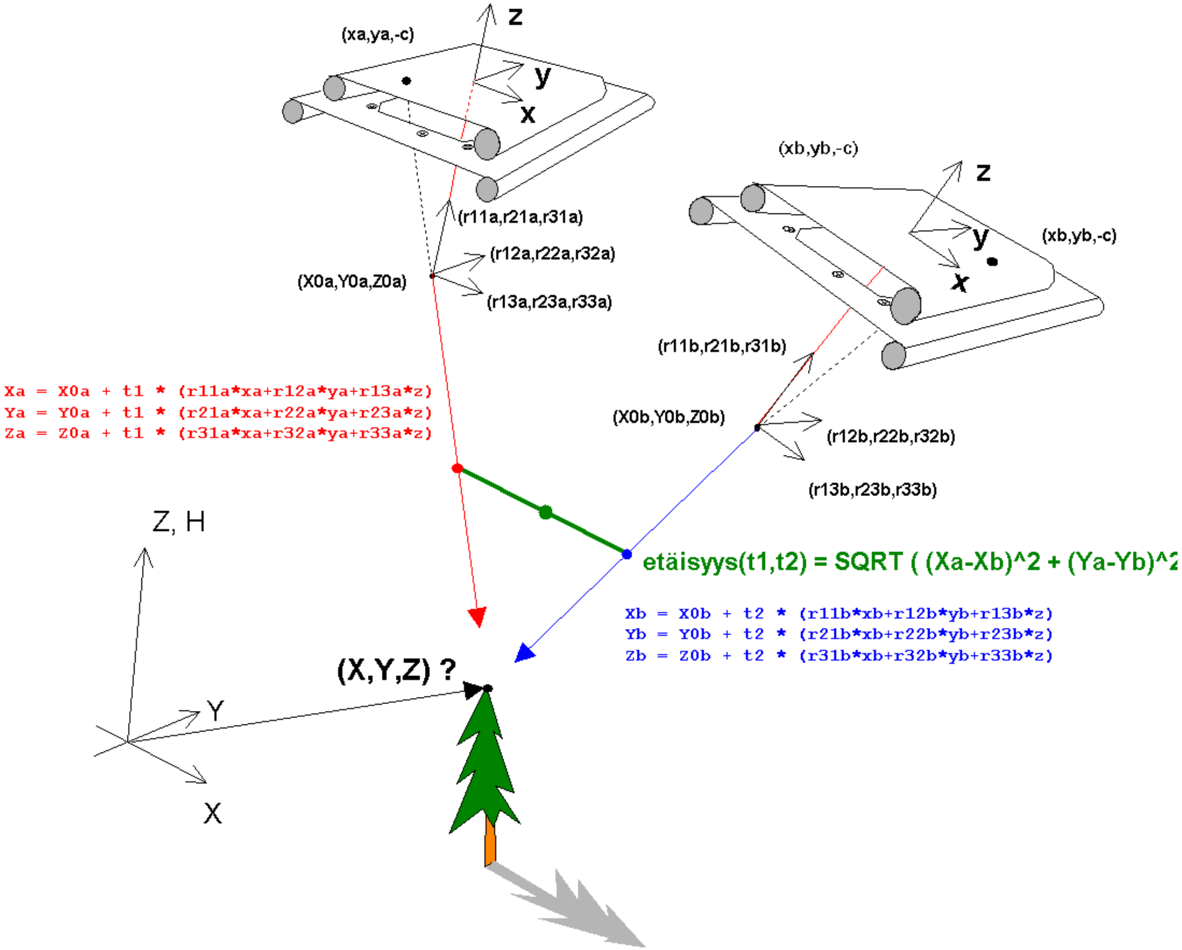
```
X = X0 + t * (r11*x+r12*y+r13*z)
Y = Y0 + t * (r21*x+r22*y+r23*z)     Image 1
Z = Z0 + t * (r31*x+r32*y+r33*z)

X = X0 + t * (r11*x+r12*y+r13*z)
Y = Y0 + t * (r21*x+r22*y+r23*z)     Image 2
Z = Z0 + t * (r31*x+r32*y+r33*z)
```

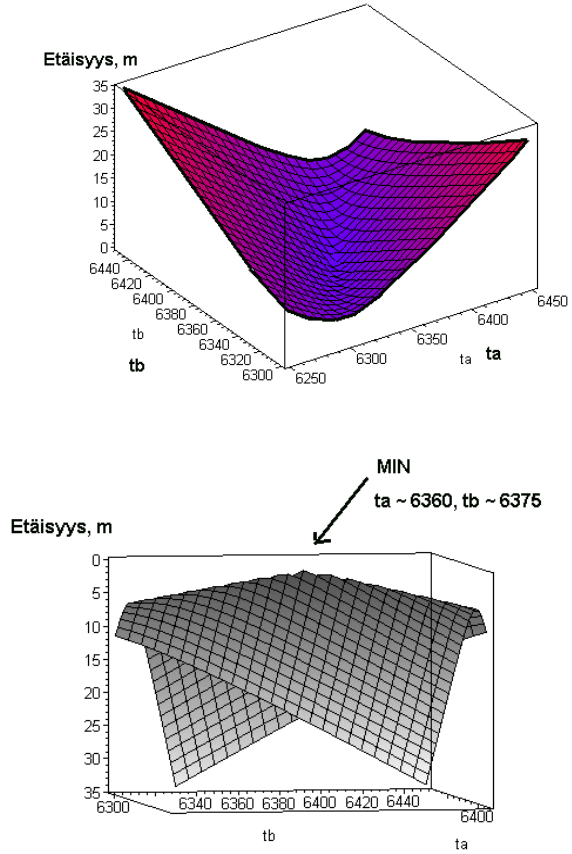
Unknowns: Position of point P, where image rays meet. The lengths of the rays define this position: t_1 and t_2.

Known: Projection centres X_0, Y_0, Z_0, cosines r_{ij} and image point in both images (x,y).

Because the orientation parameters (2 x 6, 2 x 3) and the observations are not perfect, the rays do not intersect. (Note that we have 6 equations and 5 unknowns). Solution options; solve sets of 5 equations, or use the redundancy and solve with LS adjustment (non-linear regression).

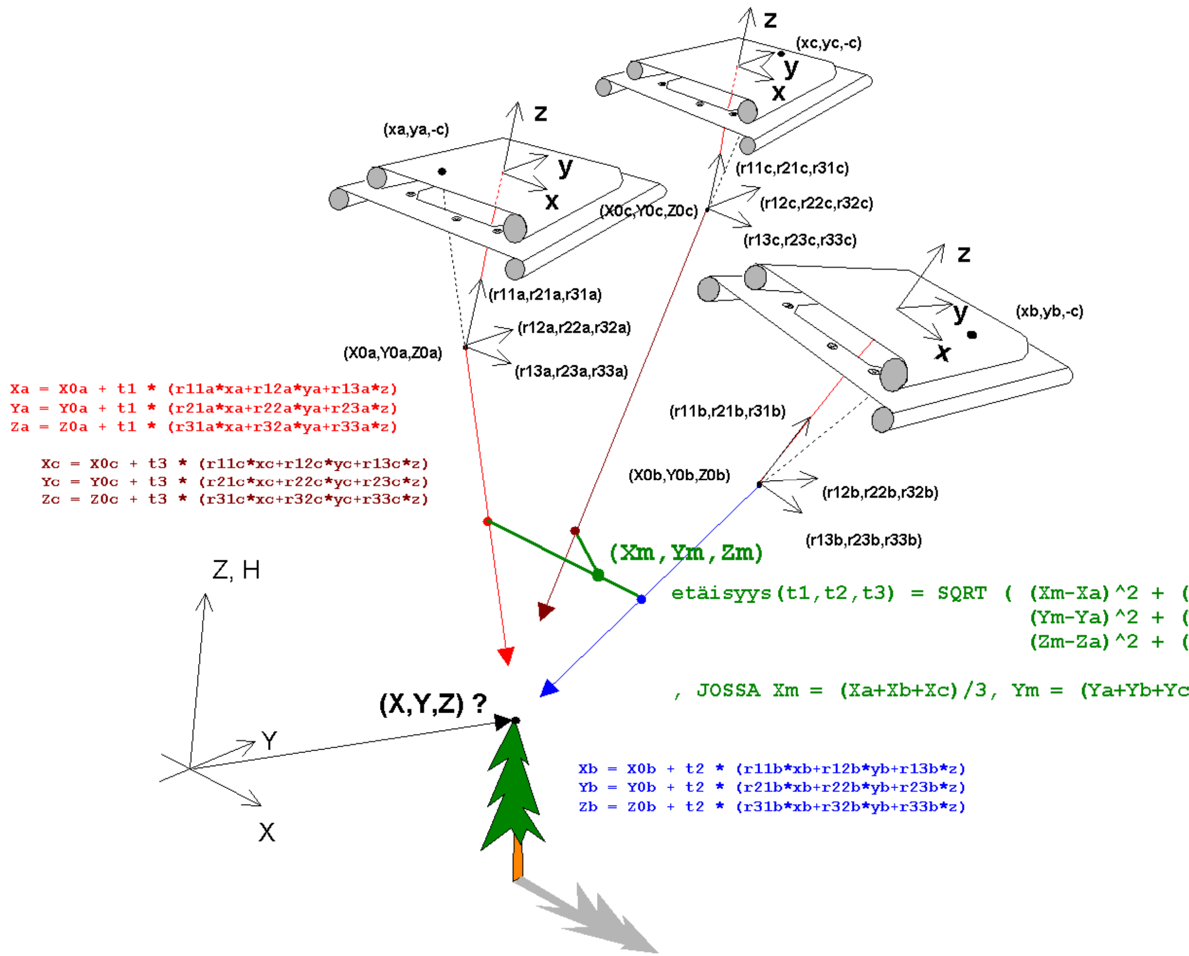
Assume error-free observations

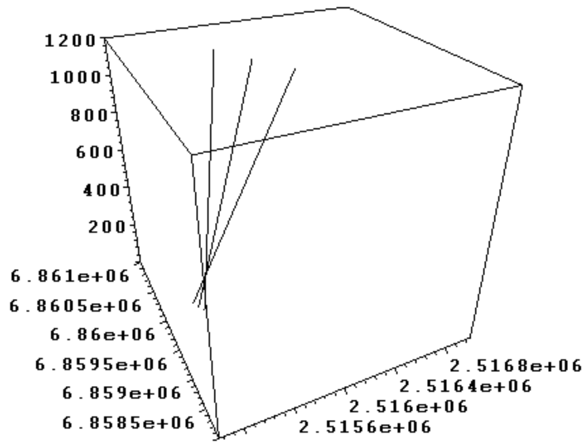




<u>Solution 1</u>: Minimize in (t_1,t_2)- domain the distance between two fixed camera rays. Optimizing $\Re^n \rightarrow \Re$, for example with the gradient method.

3 images → XYZ; error-free observations



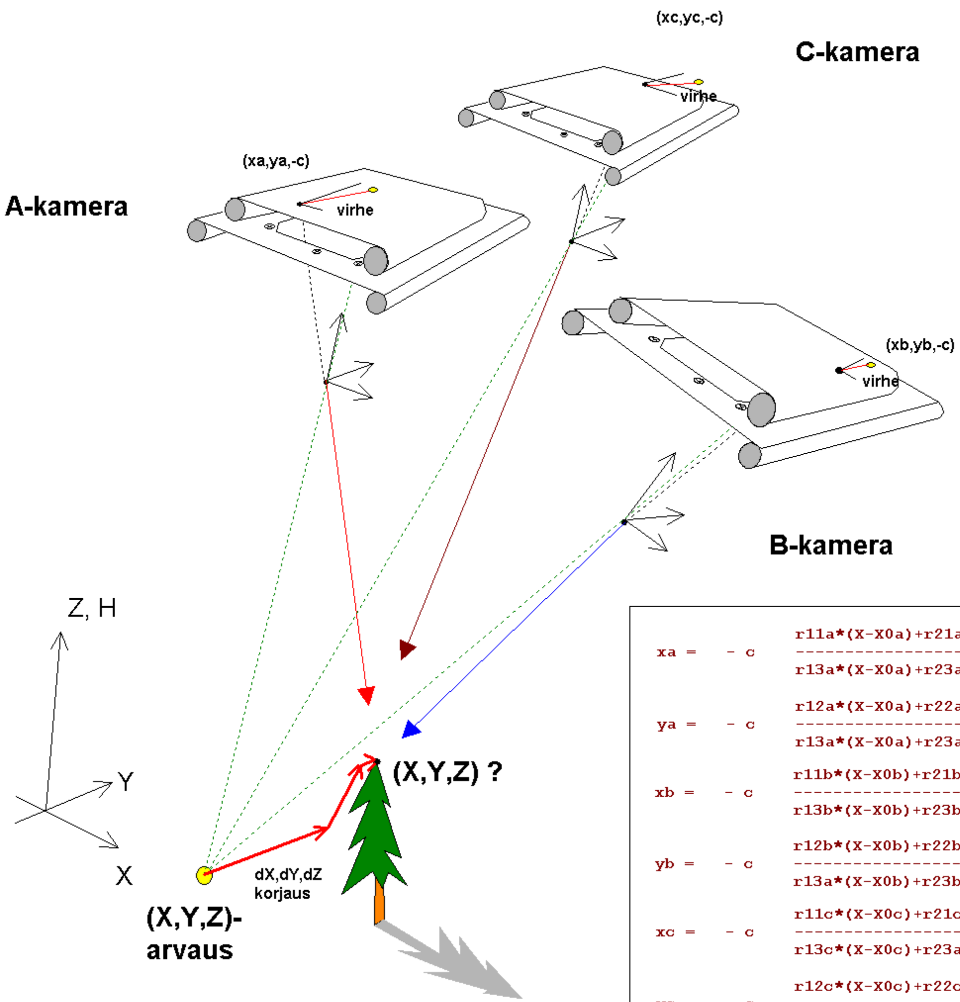


Three images

```
> ratk:=extrema(dist3,{},{t1,t2,t3},'s');
ratk := {.224} (minimal norm 22.4 cm)
> s;
{t1=6359.099310,t2=6376.119592,t3=6412.668636}
```

Solution as mean of t1/t2/t3: **2515366.596, 6859874.913,165.835**

3 images → XYZ   Some errors in OBS



$$xa = -c \, \frac{r11a*(X-X0a)+r21a*(Y-Y0a)+r31a*(Z-Z0a)}{r13a*(X-X0a)+r23a*(Y-Y0a)+r33a*(Z-Z0a)}$$

$$ya = -c \, \frac{r12a*(X-X0a)+r22a*(Y-Y0a)+r32a*(Z-Z0a)}{r13a*(X-X0a)+r23a*(Y-Y0a)+r33a*(Z-Z0a)}$$

$$xb = -c \, \frac{r11b*(X-X0b)+r21b*(Y-Y0b)+r31b*(Z-Z0b)}{r13b*(X-X0b)+r23b*(Y-Y0b)+r33b*(Z-Z0b)}$$

$$yb = -c \, \frac{r12b*(X-X0b)+r22b*(Y-Y0b)+r32b*(Z-Z0b)}{r13a*(X-X0b)+r23b*(Y-Y0b)+r33b*(Z-Z0b)}$$

$$xc = -c \, \frac{r11c*(X-X0c)+r21c*(Y-Y0c)+r31c*(Z-Z0c)}{r13c*(X-X0c)+r23a*(Y-Y0c)+r33c*(Z-Z0c)}$$

$$yc = -c \, \frac{r12c*(X-X0c)+r22c*(Y-Y0c)+r32c*(Z-Z0c)}{r13c*(X-X0c)+r23c*(Y-Y0c)+r33c*(Z-Z0c)}$$

<u>Solution:</u>

LS adjustment (non-linear regression)

Image rays imperfect

1. Initial approximation (X,Y,Z).

2. Solve corrections [ΔX, ΔY, ΔZ], using linearized observation equations.

At solution, the l2-norm of image residuals is at minimum.

3 images → XYZ

$$\begin{cases} x_{(obs)} - v_x = f_x(x_0, c, \omega, \varphi, \kappa, X_0, Y_0, Z_0, X^0, Y^0, Z^0) + \left(\frac{\partial x}{\partial X}\right)^0 dX + \left(\frac{\partial x}{\partial Y}\right)^0 dY + \left(\frac{\partial x}{\partial Z}\right)^0 dZ \\ y_{(obs)} - v_y = f_y(y_0, c, \omega, \varphi, \kappa, X_0, Y_0, Z_0, X^0, Y^0, Z^0) + \left(\frac{\partial y}{\partial X}\right)^0 dX + \left(\frac{\partial y}{\partial Y}\right)^0 dY + \left(\frac{\partial y}{\partial Z}\right)^0 dZ \end{cases}$$

Jokainen kuvahavainto (x,y) tuottaa yhtälöparin, joka linearisoidaan tuntemattomien kohdekoordinaattien suhteen.

$$\mathbf{A_i} = \begin{bmatrix} \left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}}\right)_\mathbf{i} & \left(\frac{\partial \mathbf{x}}{\partial \mathbf{Y}}\right)_\mathbf{i} & \left(\frac{\partial \mathbf{x}}{\partial \mathbf{Z}}\right)_\mathbf{i} \\ \left(\frac{\partial \mathbf{y}}{\partial \mathbf{X}}\right)_\mathbf{i} & \left(\frac{\partial \mathbf{y}}{\partial \mathbf{Y}}\right)_\mathbf{i} & \left(\frac{\partial \mathbf{y}}{\partial \mathbf{Z}}\right)_\mathbf{i} \end{bmatrix}$$

Osittaisderivaattojen arvot (lähtölikiarvoilla X^0, Y^0, Z^0) sijoitetaan **A**-rakennematriisiin, jossa N×2 riviä, 3 saraketta. (N=kuvien lkm)

$$\mathbf{y_i} = \begin{bmatrix} \mathbf{x_{(obs)_i}} - f_{x(i)}(x_0, c, \omega, \varphi, \kappa, X_0, Y_0, Z_0, X^0, Y^0, Z^0) \\ \mathbf{y_{(obs)_i}} - f_{y(i)}(y_0, c, \omega, \varphi, \kappa, X_0, Y_0, Z_0, X^0, Y^0, Z^0) \end{bmatrix}$$

Lähtölikiarvoilla saatujen koordinaattien ja havaittujen koordinaattien erotuksista kootaan virhevektori (**y**)

$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\, \mathbf{A}^T\mathbf{y}$ $\quad \mathbf{x} = \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \end{bmatrix}$

PNS (OLS) –ratkaisuna saadaan parannukset, **x**

3 images → XYZ LS-adjustment

Example Images 1230, 1231, 1232

Three images → Six obs equations → A(6x3), y(6x1)

Iniatial apprx.: 2 515 000, 6 860 000, 200

Linearize to get (**A** and **y**)
Correction vector: $\mathbf{x=(A^TA)^{-1} A^Ty}$, [353.940, -121.008, -33.526]
Adjusted: 2515353.94, 6859878.99, 166.47

Round 2. Update (**A** and **y**)
Correction vector: $\mathbf{x=(A^TA)^{-1} A^Ty}$, [12.629, -4.089, -0.680]
Adjusted: 2515366.57, 6859874.90, 165.79

Round 3. Update (**A** and **y**)
Correction vector: $\mathbf{x=(A^TA)^{-1} A^Ty}$, [0.008, -0.002, 0.000]
Adjusted: 2515366.58, 6859874.90, 165.79
cf. (t1,t2,t3): 2515366.60, 6859874.91, 165.84

DEMO 3
eteenpainleikkaus.xls

N images → XYZ (accuracy)

$$\sigma_0 = \sqrt{\frac{\mathbf{v}^{\mathbf{T}}\mathbf{v}}{2N-3}}$$

"sigma-null", [μm]

$$q_{kk} = diag((\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1})$$

SDx SDy SDz (cf . GPS #DOP valus)

$$\hat{\sigma}_{xk} = \sigma_0 \sqrt{q_{kk}}$$

"Naive" as $\sigma_0 \to 0 \to \sigma_{xk} \to 0$

Orientation errors not included (error accumulation)

N images → XYZ (accuracy)

So how accurate xyz can we measure with images?

Errors are made

1) Pointing the exactly same target in N images
   - N non-intersecting rays

2) Geometry of the cameras; we usualy favor 2 coordinates at the expense of a third

3) Distance to target; inaccuracy in ray direction causes a larger effect for larger distances.

4) **Sensor** (lens) can fool us, the rays are not collinear

A few notes on sensors (geometry)

1-lens vs. Multi-lens (CCD) systems

Motion compensation (75-100 m/s, 1/200 s, ~ 20-40 cm)

Metric cameras & other cameras (stability)

DSO: GPS or GPS/imu

MTF of lens; pixel size.

How to automate image based 3D measurements?

Finding automatically in the N images the corresponding image features is the KEY to success!

How to?

Constrain & Use Top-Down approach

- epipolar geometry
- a priori knowledge about the geometry

$\Rightarrow$ matching area in images is reduced

- measure similarity in images to quantify correspondence
- solve for XYZ
- filter the point cloud for outliers

How to automate image based 3D measurements?

- similarity in images to quantify correspondence

....

Usually works fr smooth surfaces;
Edges, sharp features are difficult
Occlusion destroys everything

Tree crowns are sharp peaks; canopy is rough; lots of occlusions

⇒ Image matching usually fails in giving heigths correctly; and canopy gaps are missed

⇒ Go for treetops (Korpela, 2000, 2004, 2007)
Build an interest operator for tree tops
Concentrate search in the upper canopy

Problems: trees greatly vary in shape, size, color.

A look at LiDAR (geometry)

Pulsed systems; some key features

- Quantization of the returning echo; discrete or full waveform

- 6 EO parameters for each pulse; error structures: time, strip, project level

- Range discrimination accuracy; blind range?;

- XYZ accuracy of a return (point); pulse; which?

- backscatter photon-count measurement accuracy (amplitude)

- Minimum detectable object

- divergence / footprint (% of energy)

- scan angle; what do we measure?

LiDAR vs. Photogrammetry (geometry)

- LiDAR is an active sensor;

- Geometry relies on DSO, if it fails, no data

- XYZ data is acquired ”on-the fly”; no correspondence problem, usually no outliers in the point cloud; no need for texture in the object!

- Point patterns are irregular; not always optimal, Images have a higher sampling rate from the same flying height, regular sampling $\Rightarrow$ no problem in smooth surfaces, but tree reconstruction requires a high sampling rate.

- LiDAR systems can extract several XYZ-points per pulse, thus penetrate the vegetation; reaching ground. This is a fundamental diffrence nd benefit!

SUMMARY (geometry)