SHEET-OF-LIGHT Laser profile triangulation

— Laser planes (⇒ lines) projected on object ⊕ camera records projected lines : Z values of profiles
↓
— Profiles recorded with known advancement velocity yield object.

— We can have a setup which is:

- <u>CALIBRATED</u>, returns values
  - Disparities (≠ stereo disparity!) : ROW values of each profile
  - X, Y, Z images in WCS
  - 3D model
- UNCALIBRATED, return values $\begin{cases} \text{- disparity image} \\ \text{- score of reliability} \end{cases}$

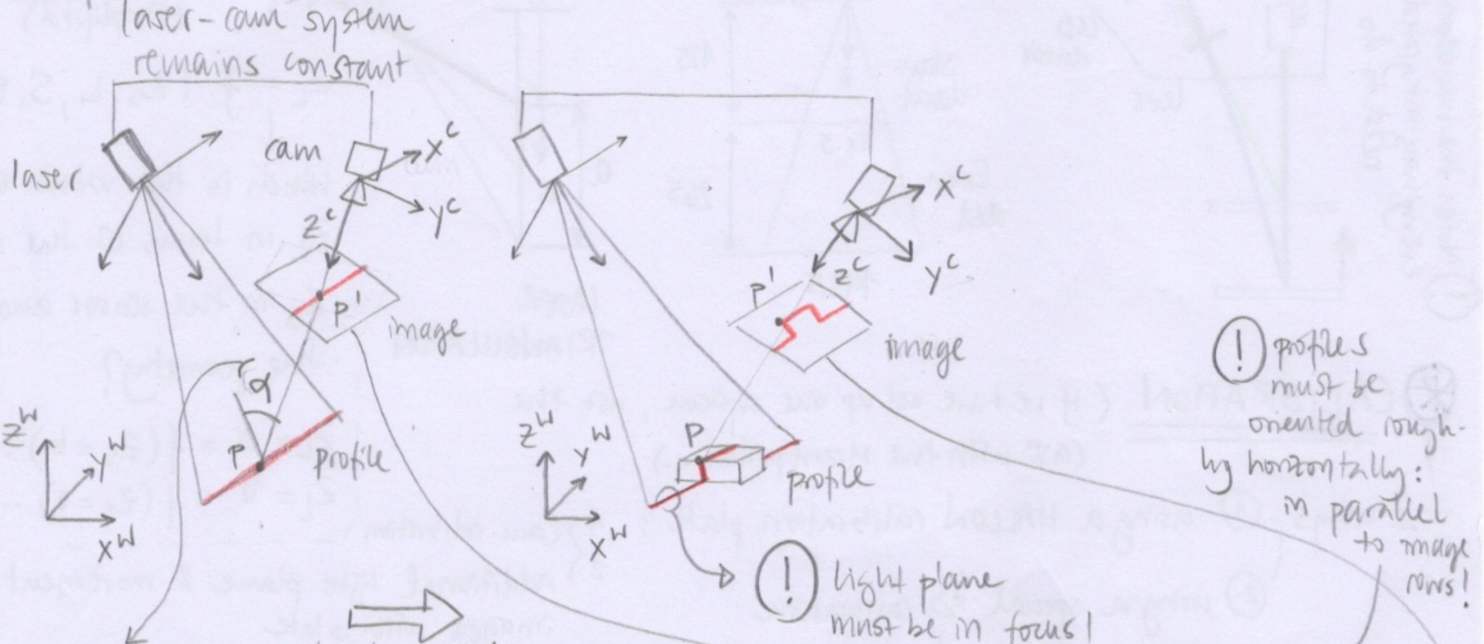
Note: all these calibration & measurement procedures are necessary if the cam (2D) & laser are not integrated in a single unit! If they are, we get calibrated pointclouds!

⊛ SETUP

→ <u>Elements</u>
1. Laser plane/line projector
2. camera which records line profile
3. positioning system, eg. a linear conveyor belt

(1, 2: these can & usually go together)

→ <u>Principle</u>



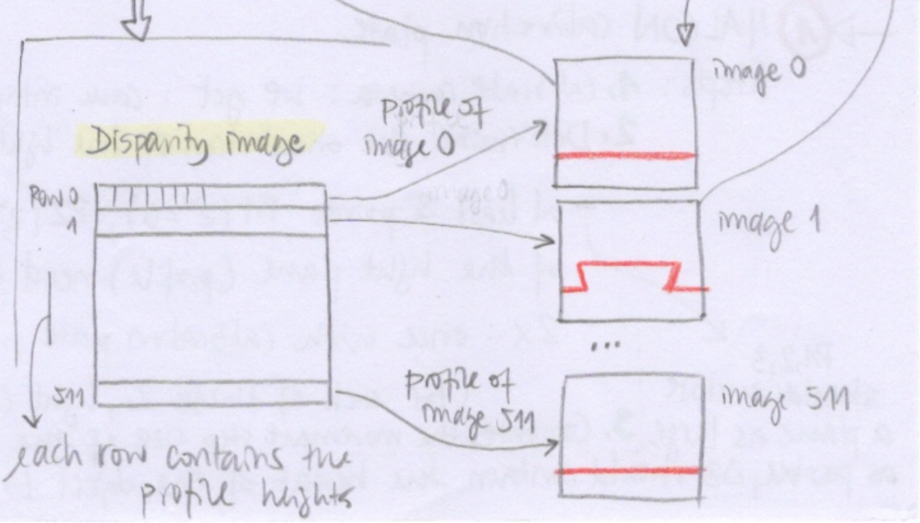
ANGLE OF TRIANGULATION: angle between laser light plane & camera axis

$-$ $\alpha = 30°$ $+$

less accuracy — more accuracy

recommended: 30°–60°

if more, occlusions increase

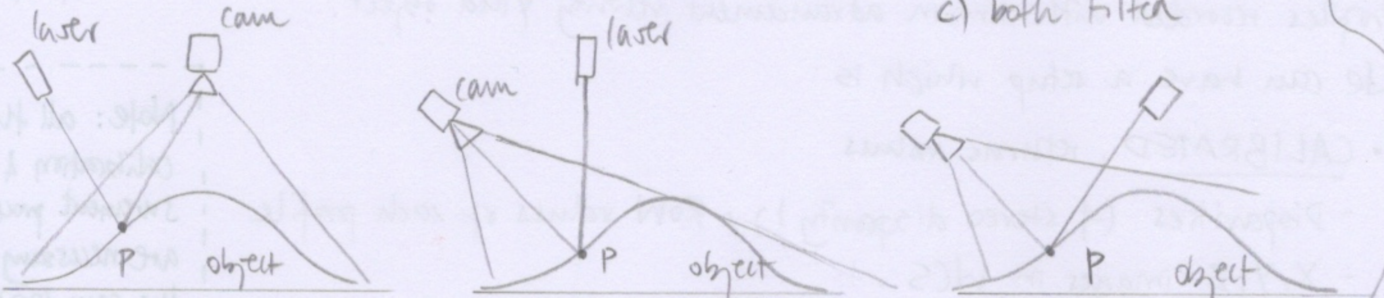


→ <u>Camera - laser configurations</u> : choose according to object geometry, so as to decrease occlusions / shadowing (eg, laser not projected or line not seen)

a) laser: tilted
cam: orthogonal

b) laser: orthogonal
cam: tilted

c) both tilted



Note: each configuration results in different projections of the profile on the image!

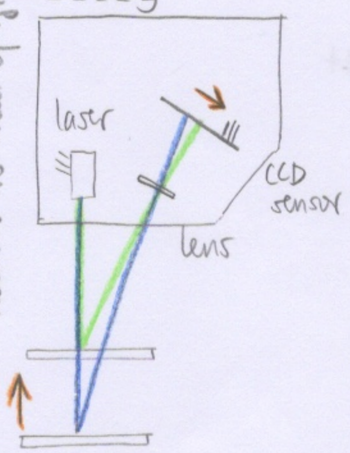
→ <u>Microepsilon Configuration</u> (example of a concrete configuration)

scanCONTROL 2900-100/BL [→ see my notes on laser triangulation!

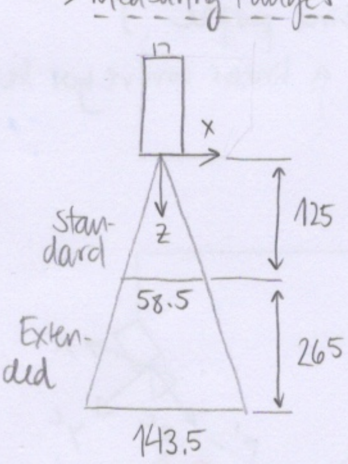
blue, PoE, 1280 px / profile, 12 μm (reference resolution?), ≤ 300Hz

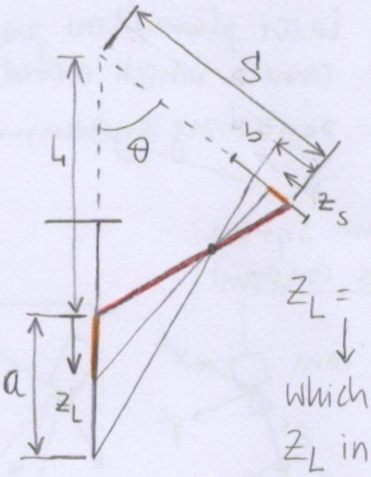
(!) Note: the microepsilon is a device with everything integrated — we really don't need to do Sheet-of-light

→ <u>Geometry</u>



→ <u>Measuring Ranges</u>





LASER TRIANGULATION

→ Geometrical definition of the problem (simplified)

$$Z_L = f(z_s, L, S, \theta, a, b)$$

which is the value of Z_L in terms of the distance Z_s in the sensor and the geometry?

$Z_L = a = f(z_s = b) \ldots$

$Z_L = 0 = f(z_s = 0) \ldots$

* <u>CALIBRATION</u> (if we have set up our system, not the case with the MicroEpsilon...)

— 2 ways ① using a HALCON calibration plate : 1) cam calibration
2) additional light plane & movement images with plate

② using a special 3D calibration object printed and scanned once while still uncalibrated; then calibration automatically performed

→ ① HALCON calibration plate

Steps: **1.** Calibrate camera : we get : cam intrinsic params + extrinsic (pose) → conventional cam calibration

2. Determine the orientation of the light plane wrt WCS : from it, we have the angle of triangulation α

- at least 3 points $P1\,(z^w = 0)$, $P2\,(z^w = 0)$, $P3\,(z^w \neq 0)$ of the light plane (profile) need to be acquired

2x : once with calibration plate, once with laser line (for each of these 2, light conditions change!)

P1,2,3 should enclose a plane as large as possible, Δz should contain the height of the object to scan later

3. Compute the movement step size of the conveyor belt → for that we need at least 2 images, in each cal plate has been linearly moved by conveyor belt a known number of steps

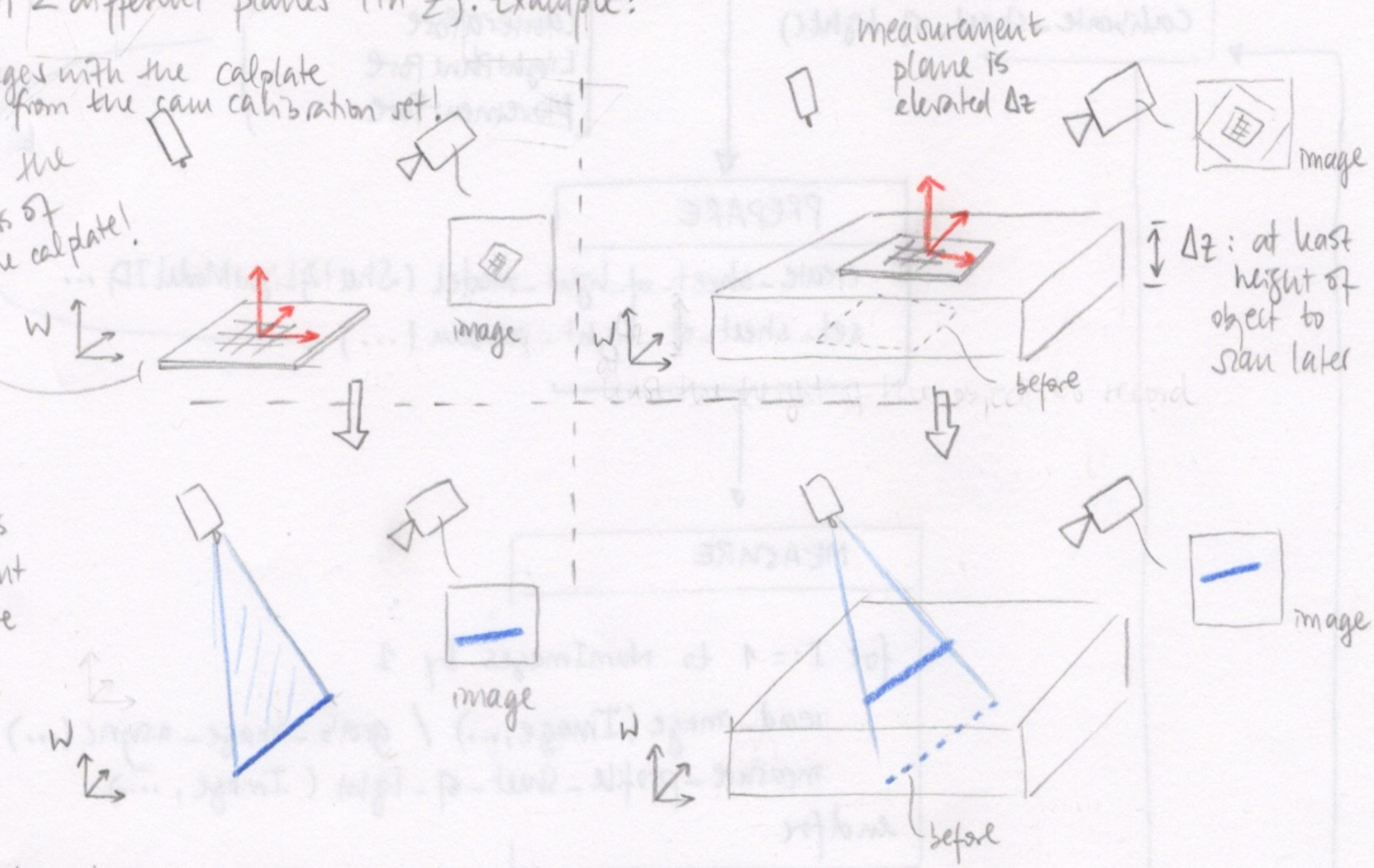
Remark on step **2**: it's confusingly explained here and in the solution guide, but as far as I understood from the example calibrate_sheet_of_light_calplate.hdev

- For step 2 we need 2×2 images that show pair-wise the laser line and the calplate in 2 different planes (in Z). Example:

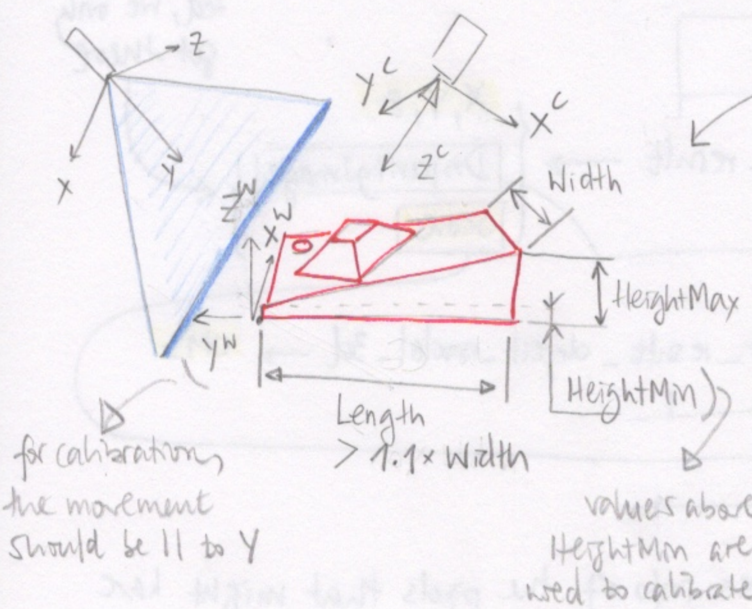
the 2 images with the calplate can be 2 from the cam calibration set!

consider the thickness of the calplate!

same config as before, but we acquire laser line



→ **2** (calibrate using a 3D printed object (not as accurate as calibrating with Halcon plates)



Steps:

1 Create a CAD/DXF model in Halcon with custom parameters/sizes
→ create_sheet_of_light_calib_object()
NOTE: size should be similar to the objects to scan later!

2. Manufacture object / print it in 3D

3. Calibrate:
- Measure profiles of cal object
  measure_profile_sheet_of_light(...)
- Get/Set disparity image
  get_sheet_of_light_result(...)
  set_profile_sheet_of_light(...)
- Calibrate: calibrate_sheet_of_light

(*) **USE** : simplified summary of workflow

see previous sections

2 possible calibrations
1) calplate
2) 3D object

CALIBRATE

get_sheet_of_light_param():
[CameraParameters, CameraPose, LightPlanePose, MovementPose]

...
calibrate_sheet_of_light()

PREPARE

create_sheet_of_light_model(SheetOfLightModelID, ...
set_sheet_of_light_param(...)

MEASURE

```
for I := 1 to NumImages by 1
    read_image(Image,...) / grab_image_async(...)
    measure_profile_sheet_of_light(Image, ...)
endfor
```

GET RESULTS

get_sheet_of_light_result → { X, Y, Z ; DisparityImage ; Score }

if uncalibrated, we only get these

get_sheet_of_light_result_object_model_3d → OM3

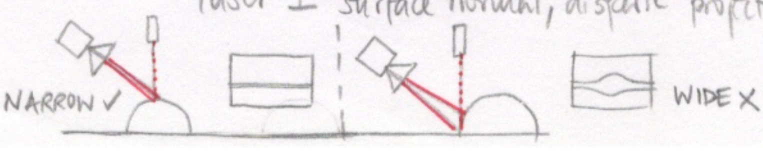
apply_sheet_of_light_calibration

if we record uncalibrated disparity images and calibrate afterwards, we can later use the calibration on the uncalibrated disparity images to get all calibrated results!

Score

- It gives info of the pixels that might have artifacts: threshold(Score) → Region: use/reject that domain in Disparity image!
- 2 types of artifacts:

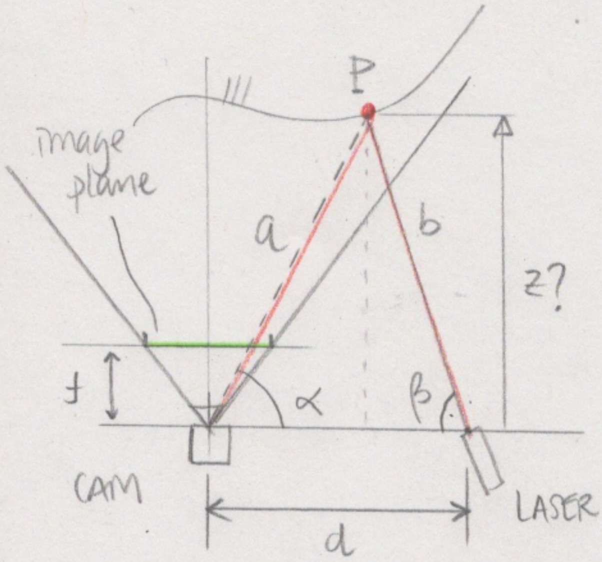
(1) Geometry of surface: if laser ⊥ surface normal, disperse projection



(2) Laser speckle
noisy distribution due to roughness

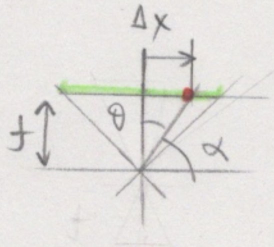
Laser triangulation (my notes - my derivation, I don't know if it's correct...)

Basic principle for lasers & also stereo vision



- Goal: we want to obtain z.
- We know (data): d, distance between laser & cam
  β, inclination of the laser wrt cam

1) We obtain α from cam image:



$$\tan\theta = \frac{\Delta x}{f}$$

$$\left[\alpha = 90^\circ - \operatorname{atan}\left(\frac{\Delta x}{f}\right)\right]$$

2)
$$\left.\begin{aligned} z &= \overset{?}{a}\sin\overset{\checkmark}{\alpha} \\ z &= \underset{?}{b}\sin\beta^{\checkmark} \end{aligned}\right\} \longrightarrow \overset{?}{a}\sin\overset{\checkmark}{\alpha} = \overset{?}{b}\sin\overset{\checkmark}{\beta}$$

$$a = b \cdot \frac{\sin\beta}{\sin\alpha}$$

$$b = a\,\frac{\sin\alpha}{\sin\beta}$$

3) a, b?

$$a\cos\alpha + b\cos\beta = \overset{\checkmark}{d}$$

$$a\cos\alpha + a\,\frac{\sin\alpha}{\sin\beta}\cdot\cos\beta = d$$

$$\left[a = \frac{d}{\cos\alpha + \frac{\sin\alpha\cos\beta}{\sin\beta}}\right]\checkmark \longrightarrow$$

[2] $\left[z = a\cdot\sin\alpha \ \checkmark\right]$

$$\left[\left[z = \frac{\sin\alpha\sin\beta}{\sin\alpha\cos\beta + \cos\alpha\sin\beta}\cdot d\right]\right]$$

Note: actually, I think this technique was used to compute the distance of a ship from the coast?