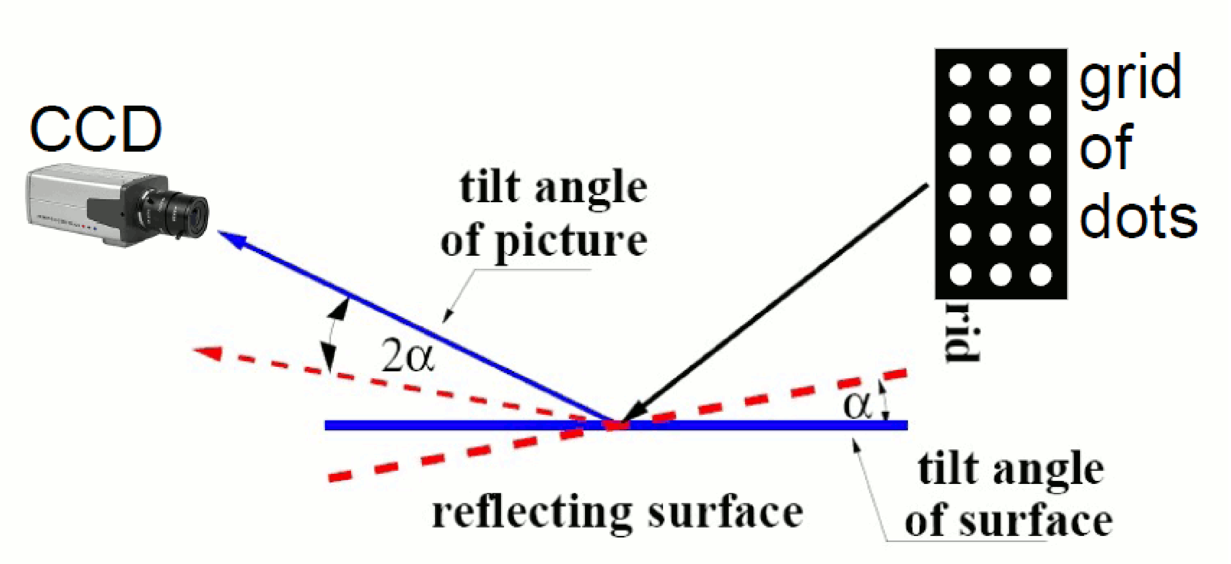
optical deformation measurements - setup plans & software

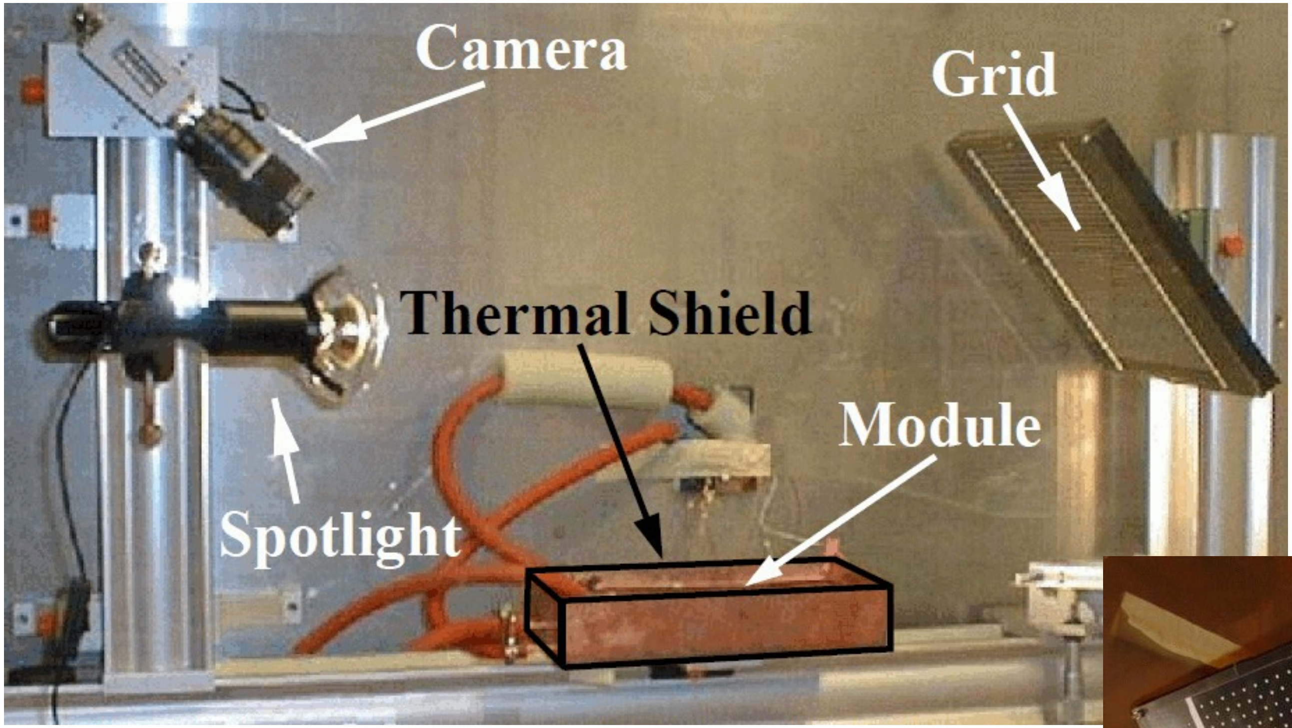
basic principle

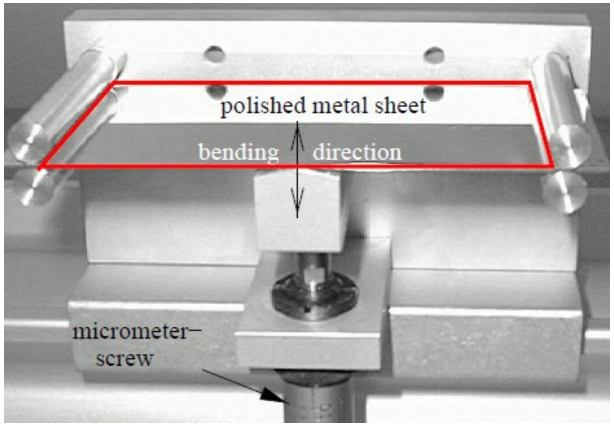


- surface under test reflects a grid of dots or LEDs, observed by a CCD
- tilt of surface elements (deformations) lead to a displacement of the dots on the CCD
- full 3D “sampled” surface reconstruction possible –
  - relative to a reference shape/image (typically at room temperature)
  - accuracy ~10 µm can be reached
  - only “intrinsic” deformation – global offsets or tilts cannot be reconstructed

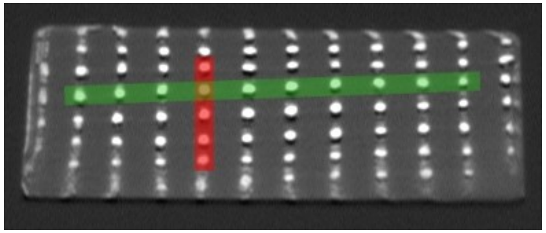
the Aachen setup

Built in Aachen-1 by S. König (now PSI) 2000/1,
original idea/implementation from H.G. Moser et al. (MPI Munich)

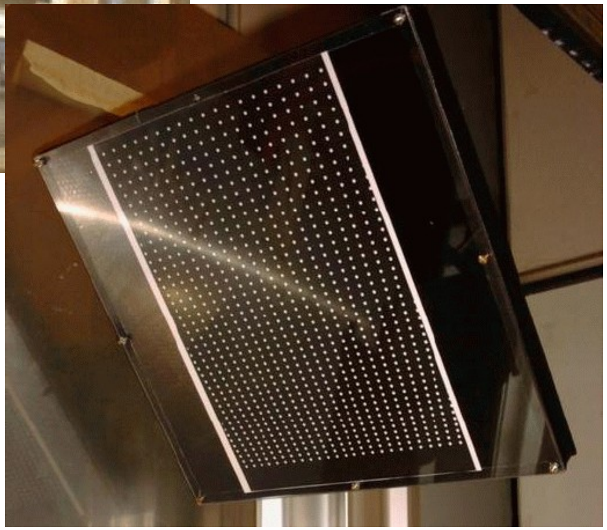




calibration device



First prototype
of FR4 hybrid
(copper coating)



the
dot
grid

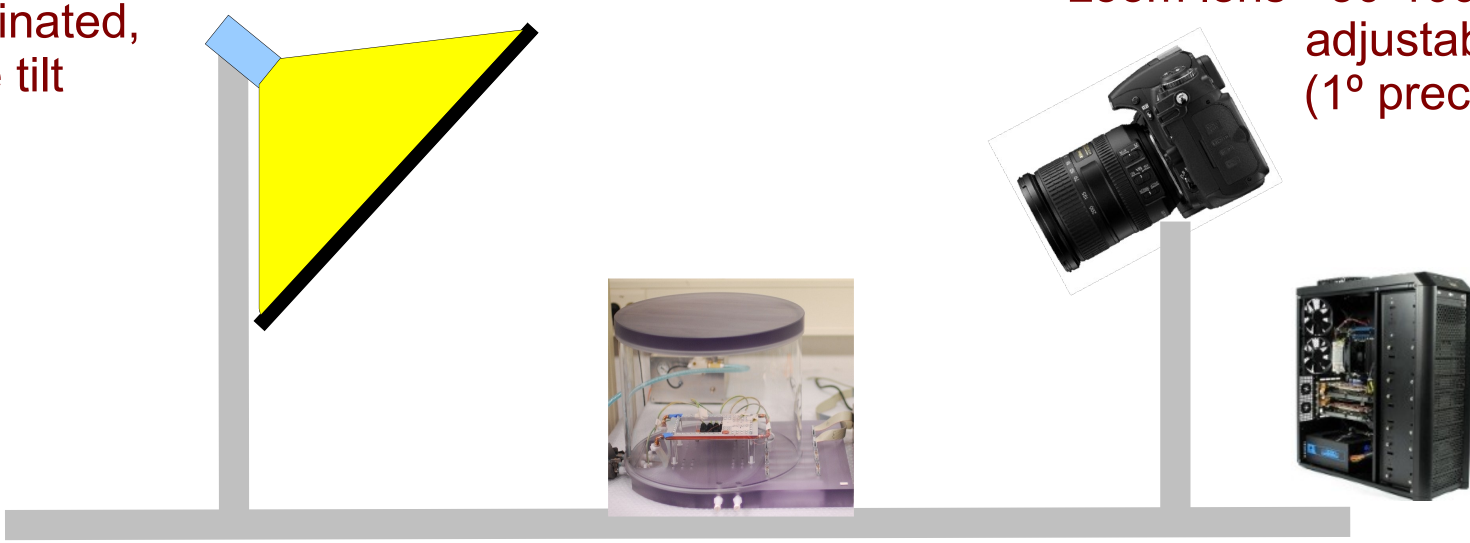
plans for the DESY setup

Setup parameters to be optimized in simulation -
- distances grid-sensor-camera, focal lens range (→ size of DUTs?)
- grid dimensions and spacing

started now since software is in a useable state!

dot grid:
rigid plate with bores,
back illuminated,
adjustable tilt

camera:
cheapest color SLR,
zoom lens ~30-100 mm,
adjustable tilt
(1° precision)



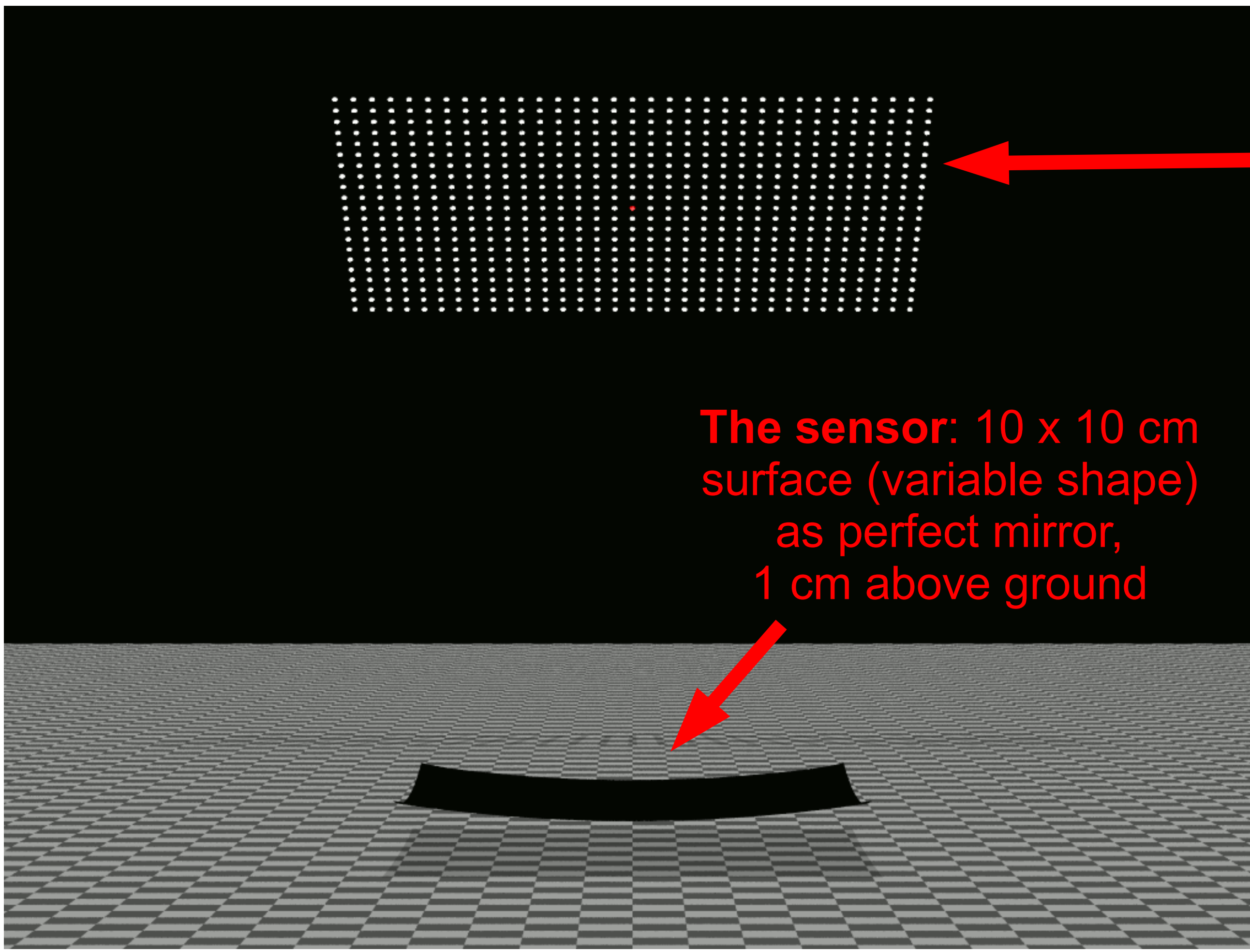
new
fast
PC

alu profiles

“old” temperature setup,
elevated table (no cover)

simulating the setup

Simulation is done with POVRay (open source raytracer program)



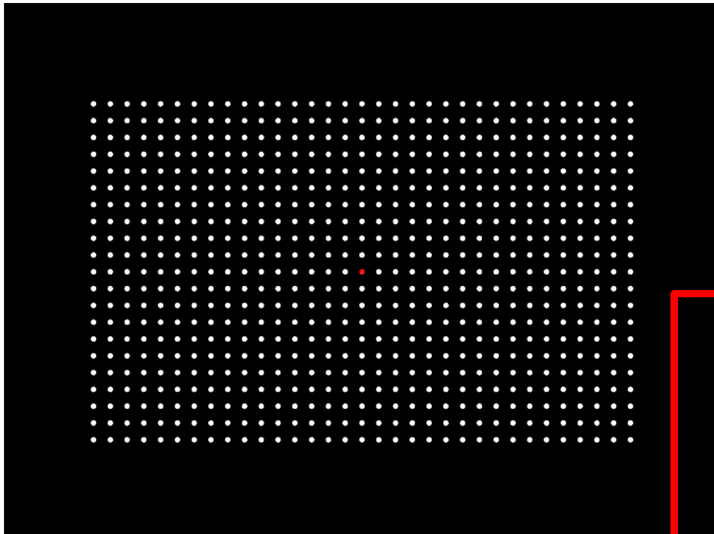
The dot grid: 16 x 10 cm,
33 x 21 white LEDs
(middle one is red)
80 cm away from the camera,
40 cm above the ground,
45º tilt

The camera:
40 cm above ground,
45º tilt,
800x600 pix (10 μ pitch),
f = 30 mm

simulating the setup

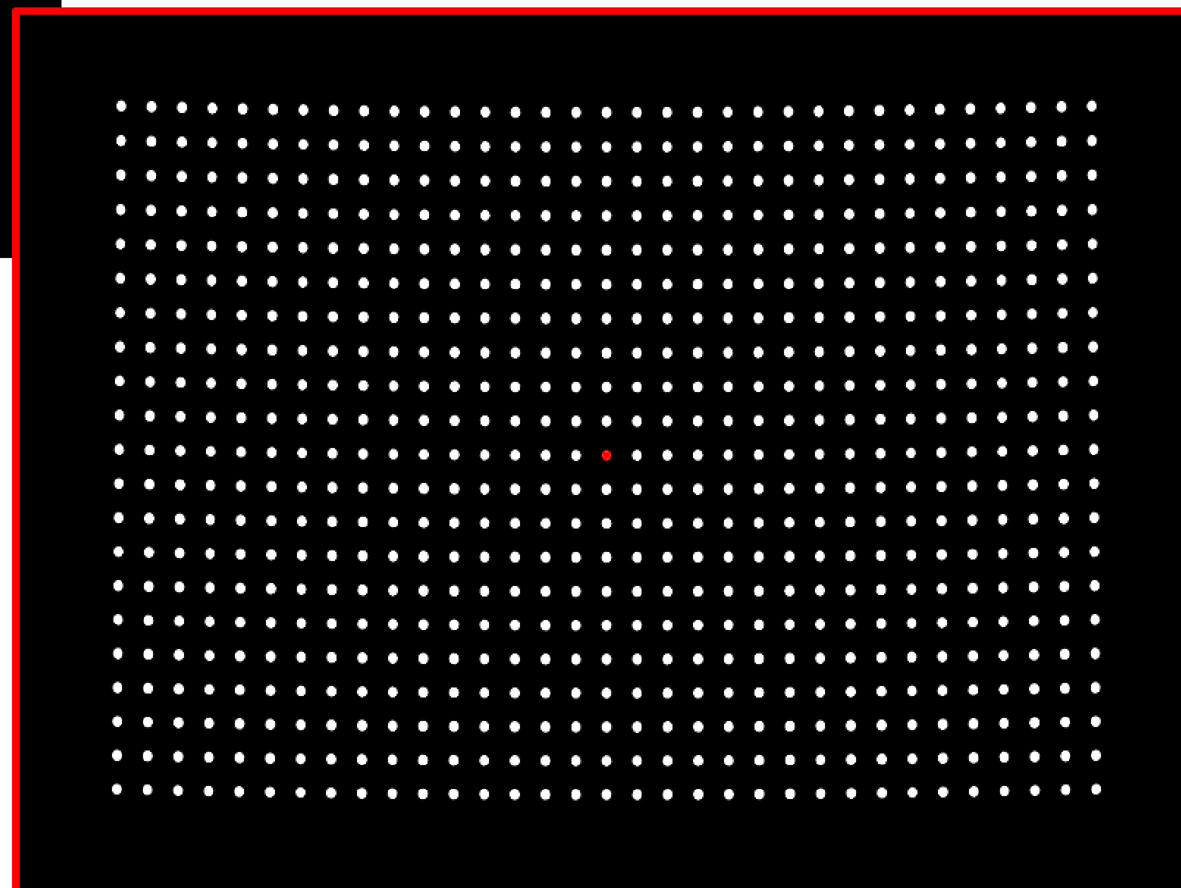
what POVRay is
normally used for...

surface reconstruction: raw input

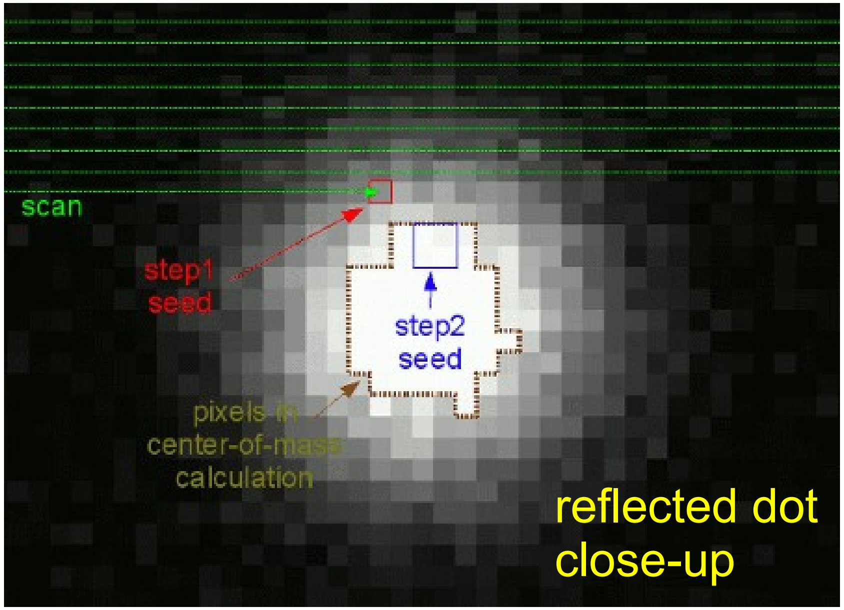


"flat" reference surface

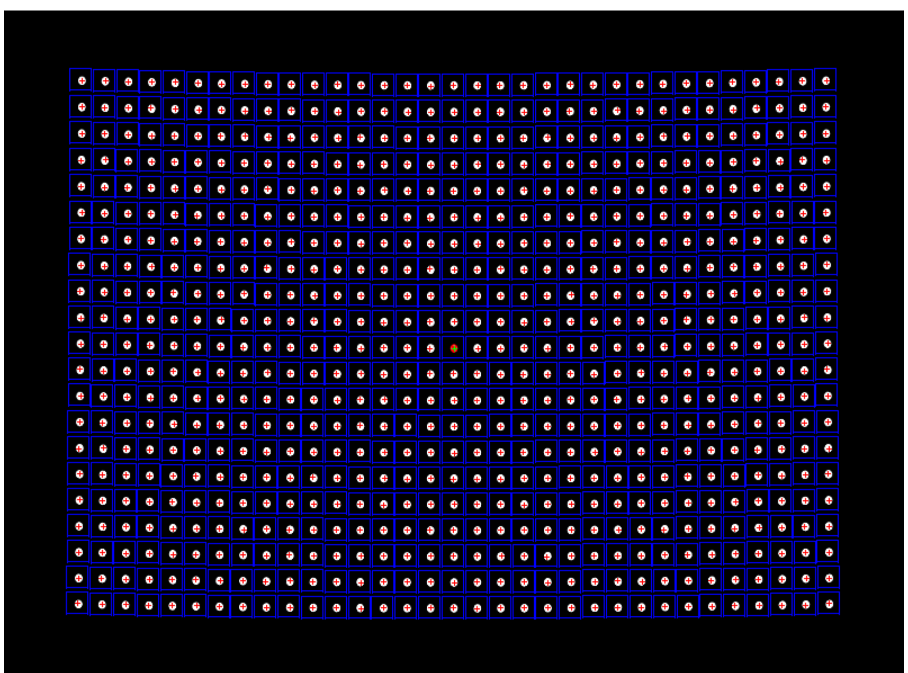
"current" image
300 µm bowing along both x,y:
$z = 300\ \mu m * (x^2 + y^2)$

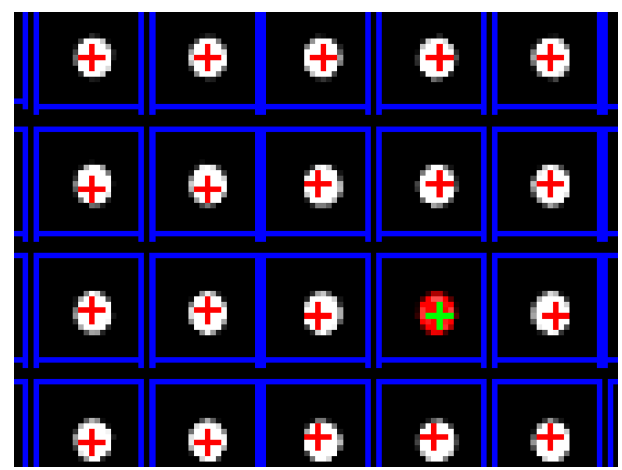


surface reconstruction: point finding



- scan along x,y row-wise and compute grayscale ADC values
- stop at pixel > threshold **T1** (step1 seed)
  - evaluate group of pixels in vicinity
  - if average > threshold **T2** (step2 seed)
    - accept point
    - determine point position by center-of gravity of all pixels around seed2 with grayscale ADC > threshold **T3**
    - average red/cyan ratio of pixels to determine if the point is “red”

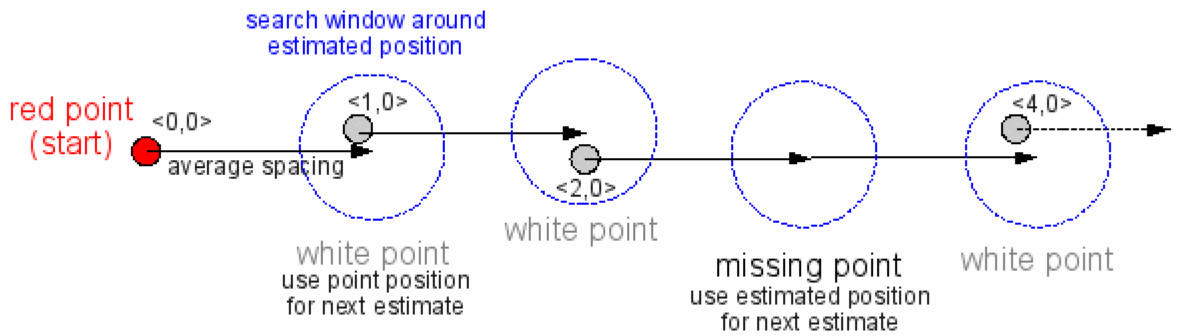




surface reconstruction: point indexing

Must consider that points might be missing:
- gaps in multi-component surfaces
- points not properly reflected (diffraction, surface quality, …)

Indexing assigns position indices (0..N, 0..M) to the reconstructed points:
- 1:1 matching of reflected points to the grid
- red point in the middle is used as reference (0,0)
- average spacing along x and y is determined to estimate
  the positions of the next point(s)



Indices are used to match points in reference and “current” image

Requires proper tuning of reconstruction & setup parameters
(size of search window, distance of dots in grid, etc.)

surface reconstruction: calculating the surface slope

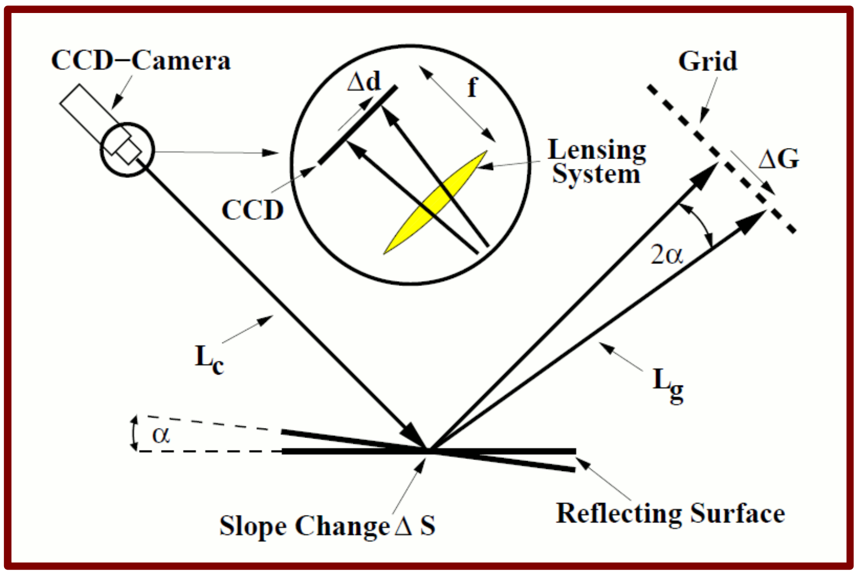
Parameters for determining
the surface slope:

- focal length of camera lens f
- CCD pixel pitch P
- distance camera-sensor Lc
- distance sensor-grid Lg

depend on viewing angle

Point shift on sensor Δd
(between current & reference image)
is measured

$$\tan\left[2 \cdot \arctan\left(\Delta S\right)\right] \cdot L_g\left(\delta\right) = \frac{\Delta d \cdot P}{f} \cdot \left[L_g\left(\delta\right) + L_c\left(\delta\right)\right]$$



Gives the surface slope changes ΔS
between reference and current state
at the position of a point
→ spatial integration required

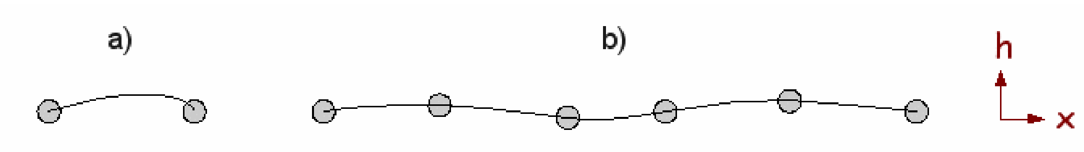
Viewing angle corrections to Lc and Lg
currently not applied (need to calculate)

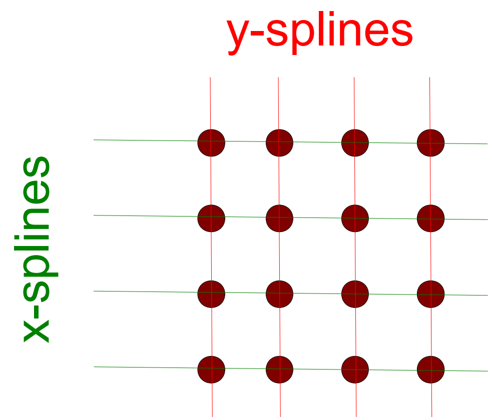
surface reconstruction: spline fitting

Indexed points are grouped in “rows” (along x) and “columns” (along y)
→ fit a quadratic spline to each group

$$h(x) = A \cdot x^2 + B \cdot x + C$$

h := surface height (deformation amplitude)





For all splines along a “row” or “column” :

$$h_j(x_{j+1}) = h_{j+1}(x_{j+1}) \text{ for } j = 1..N-1$$

continuity at the intersections

$$\frac{\delta h_j}{\delta x}(x_{j+1}) = \frac{\delta h_{j+1}}{\delta x}(x_{j+1}) = h_{j+1} \text{ for } j = 1..N-1$$

continuity of 1st derivative

$$\frac{\delta h_1}{\delta x}(x_1) = s_1, h_1 = 0$$

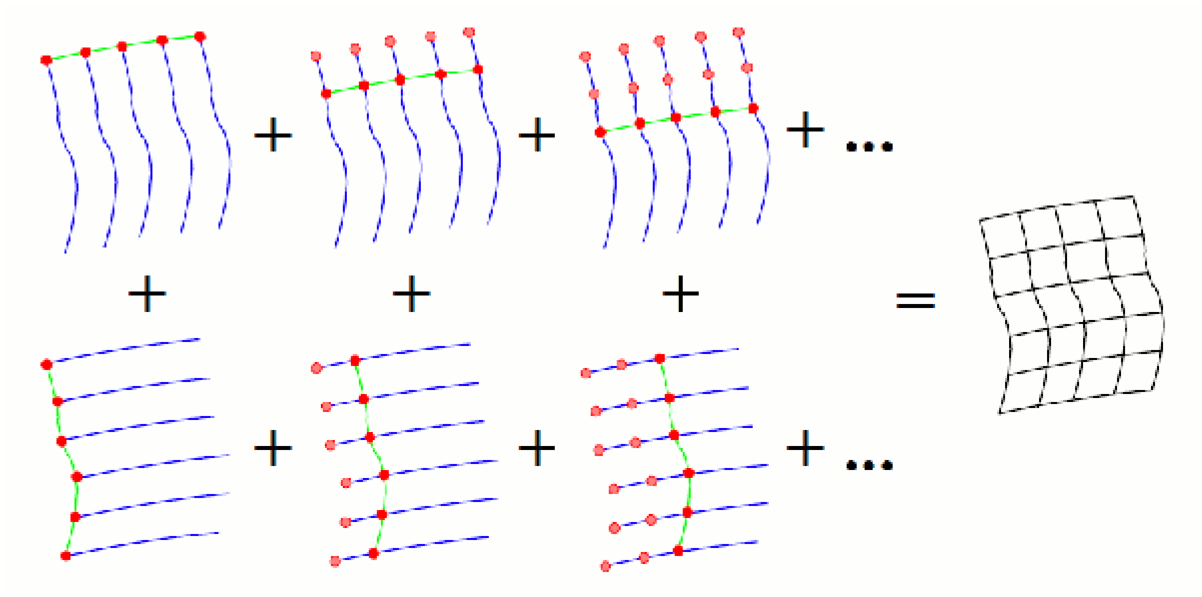
freedom of choice for global height (parameter C)
which cannot be reconstructed
since we have changes in slope ΔS only

→ equation system can be solved to obtain surface profile along spline

surface reconstruction: spline cross-mounting

Splines are “cross-mounted” on each other:

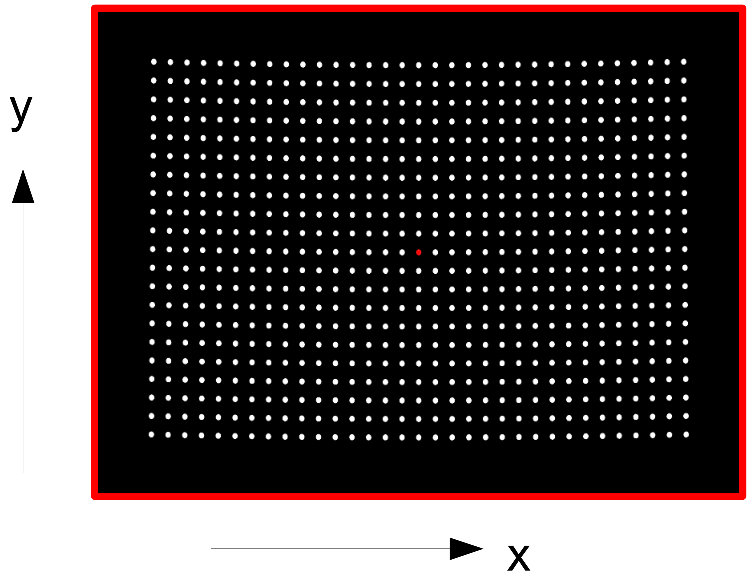
- take 1st spline along x and adjust the height of all perpendicular y-splines at the shared point (“mount” them on the x-spline)
- take 2nd spline along x ...
- …
- repeat procedure in perpendicular direction



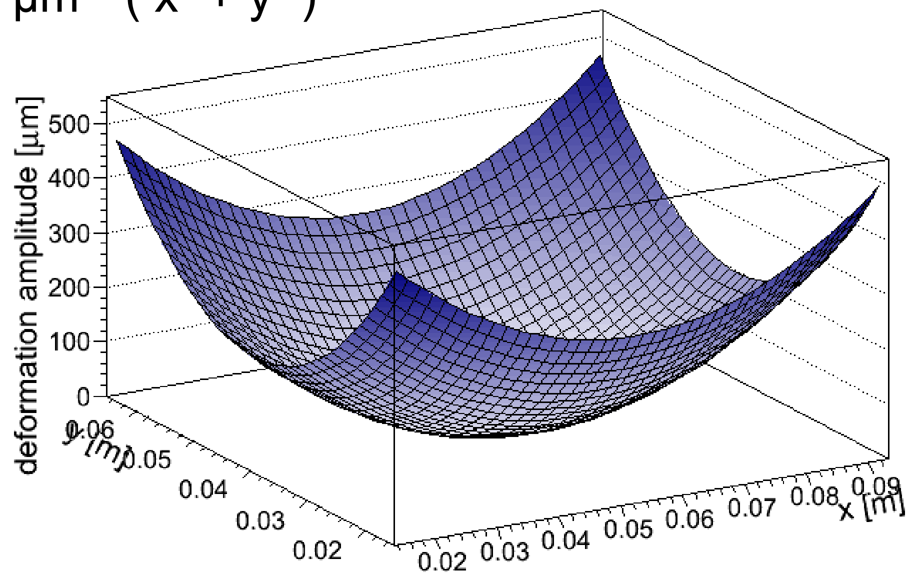
The average of all adjustments determines the global height of a spline
→ smooth surface

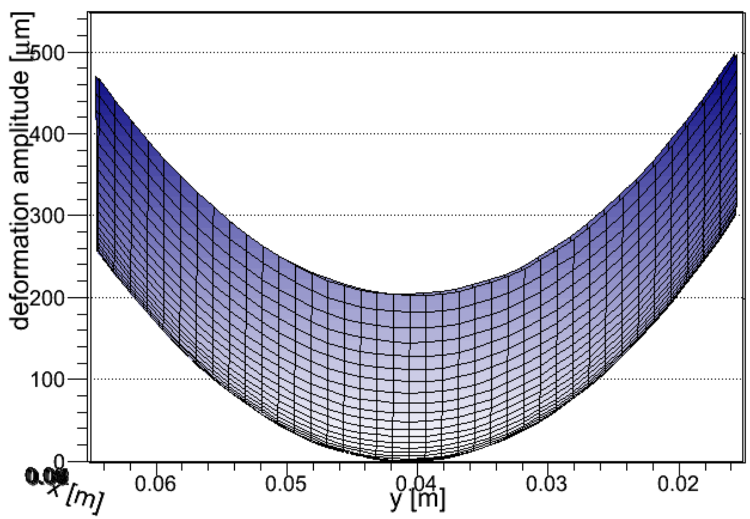
Finally, correct for global surface height offset (lowest point set to 0)
and tilt (rotate surface such that average normal is perpendicular to x-y-plane, TODO)

simulation examples: 300 µm bowing in x and y



Simulated surface:
$z = 300\ \mu m * (x^2 + y^2)$

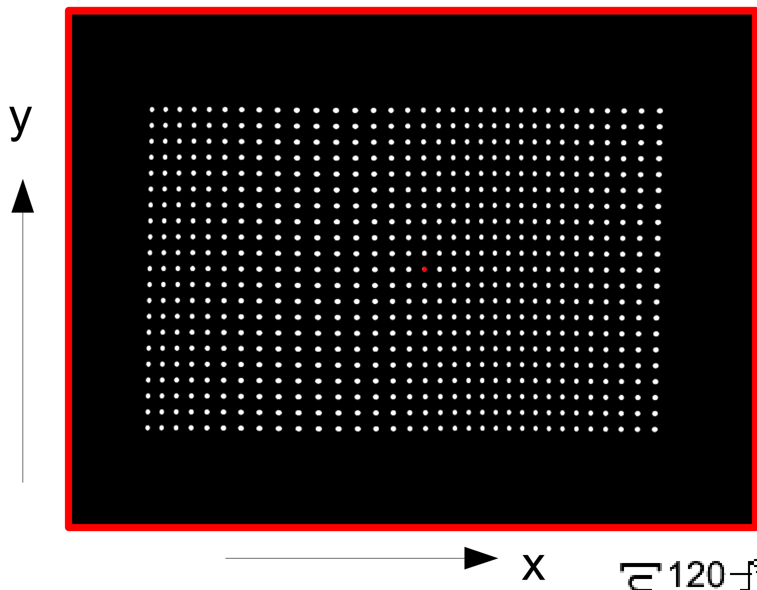




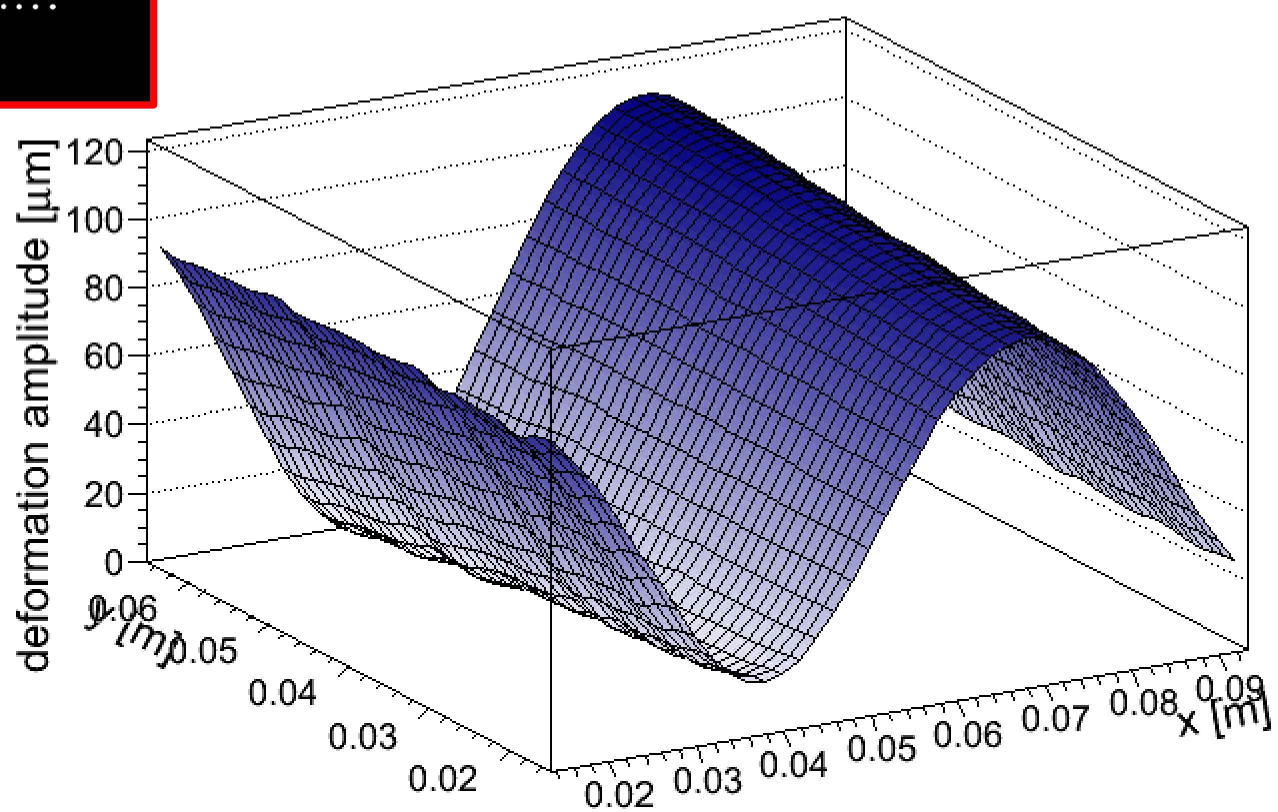
Deformation amplitude not easy to compare since sensor is not fully illuminated with points

Some slight asymmetry along y (away from camera) due to yet missing viewing angle correction of distances

simulation examples: a sine "wave" along y



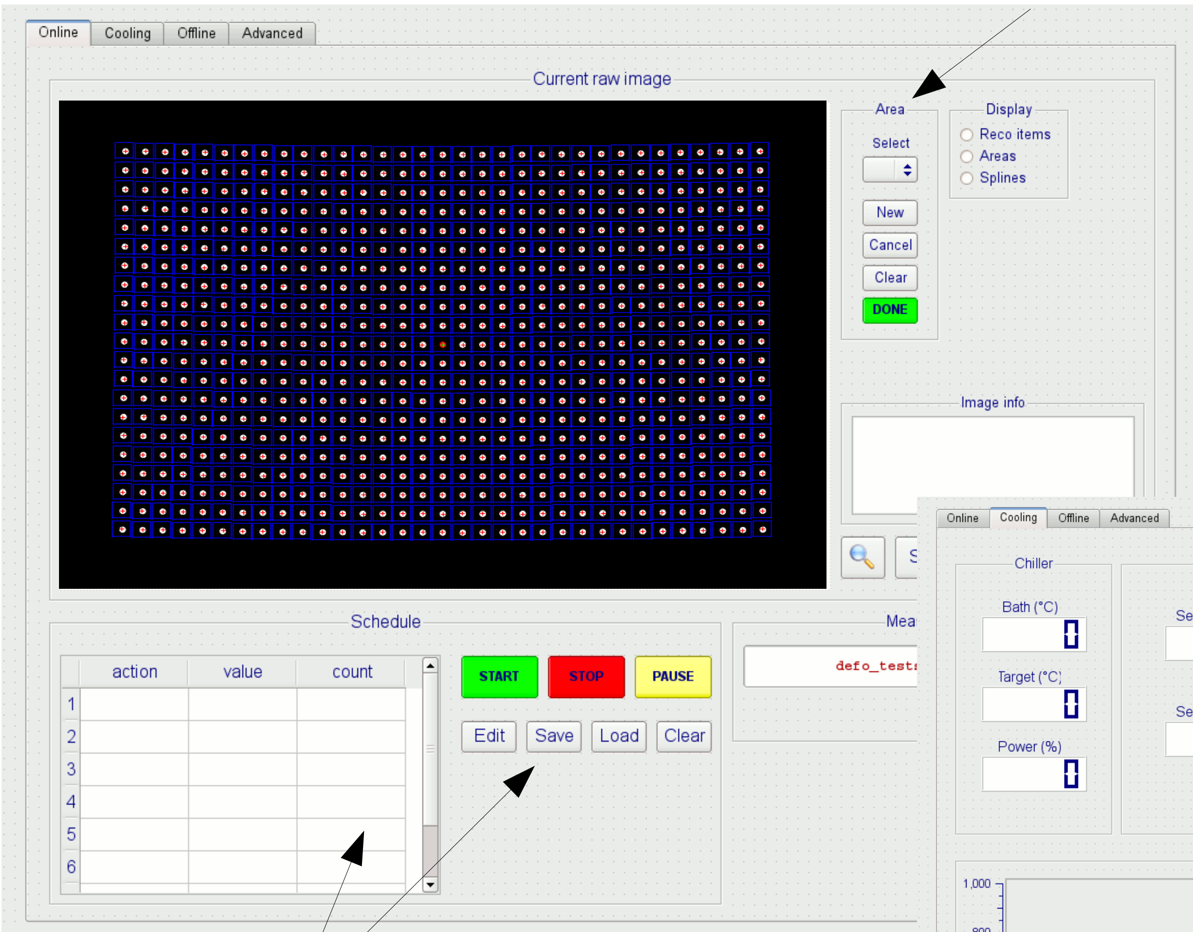
Simulated surface:
z = 50 µm * sin(6 / cm * x)



GUI in preparation

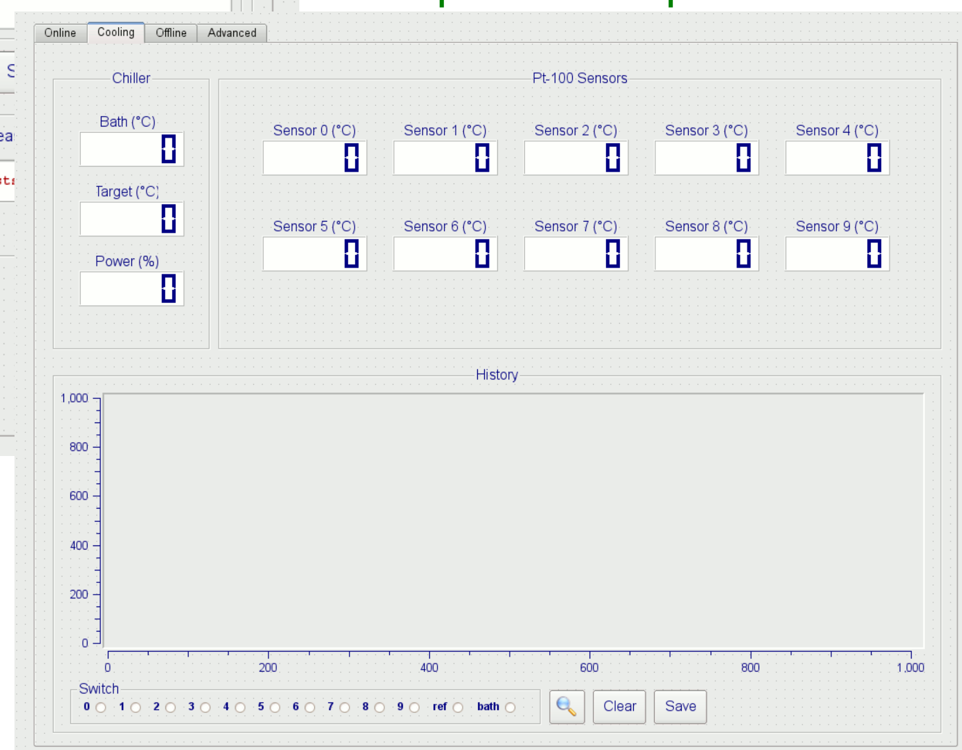
raw image window

several image “areas” (parallel processing)



Devices to control:
- camera
- chiller
- multimeter (pt-100)
- reference thermometer (?)

pt-100 temp sensors



programmable measurement schedule