Internship defense

Development of an automatic tool for defect detection on ultrasonic testing images

Presented by El Houssaini Youness

Outline:

- Context
- Data collection
- Data cleaning
- Segmentation
- 3D reconstruction
- Perspective

Context?

Company:

The Federal Ministry for Economic Affairs and Energy

The German multinational conglomerate company

The Combined research for turbomachinery

"Fraunhofer Society for the Advancement of Applied Research"
1/72 → Fraunhofer Institute for Mechanics of Materials

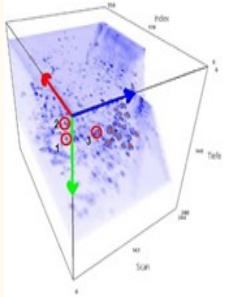
Motivation:

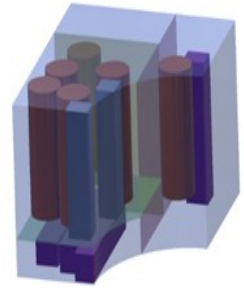
- Casting and forging processes of rotor discs lead to manufacturing defects (blowhole, shrinkage cavity, non metallic inclusion)
- Currently forging defects are assessed as equivalent sharp cracks by fracture conservative mechanics methods
- Consideration of the crack initiation phase should help to improve lifetime prediction

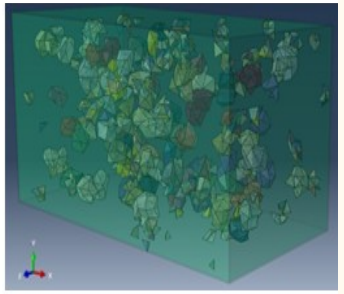
Project:

Lifetime : N = $?_{Nuc}$ + $?_{FCG}$

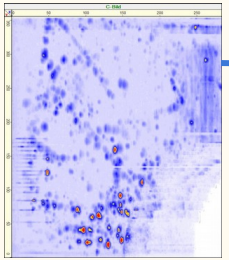
UT testing → Specimen planing → fatigue tests → REM + EDX → FE-Simulation

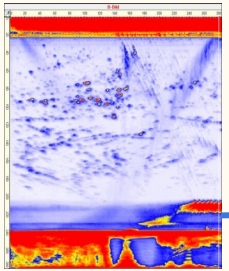






Objective:



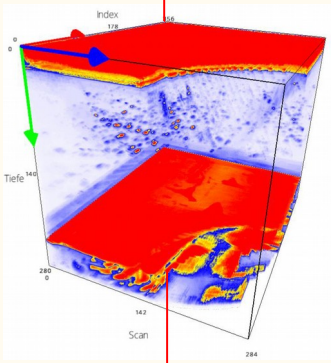


- XY UT image
- XZ UT image

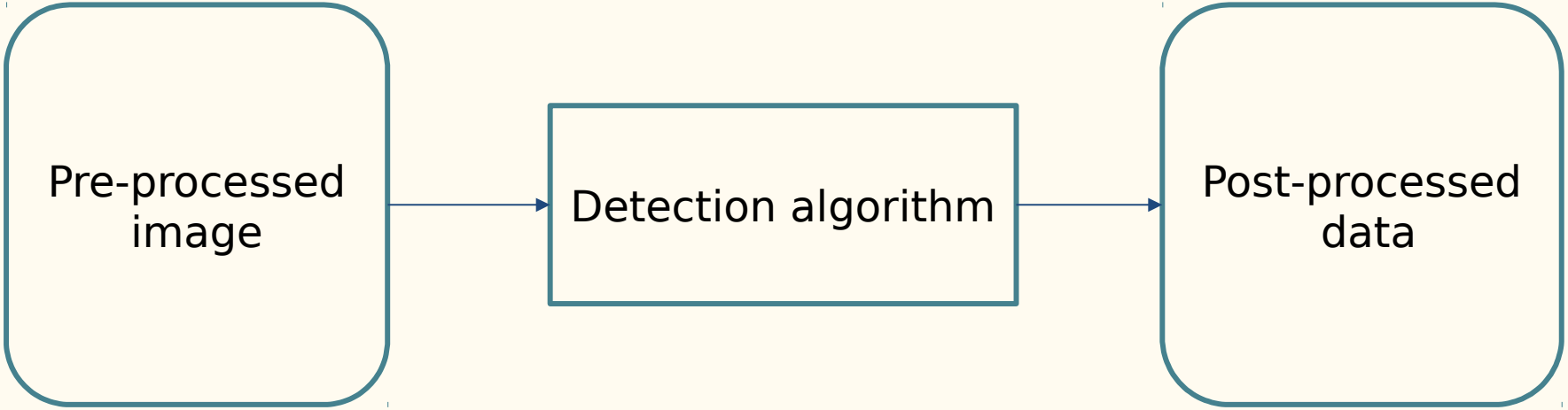
Defects detection tool

Estimation of defects:

- Position
- shape
- Distribution

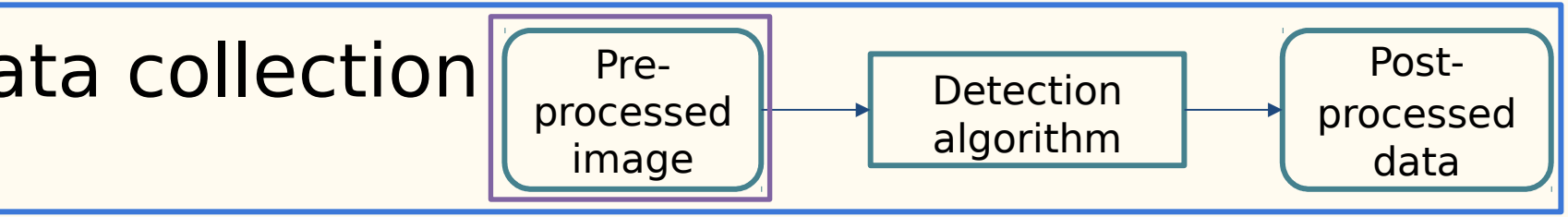


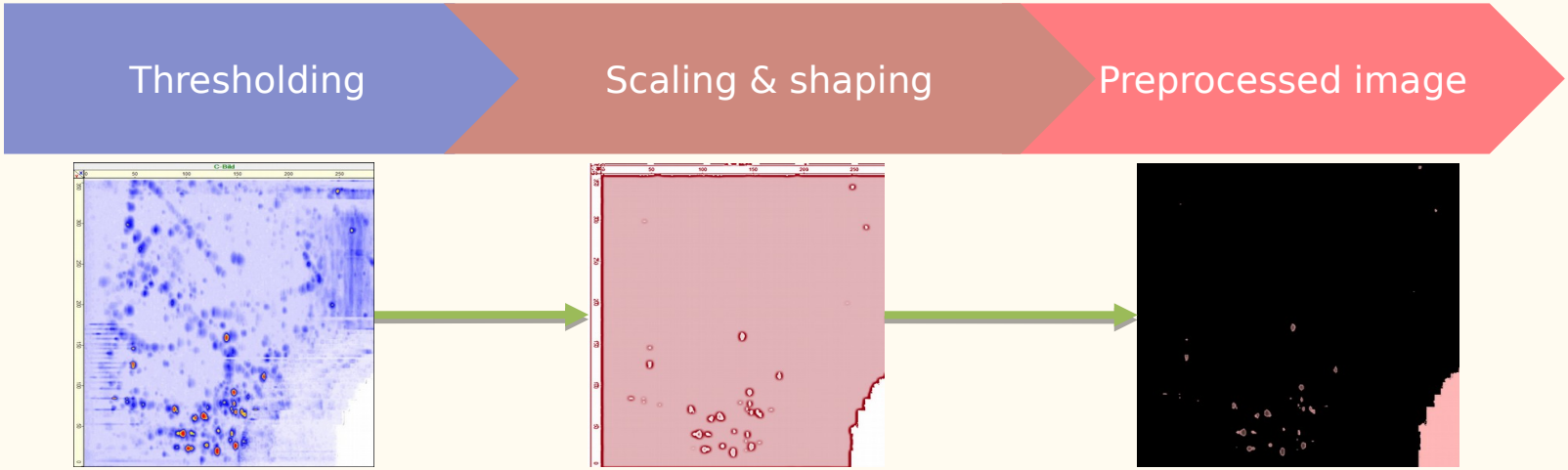
Overview


Pre-processed image
Detection algorithm
Post-processed data

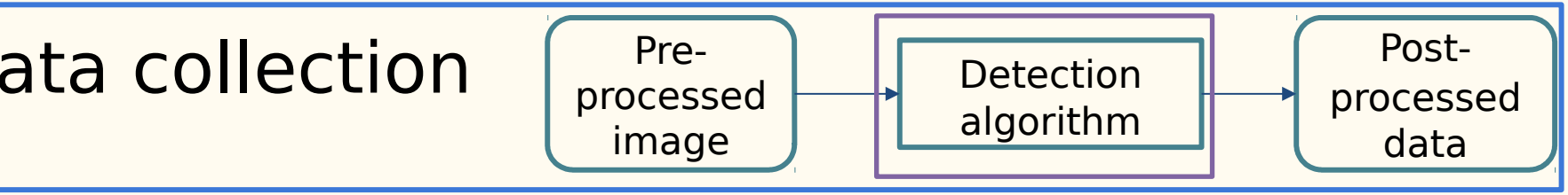
Data collection

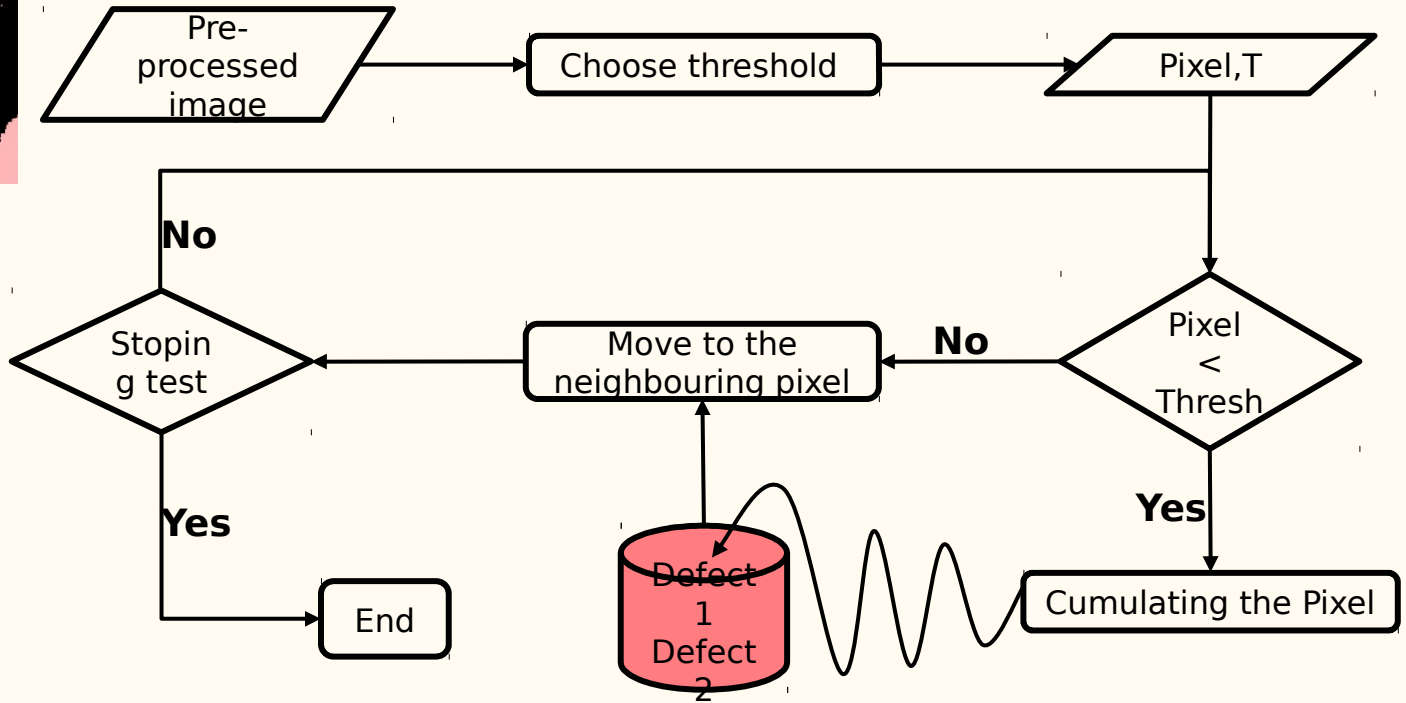
Data collection



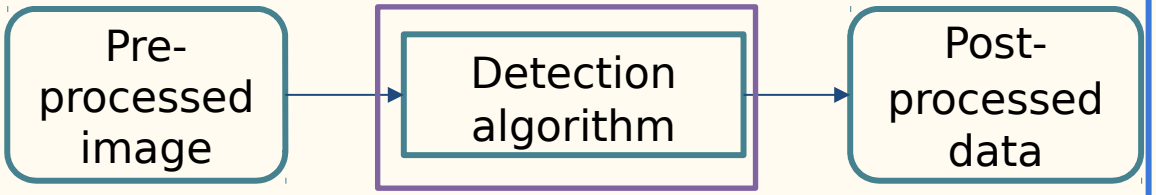


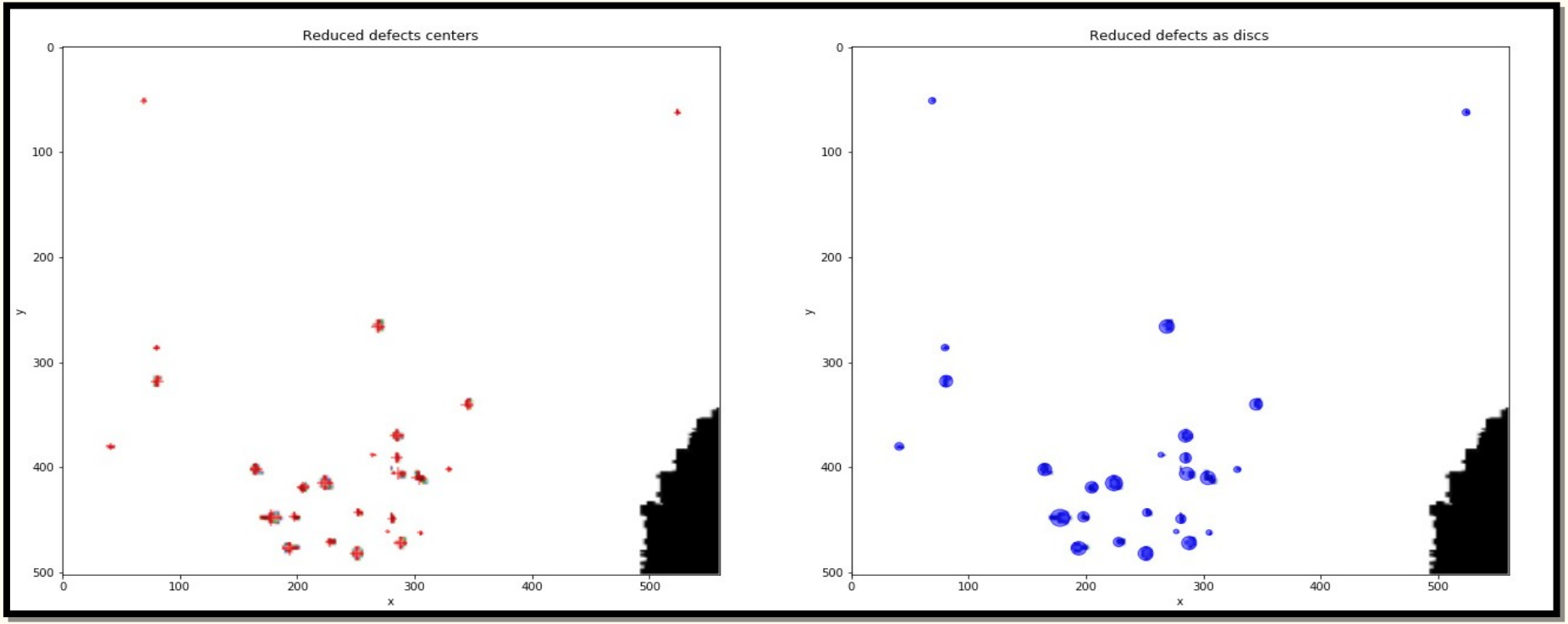
Data collection



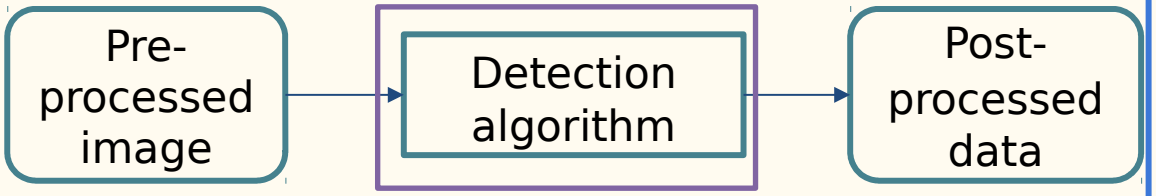


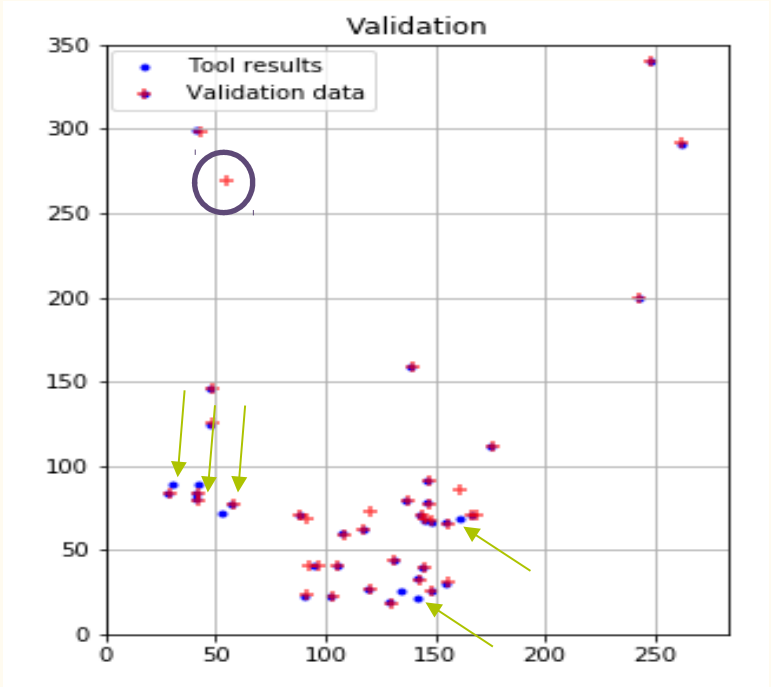
Results

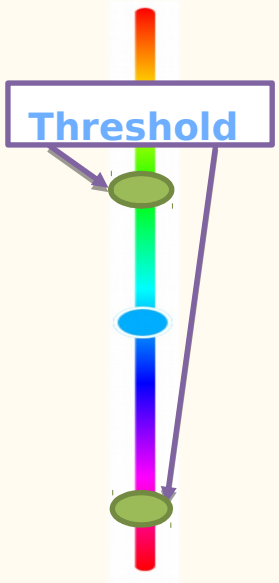

Pre-processed image
Detection algorithm
Post-processed data

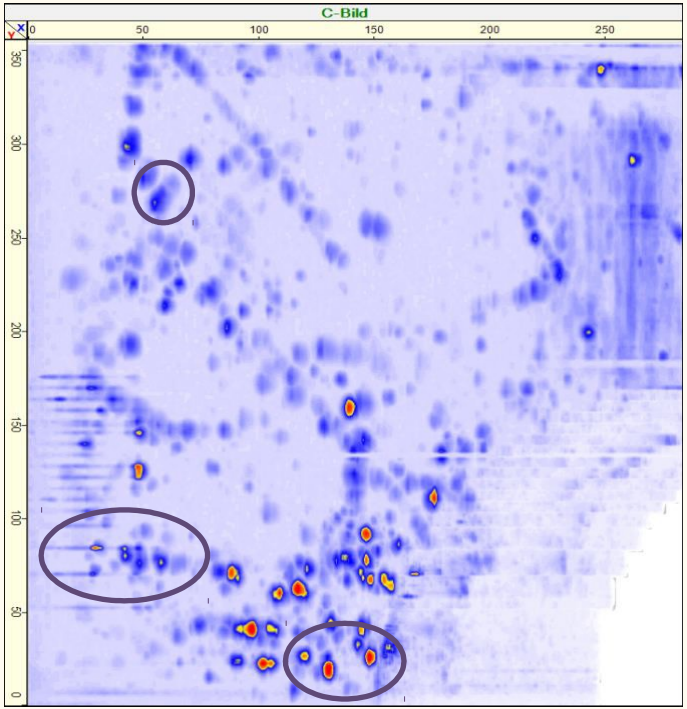

Reduced defects centers
Reduced defects as discs
x
y
0
100
200
300
400
500

Validation


Pre-processed image
Detection algorithm
Post-processed data

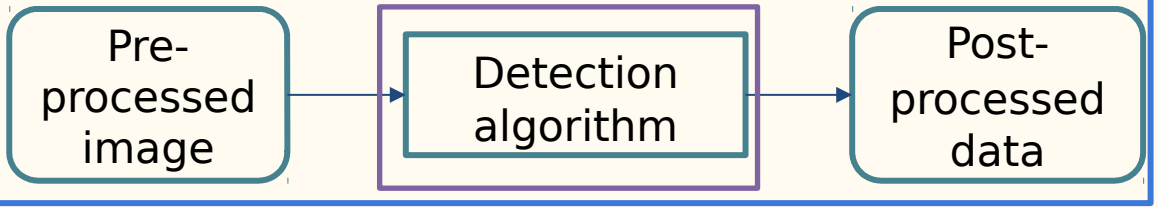

Validation
Tool results
Validation data
0
50
100
150
200
250
300
350

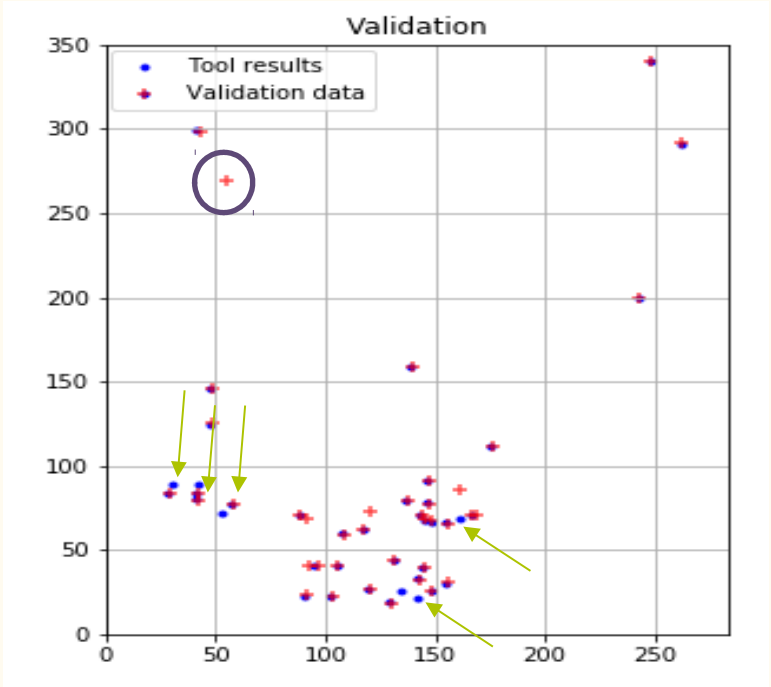

Threshold


C-Bild

Data cleaning

Data cleaning

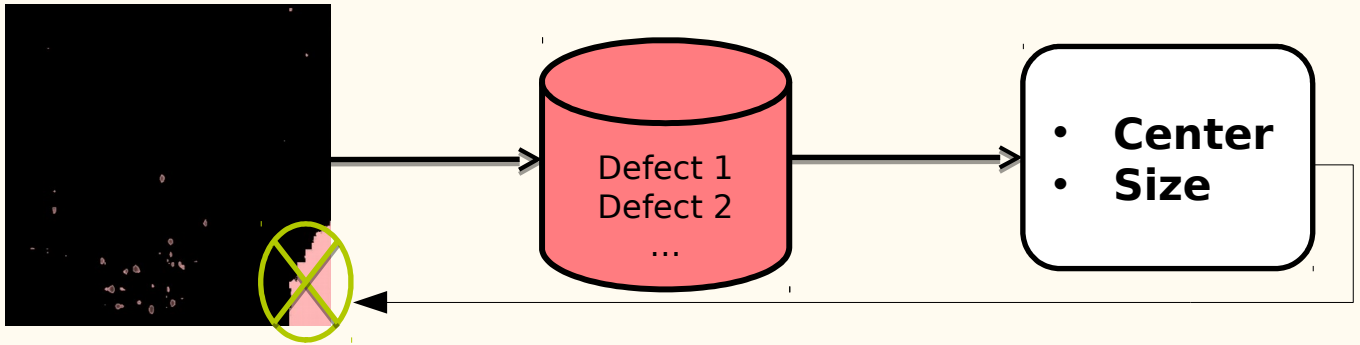




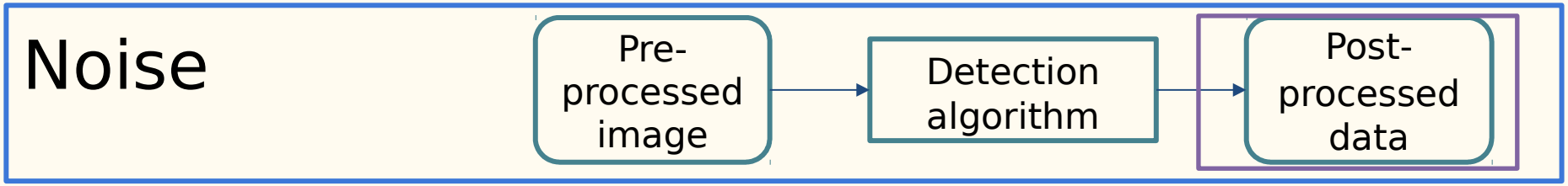
- More noise defects: **+ 6**
- Less detected points: **36/40**

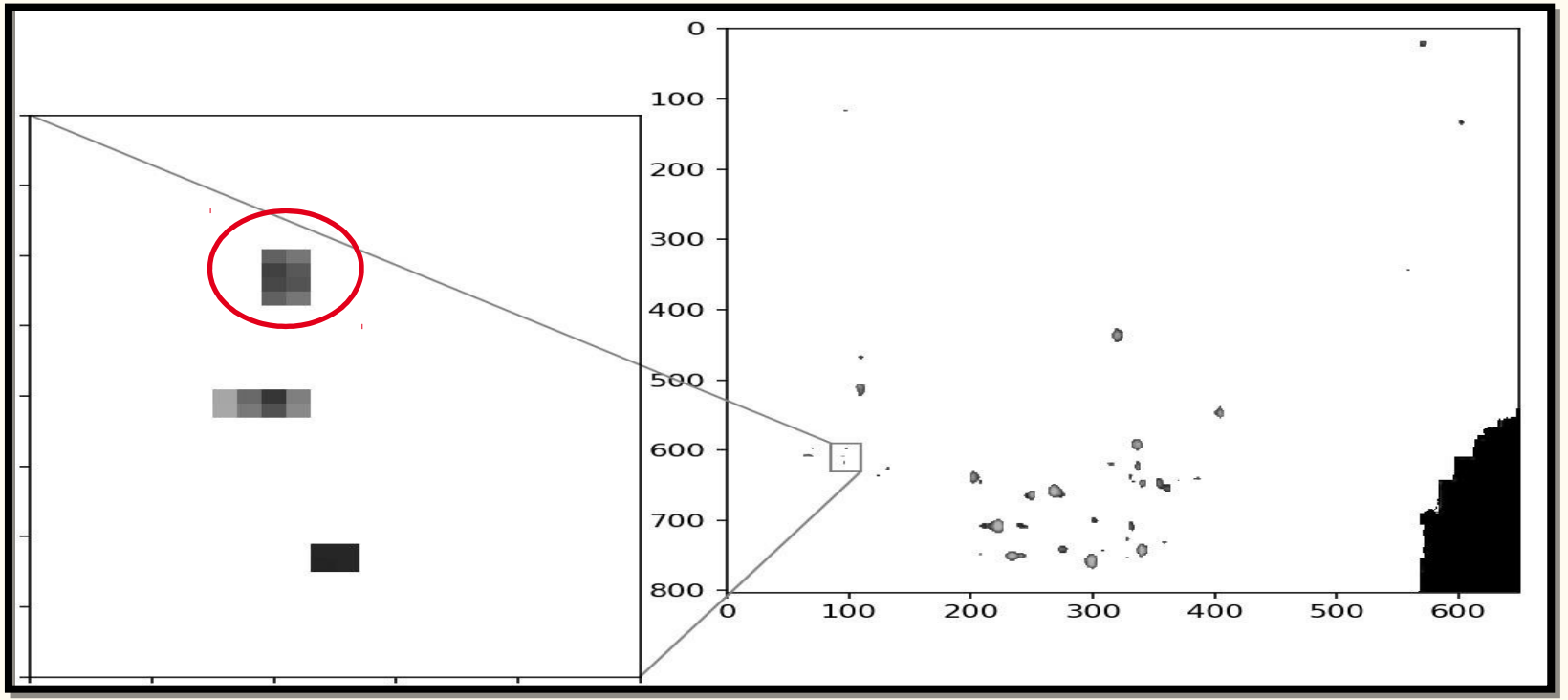
Artifacts


Pre-
processed
image
Detection
algorithm
Post-
processed
data

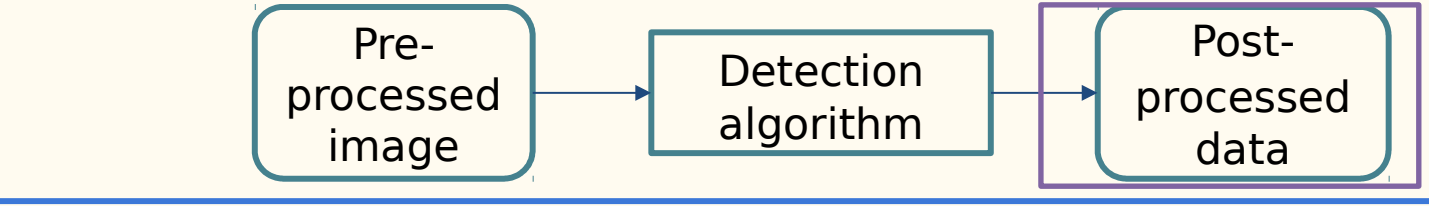

Defect 1
Defect 2
...
• Center
• Size

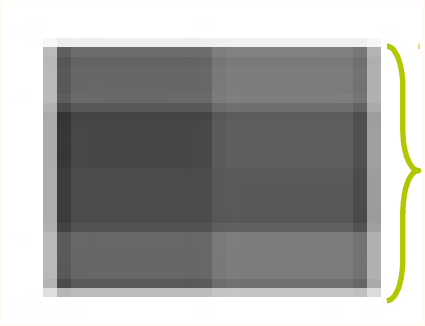
Noise


Pre-
processed
image
Detection
algorithm
Post-
processed
data


0
100
200
300
400
500
600
700
800
0
100
200
300
400
500
600

Noise





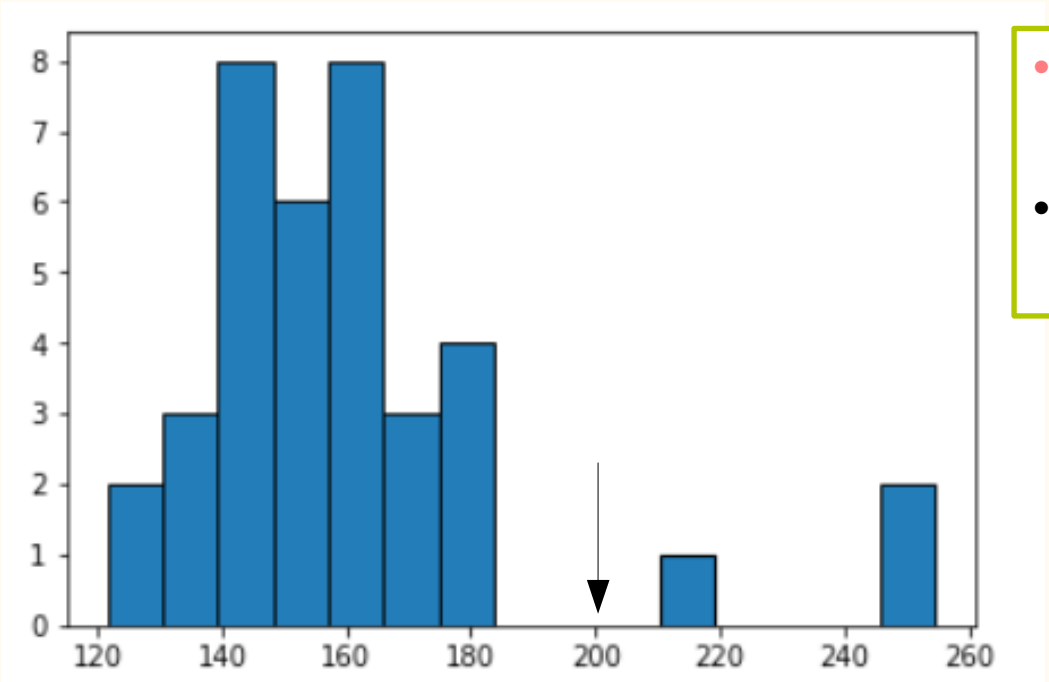
```
defect
```

```
array([[ 97., 205.],
       [ 98., 205.],
       [ 98., 204.],
       [ 97., 204.]])
```

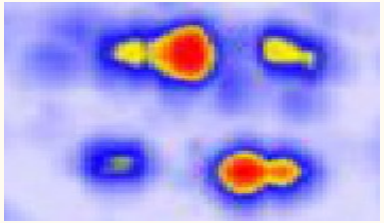
```
get_defect_intensity(imgray,defect)
```

```
array([218, 217, 217, 217], dtype=uint8)
```

Noise

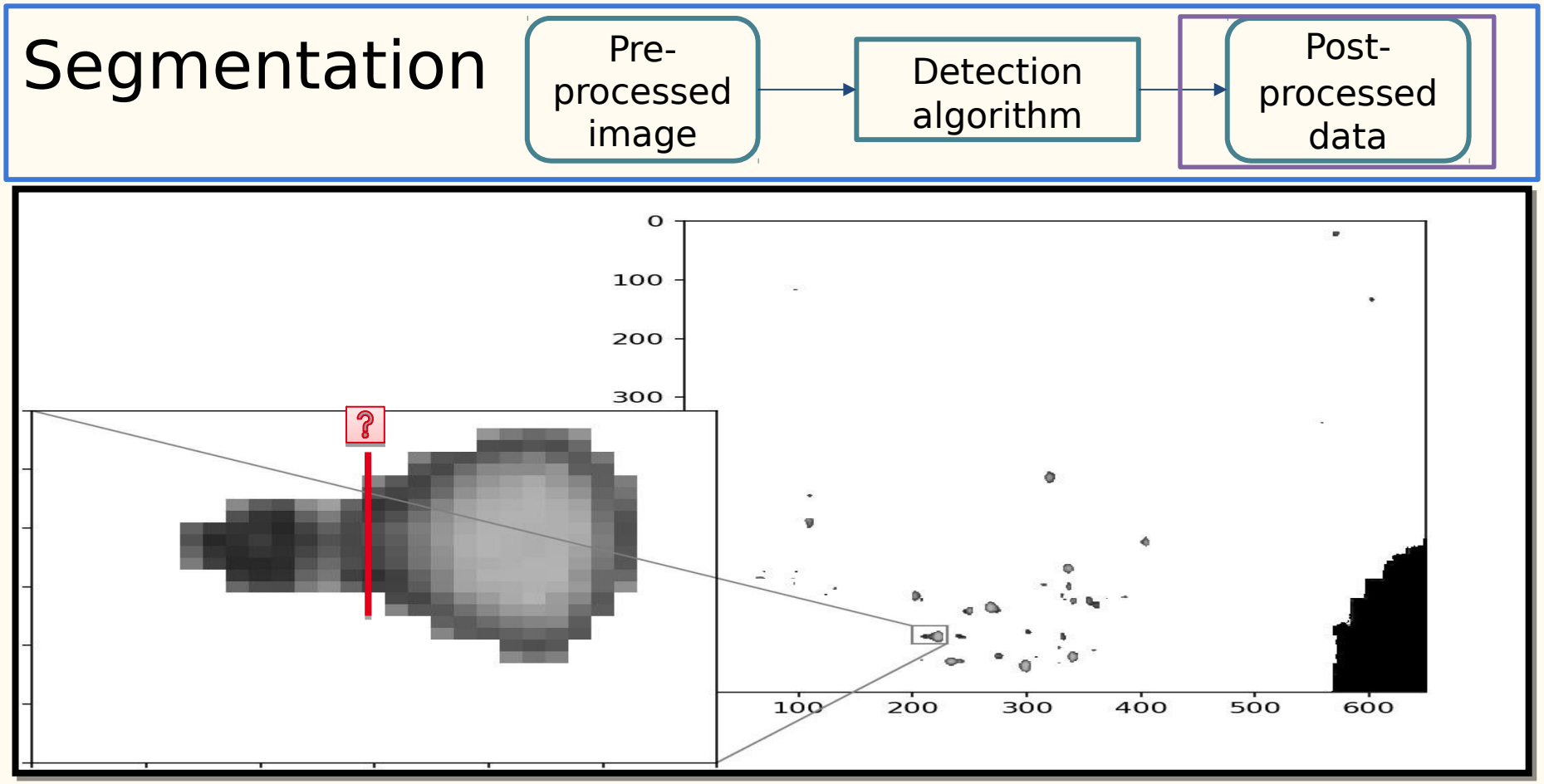


- More noise defects:
  ~~+~~ **6**
- Less detected defects:
  36/40



Segmentation

Segmentation



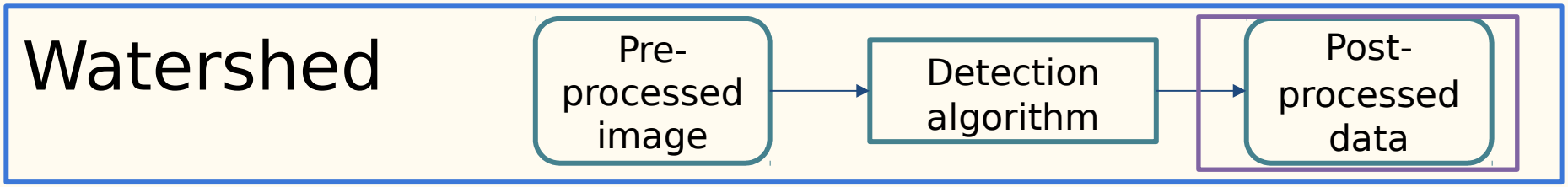
Extrema

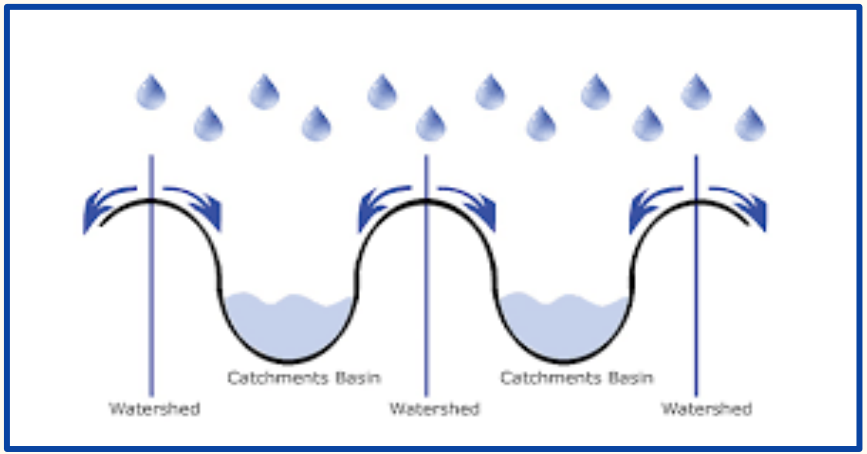
Pre-processed image → Detection algorithm → Post-processed data

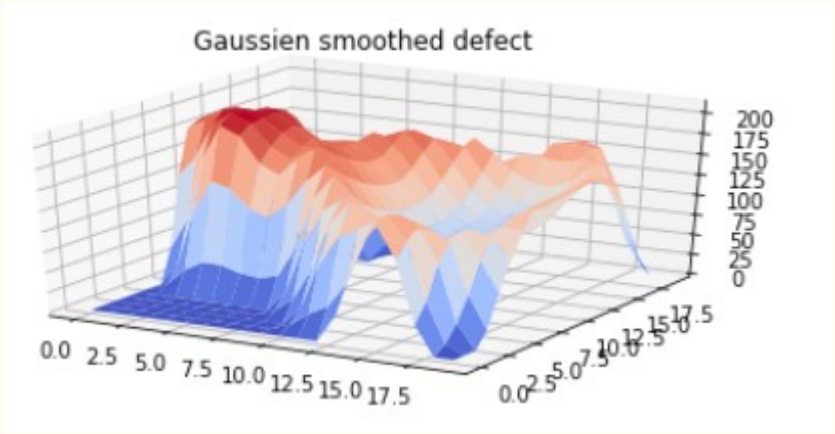
Defect matrix → ● Sobel operator G ● Directions (x,y) → Magnitude $\sqrt{G_{(1,0)}{}^2+G_{(0,1)}{}^2+G_{(1,1)}{}^2+G_{(1,-1)}{}^2}$ → Minima algorithm → Minima

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	298.84	636.30	889.92	963.19	750.43	153.03	741.13	1010.66	413.96
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	624.84	940.88	920.21	613.75	238.74	272.86	214.04	935.31	742.40
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	301.29	652.25	960.98	729.08	57.04	289.81	332.31	323.64	342.66	477.28	994.19
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	682.23	1045.37	818.40	251.44	304.18	346.11	306.04	230.32	328.26	170.07	895.66
4	369.87	736.16	996.32	965.62	822.35	756.48	900.39	1099.91	841.68	109.28	390.97	379.90	259.76	172.28	114.85	269.14	414.37	522.66
5	788.68	1154.92	1206.64	1142.02	1002.65	927.50	1093.82	932.38	371.33	339.20	440.83	302.87	144.34	77.20	56.25	194.09	419.13	224.19
6	1193.45	1010.76	258.26	354.14	451.39	240.08	389.98	52.65	267.68	390.30	382.36	232.56	62.32	32.92	26.94	151.17	400.59	248.37
7	1007.78	550.02	9.59	228.95	370.04	124.64	251.02	112.12	307.94	371.22	343.64	214.71	49.03	13.64	20.74	127.70	370.51	370.29
8	352.96	18.06	64.92	108.94	295.12	117.33	132.46	83.35	275.97	343.00	313.27	188.72	38.88	11.40	16.85	75.51	322.96	452.43
9	257.94	48.87	22.89	57.20	260.67	48.97	75.88	129.85	270.88	329.87	290.82	177.84	32.80	27.24	7.62	39.67	304.36	501.13
10	988.17	551.37	107.01	127.71	372.72	183.47	157.31	109.37	272.35	334.44	284.67	179.69	60.61	10.00	17.15	31.30	318.78	523.32
11	1128.22	992.55	379.44	369.25	547.40	339.29	373.19	91.88	326.50	396.06	332.35	201.00	81.17	19.90	16.37	57.93	356.51	475.64
12	653.02	1062.23	1159.00	1135.24	1011.74	894.85	947.64	530.52	172.31	387.82	418.63	276.44	128.83	66.14	42.10	123.28	388.53	353.12
13	289.04	627.64	893.15	895.84	751.82	698.79	1048.50	948.13	176.82	240.09	438.85	340.91	186.54	121.74	88.15	198.00	392.19	201.17
14	0.00	0.00	0.00	0.00	0.00	0.00	652.54	1025.28	896.86	372.73	341.34	383.94	239.26	199.15	172.13	289.47	383.41	247.71
15	0.00	0.00	0.00	0.00	0.00	0.00	296.39	643.33	1029.64	833.19	84.78	353.82	347.66	293.96	265.66	352.70	250.06	691.41
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	667.80	1025.92	988.22	595.94	190.29	351.82	357.88	276.32	312.13	1020.09
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	311.09	667.80	934.12	1002.05	753.05	80.98	32.37	110.50	801.70	932.58

Watershed


Pre-processed image
Detection algorithm
Post-processed data


Catchments Basin
Catchments Basin
Watershed
Watershed
Watershed


Gaussien smoothed defect
200
175
150
125
100
75
50
25
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Watershed

Pre-processed image → Detection algorithm → Post-processed data

Labeling the defect (i.e **E**)

Find the minima set S $\{M_k\}$, $k \in S$

M_k,(i,j),E

False

$M_k < I(i,j) < M_k + E$ & *(i,j) not watershed*

False

K++ / E++ / move to adjacent pixel

Stoping test

True

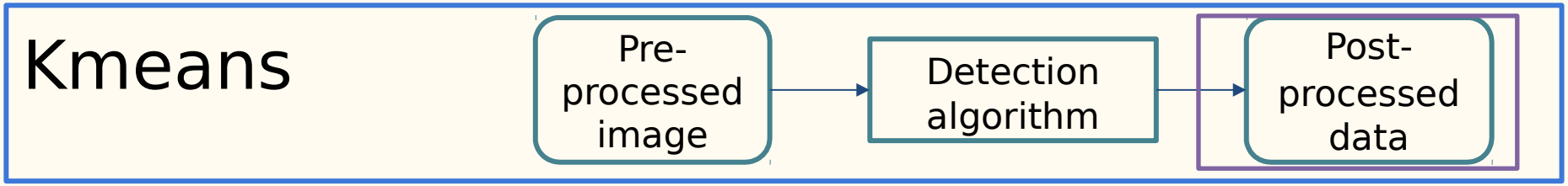
End

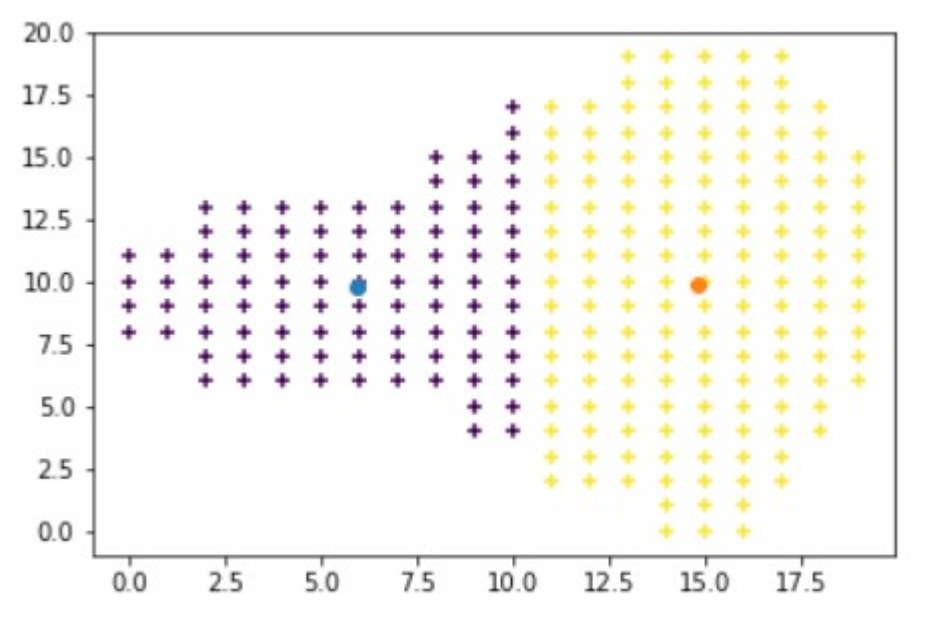
True

Sub-defect 1
Sub-defect 2
...

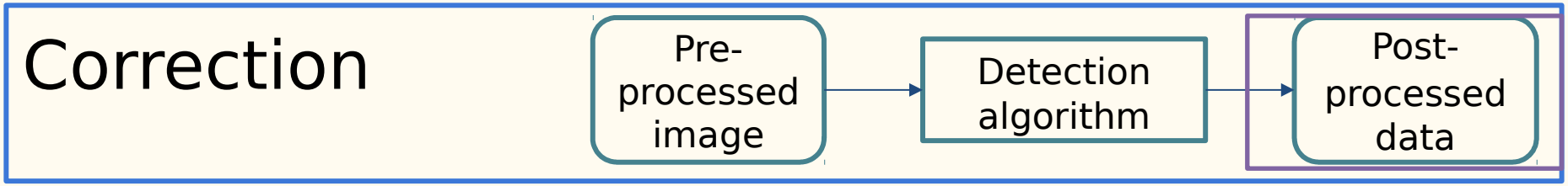
Accumulating the sub-defect

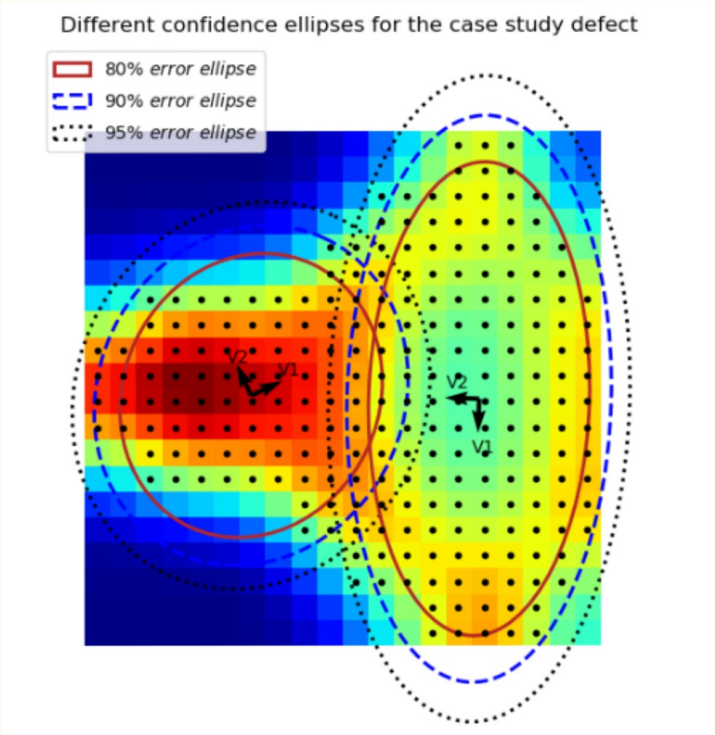
Kmeans


Pre-processed image
Detection algorithm
Post-processed data


20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Correction





$$\left(\frac{X}{\sigma_x}\right)^2+\left(\frac{Y}{\sigma_y}\right)^2=S$$

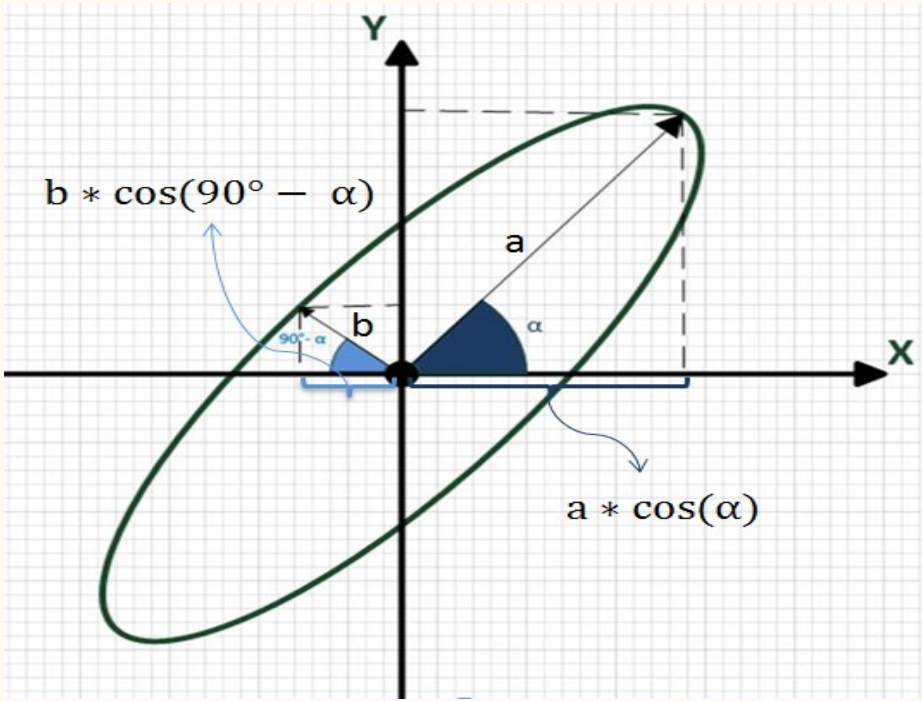
$$P(s<5.991)=1-0.05=0.95$$

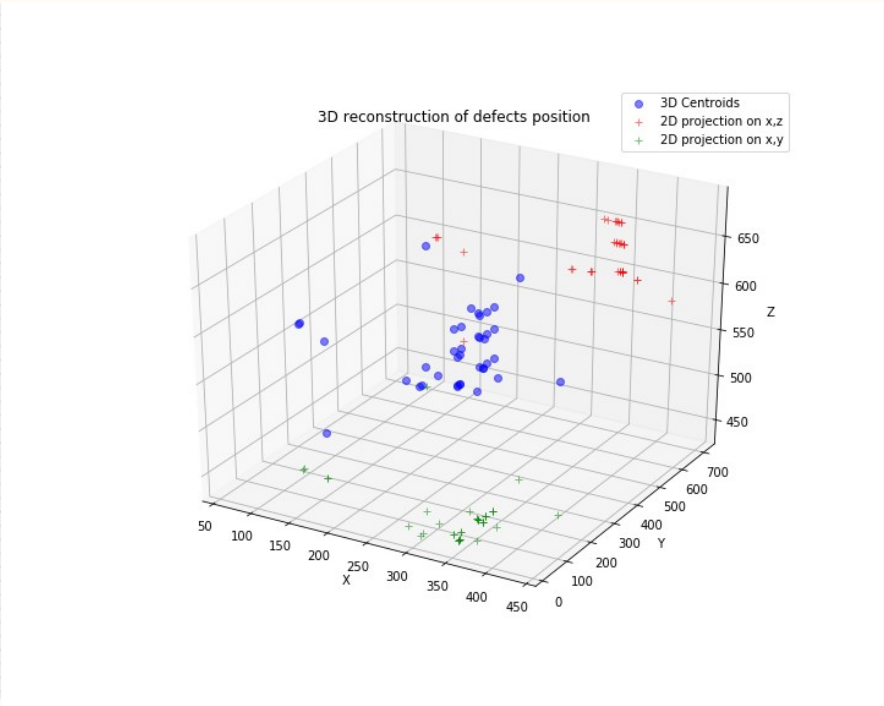
$$length, width=2\sqrt{\lambda_i * S}$$

$$\alpha=\arctan\left(\frac{V_1(y)}{V_1(x)}\right)$$

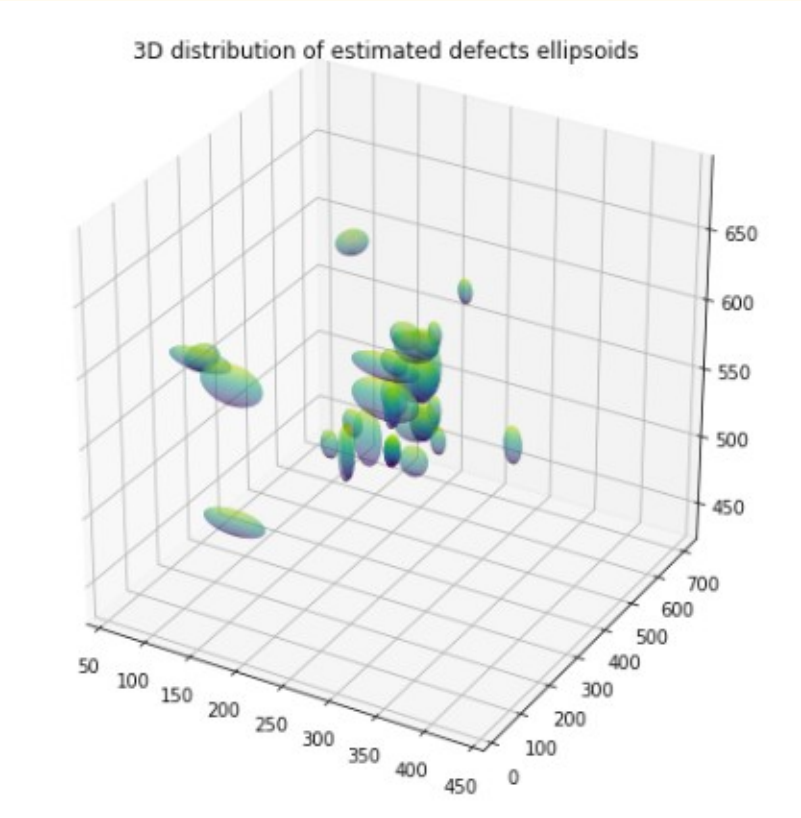
3D reconstruction

3D reconstruction





3D reconstruction

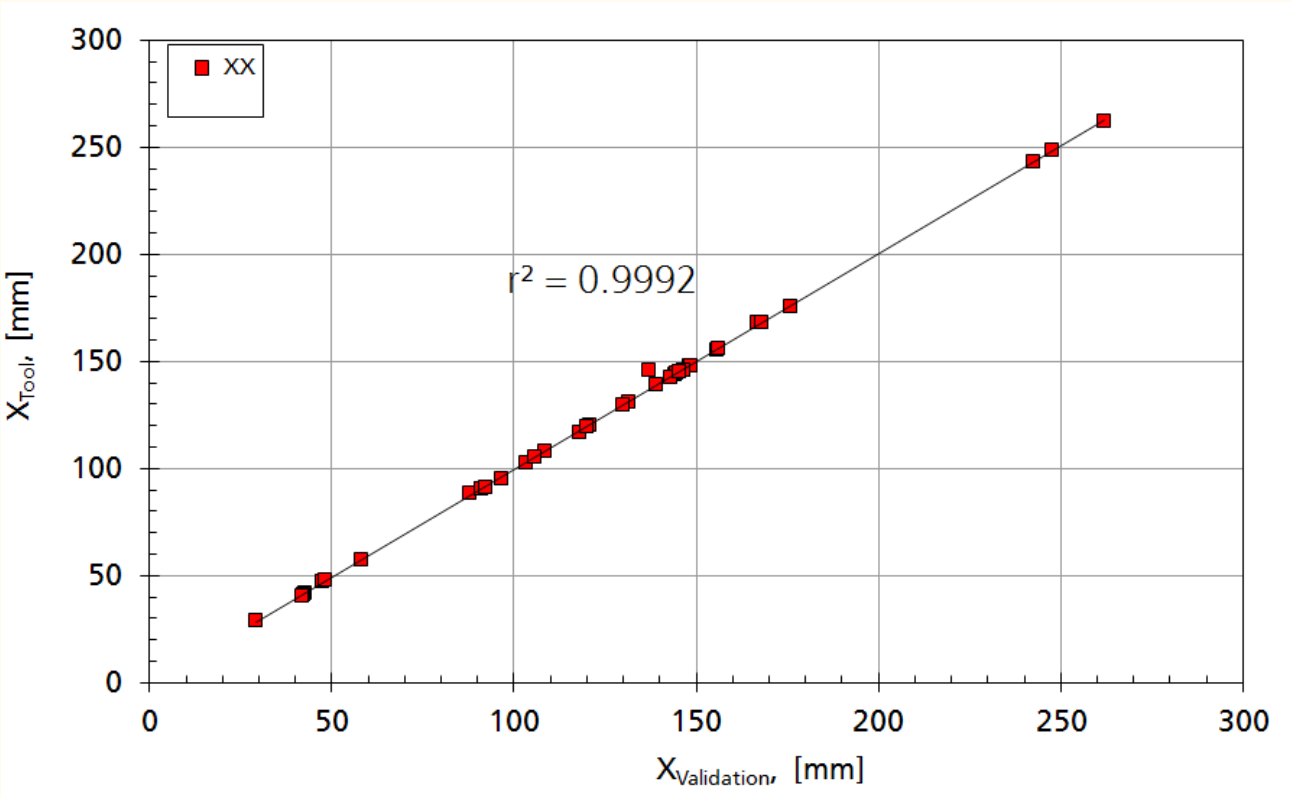


$$\left(\frac{X}{\sigma_x}\right)^2+\left(\frac{Y}{\sigma_y}\right)^2+\left(\frac{Z}{\sigma_z}\right)^2=S$$

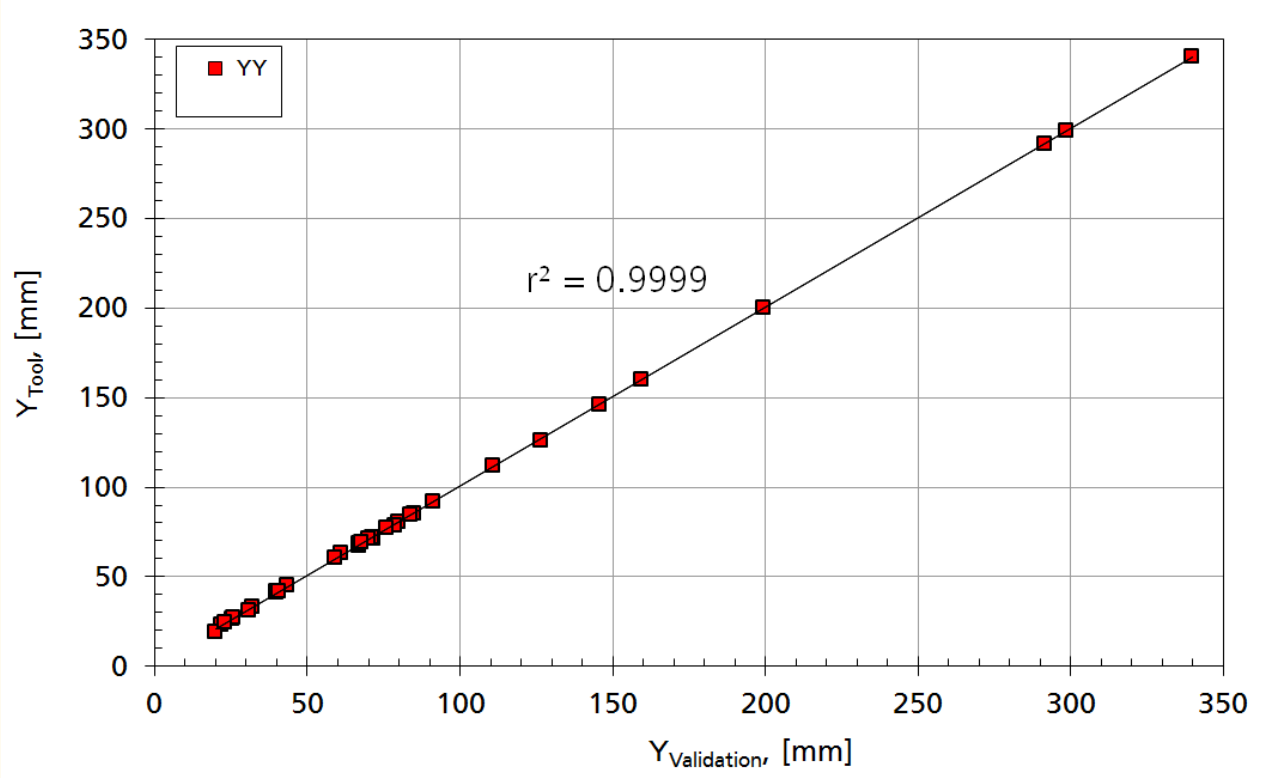
$$P(s<7.815)=1-0.05=0.95$$

$$length, widths=2\sqrt{\lambda_i * S}$$

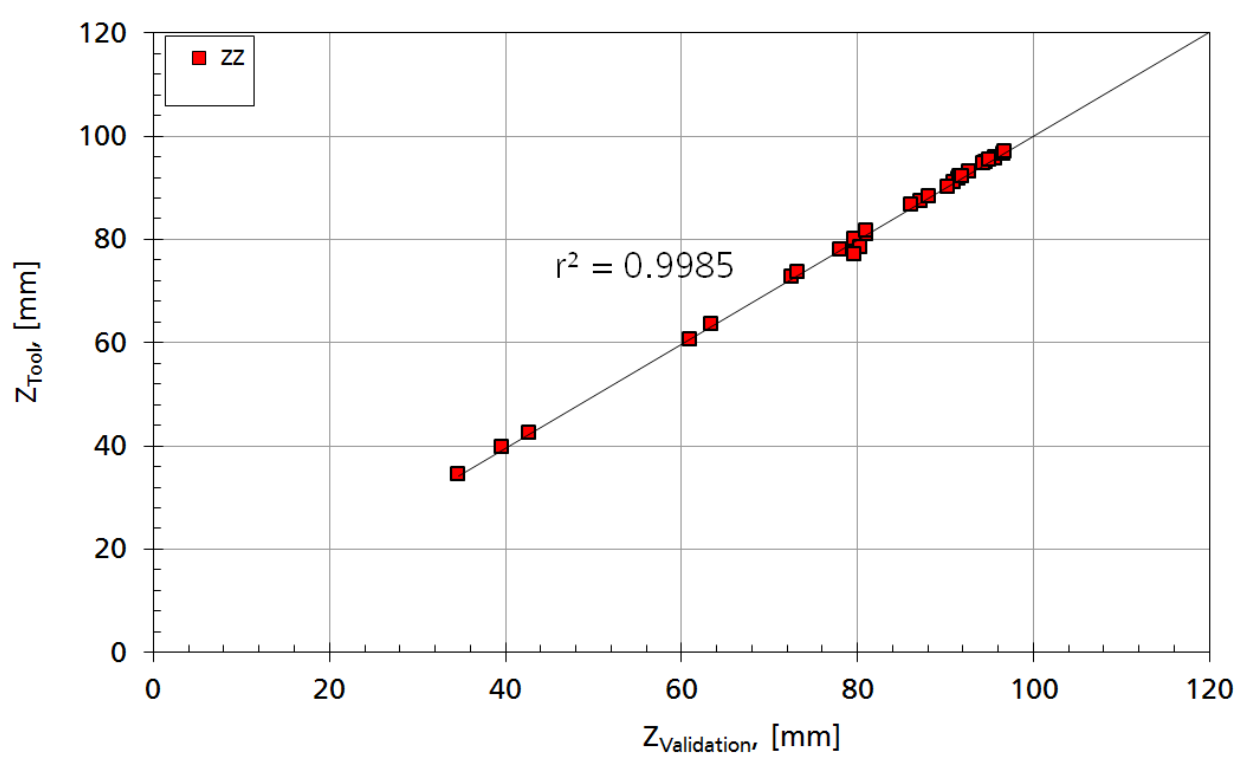
3D reconstruction



3D reconstruction



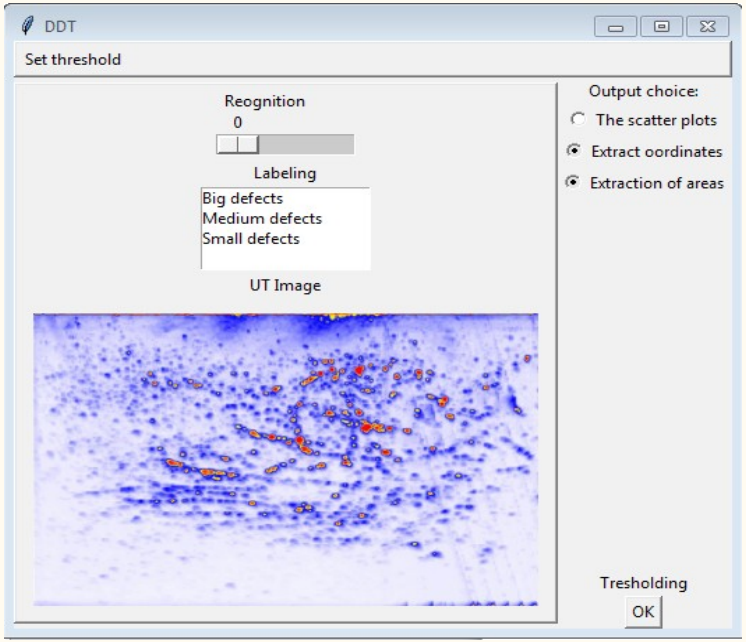
3D reconstruction



Perspective

Perspective

- Finding the optimal key parameters (e.g. local kernal, error certainty..)
- Performance enhancement (i.e. 2min)
- GUI enhancement

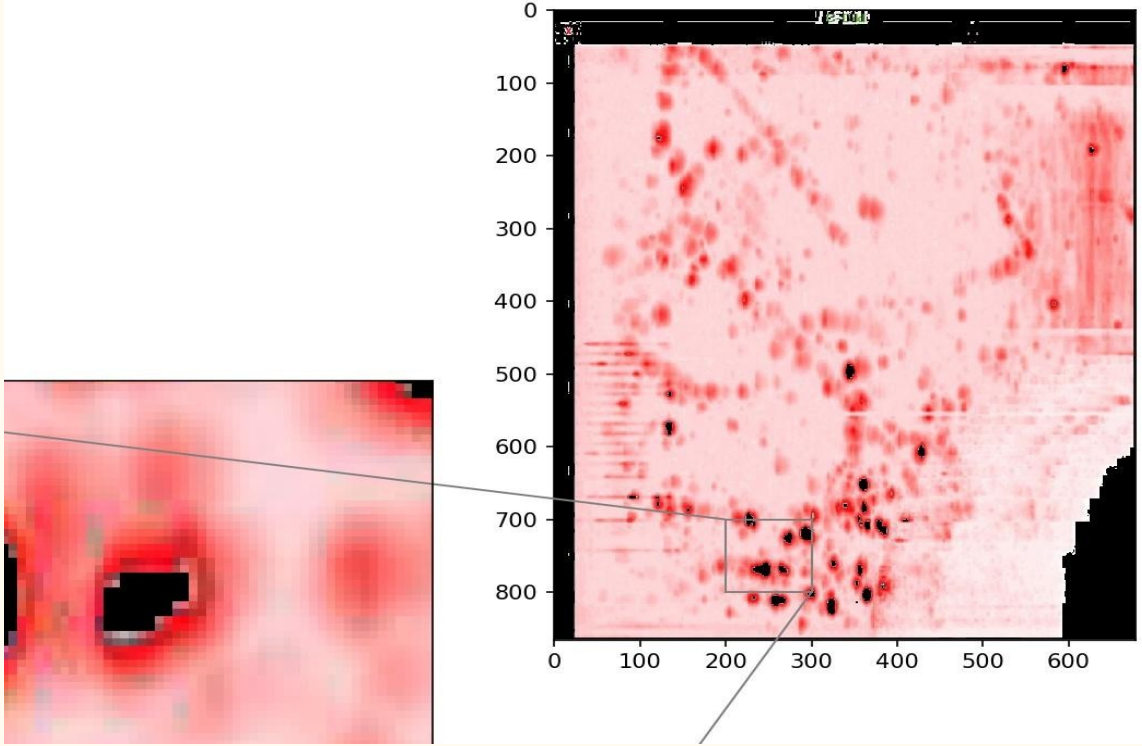


Thank you for your attention!

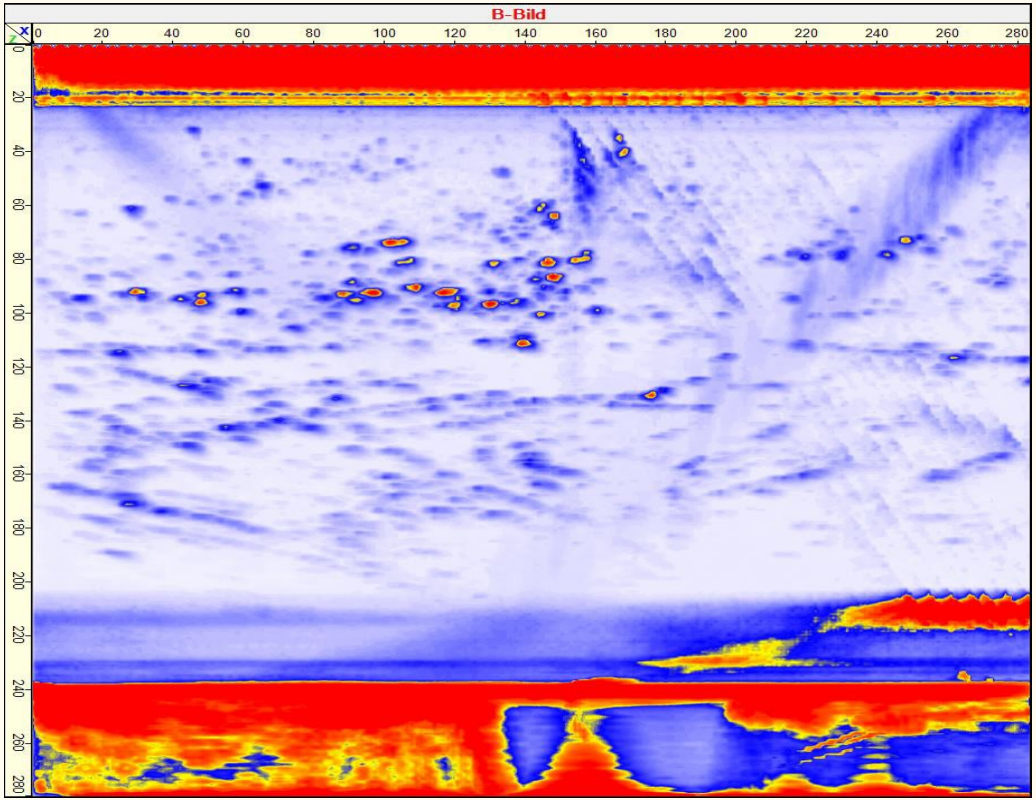
Sources:

- https://www.qualitymag.com/gdpr-policy?url=https%3A%2F%2Fwww.qualitymag.com%2Farticles%2F92425-machine-vision-image-processing
- https://www.bmwi.de/Navigation/FR/Home/home.html

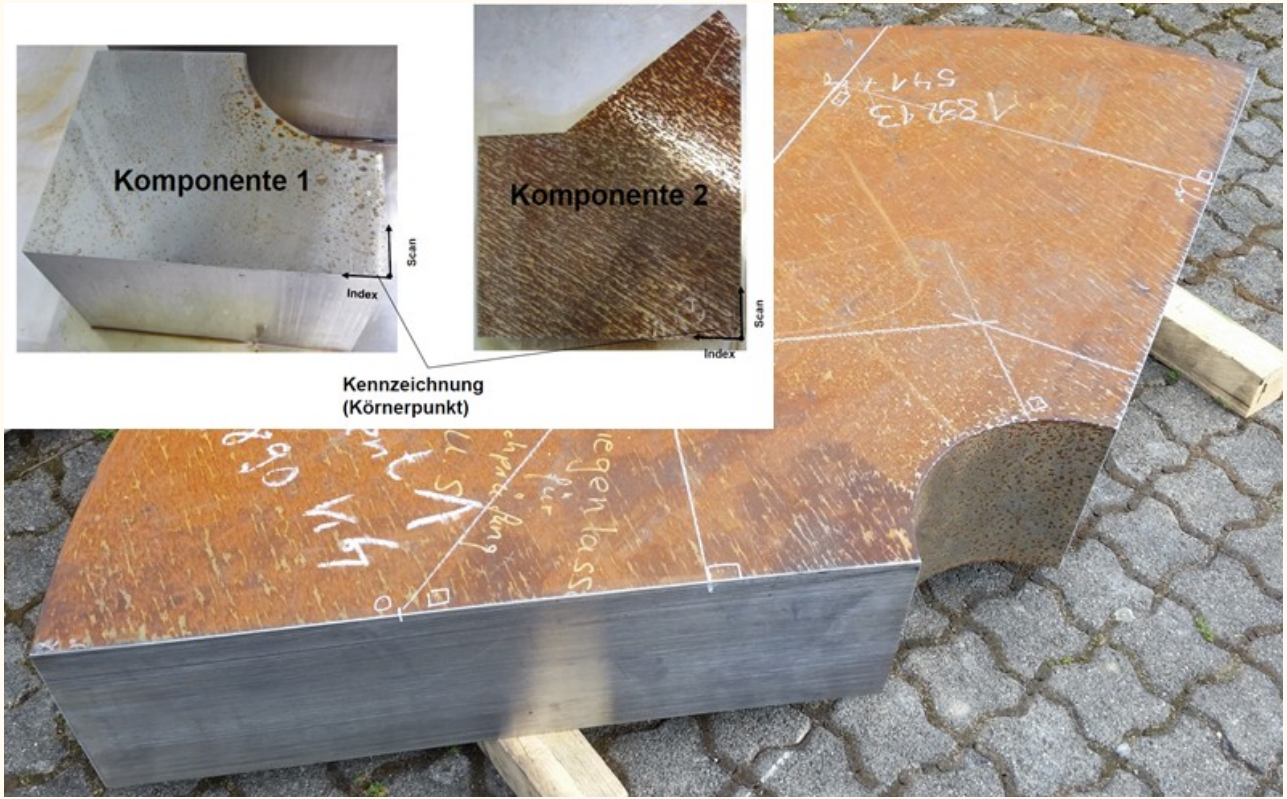
Appendice:



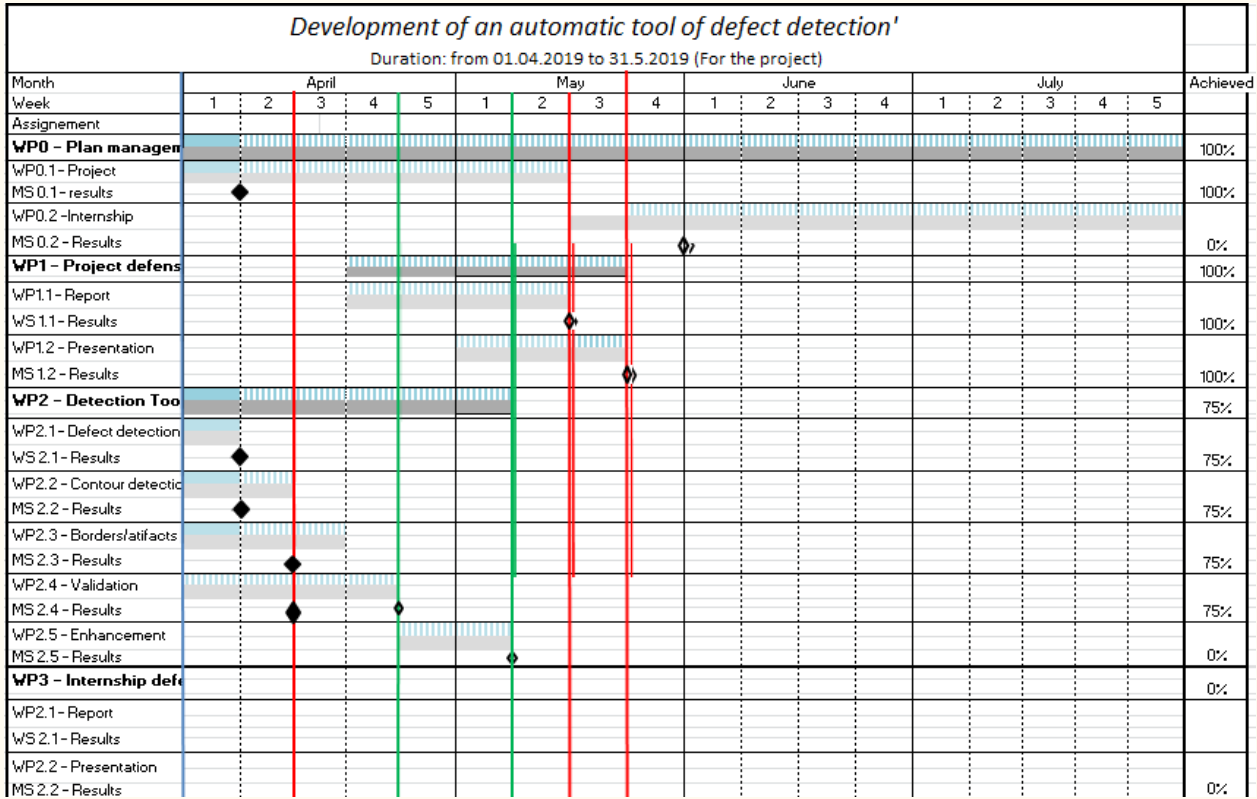
Appendice:

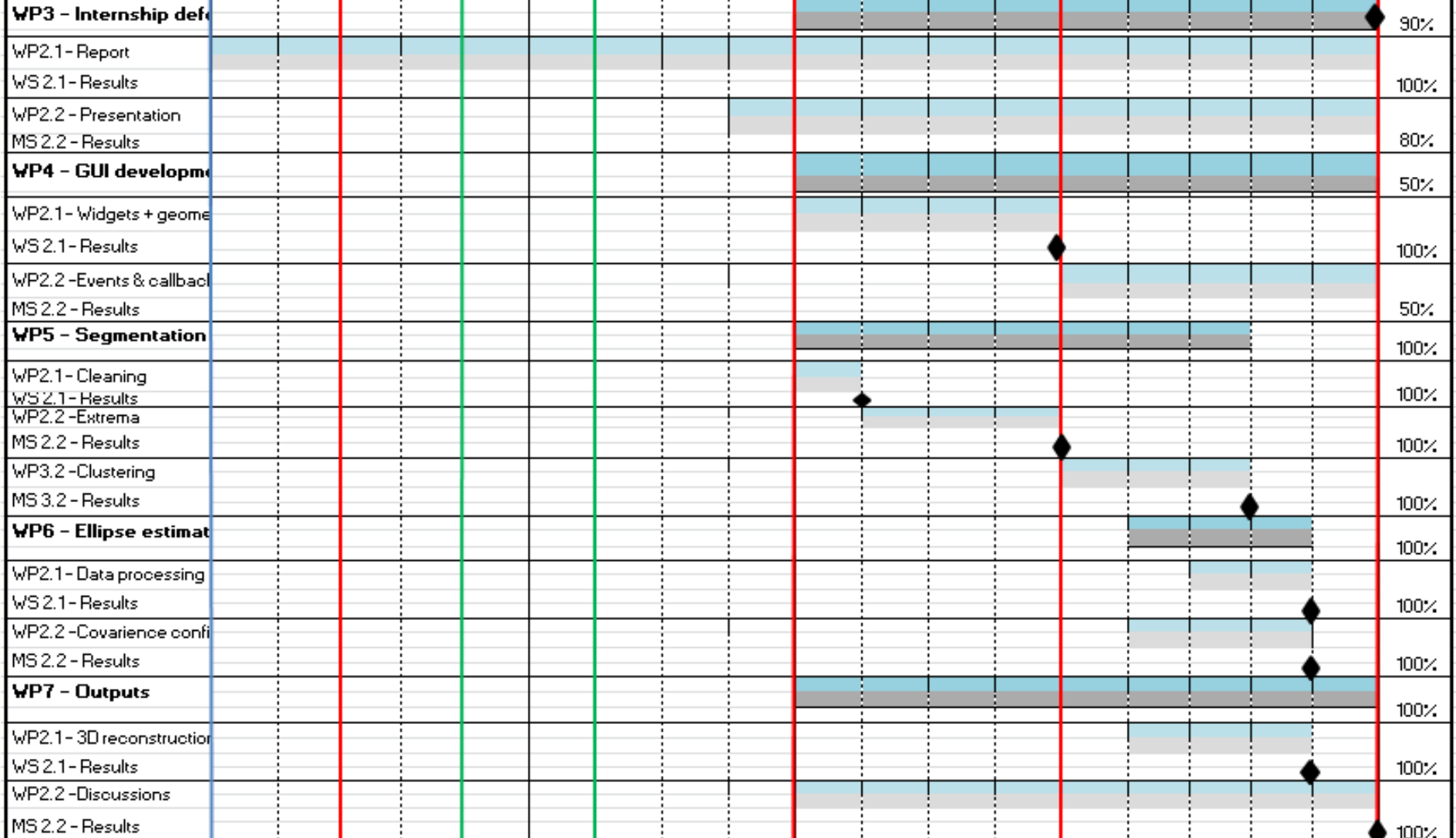

B-Bild
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280

Semester project context:

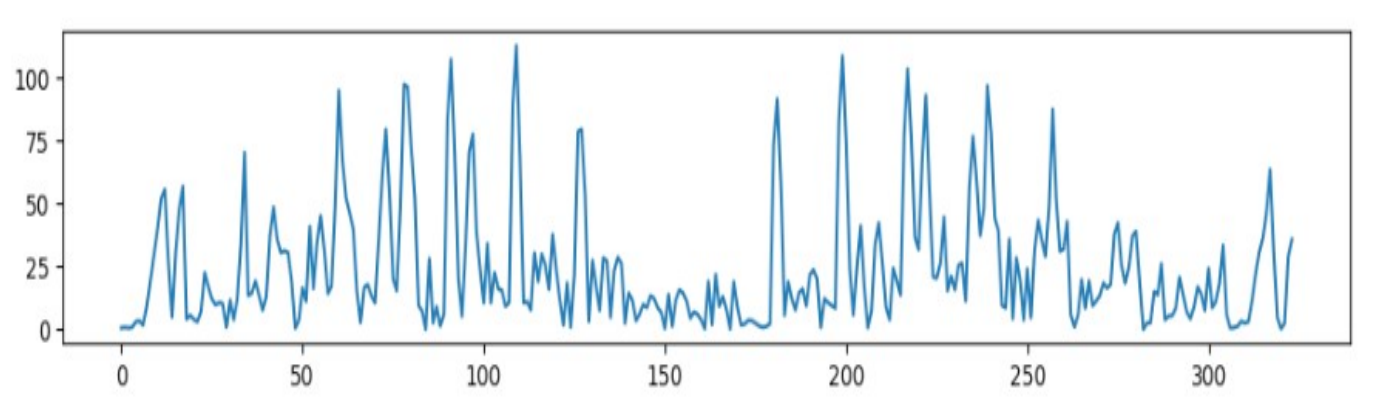


Semester project context:




WP3 – Internship defe
90%
WP2.1 - Report
WS 2.1 - Results
100%
WP2.2 - Presentation
MS 2.2 - Results
80%
WP4 – GUI developme
50%
WP2.1 - Widgets + geome
WS 2.1 - Results
100%
WP2.2 -Events & callbacl
MS 2.2 - Results
50%
WP5 – Segmentation
100%
WP2.1 - Cleaning
WS 2.1 - Results
100%
WP2.2 -Extrema
MS 2.2 - Results
100%
WP3.2 -Clustering
MS 3.2 - Results
100%
WP6 – Ellipse estimat
100%
WP2.1 - Data processing
WS 2.1 - Results
100%
WP2.2 -Covarience confi
MS 2.2 - Results
100%
WP7 – Outputs
100%
WP2.1 - 3D reconstructior
WS 2.1 - Results
100%
WP2.2 -Discussions
MS 2.2 - Results
100%

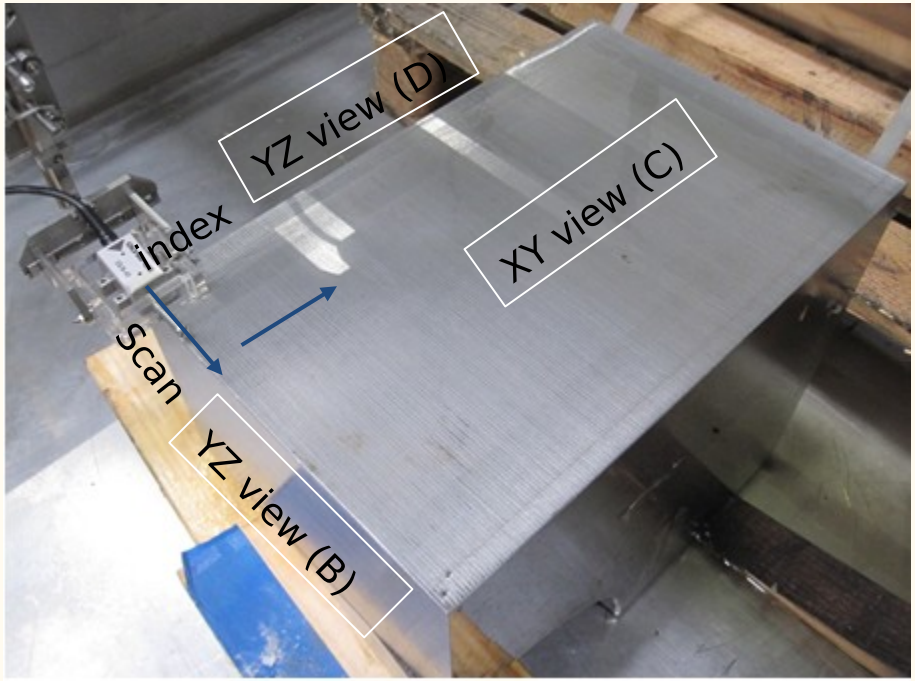
Gradient 1D



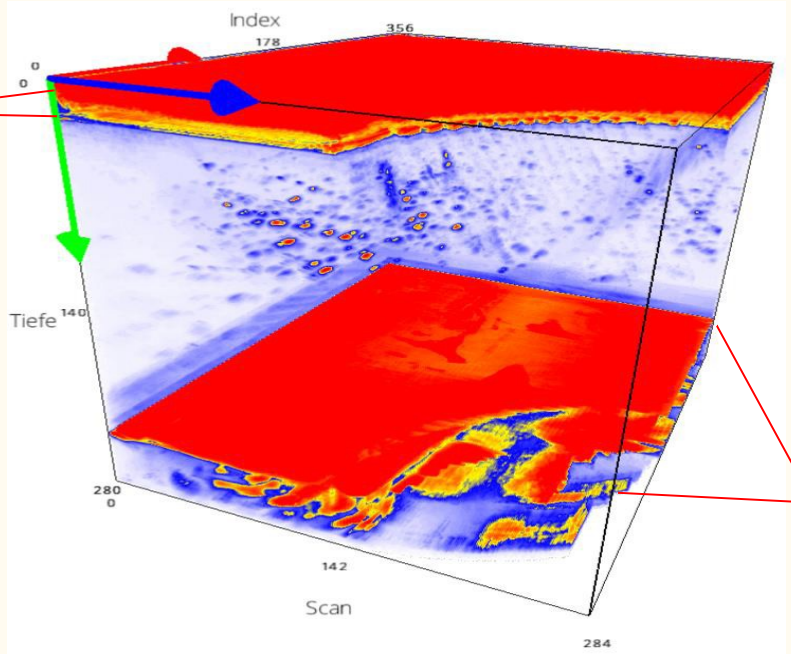
Test objects, coordinate system 0-point

Characteristics:

- Test head: 3.5L16-A3
- 0.25 mm gap (PK and component)
- Test head spring-mounted
- 0-point: center PK component corner



UT imaging result



Far field artifacts

Echo head wall

Mannual solution