DataLab

Automatic Reconstruction of X-ray Scenes with Python and DataLab

https://datalab-platform.com/

EuroSciPy 2026

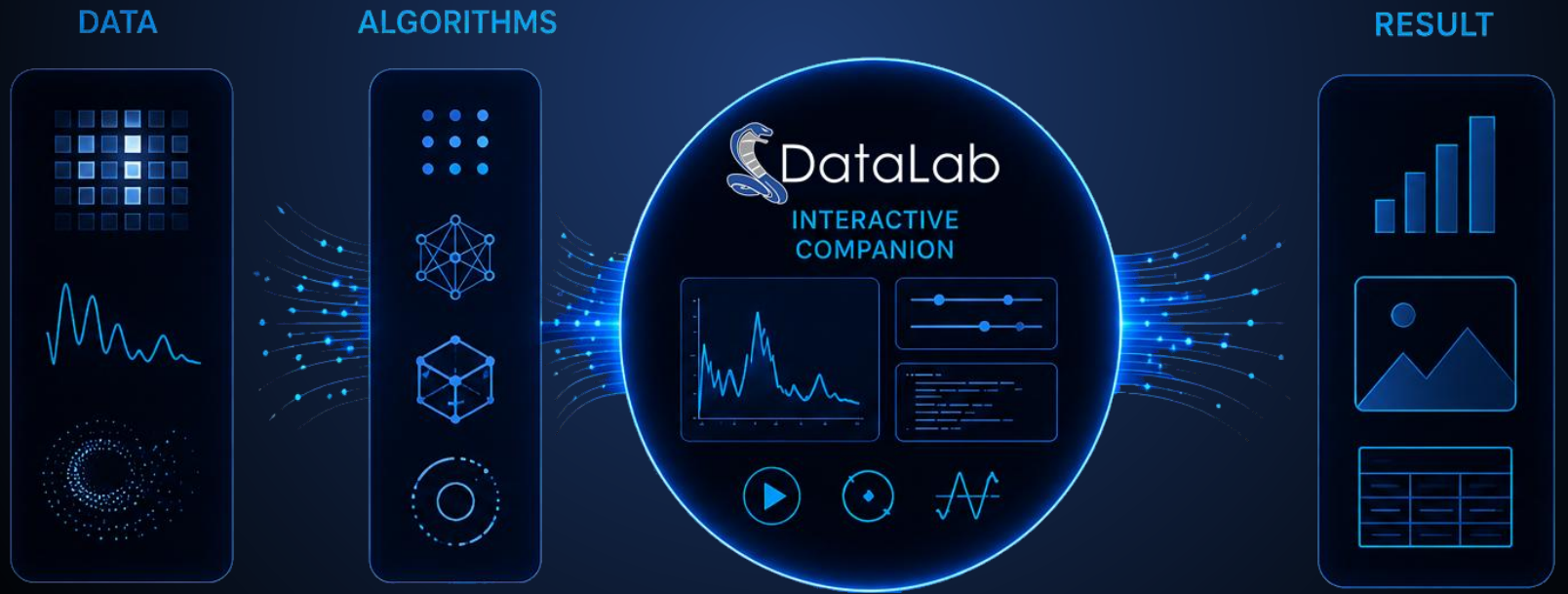
July 21 2026

CODRA

Pierre RAYBAUT

Why this talk?

A real-world case study:
Accelerating scientific algorithm development with DataLab



CODRA

Who am I?

Pierre Raybaut

Executive VP

CÓDRA

INSTITUT d'OPTIQUE GRADUATE SCHOOL

2000

LABORATOIRE CHARLES FABRY

cea

2018

CÓDRA

Industrial supervision / Control-command
Technical business information systems
Scientific software

THALES AVIONICS

CÓDRA

Who am I?

Pierre Raybaut

Executive VP

CODRA

Open-source background

- 2023 — DataLab — Scientific Data Processing Platform
- 2012 — WinPython — 1st Portable Scientific Python Distribution for Windows
- 2009 — Spyder — 1st ready-to-use Scientific Python IDE for all platforms
- 2008 — python(x,y) — 1st all-in-one Scientific Python Distribution for Windows

CODRA

What is **DataLab**?

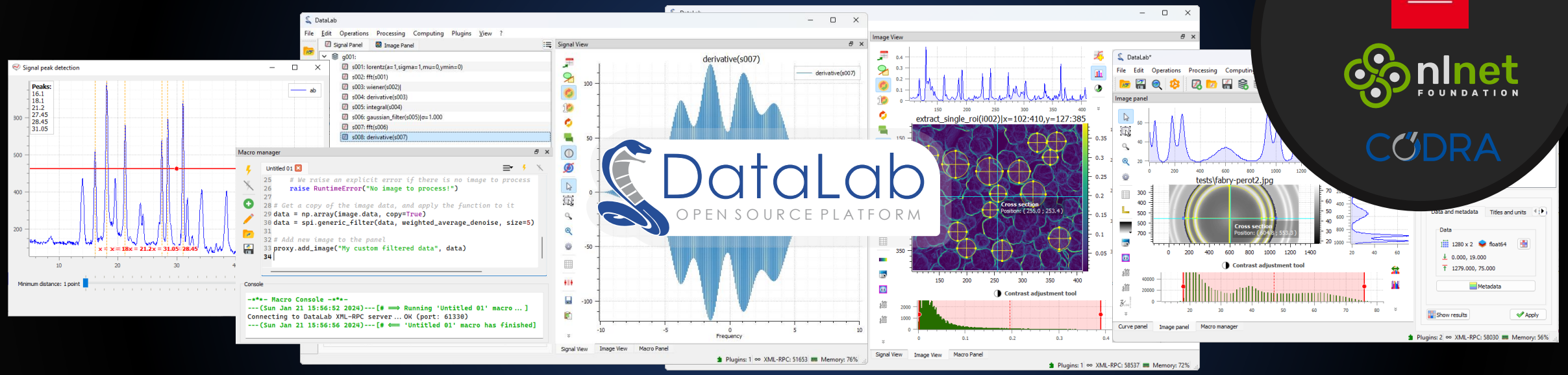
https://datalab-platform.com

Project funded by

cea

nlnet FOUNDATION

CODRA



- Open-source platform for processing and visualizing 1D signals or 2D images
- Built on Python and scientific libraries NumPy SciPy scikit-image OpenCV PlotPyStack Sigima

200+ integrated processing features

CODRA

DataLab in a nutshell

- Ergonomic
- Interoperable (remote-control API)
- Extensible (Python API: plugins, macros)
- Robust (industrial-grade software)

- Already used in production (e.g. Laser Mégajoule)

CODRA

DataLab: in research, industry and education...

Spectral measurement processing (physical sensors)

Medical or microscopy image analysis

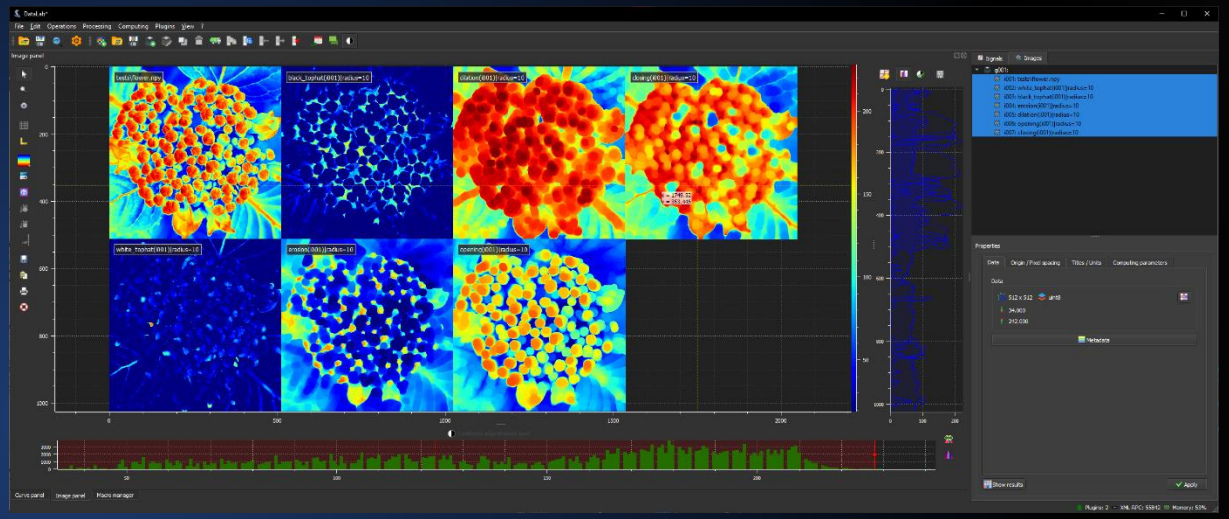
Experimental data visualization

R&D processing-chain validation

Interactive debugging of internal processing workflows

CODRA

DataLab use-case categories



Signal and image processing

CODRA

DataLab use-case categories

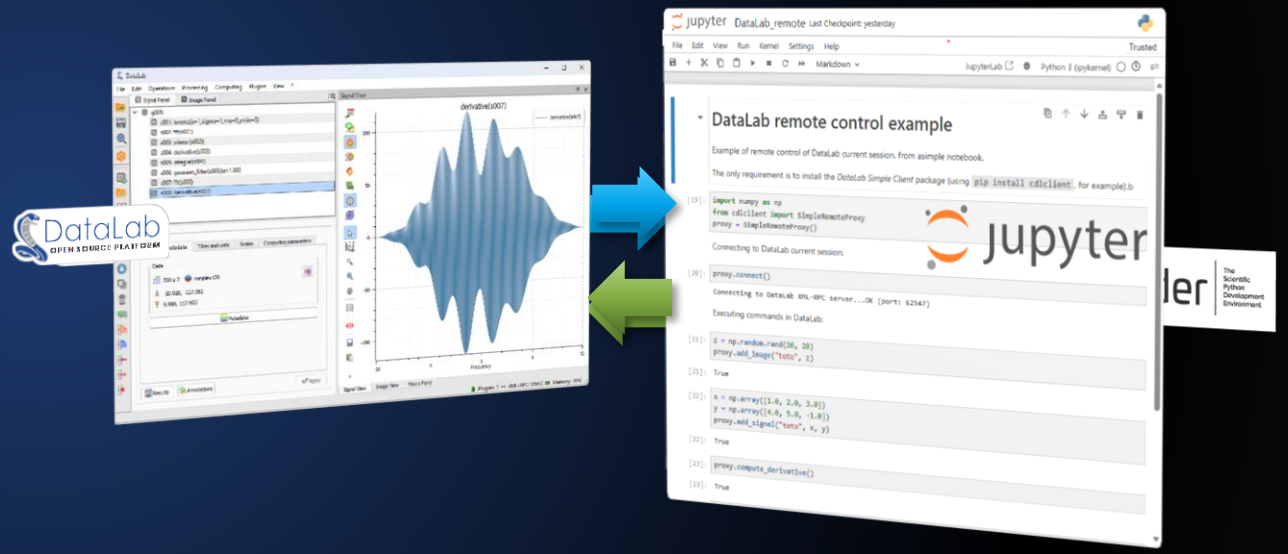
DataLab use-case categories

Analysis
Explore data, design and tune processing algorithms

Prototyping
Prototype ideas and algorithms rapidly

Advanced debugging
Inspect and debug internal processing workflows

Integration
Integrate DataLab in larger applications and workflows



Interact with your own code

CODRA

DataLab use-case categories

DataLab use-case categories

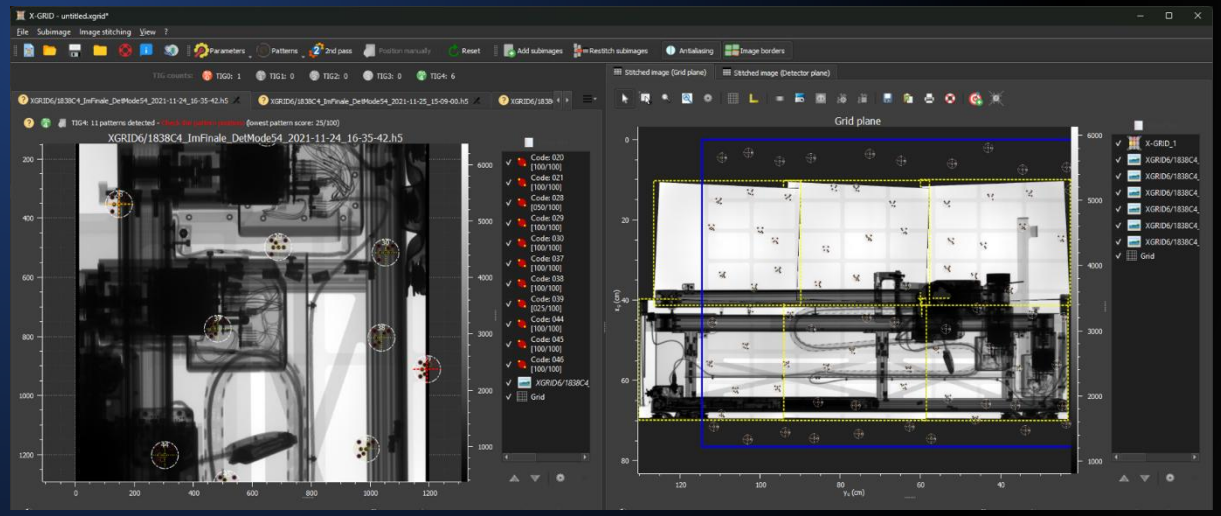
Analysis
Explore data, design and tune processing algorithms

Prototyping
Prototype ideas and algorithms rapidly

Advanced debugging
Inspect and debug internal processing workflows

Integration
Integrate DataLab in larger applications and workflows

X-GRID application (that's today's topic!)



DataLab use-case categories

DataLab use-case categories

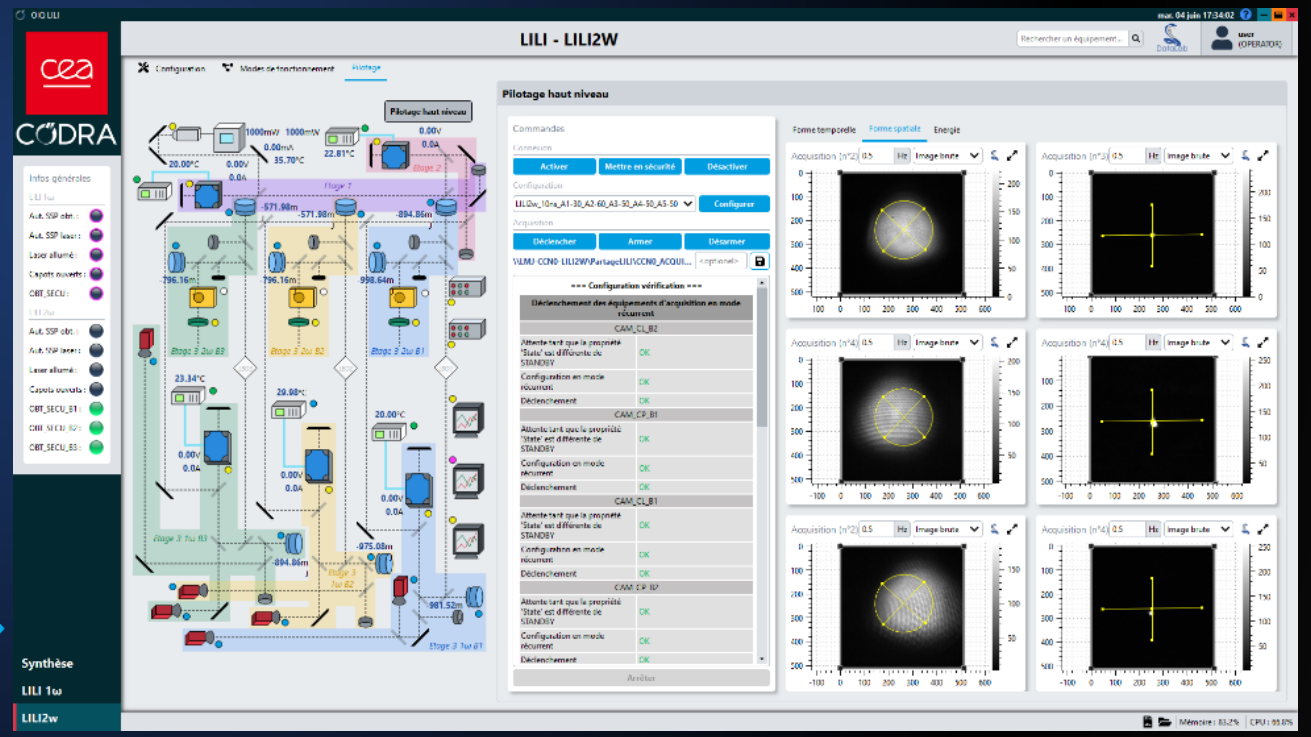
Analysis
Explore data, design and tune processing algorithms

Prototyping
Prototype ideas and algorithms rapidly

Advanced debugging
Inspect and debug internal processing workflows

Integration
Integrate DataLab in larger applications and workflows

Embedded in Laser Megajoule's control system



From DataLab use cases to X-GRID

DataLab use-case categories

1

Analysis
Explore data, design and tune processing algorithms ✓

Prototyping
Prototype ideas and algorithms rapidly ○

2

Advanced debugging
Inspect and debug internal processing workflows ✓

Integration
Integrate DataLab in larger applications and workflows ○

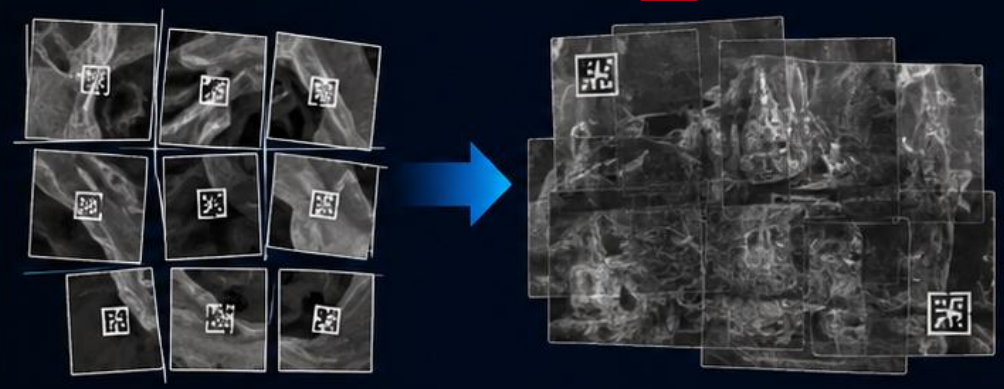
We will now see DataLab in action on this case study

Industrial case study

X-GRID
Automatic reconstruction of large X-ray scenes

X-GRID
X-ray image stitching based on automatic grid patterns detection

cea CODRA



CODRA

X-GRID: context and motivation

- Use case: **radiography of large objects** → typically 1–2 meters wide
- **Field-deployable** acquisition → indoors, outdoors, wherever inspection is needed
- Mobile X-ray equipment → imaging plates or digital detector, typically 35 cm × 43 cm
- Consequence → the full scene cannot be captured in a single shot
- Need → reconstruct the complete radiographic scene from multiple smaller-field X-ray images

- **X-GRID** is an application specified and developed by CODRA for CEA,
  to automatically reconstruct complete radiographic scenes from partial X-ray images

Contact @ CEA:
Marc Rabec le Gloahec

X-GRID

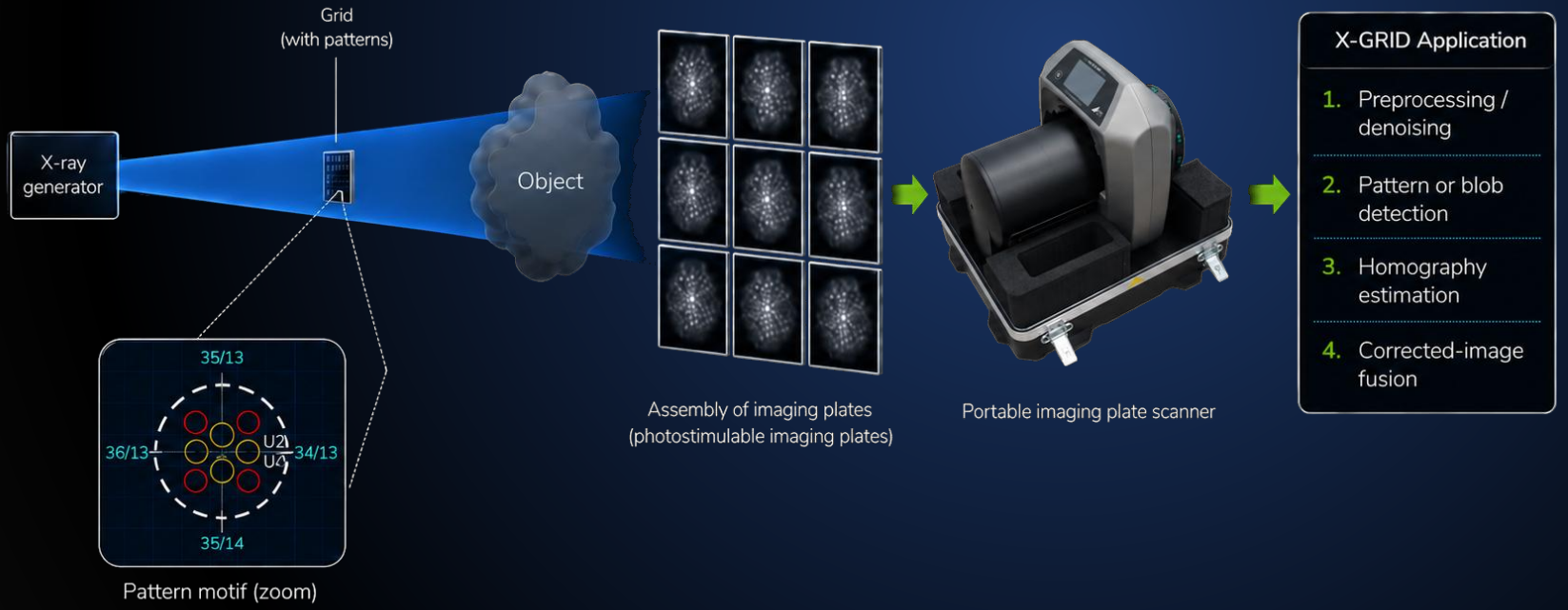
X-ray image stitching based on automatic grid patterns detection

CODRA

CODRA

Reconstructing large X-ray scenes

- **X-GRID** is an application specified and developed by CODRA for CEA, dedicated to the automatic reconstruction of radiographic scenes from partial X-ray images



X-GRID

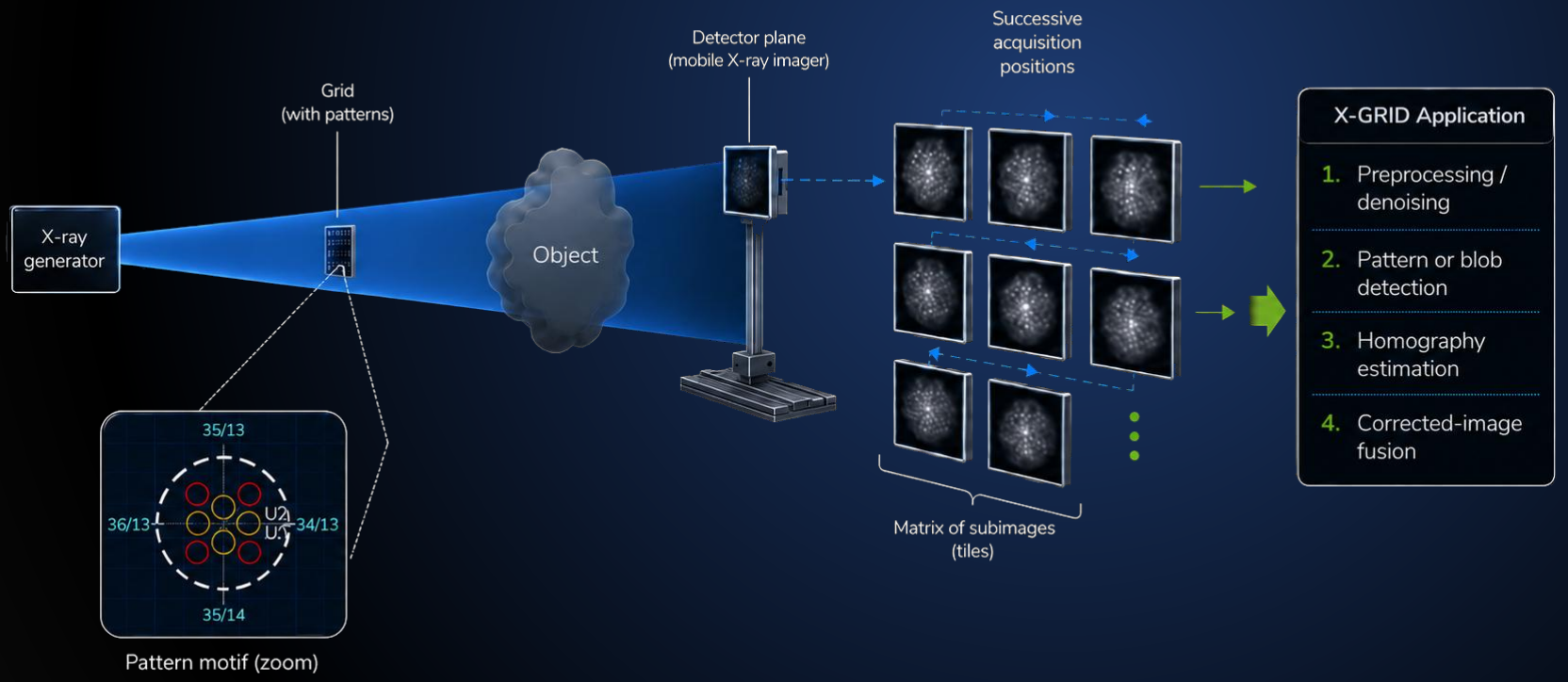
X-ray image stitching based on automatic grid patterns detection

cea CODRA

CODRA

Reconstructing large X-ray scenes

- **X-GRID** is an application specified and developed by CODRA for CEA, dedicated to the automatic reconstruction of radiographic scenes from partial X-ray images



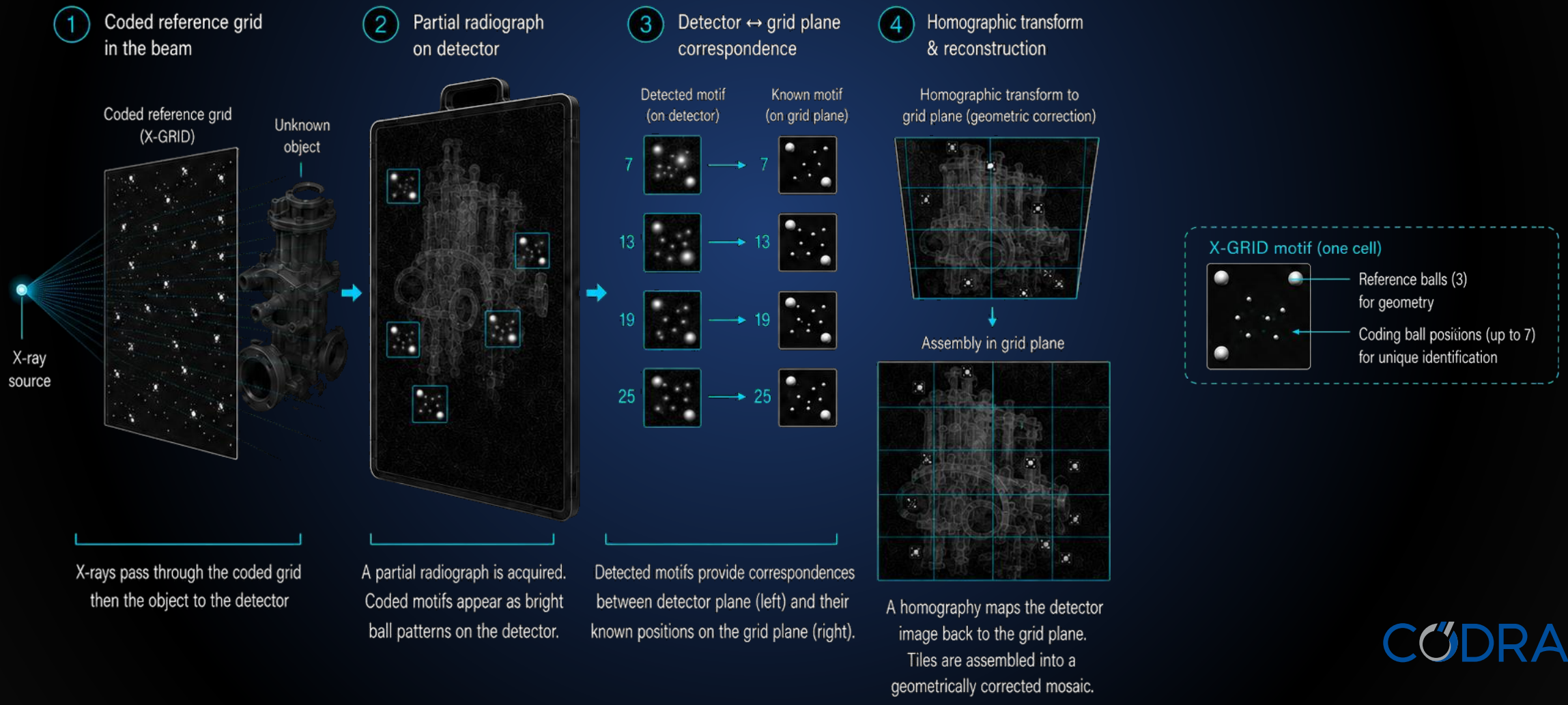
X-GRID

X-ray image stitching based on automatic grid patterns detection

cea CODRA

CODRA

The key idea: a coded reference grid

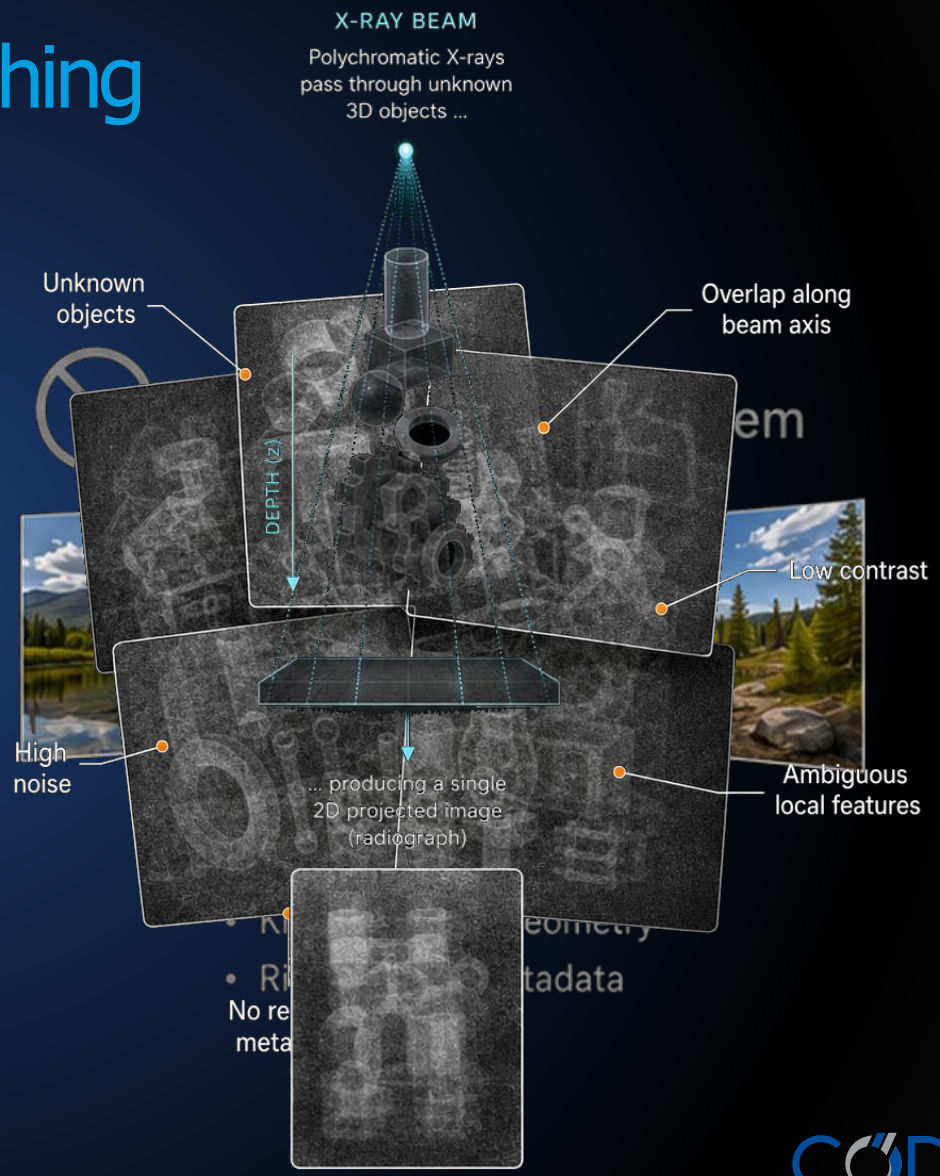


Why this is not standard image stitching

- X-ray images are not outdoor photographs
- They are projections through unknown objects, with structures superimposed along the beam axis

- Typical difficulties:
  - Low contrast
  - High noise
  - No natural texture continuity
  - No reliable acquisition metadata

- So this is not a panorama problem
- It requires a dedicated reconstruction approach

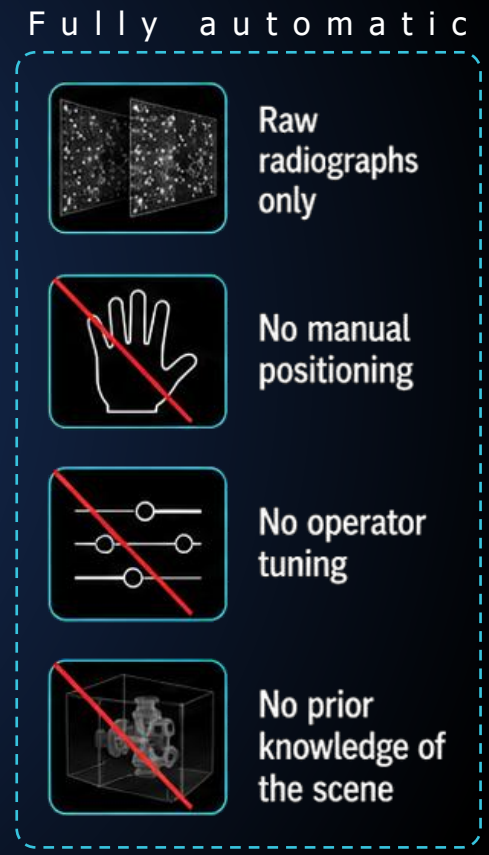


From physical principle to software reconstruction

- The coded grid makes reconstruction possible

- The software challenge is:
  - To go from **raw radiographs** to a **geometrically corrected mosaic**:
    - Detect/decode
    - Estimate homography
    - Correct/assemble
  - And to do it **fully automatically**:
    - No manual positioning
    - No operator tuning
    - No prior knowledge of the scene

- This is where CODRA developed **X-GRID** — and where **DataLab** became essential



About X-GRID application

- X-GRID is a python desktop application that strongly leverages on:
  - **Qt** for modern user interfaces Qt
  - Powerful open-source processing libraries: **SciPy**, **scikit-image**, **OpenCV**

    scikit-image OpenCV SciPy
  - **PlotPyStack** libraries (guidata, PlotPy) for efficient 2D visualization

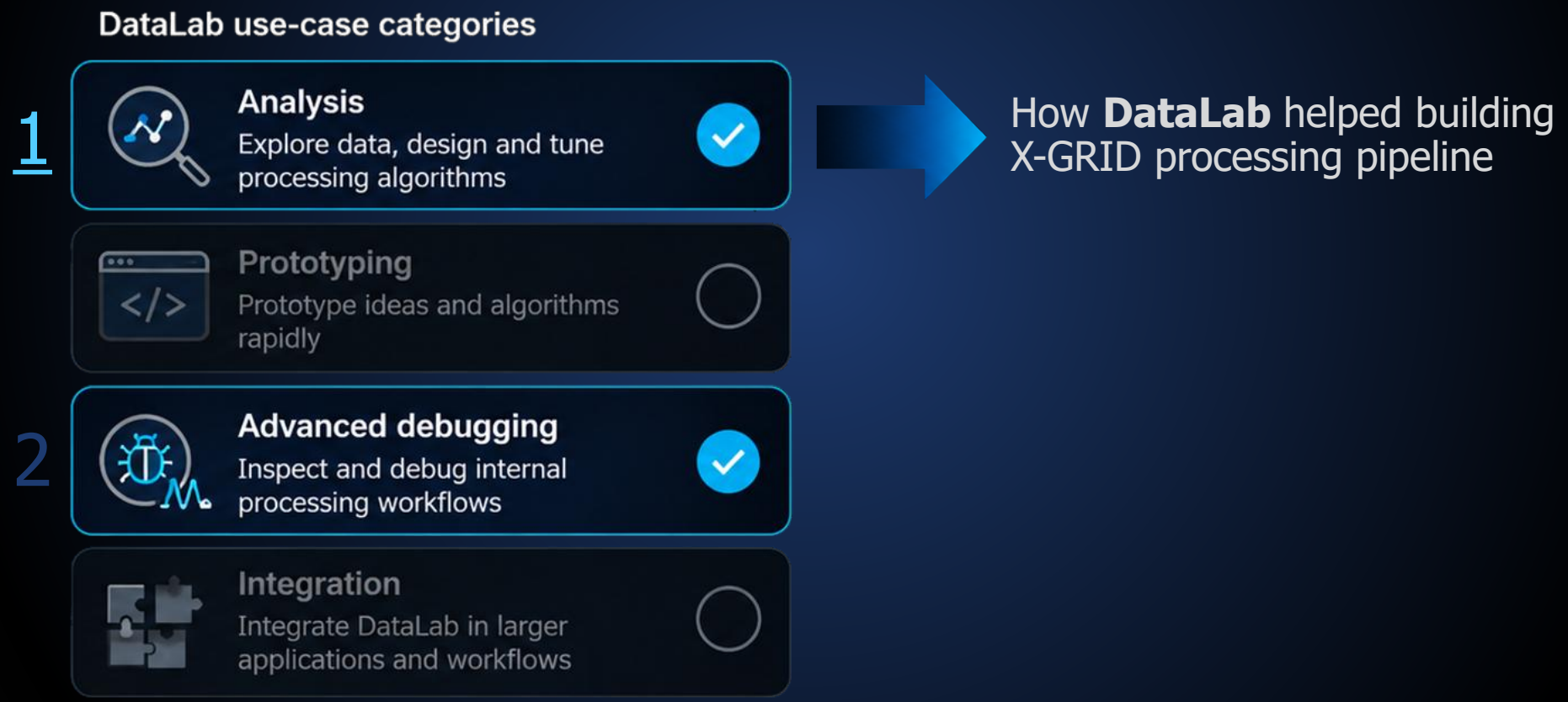
    PlotPyStack https://github.com/PlotPyStack

    The only practical solution for working smoothly with images this large:
    - Each sub-image can be up to 8600 pixels × 7000 pixels × 16 bits ~120MB (uncompressed data in memory)
    - A typical assembly contains 4 to 8 sub-images
  - **DataLab** for prototyping and advanced debugging (remote control)

    DataLab https://datalab-platform.com

CODRA

DataLab's Use case 1 – Building X-GRID processing pipeline



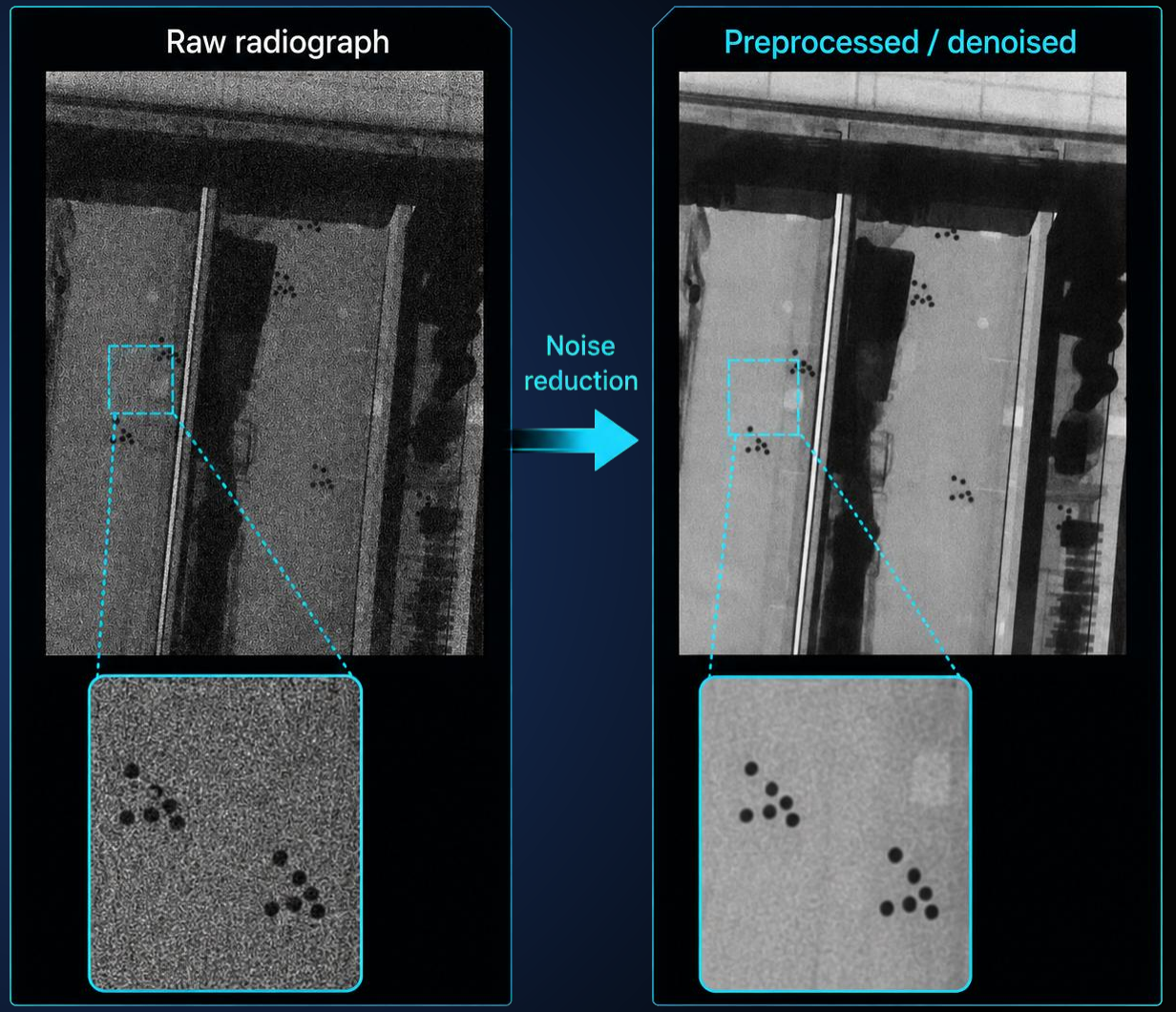
How **DataLab** helped building X-GRID processing pipeline

CODRA

DataLab's Use case 1

Pipeline: from Raw radiographs
to geometrically corrected mosaic

1. Preprocessing / denoising
2. Coded-pattern detection & decoding
3. Homography estimation
4. Geometric correction & image fusion



DataLab **What DataLab made possible**

DataLab made it easy to rapidly explore and tune a denoising strategy optimized for robust blob detection.

scikit-image
- exposure.equalize_hist
- restoration.denoise_bilateral
- restoration.denoise_tv_chambolle

SciPy
- ndimage.median_filter
- ndimage.uniform_filter

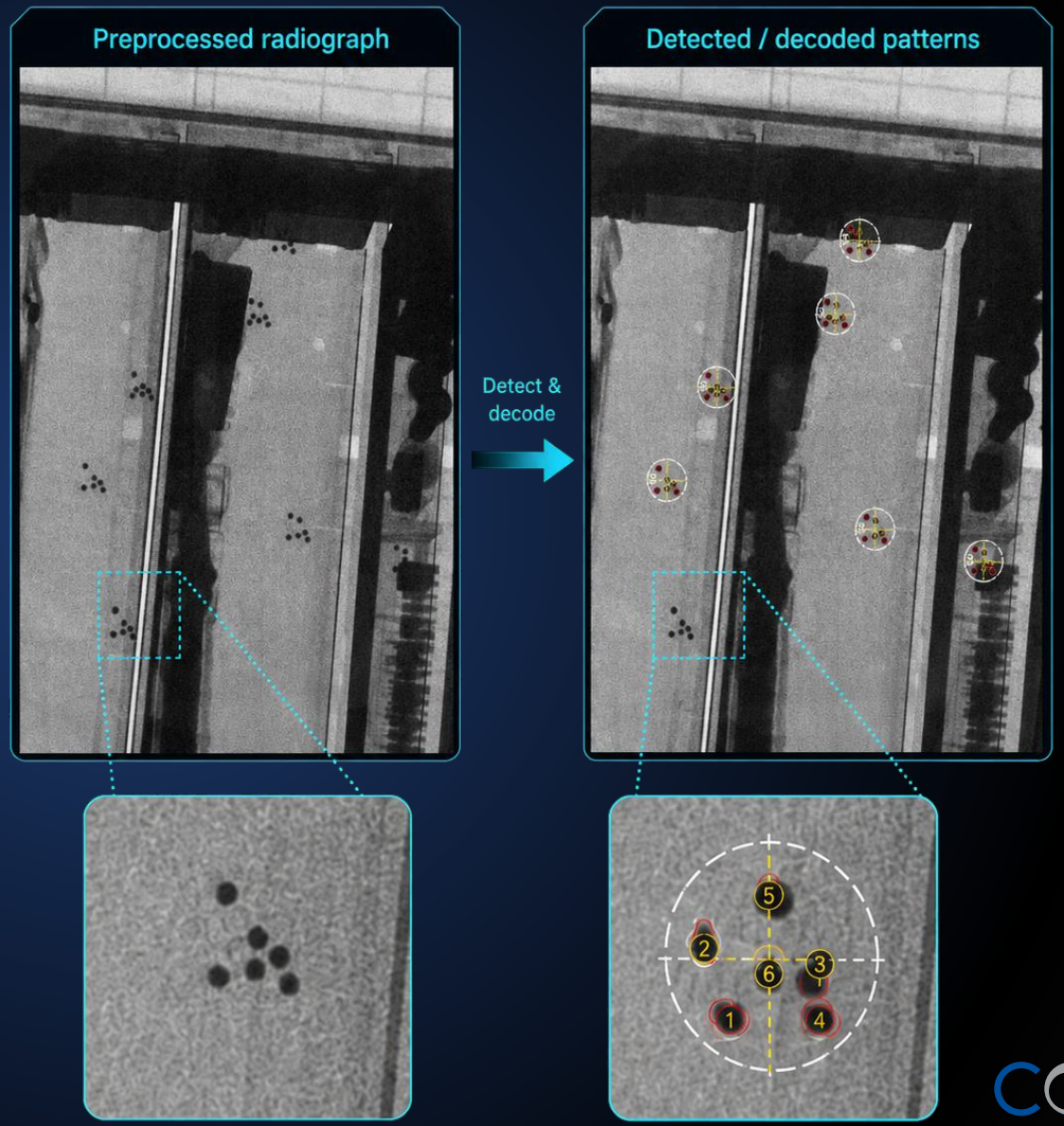
Easy comparison of variants, interactive parameter tuning, and combination of operators from multiple open-source libraries.

CODRA

DataLab's Use case 1

Pipeline: from Raw radiographs
to geometrically corrected mosaic

1. Preprocessing / denoising
2. Coded-pattern detection & decoding
3. Homography estimation
4. Geometric correction & image fusion



DataLab

What DataLab made possible

DataLab helped tune blob-detection parameters, select OpenCV SimpleBlobDetector, and design the full coded-pattern detection algorithm.

OpenCV

OpenCV SimpleBlobDetector

blob detector selection and parameter tuning

- Remote control
- Advanced debugging
- Design of the full coded-pattern detection algorithm

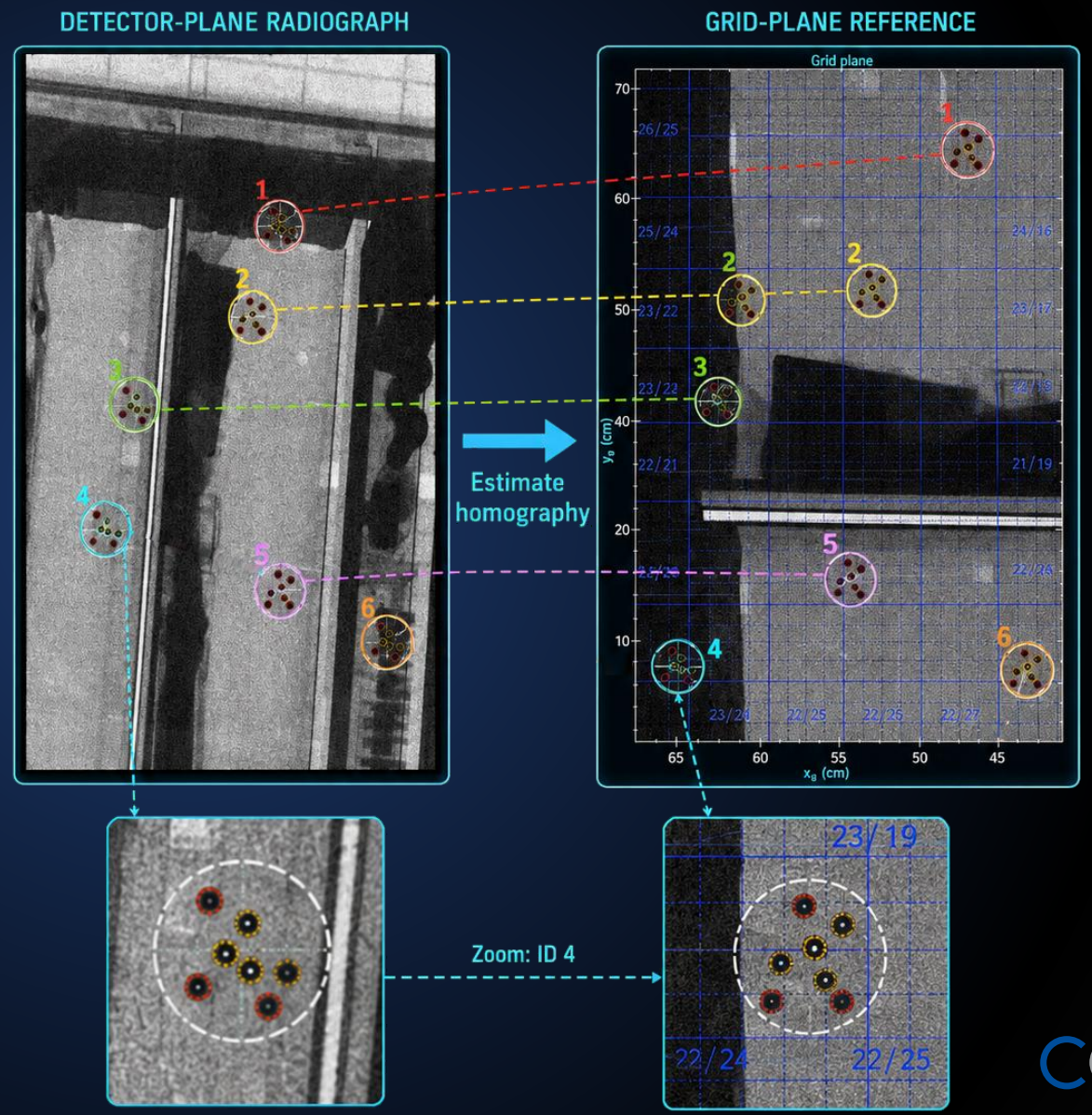
DataLab made it possible to go beyond isolated blob detection and iteratively build the complete coded-pattern detection and decoding workflow.

CODRA

DataLab's Use case 1

Pipeline: from Raw radiographs
to geometrically corrected mosaic

1. Preprocessing / denoising
2. Coded-pattern detection & decoding
3. Homography estimation
4. Geometric correction & image fusion



DataLab | **What DataLab made possible**

DataLab helped tune geometric-transformation parameters,
select OpenCV findHomography,
and design the full homography-estimation workflow.

OpenCV

OpenCV findHomography

projective transform estimation

- Remote control
- Advanced debugging
- Design of the full homography-estimation workflow

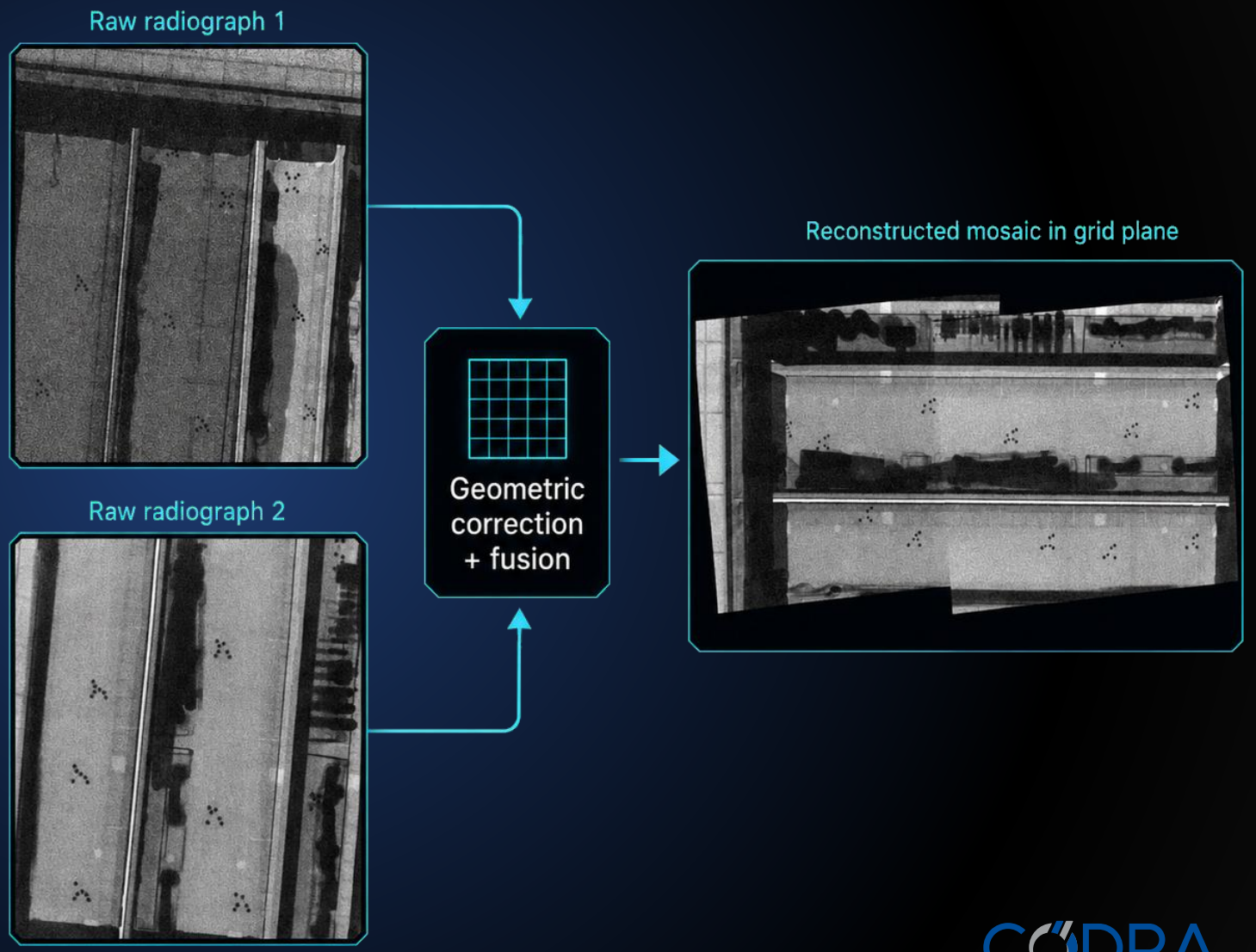
DataLab made it possible to go beyond isolated OpenCV calls
and iteratively build the complete homography-estimation workflow.

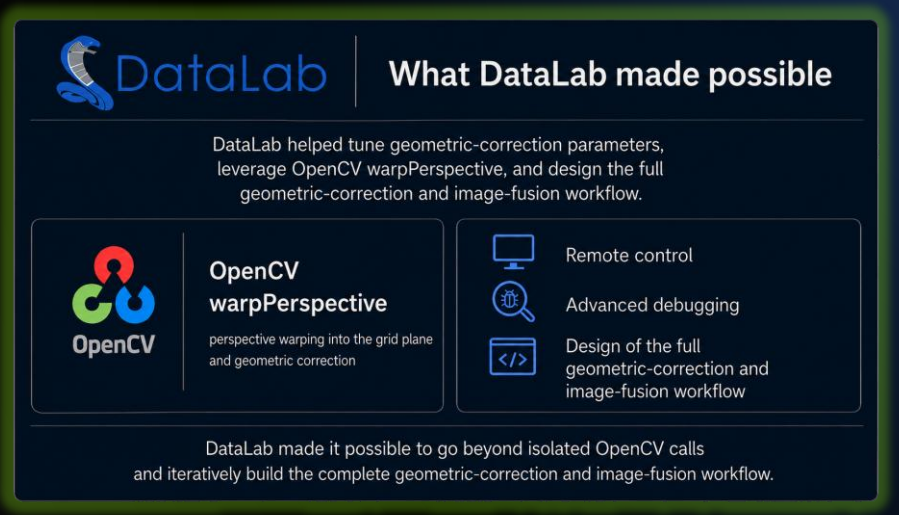
CODRA

DataLab's Use case 1

Pipeline: from Raw radiographs to geometrically corrected mosaic

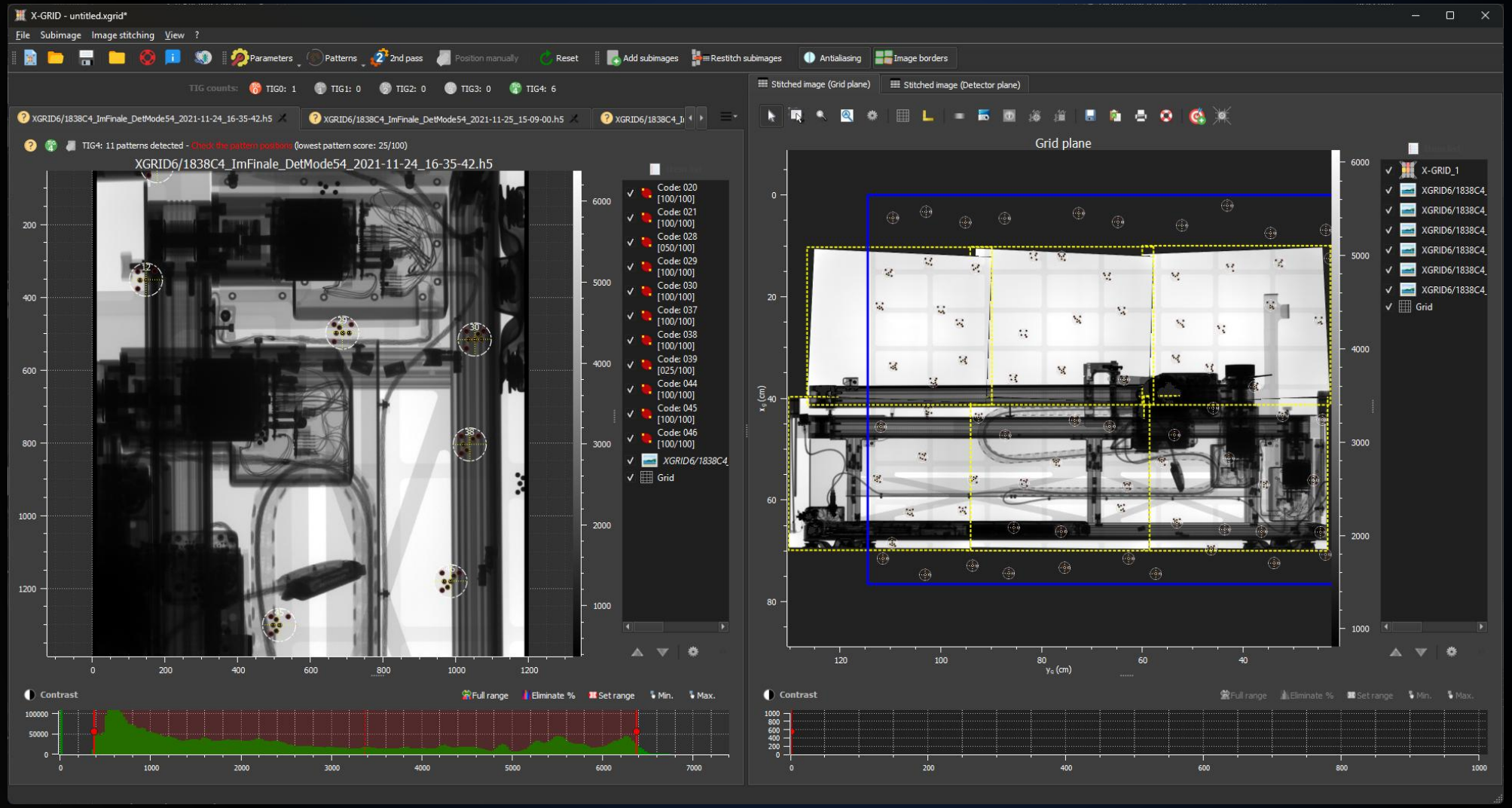
1. Preprocessing / denoising
2. Coded-pattern detection & decoding
3. Homography estimation
4. Geometric correction & image fusion



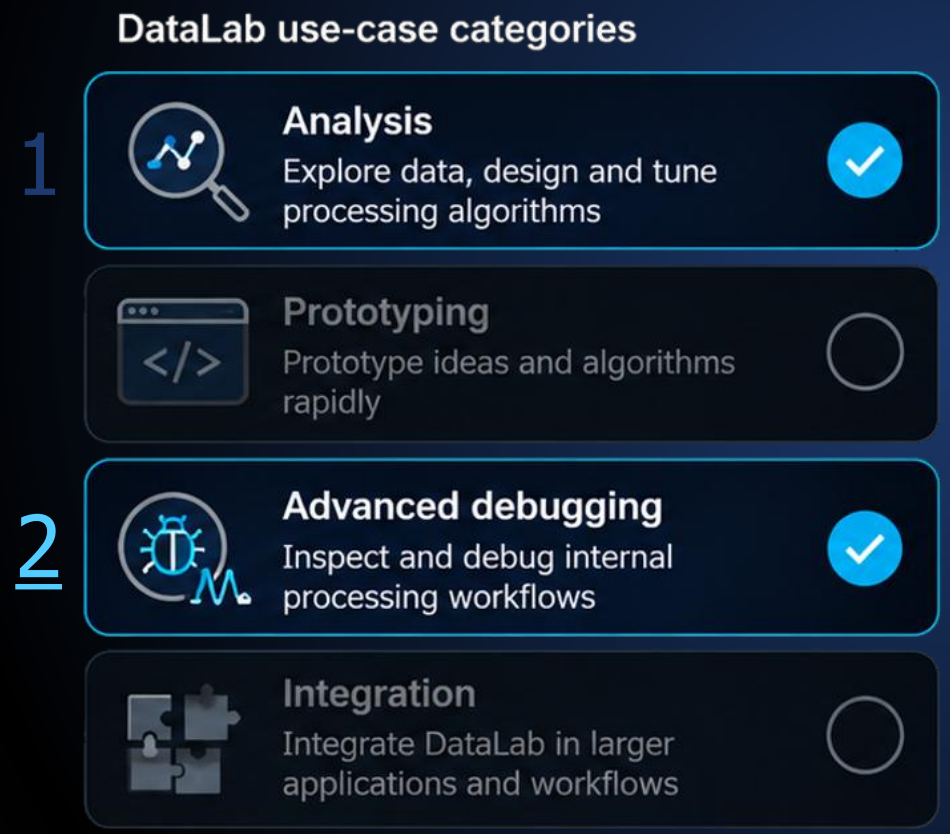


CODRA

Demo time: X-GRID in action



DataLab's Use case 2 – Advanced debugging inside X-GRID



Once the first algorithm was implemented in X-GRID, **DataLab** became a live inspection and debugging companion

Classical debugger

Code (xgrid_processing.py)

```
21  def process_image(image, n);
22      acc = 0.0
23      if cond:
24          compute(image)
25          for i in range(n):
26              acc += process(buf[i])
27          update(result)
28      return result
```

Stack

process_image()
detect_patterns()
main()
<module>()

Variables

acc = 0.345
i = 42
n = 256
ptr = <memory ...>

Arrays

12	7	3	91	...
4	0	8	16	...
23	5	9	11	...

References

image	<ndarray>
buf	<ndarray>
result	<ndarray>

Breakpoints

- xgrid_processing.py:24 ☑
- xgrid_processing.py:42 ☐

Running in X-GRID

Remote control

Processing pipeline

Live inspection of intermediate processing results

DataLab-connected debugger

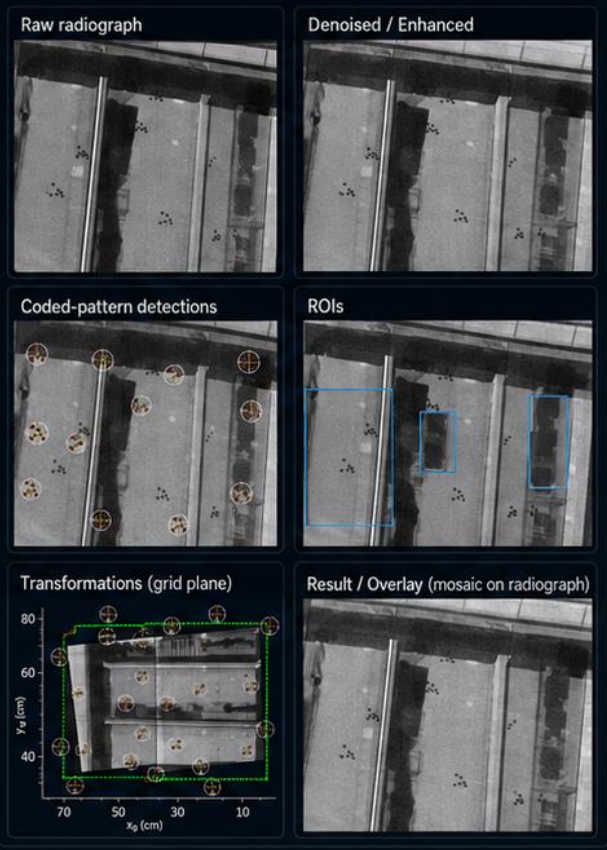
X-GRID (Application)

- Acquire radiograph
- Preprocess
- Detect coded features
- Estimate geometry
- Reconstruct mosaic
- Update results

Running

DataLab (Live inspection)

Intermediate images | Annotations | ROIs | Detections | Transformations



Timeline (processing steps)

CODRA

Remote-controlling DataLab from X-GRID

- Sigima (DataLab's engine) provides a remote control client for DataLab:

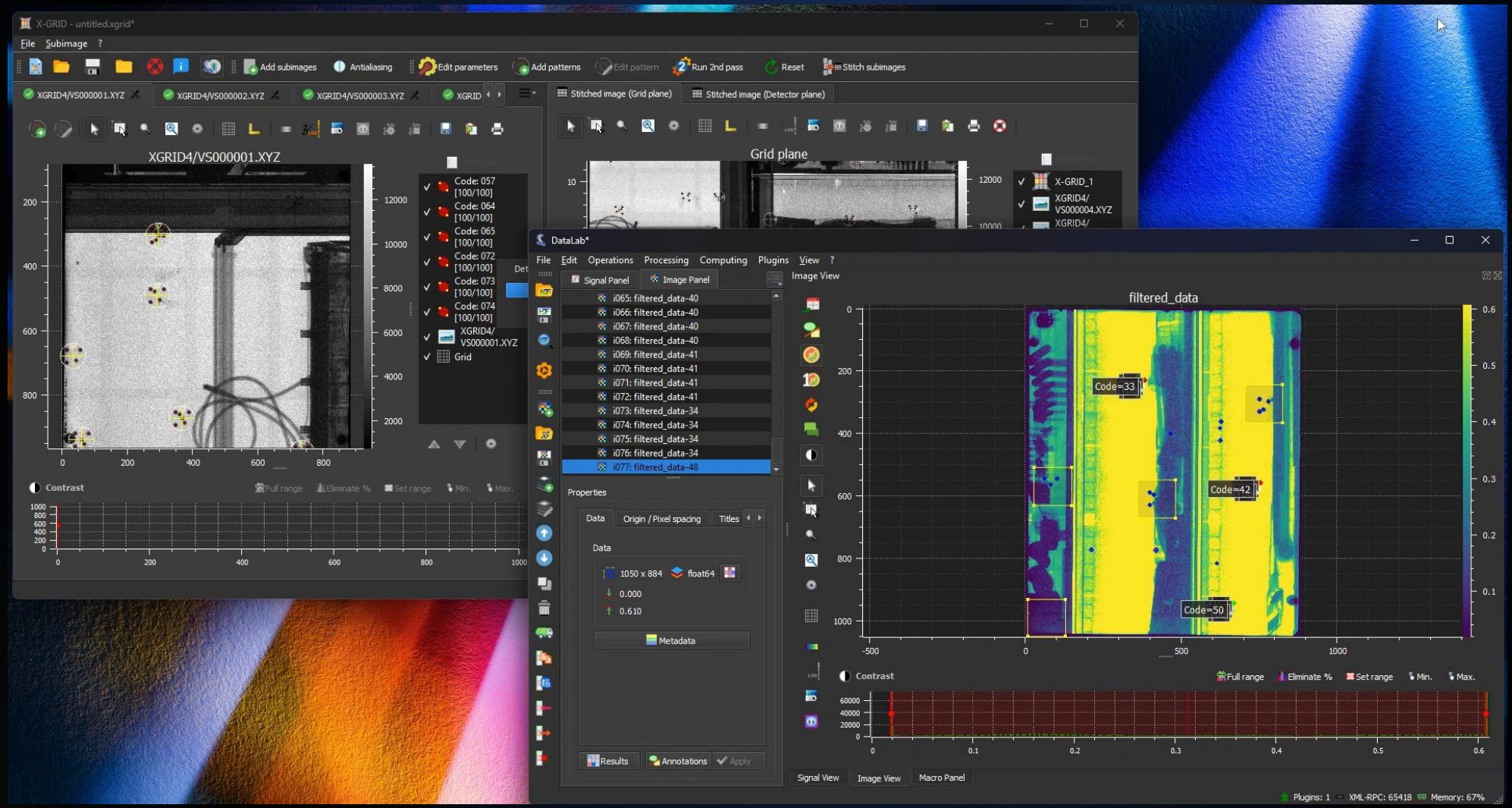
```python
# DataLab remote control client:
from sigima.client import SimpleRemoteProxy
```

- At any breakpoint, X-GRID can send intermediate data to DataLab for interactive inspection:

```
107   class Grid:
118       def __init__(
129               self.dl_proxy = SimpleRemoteProxy()
130               self.dl_proxy.connect(timeout=1.0, retries=1)
131           except Exception:  # pylint: disable=broad-except
```

```
1181              if self.debug:
1182                  self.dl_proxy.add_image(f"filtered_data-{code:02d}", filtered_data)
1183                  fitms = []
1184                  for blob in broi:
1185                      x, y = blob[:2]
1186                      fitms += [make.circle(x - ballradius, y, x + ballradius, y)]
1187                  self.dl_proxy.add_annotations_from_items(fitms)
```

Demo: X-GRID running, DataLab observing



Takeaways

1. **Scientific Python can solve hard industrial imaging problems**
   X-GRID turns noisy partial X-ray images into a geometrically corrected mosaic
2. **DataLab accelerates algorithm design**
   Interactive experimentation makes it easier to test, combine and tune processing ideas
3. **DataLab bridges exploration and production**
   Remote control brings interactive inspection directly into production software

See it. Understand it. **Improve it.**

CODRA

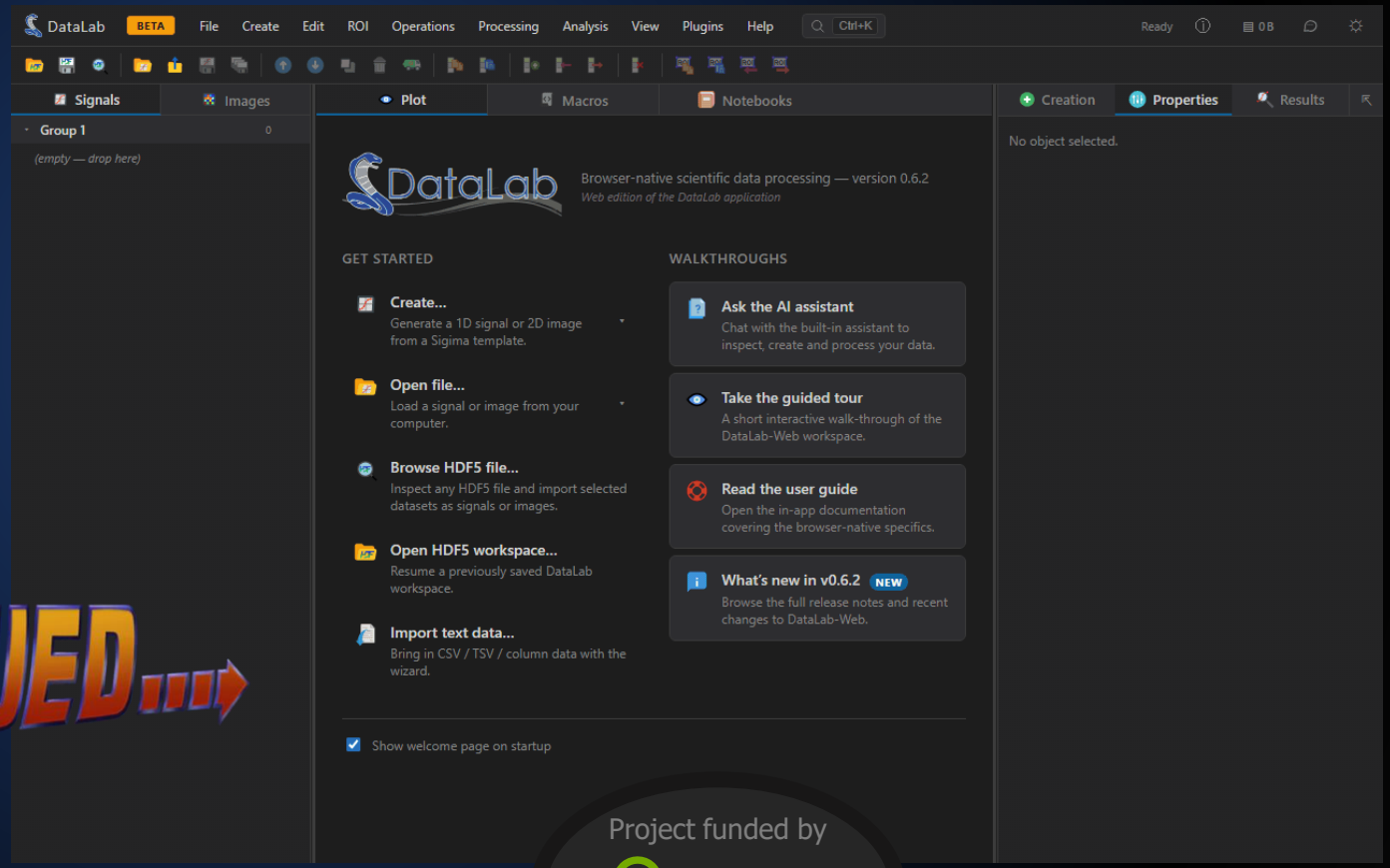
Beyond desktop: **DataLab Web**

- All DataLab features, accessible from a browser
- Local execution, no backend, no upload
- SDK TypeScript for embedding and remote control

PYODIDE WA

Sigima

TO BE CONTINUED...



Available since June 2026:
https://datalab-platform.com/web

Project funded by nlnet FOUNDATION

CODRA

Pierre Raybaut

Executive Vice President, Engineering.
Creator of Spyder (the scientific Pyth...

Thank you!

EuroSciPy 2026

CODRA