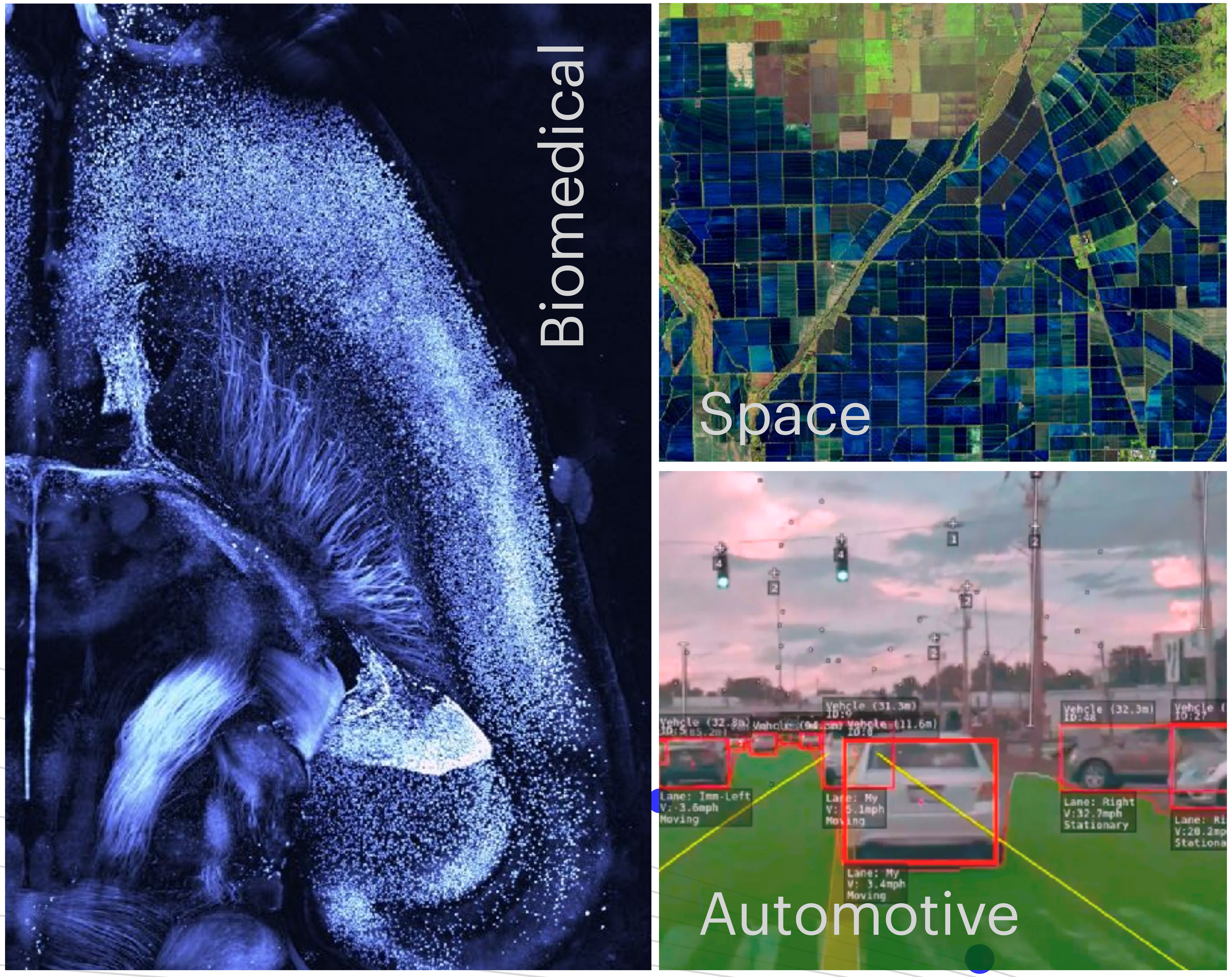
Dotphoton

EIPICSA 2



dotphoton.com

What we had: the Jetraw algorithm

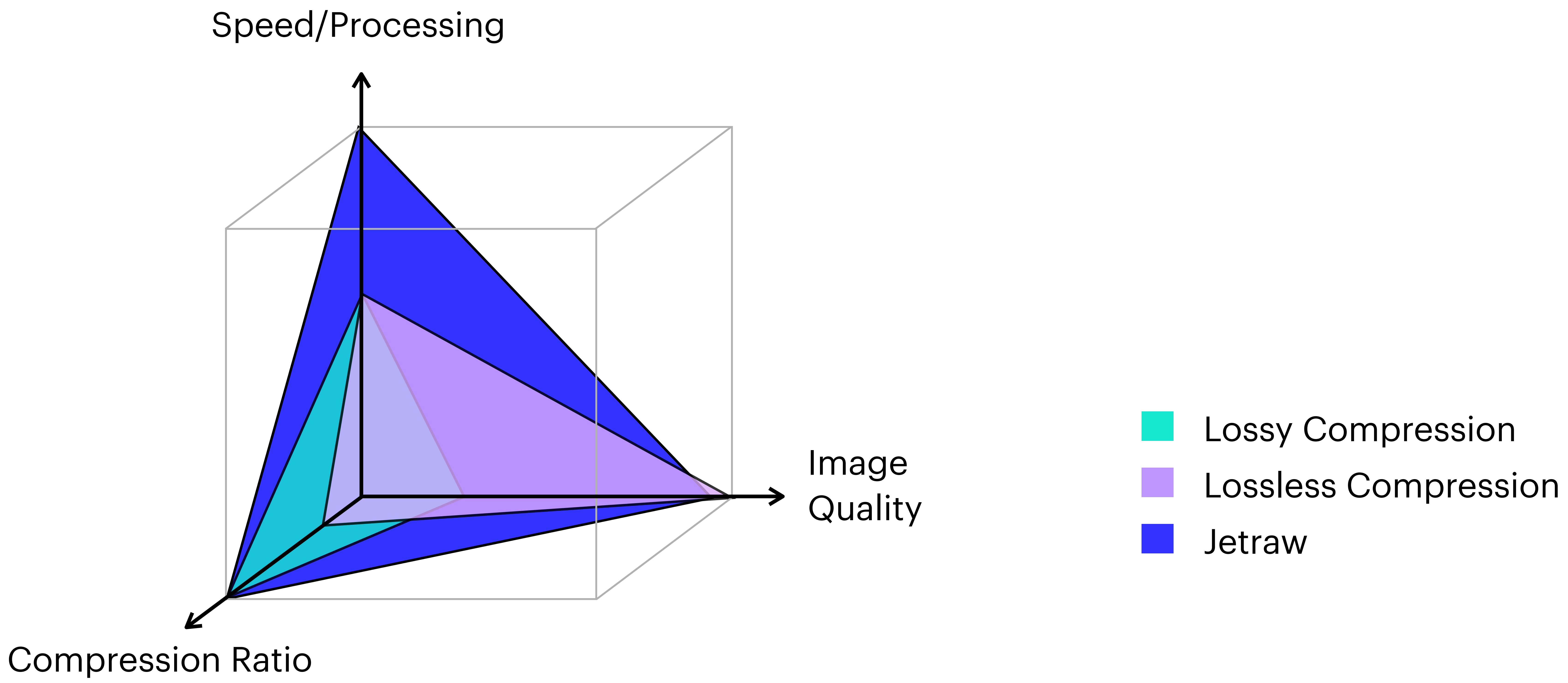
- A compression algorithm that treats images as measurements.
- Market growth due especially to an increase in Machine Learning applications.
- A proof-of-concept for the space implementation (GSTP Element 1).

What we wanted to achieve

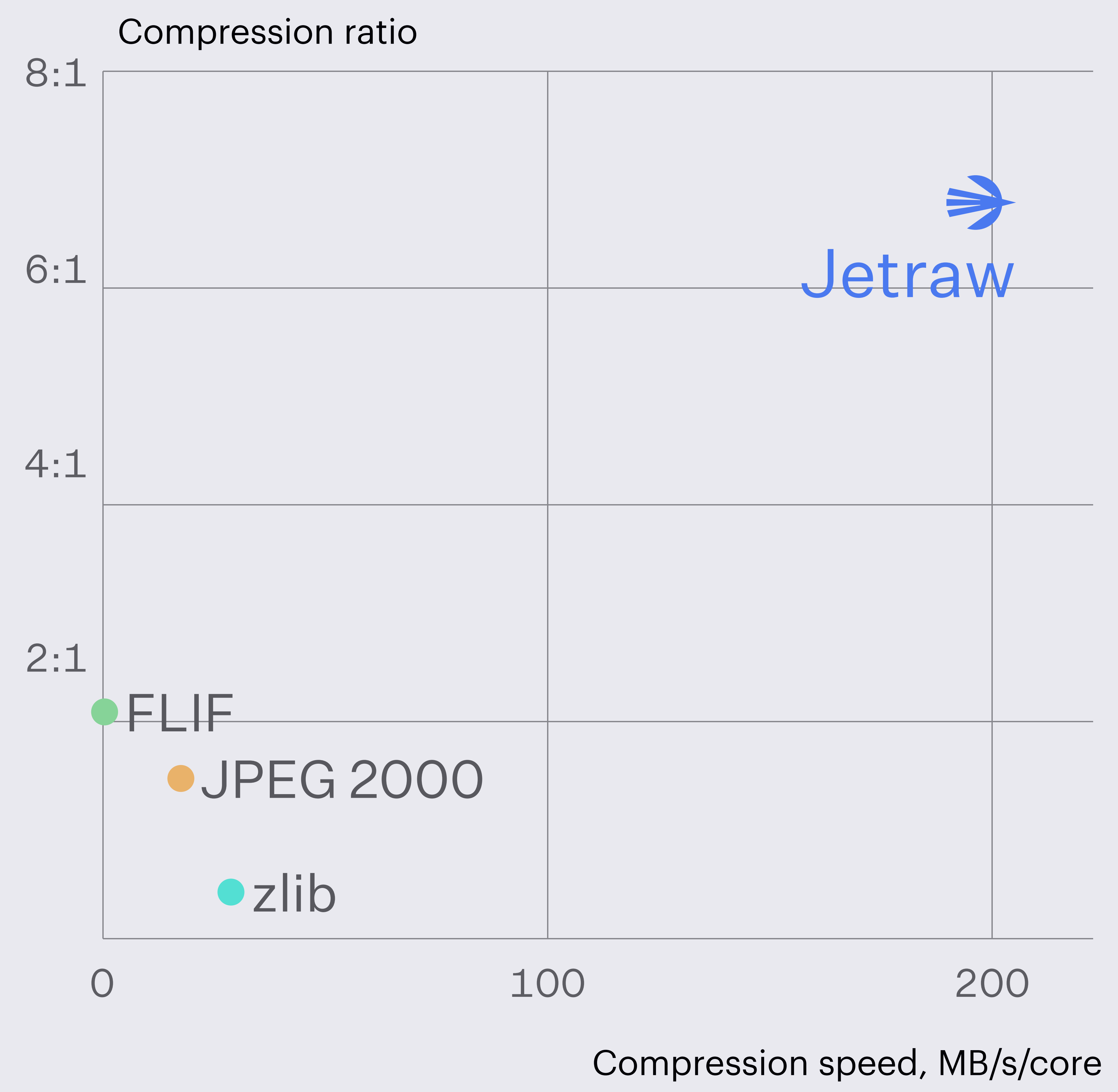
- An FPGA implementation, suitable for being integrated in space missions.
- Test and validate image compression before the mission (quality, compression ratio)
- Improve our development process and tools
- Make a portable demonstrator camera with compression for trade shows
- Demonstrate the increased performance that can be had when ML algorithms are used with raw data.
- Find customers for the products

The Jetraw algorithm

Image compression is a trade-off between quality, speed and size



Jetraw has the best in class performance in compression ratio and speed thanks to embedded sensor noise model



" It was extremely appealing from PCO's point of view to obtain a compression method that exploits the individual image capture chain of each camera model"

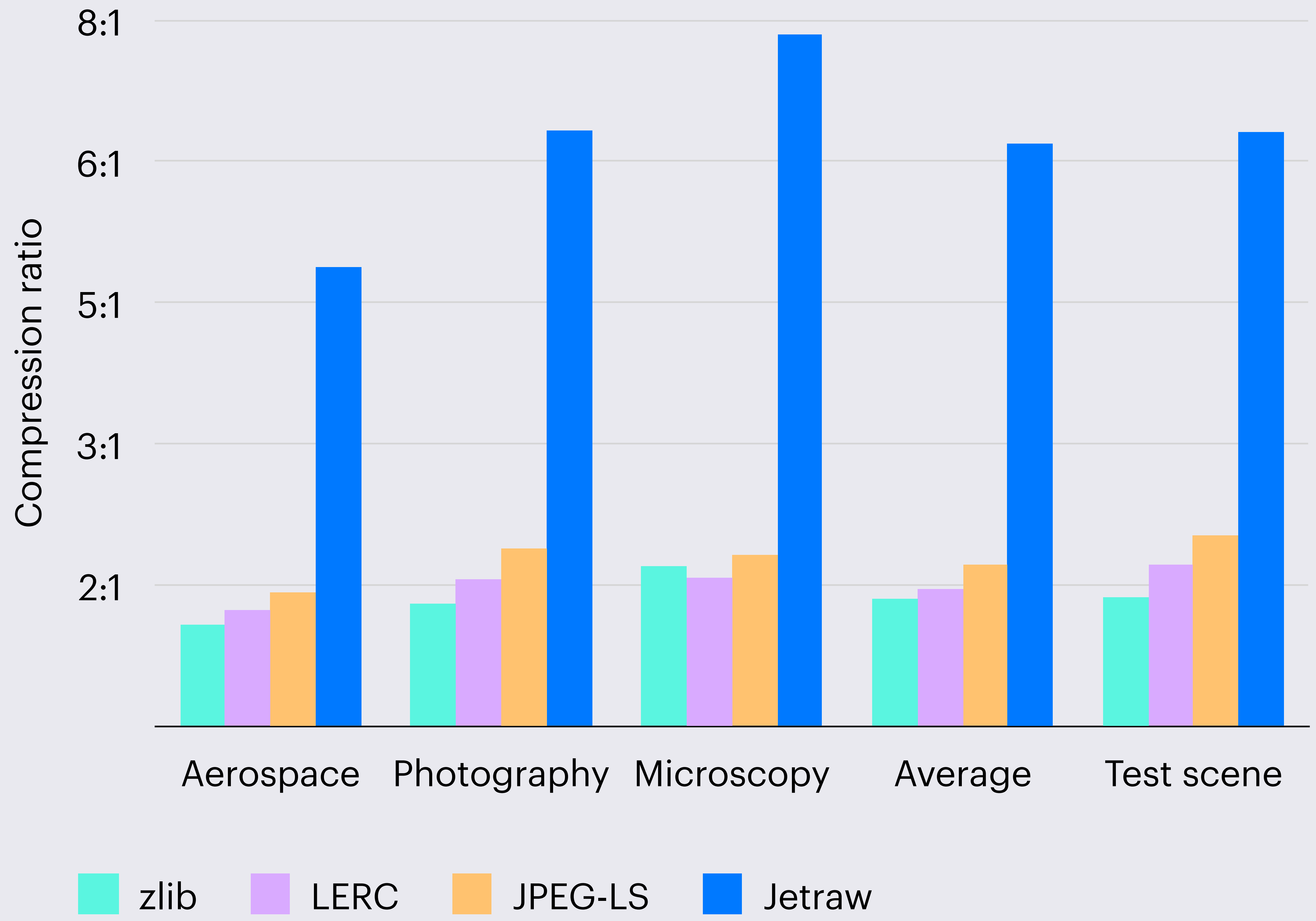
Gerhard Holst,

Head of Science & Research at PCO

pco.
An Excelitas Technologies Brand

Jetraw has the best in class performance in compression ratio and speed thanks to embedded sensor noise model

Average compression ratio by application



" We selected the Jetraw by Dotphoton due to the combination of the method's tight control on the maximum compression error, the compression ratio achieved and the algorithm speed"

Chris Dunsby,

Imperial College London

Imperial College London

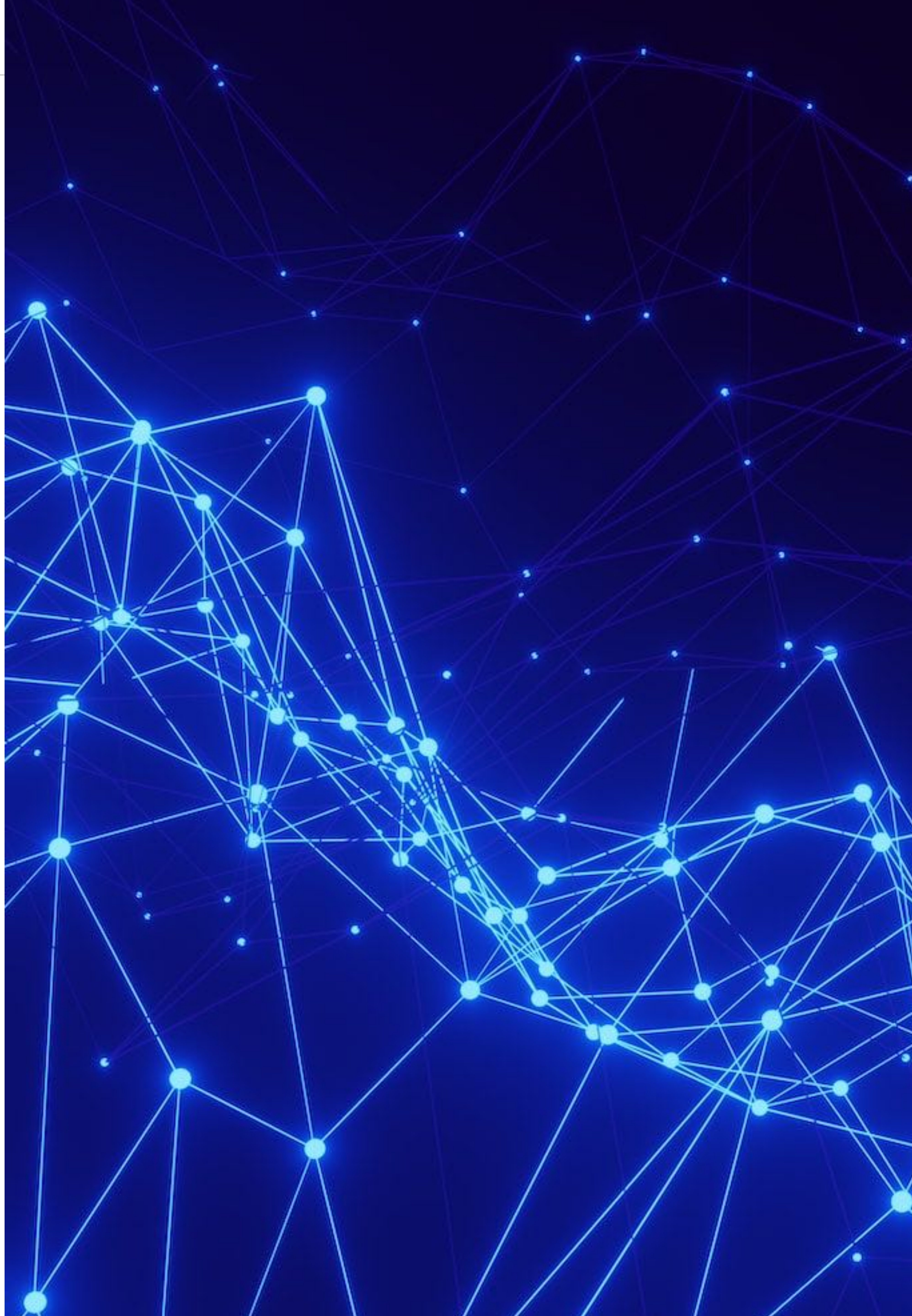
Reasons for the difference. Jetraw has been developed for raw data, seen by algorithms.

1. Other compression methods have been developed for the human eye, To maximise PSNR, to reduce mean-square difference between original and compressed file.

2. PSNR assumes that the original image is noise-free.

3. PSNR only depends on signal amplitude, not correlations.

4. For modern sensors and applications, Jetraw is more suitable.



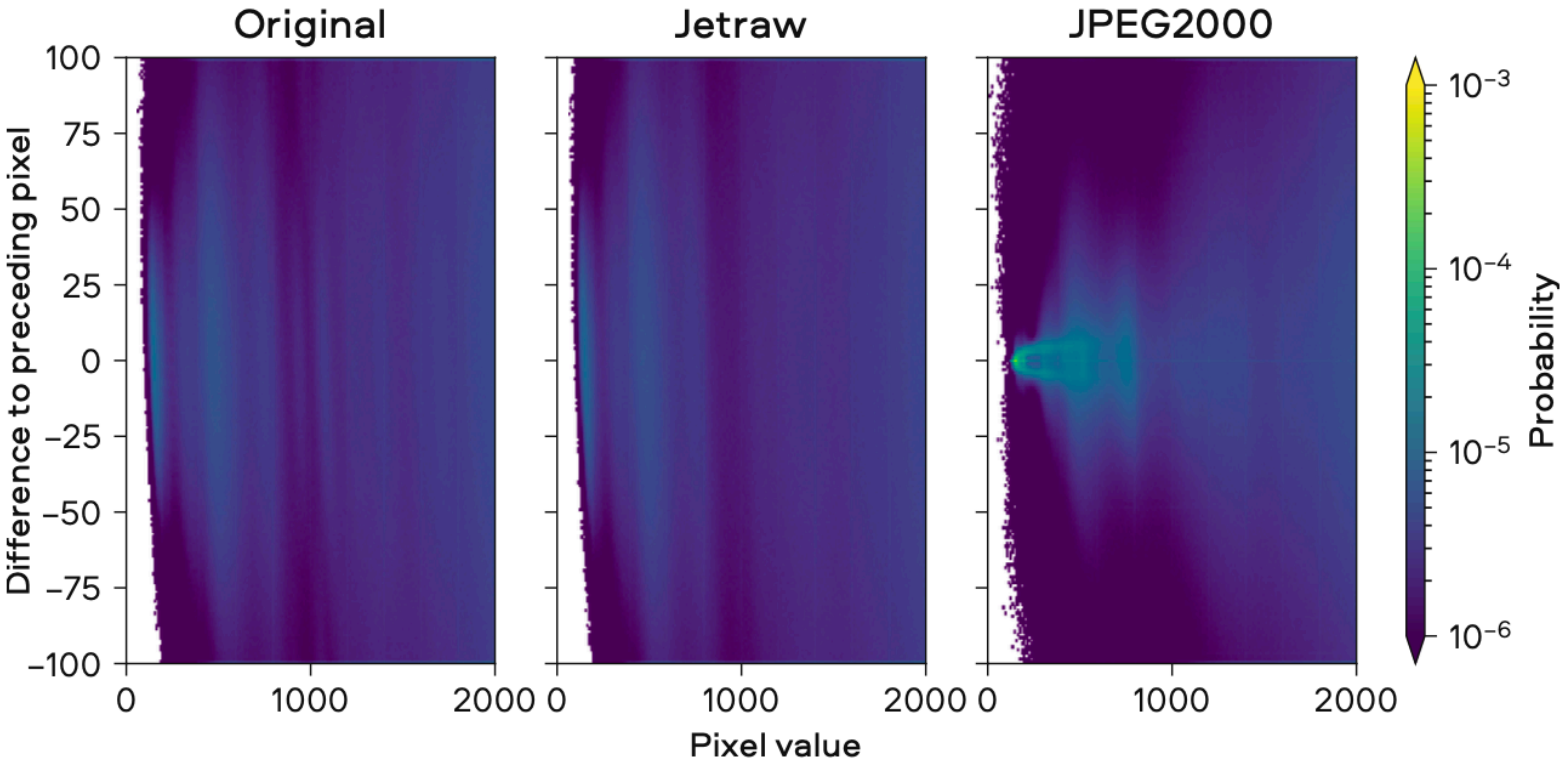
Noise plays an important role in modern images. Other compressors allocate more storage to noise than to information

Original 2: PSNR 52dB

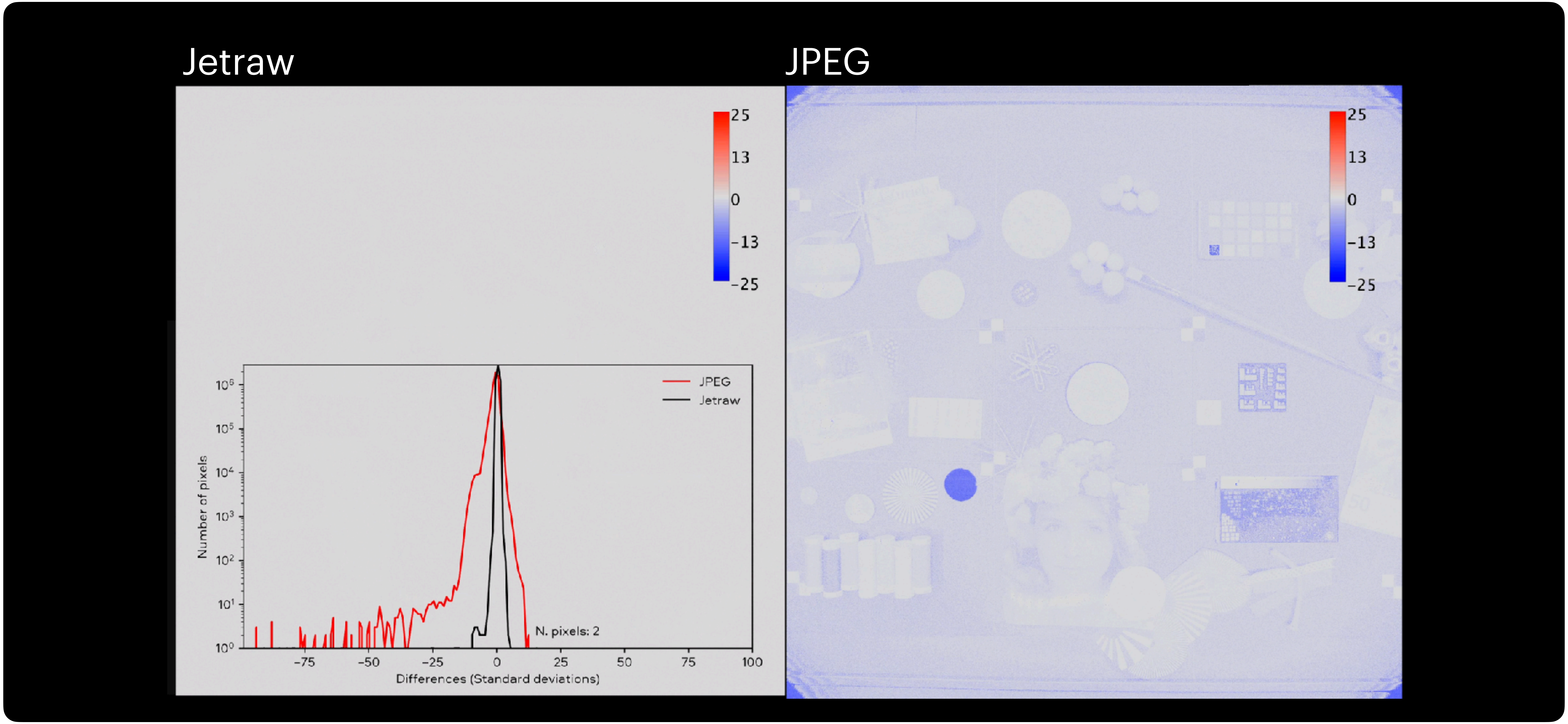
Original

JPEGXL: PSNR 53dB

Other compression methods destroy correlations



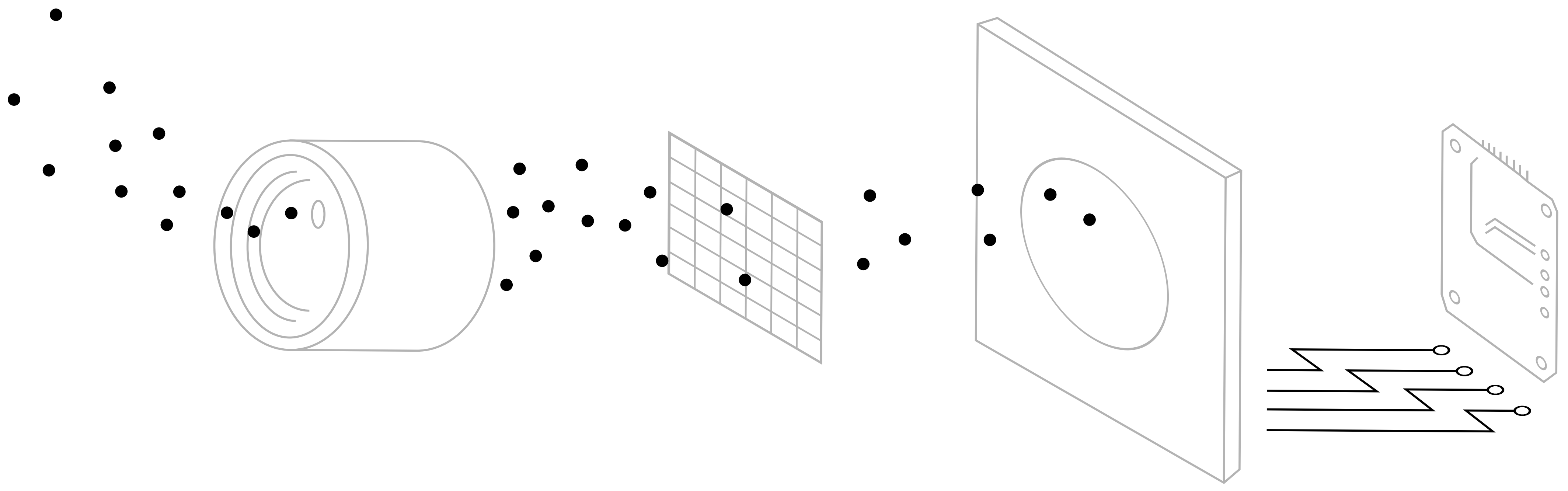
Other compression methods are biased



Raw is a well defined measurement

Digital image generation

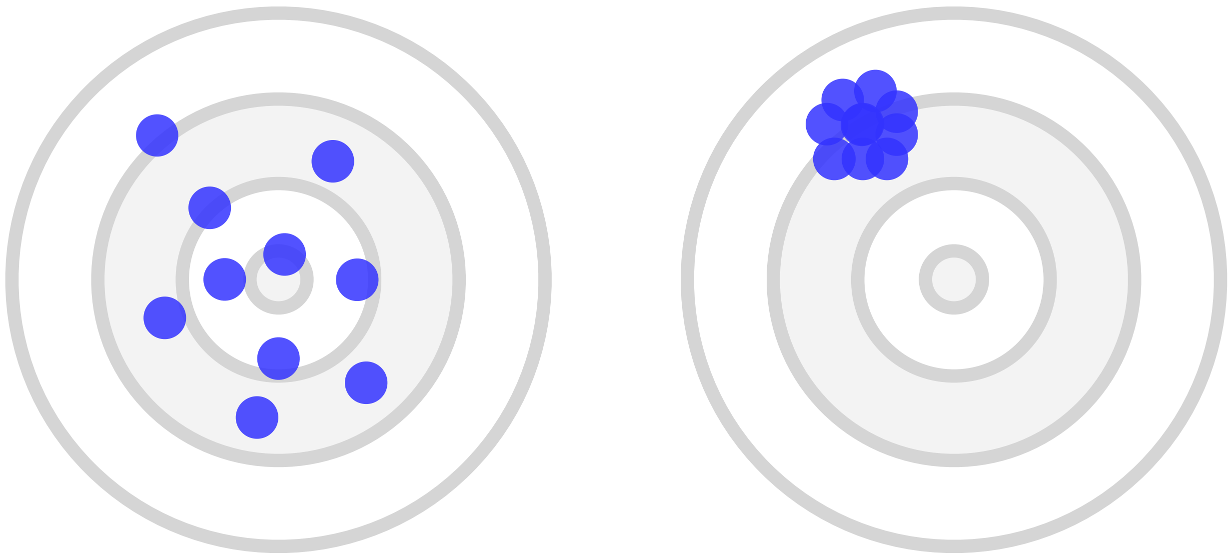
Requirement: Traceability of the image generation process (camera or sensor)



Errors are:

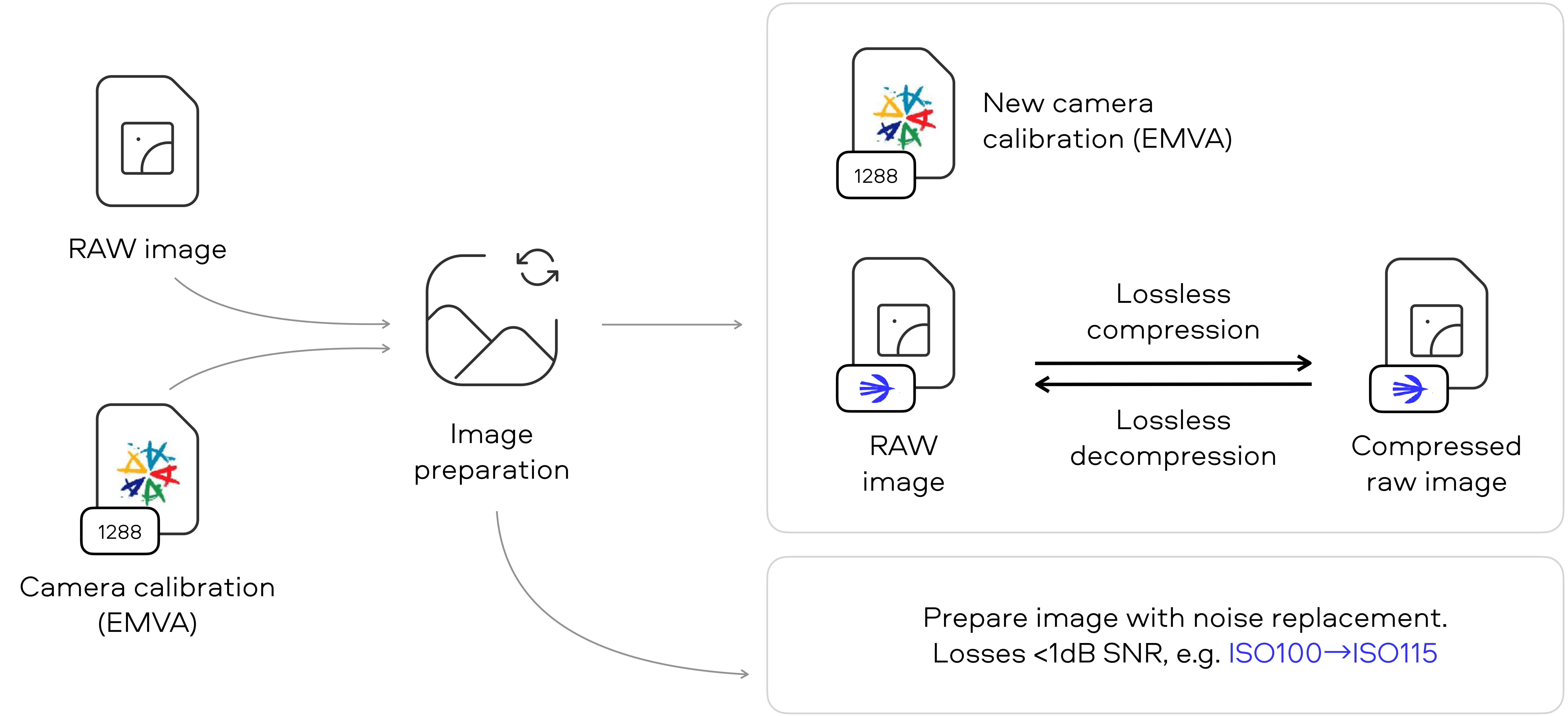
- ✔ Specified
- ✔ Independent
- ✔ Unbiased

Precision & accuracy



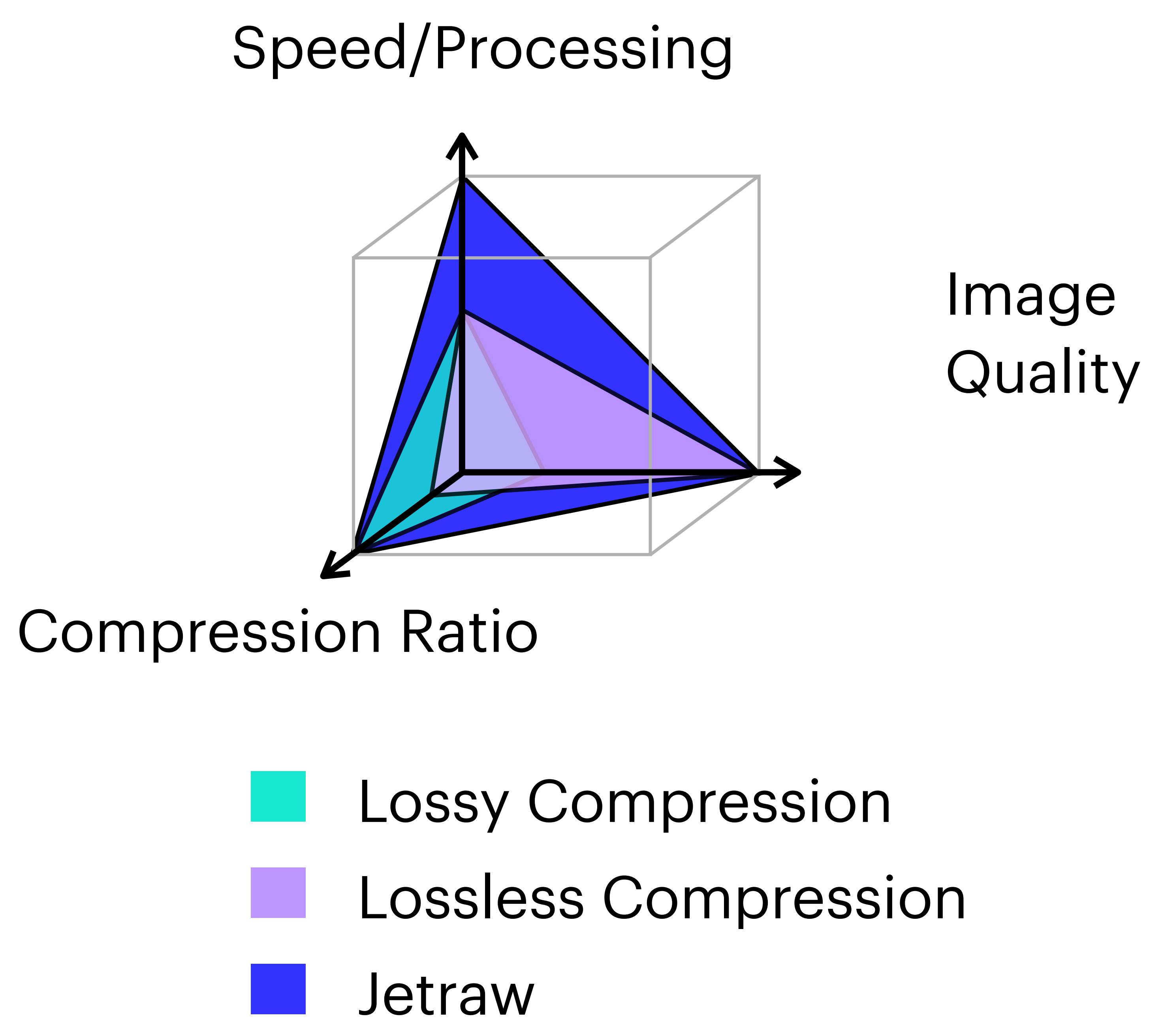
Raw data	Processed data
Low precision	High precision
High accuracy	Low accuracy

Steps to compression and decompression



The tradeoff

Disadvantages of Jetraw	Further advantages of Jetraw
✔ Only works for raw data	✔ Engineering with accurate tolerances → easier testing and failure detection
✔ The sensor and settings must be know at compression time	✔ Generate accurate synthetic data → test system “in silico”
✔ Several complexities in implementing the algorithm	✔ Use uncertainty data to improve algorithm performance → “better than lossless”.



Summary

Loss level	Tightly-controlled 1.2dB SNR (equiv. increase ISO100→ISO115)
Typical compression ratio	7:1 (always in the range 4:1 to 9:1)
Typical speed	FPGA → up to 100Gbps (6.25 Gpx/s) CPU (Intel/AMD/ARM) → 1.6Gbps/core (100Mpx/s/core)
Data formats	Any uncompressed format, typically TIFF or proprietary raw formats

Practical example 1: one 8Mpx camera, 30 fps

	Data rate	**Disk video capacity time (4TB ssd)**	**Upload time (8h of video, over 1Gbps)**
Without Jetraw	480 MB/s 1.7 TB/h	2.3 hours	16h
With Jetraw	68 MB/s 0.25 TB/h	16 hours	3h

Deployment possibilities

	Advantages	Development status
FPGA (on instrument)	Improve performance from the start	Ready
FPGA (datacenter)	High throughput with low CPU and network load	In development
Software library	Easy deployment, fast enough	Ready
Transparent compression proxy	Easy large scale deployments	Beta

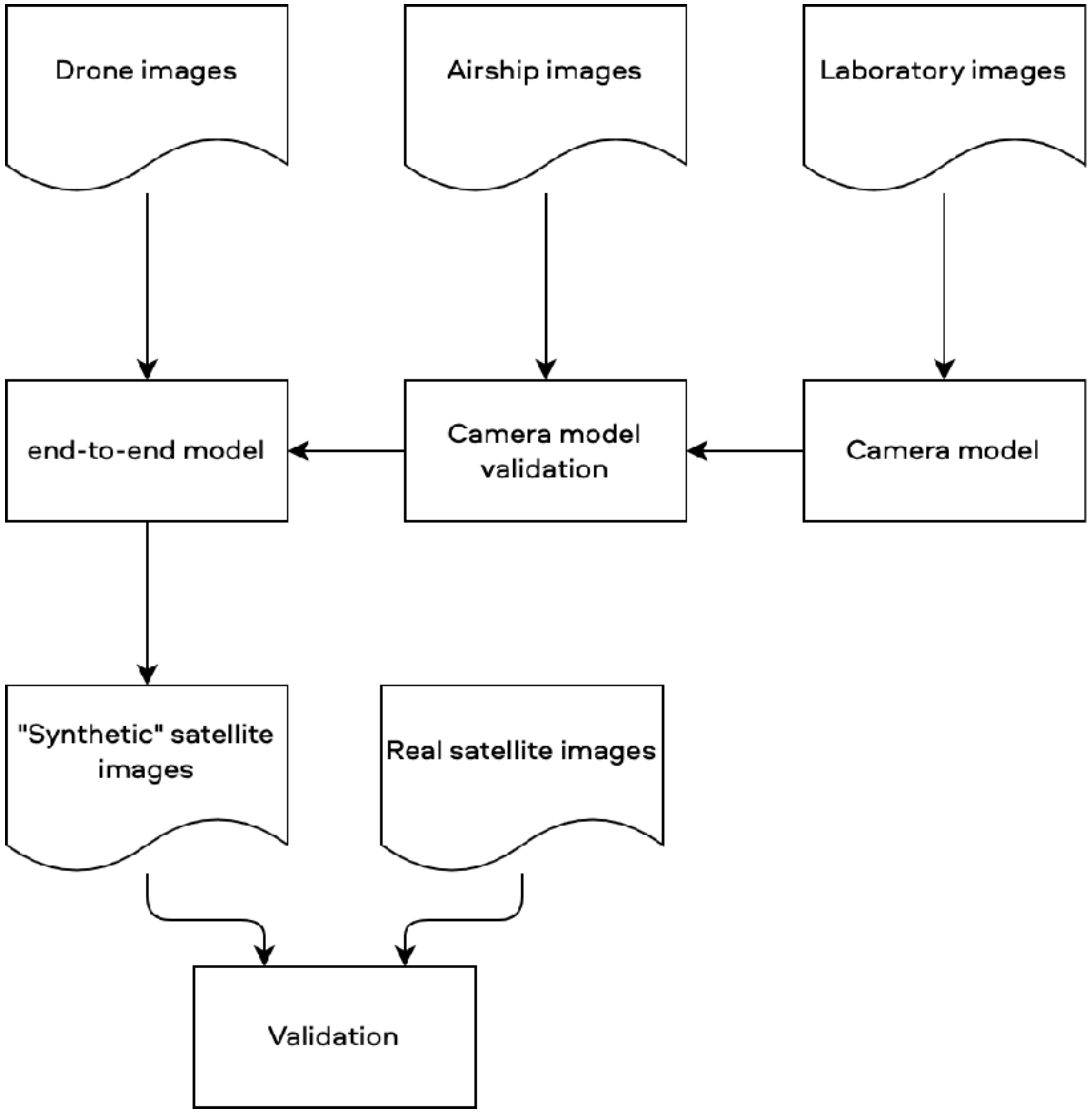
FPGA performance and resource usage

Resource usage (for an image 5000x5000 pixels, 16bpp) Vivado 2020.2, ZCU102, 200MHz.						
Pixels/ clock	LUT	FF	BRAM	DSP	Mpx/s	Power (W)
1	3968	1874	7.5	16	200	0.149
2	5608	3063	9	32	400	0.225
4	9612	5621	14	64	800	0.448
8	17644	10718	24	128	1600	0.935
16	33944	20814	39.5	256	3200	1.528
32	70790	41828	70.5	512	6400	3.255

Validation tools

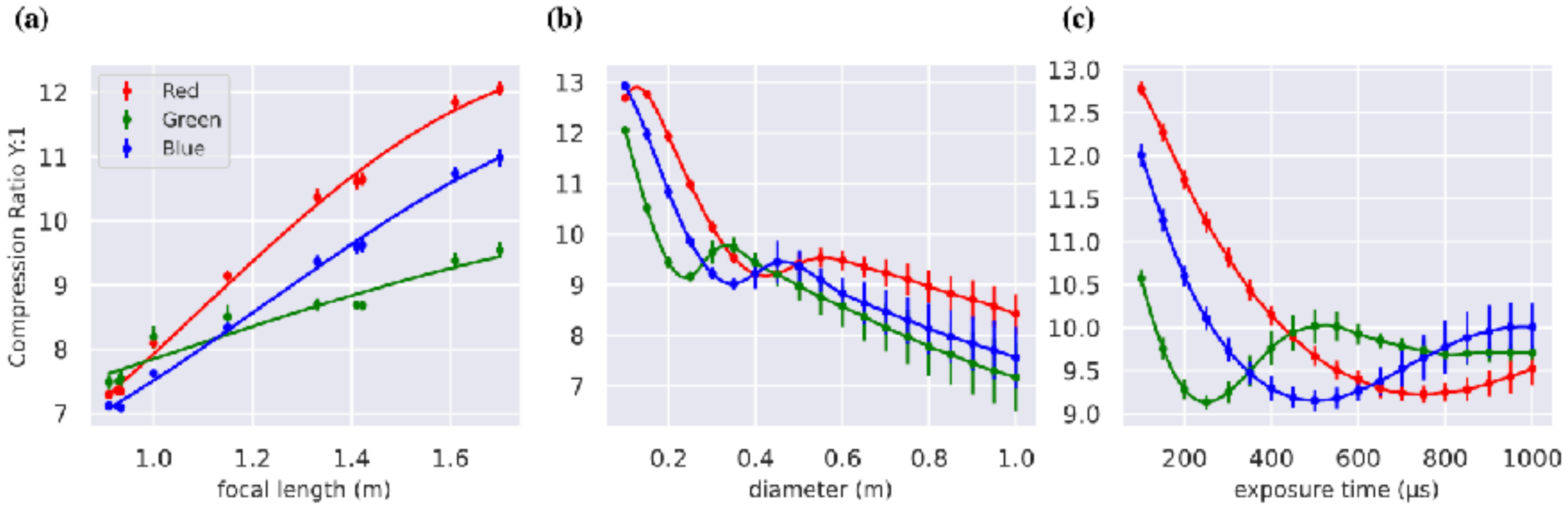
Testing, verification and validation tools

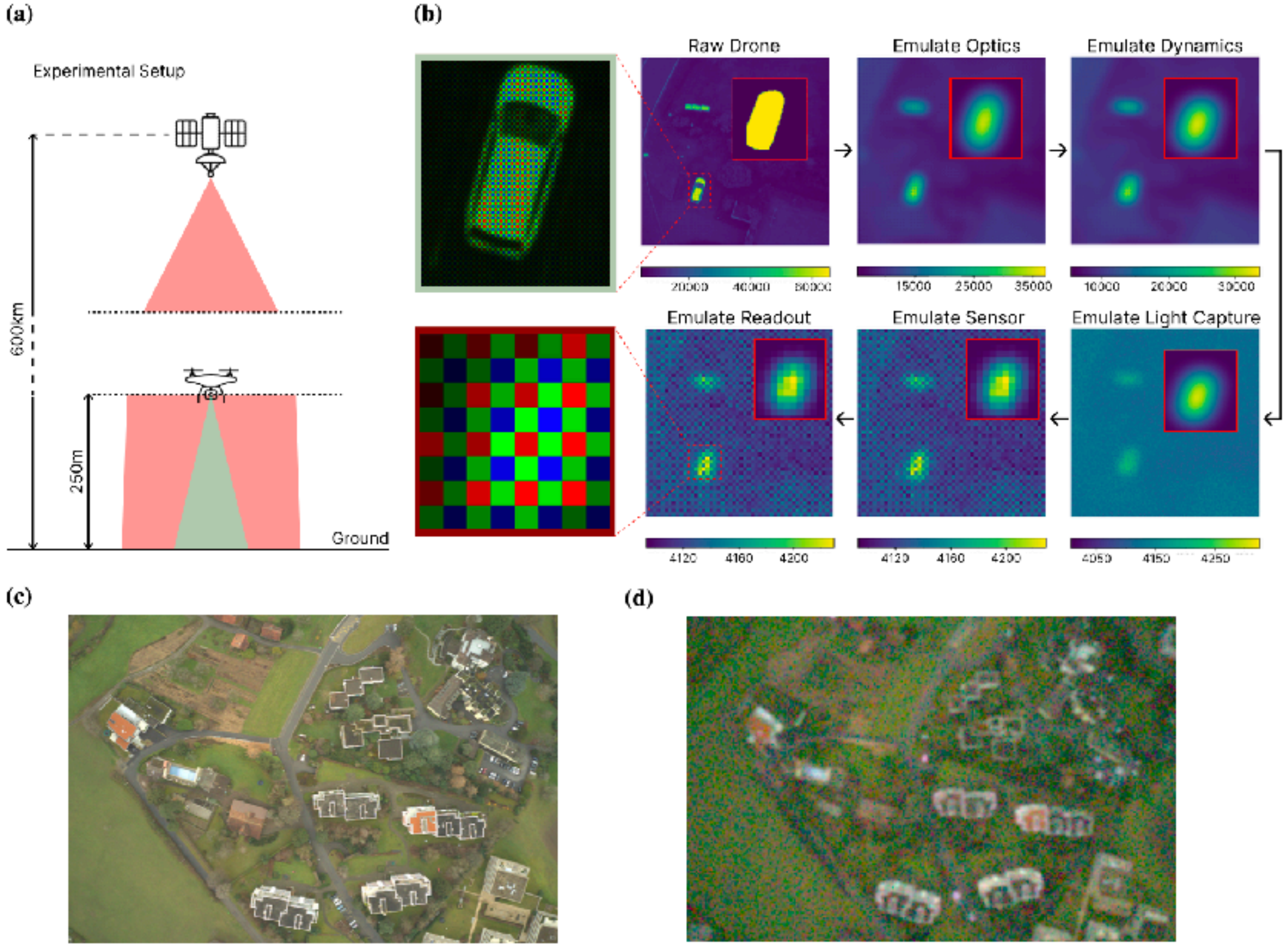
End-to-end model overview

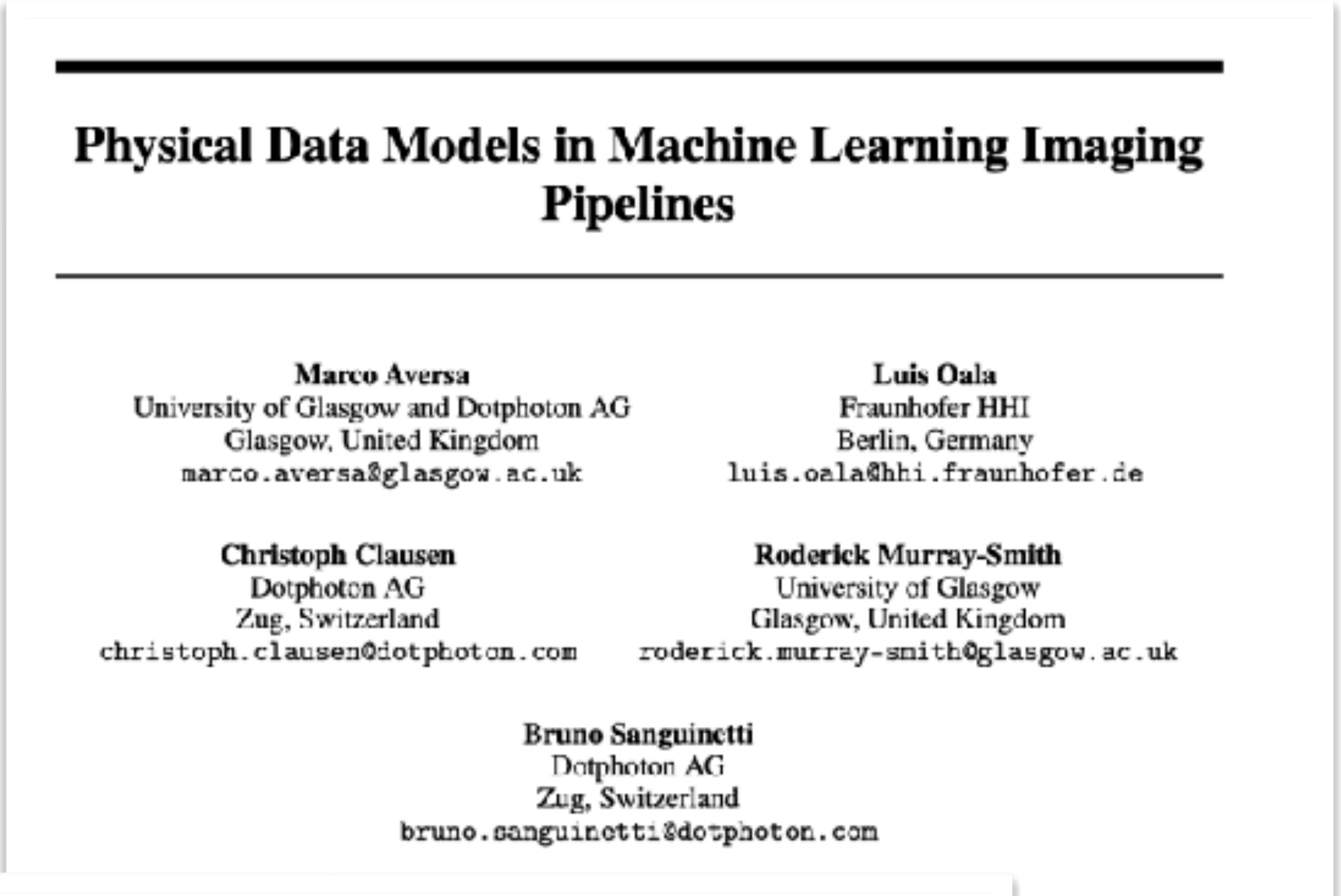


End-to-end model results

- Inspired from "Pyxel", using the "Poppy" physical optics propagation code
- Tolerances discussed with MediaLario







Physical Data Models in Machine Learning Imaging Pipelines

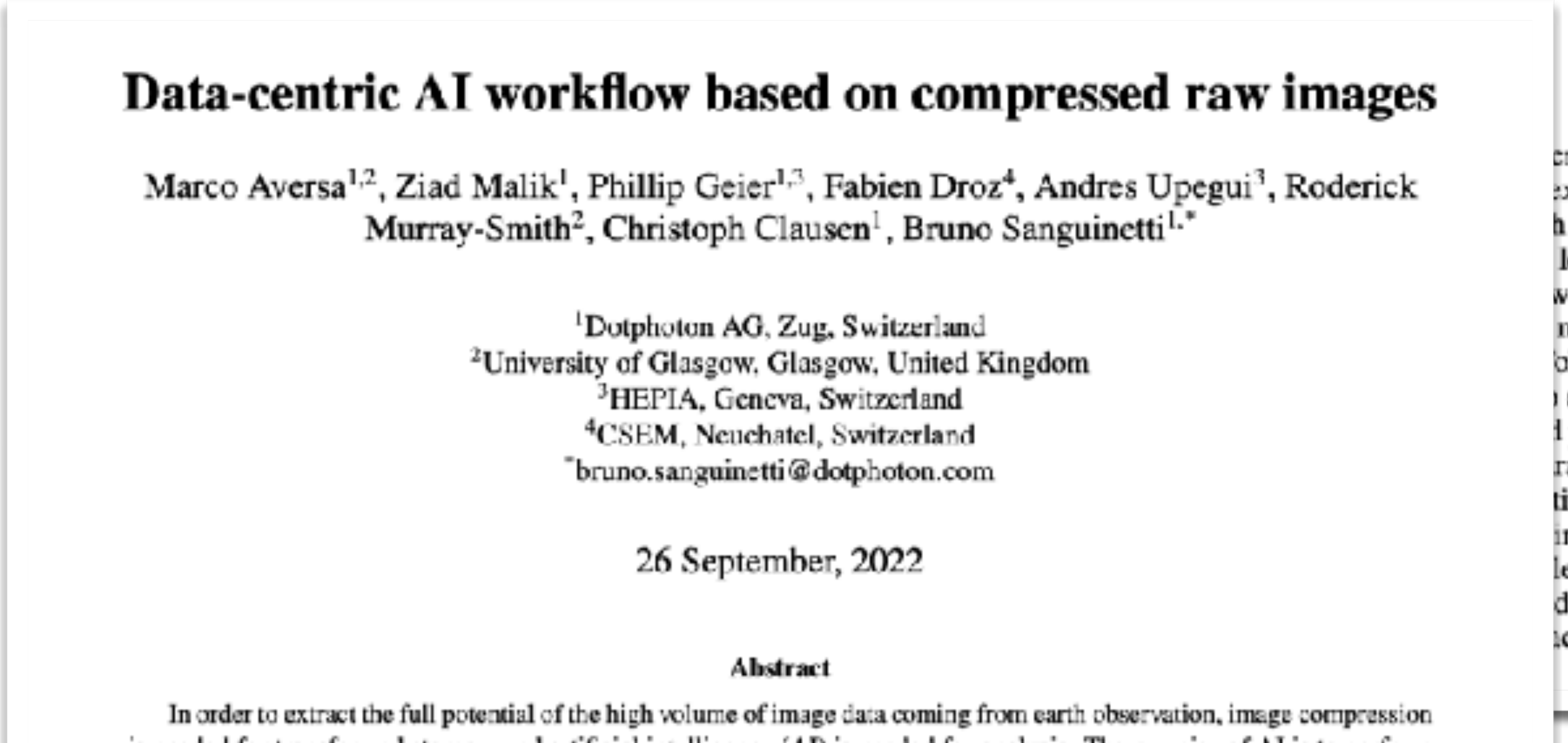
Marco Aversa
University of Glasgow and Dotphoton AG
Glasgow, United Kingdom
marco.aversa@glasgow.ac.uk

Luis Oala
Fraunhofer HHI
Berlin, Germany
luis.oala@hhi.fraunhofer.de

Christoph Clausen
Dotphoton AG
Zug, Switzerland
christoph.clausen@dotphoton.com

Roderick Murray-Smith
University of Glasgow
Glasgow, United Kingdom
roderick.murray-smith@glasgow.ac.uk

Bruno Sanguinetti
Dotphoton AG
Zug, Switzerland
bruno.sanguinetti@dotphoton.com



Data-centric AI workflow based on compressed raw images

Marco Aversa[1,2], Ziad Malik[1], Phillip Geier[1,3], Fabien Droz[4], Andres Upegui[3], Roderick Murray-Smith[2], Christoph Clausen[1], Bruno Sanguinetti[1,*]

[1]Dotphoton AG, Zug, Switzerland
[2]University of Glasgow, Glasgow, United Kingdom
[3]HEPIA, Geneva, Switzerland
[4]CSEM, Neuchatel, Switzerland
[*]bruno.sanguinetti@dotphoton.com

26 September, 2022

Abstract

In order to extract the full potential of the high volume of image data coming from earth observation, image compression

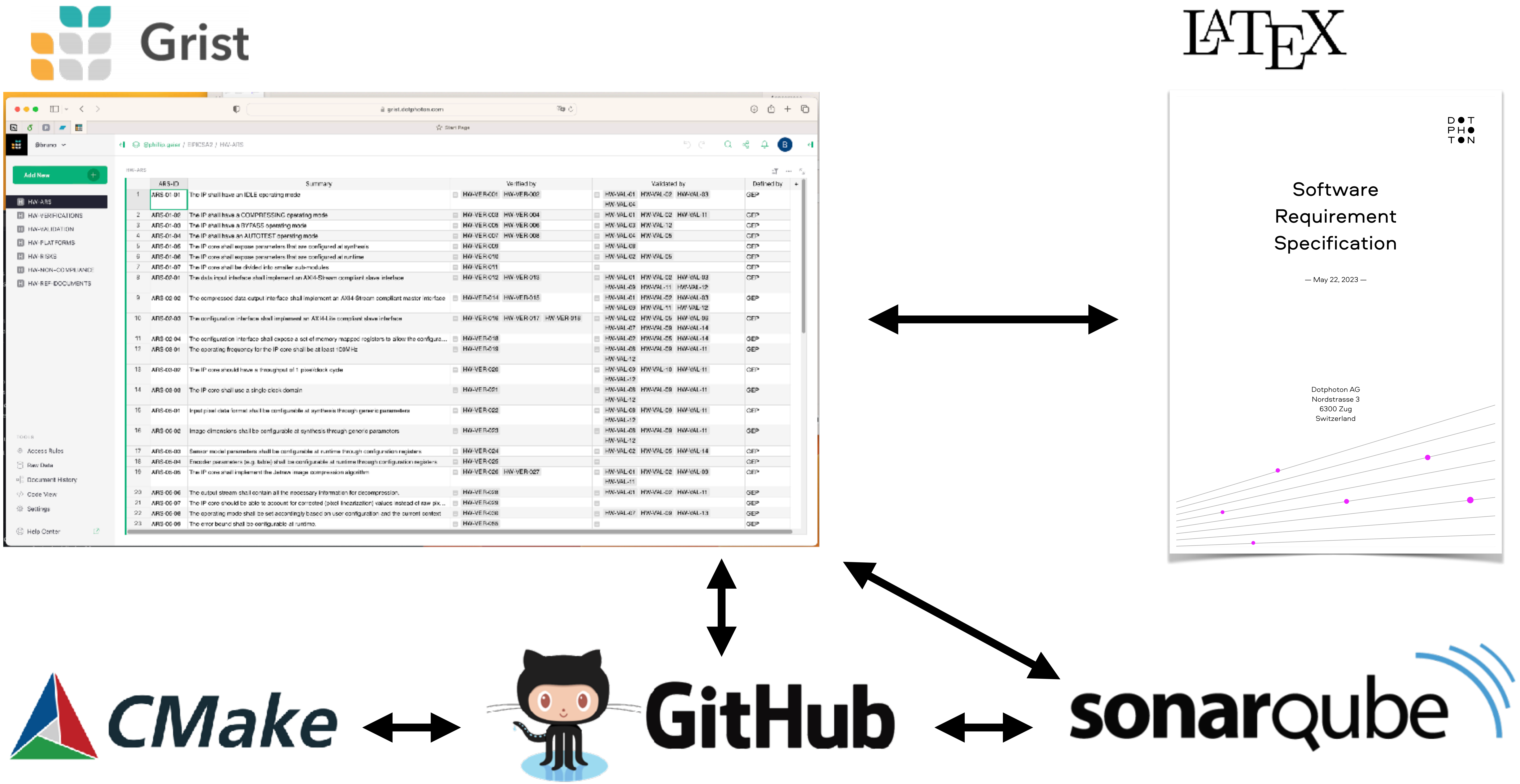

THUNDER & COLT
EXPERIMENTAL
E-AIRSHIP

FPGA validation

- Questasim
- “Design-tester” automated AXI4-stream design tool in python
- vunit
- Test platform 1, based on KU105, images sent from computer (quality testing)
- Test platform 2, based on KU105, artificial test patterns generated on FPGA (perf testing)
- Demonstrator, based on KD260 (“Krya Vision”)

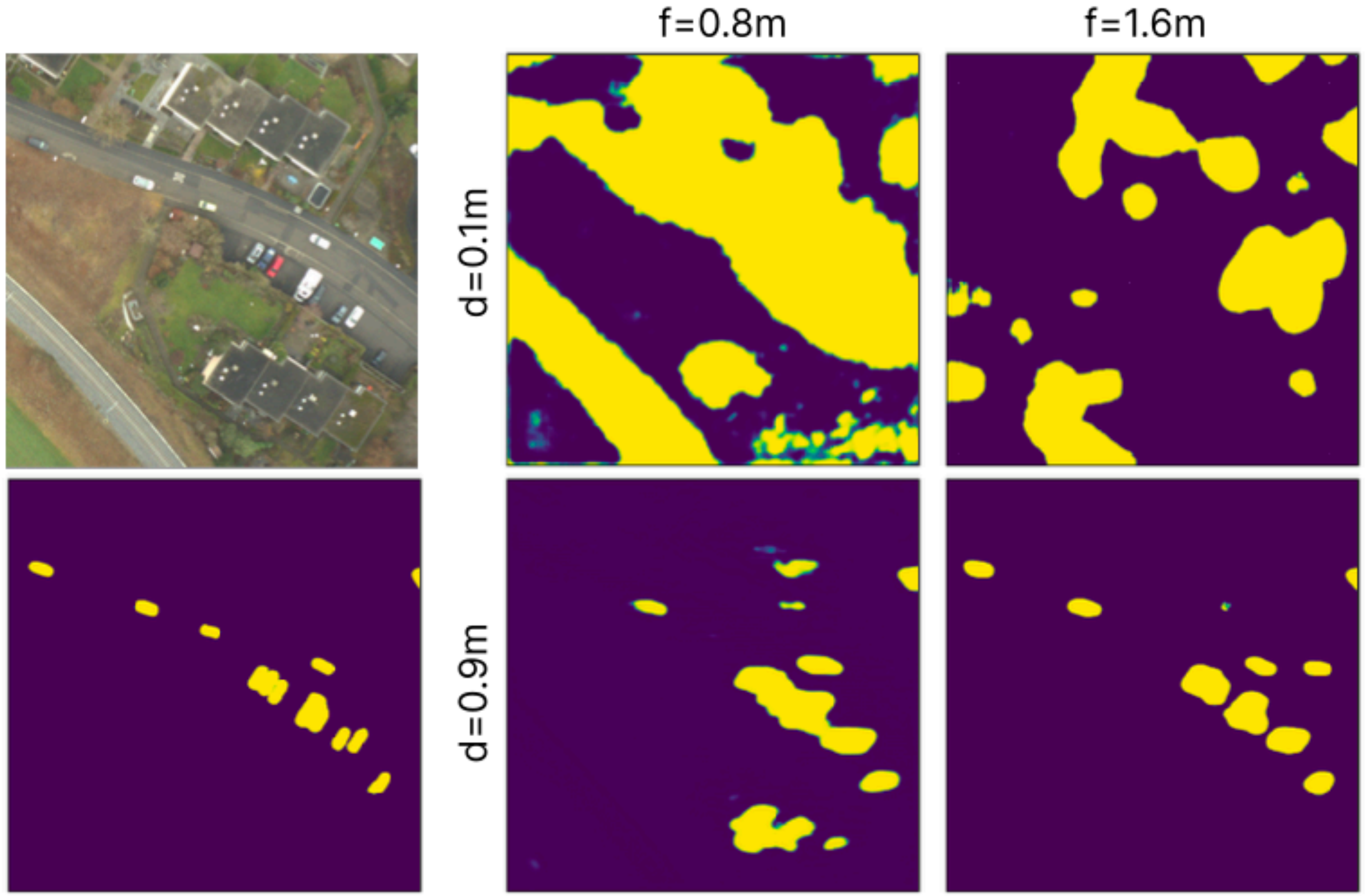
New development process and tools

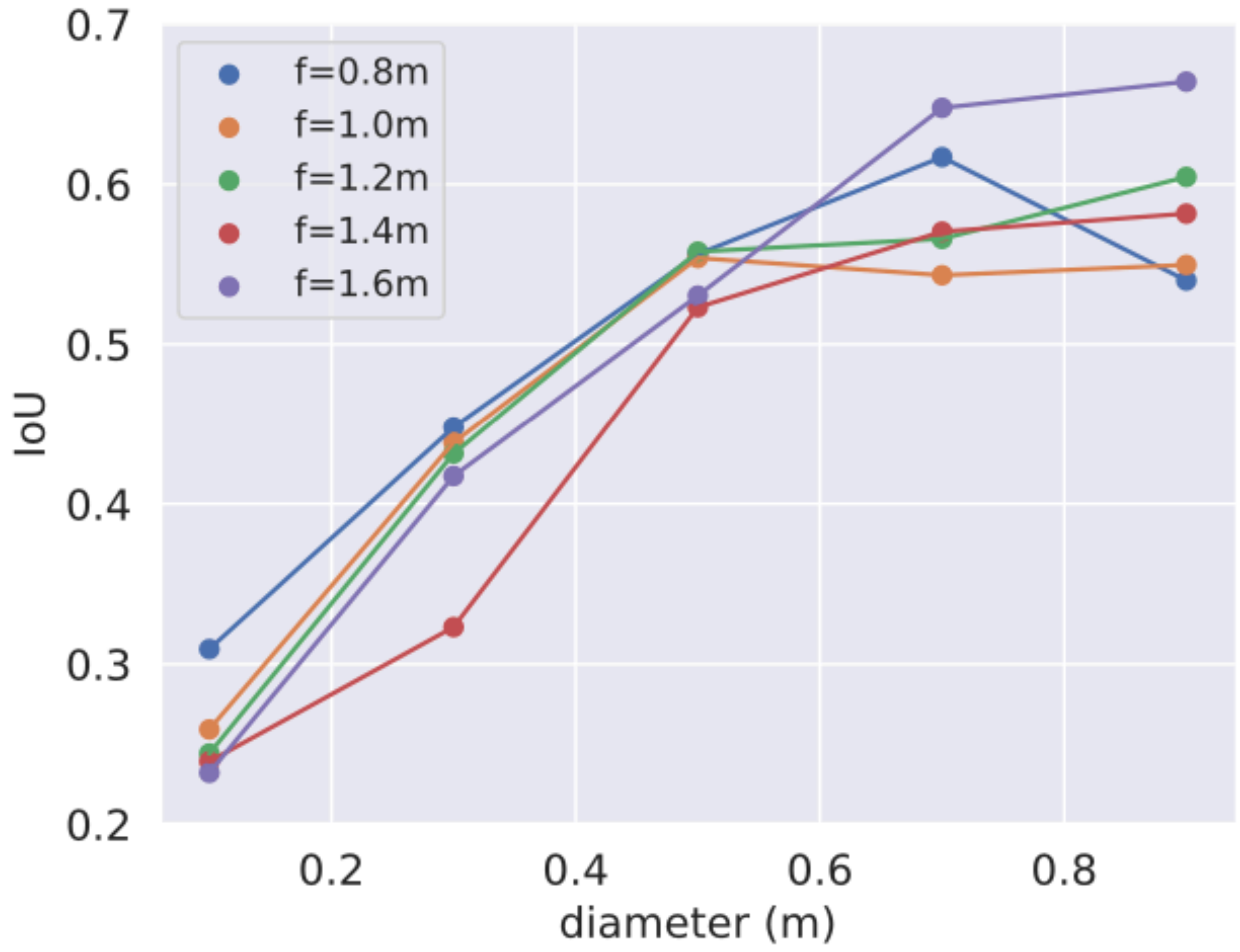
- “Notion” did finally not work out, difficult to do configuration management.
- New development process based on integrating the following:
  - Github (version tracking)
  - cmake (configuration management)
  - Sonarqube (CI/CD, static code analysis)
  - Grist (Requirements management, traceability and project management)
  - Latex (Documentation)


Grist
LaTeX
Software Requirement Specification
— May 22, 2023 —
Dotphoton AG
Nordstrasse 3
6300 Zug
Switzerland
CMake
GitHub
sonarqube

Demonstrate improved performance of ML algorithms with RAW data

OBPDC 2022, NeurIPS 2022





OBPDC 2022, NeurIPS 2022

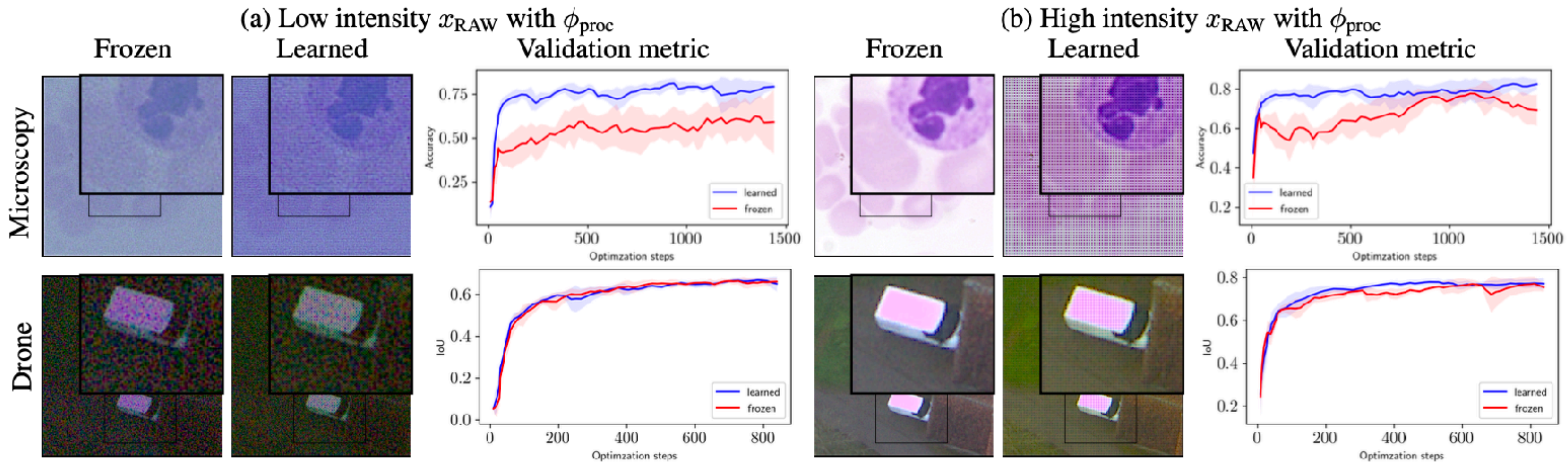


Figure 3: Low (a) and high (b) intensity images processed by a *frozen* and a *learned* pipeline. This dataset drift adjustment would not be possible with processed data typically used for machine learning experiments. The plots in the rightmost column of each block display the mean of validation metrics over five cross-validation runs. Error bars are reported as one standard deviation. Optimization steps 1439 and 915 correspond to epoch 60 into training.

Demonstrator: Live!