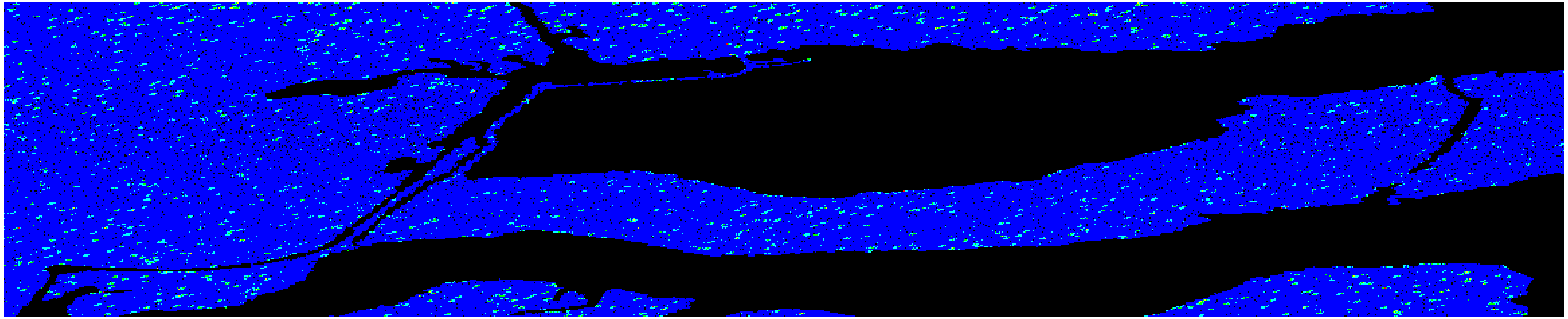
ALUs coregistration error

- Error persists, even if ALUs and SNAP use the same tile size

Legend	Relative error
RED	one pixel zero, second is not
ORANGE	>10%
YELLOW	> 1%
GREEN	> 0.1%
CYAN	> 100ppm
LIGHT BLUE	> 10ppm
BLUE	< 10ppm
BLACK	equal

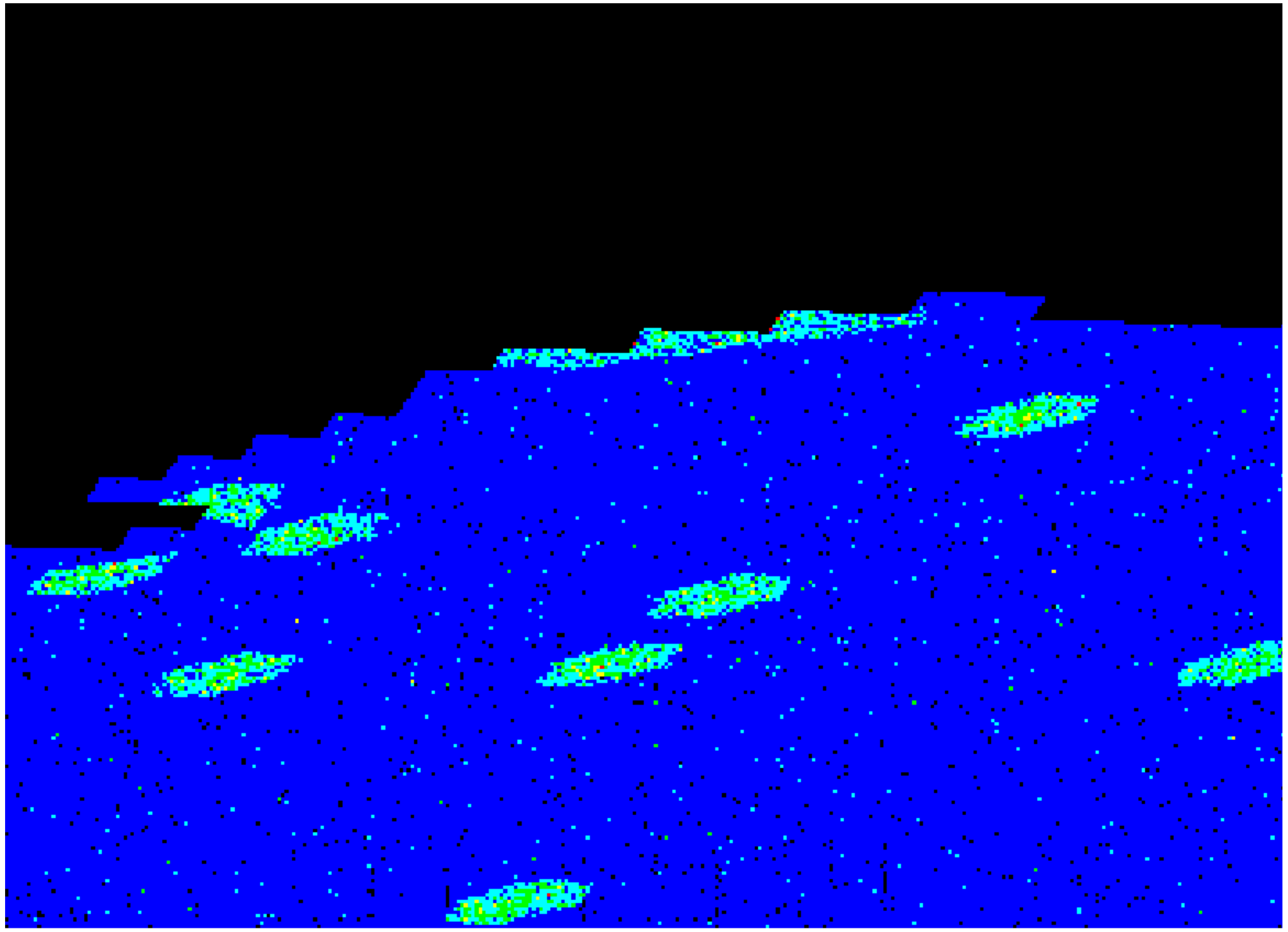


ALUs coregistration error

Two causes:

- Orbit State Vector interpolation
- FMA

Legend	Relative error
RED	one pixel zero, second is not
ORANGE	>10%
YELLOW	> 1%
GREEN	> 0.1%
CYAN	> 100ppm
LIGHT BLUE	> 10ppm
BLUE	< 10ppm
BLACK	equal



Orbit State Vector interpolation

- Snap uses JAMA – Java Matrix Package
- ALUs uses Eigen
- Method ported from snap - Vandermonde matrix, QR decomposition(placeholder reflections), however libraries produce slightly different results.

	alus	snap
X position	4353250.094539642333984	4353250.094528198242188
Y position	992288.187537193298340	992288.187539100646973
Z position	5479862.164007186889648	5479862.164011001586914
X velocity	5984.082505045458674	5984.082505041733384
Y velocity	-705.878539511002600	-705.878539509139955
Z velocity	-4614.602186229079962	-4614.602186214178801

FMA

- Fused multiply-add. A = A + B * C. This operation can be performed in single machine instruction or as two separate, multiplication and addition.
- A common operation in many math problems, e.g interpolation
- GPU/C++ more liberal in its use
- Java/SNAP does not use the CPU equivalent(Math.fma)
- FMA is more accurate(1 vs 2 roundings), but not using it would mean a bigger "error" compared to snap