AN FPGA-BASED IMPLEMENTATION OF A HYPERSPECTRAL ANOMALY DETECTION ALGORITHM FOR REAL-TIME APPLICATIONS

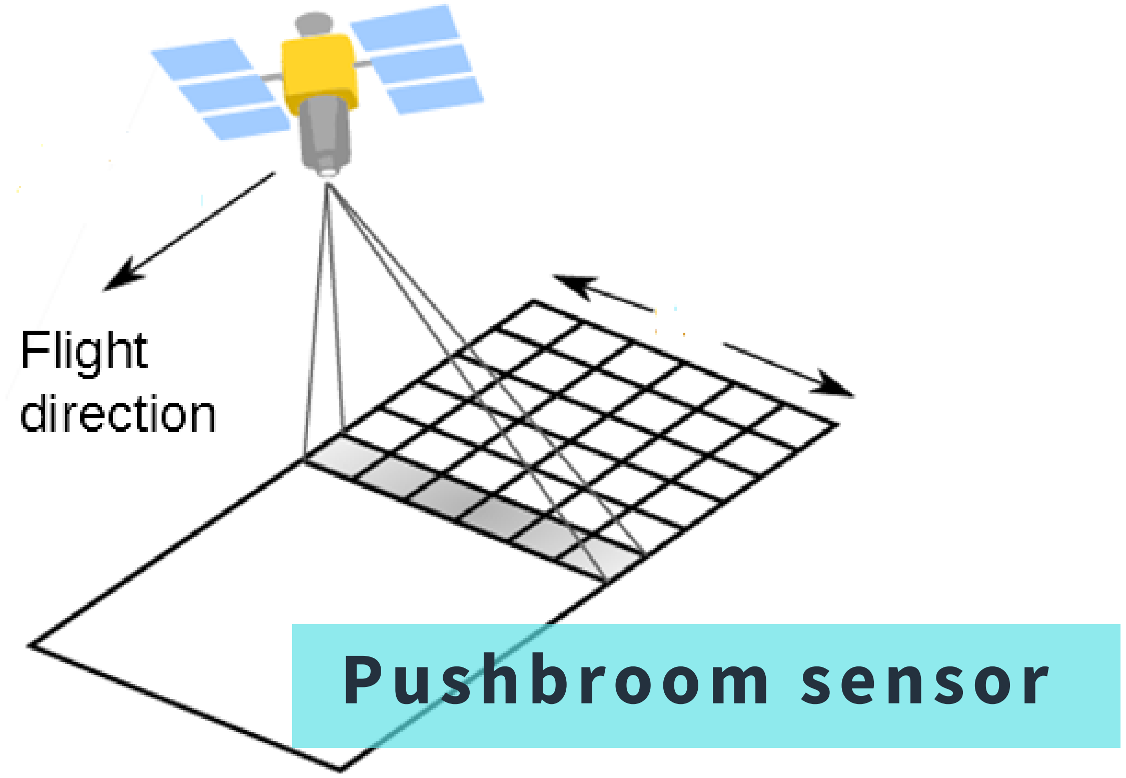
Authors:

María Díaz Martín, Julián Caba, Raúl Guerra Hernández, Jesús Barba, Sebastián López Suárez

International Geoscience and Remote Sensing Symposium (iGARSS) July, 2021

MOTIVATIONS

Sensor	Pixels per frame	Bands per frame	Frames per second	Data resolution
SPECIM FX-10	1024	240/160	330	12 bits/ 16 bits packaged

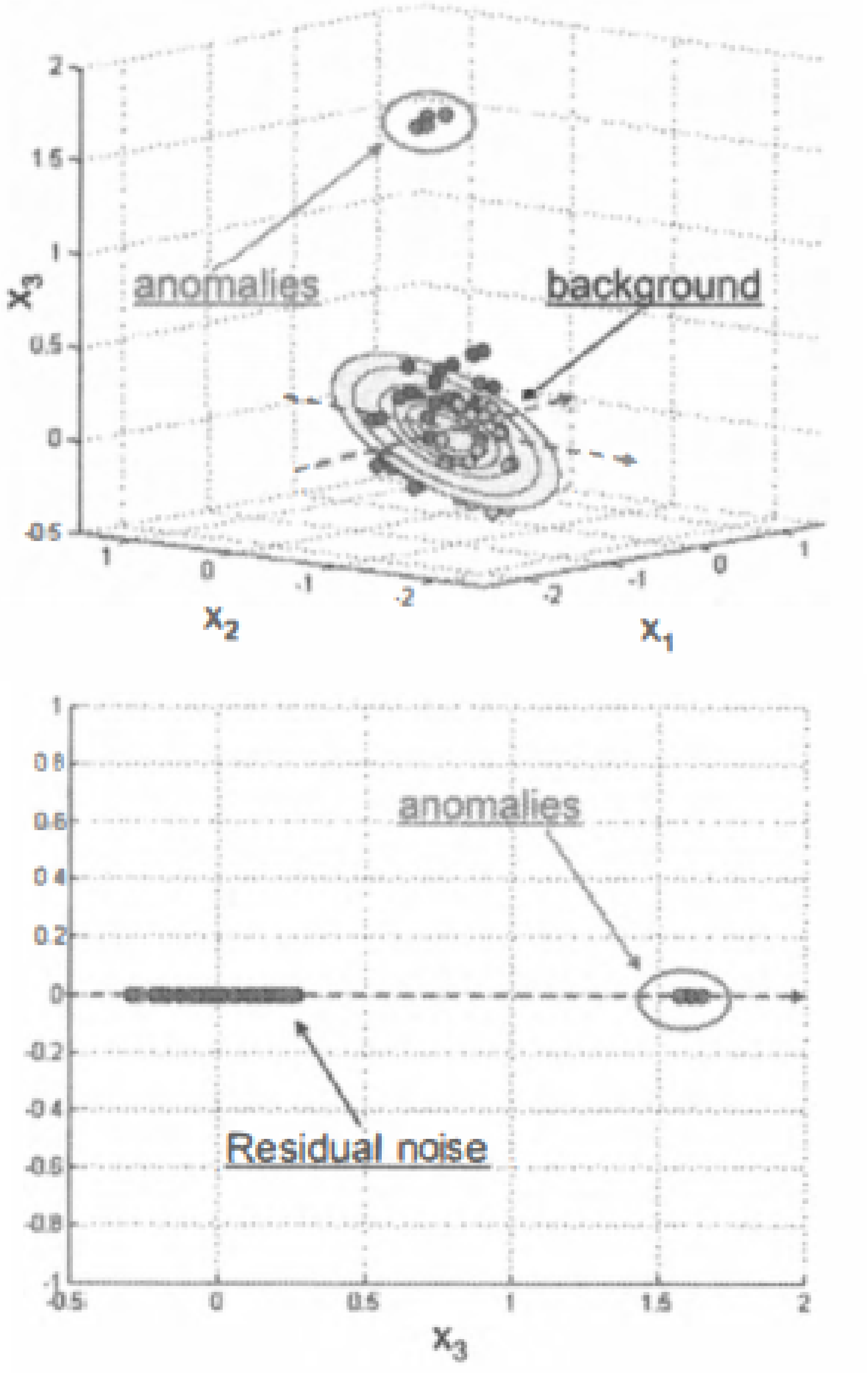


- More than 9 GB/min to be processed
- Line-by-line performance
- Short battery life
- Low-power, low-cost, low-weight and compact size onboard PC

ZedBoard
(Xilinx Zynq-7020 SoC)

SUBSPACE ANOMALY DETECTION

ULPGC Universidad de Las Palmas de Gran Canaria — UCLM UNIVERSIDAD DE CASTILLA-LA MANCHA



Images taken from Matteoli S. et al.DOI:10.1109/MAES.2010.5546306.

$$\mathbf{r_j} = \sum_{n=1}^{p} \mathbf{e_n} \cdot a_{j,n} + \mathbf{n_j}$$

$\mathbf{b_n}$ = background spectra

$\mathbf{s}$ = desired target (??)

$$\mathbf{r_j} = \sum_{n=1}^{p} \mathbf{b_n} \cdot a_{j,n} + \mathbf{s} \cdot a_{sj} + \mathbf{n_j}$$

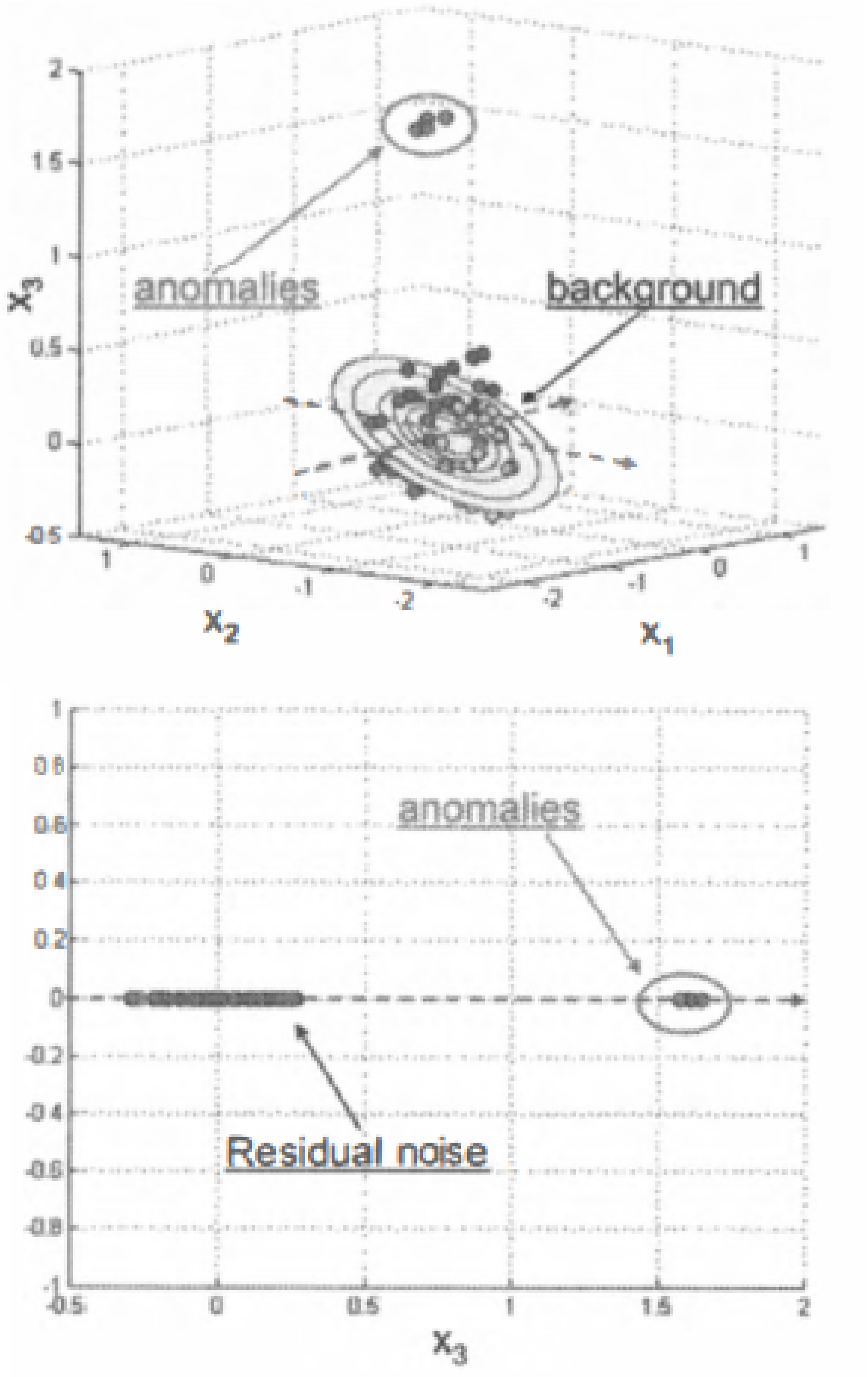
Orthogonal subspace projection (OSP)

$$\mathbf{d} = (\mathbf{r_j} - \hat{\mu}_b)' \cdot \mathbf{P} \cdot (\mathbf{r_j} - \hat{\mu}_b)$$

SUBSPACE ANOMALY DETECTION

ULPGC Universidad de Las Palmas de Gran Canaria

UCLM UNIVERSIDAD DE CASTILLA-LA MANCHA



Images taken from Matteoli S. et al.DOI:10.1109/MAES.2010.5546306.

$$\mathbf{r}_j = \sum_{n=1}^{p} \mathbf{e}_n \cdot a_{j,n} + \mathbf{n}_j$$

$\mathbf{b}_n$ = background spectra

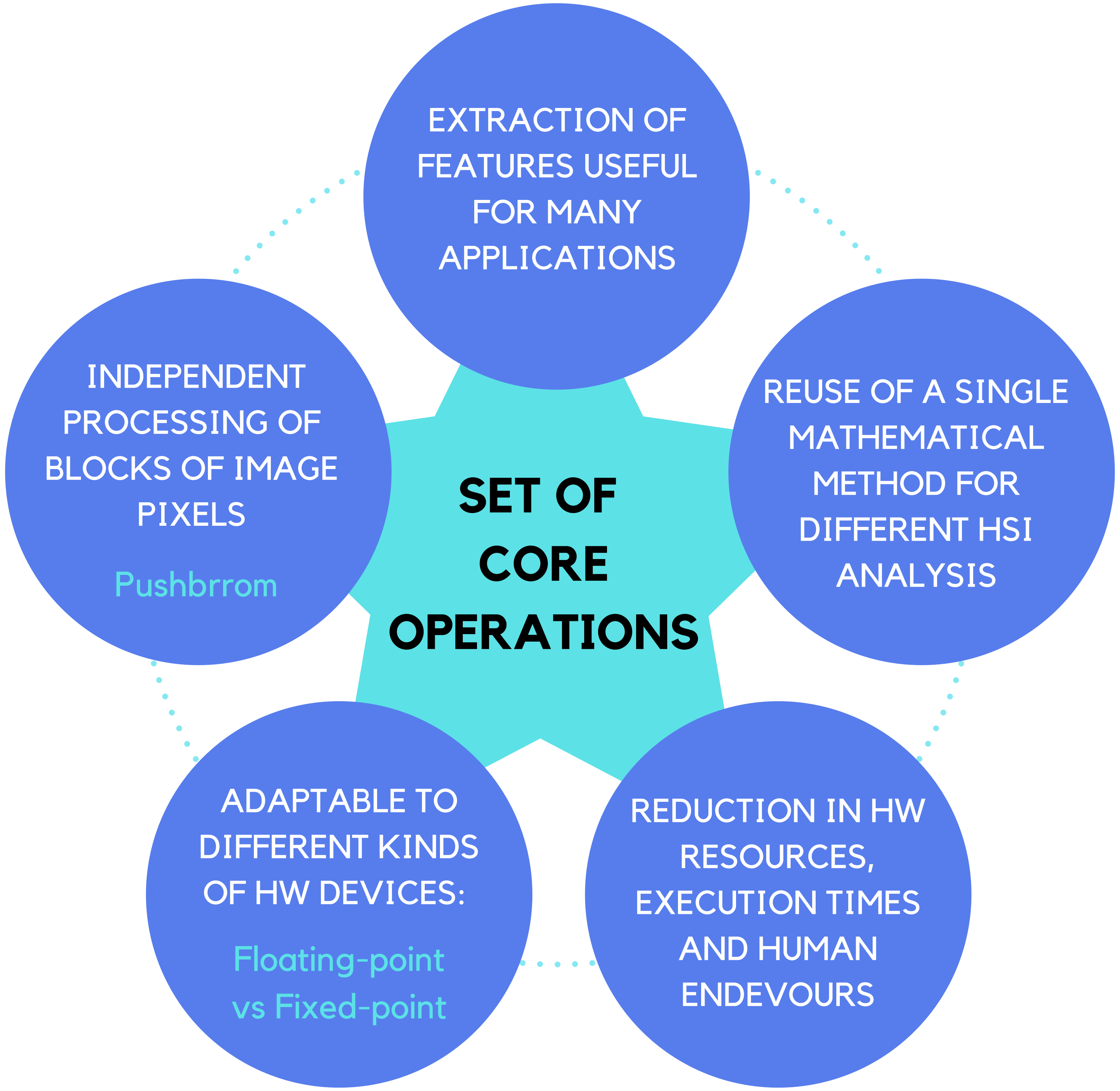
$\mathbf{s}$ = desired target (??)

$$\mathbf{r}_j = \sum_{n=1}^{p} \mathbf{b}_n \, a_{j,n} + \mathbf{s} \cdot a_{sj} + \mathbf{n}_j$$

Orthogonal subspace projection (OSP)

$$\mathbf{d} = (\mathbf{r}_j - \hat{\mu}_b)^T \cdot \mathbf{P} \cdot (\mathbf{r}_j - \hat{\mu}_b)$$

SET OF CORE OPERATIONS



01 ANOMALY DETECTION ✓

02 TARGET DETECTION

03 UNMIXING

04 CLASSIFICATION

05 LOSSY COMPRESSION

06 BAND SELECTION

SET OF CORE OPERATIONS

Inputs:

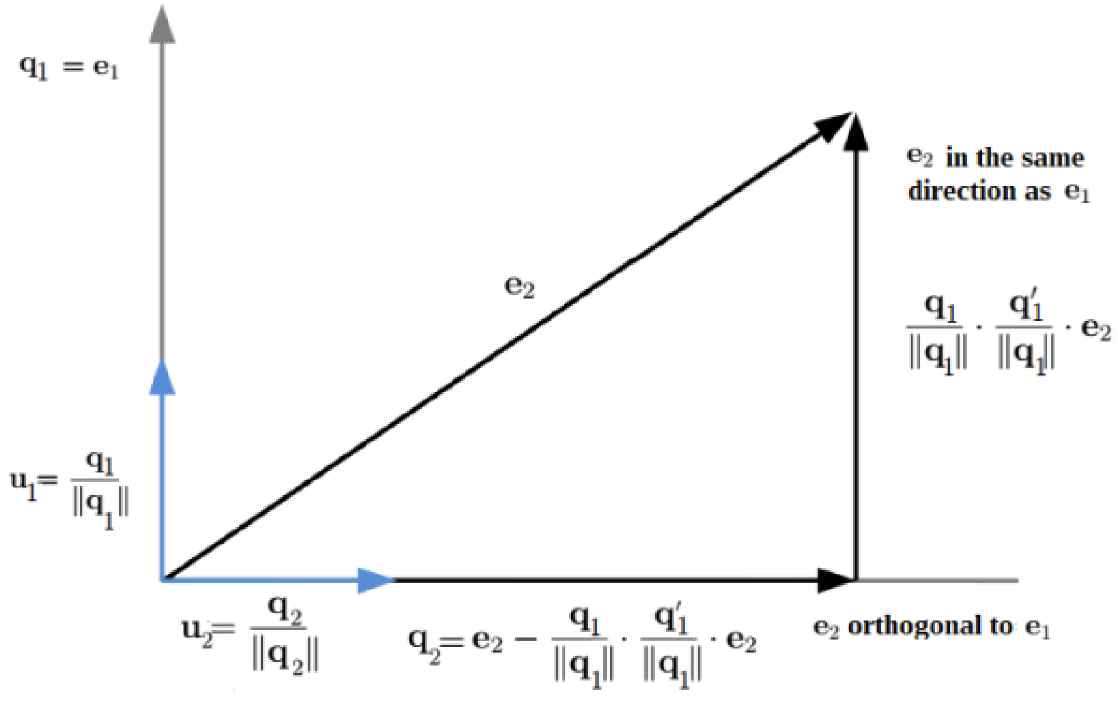
$\mathbf{M}_k = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{BS}]$

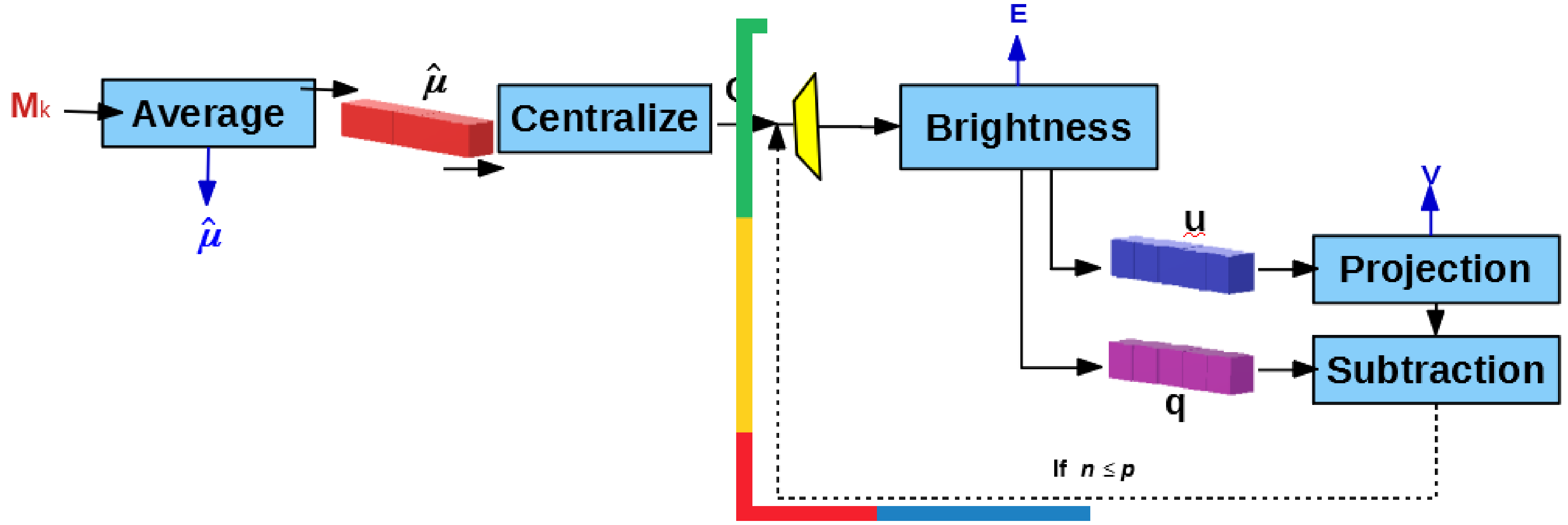
Outputs:

$\hat{\boldsymbol{\mu}}$ {Average Pixel}; $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p]$ {Characteristic pixels}; $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_p]$ {Orthogonalized vectors}; $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p]$ {Orthonormalized vectors}; $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_p]$ {Projection vectors}

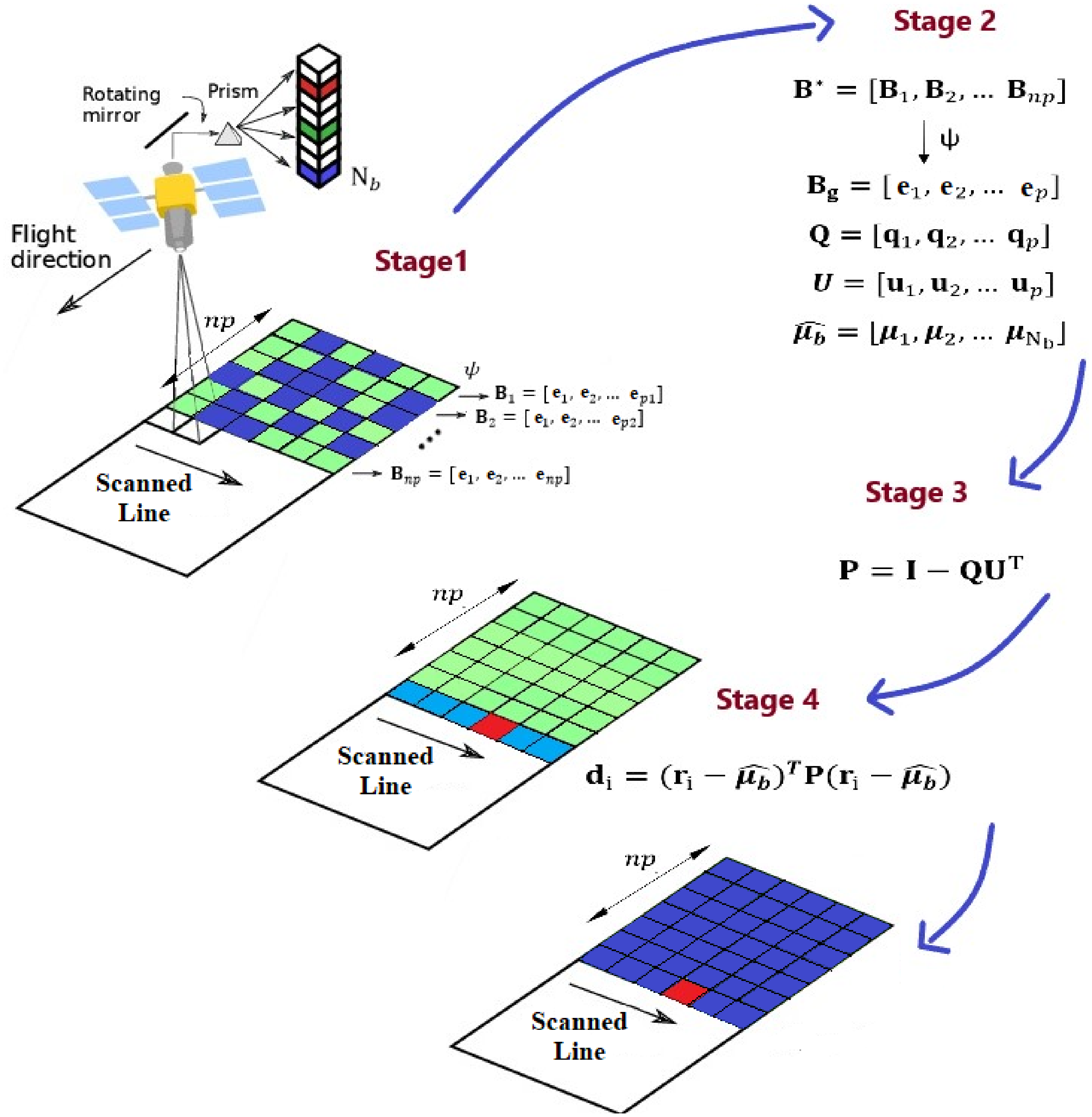
Algorithm:

```
Average pixel: μ̂;
Centralization: C = M_k − μ̂;
for n = 1 to p do
    for j = 1 to BS do
        Brightness Calculation: b_j = c'_j · c_j;
    end for
    Maximum Brightness: j_max = argmax(b_j);
    Extracted pixels: e_n = r_{j_max};
    q_n = c_{j_max};
    u_n = q_n / b_{j_max};
    Projection: v_n = u'_n · C;
    Subtraction: C = C − q_n · v_n;
end for
```





THE HW-LbL-FAD



01 **Background estimation**

- Stage 1
- Stage 2
- Stage 3

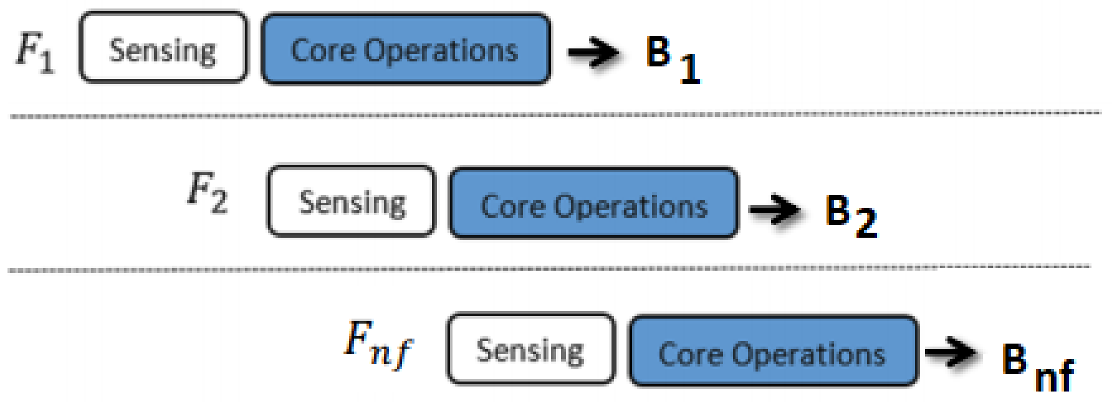
02 **Anomaly detection**

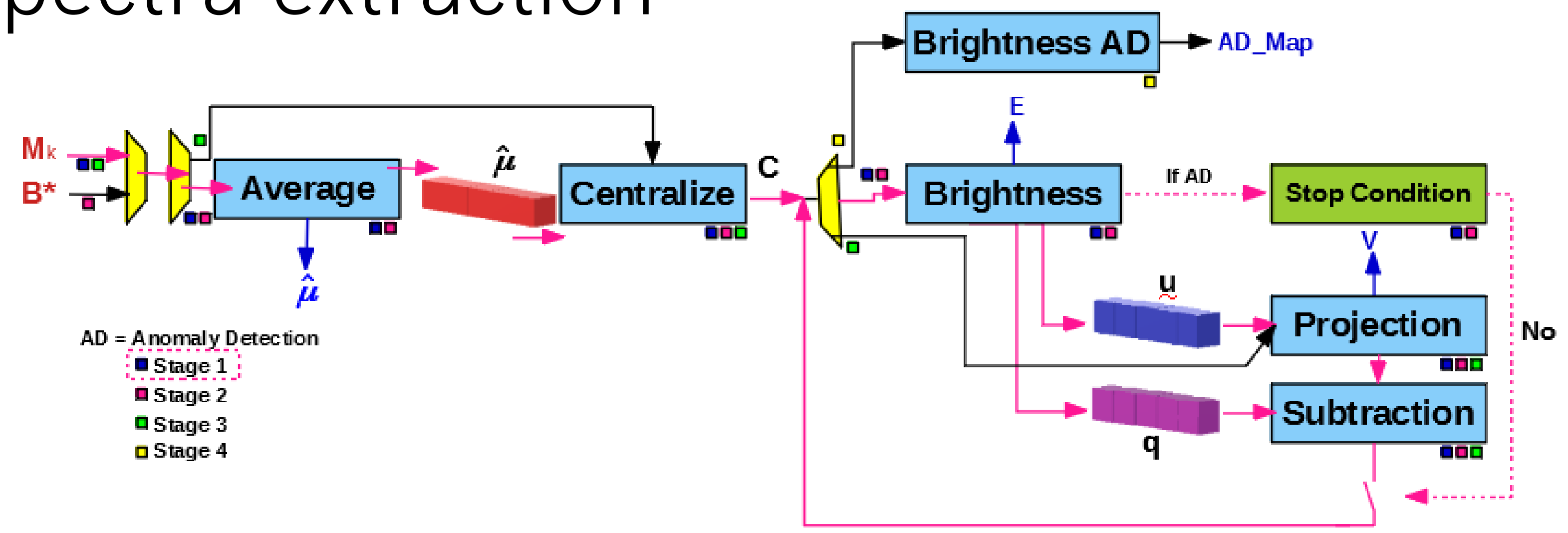
- Stage 4

THE HW-LbL-FAD

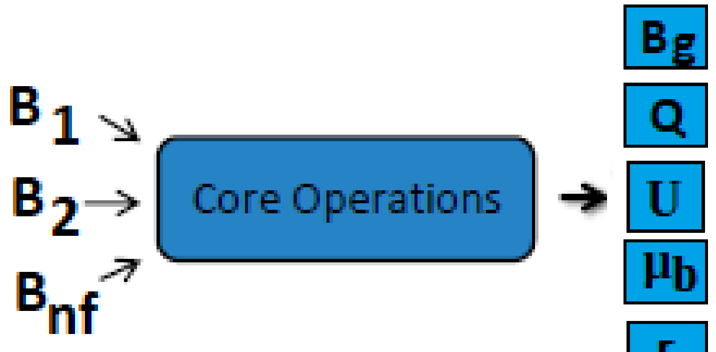
Background estimation

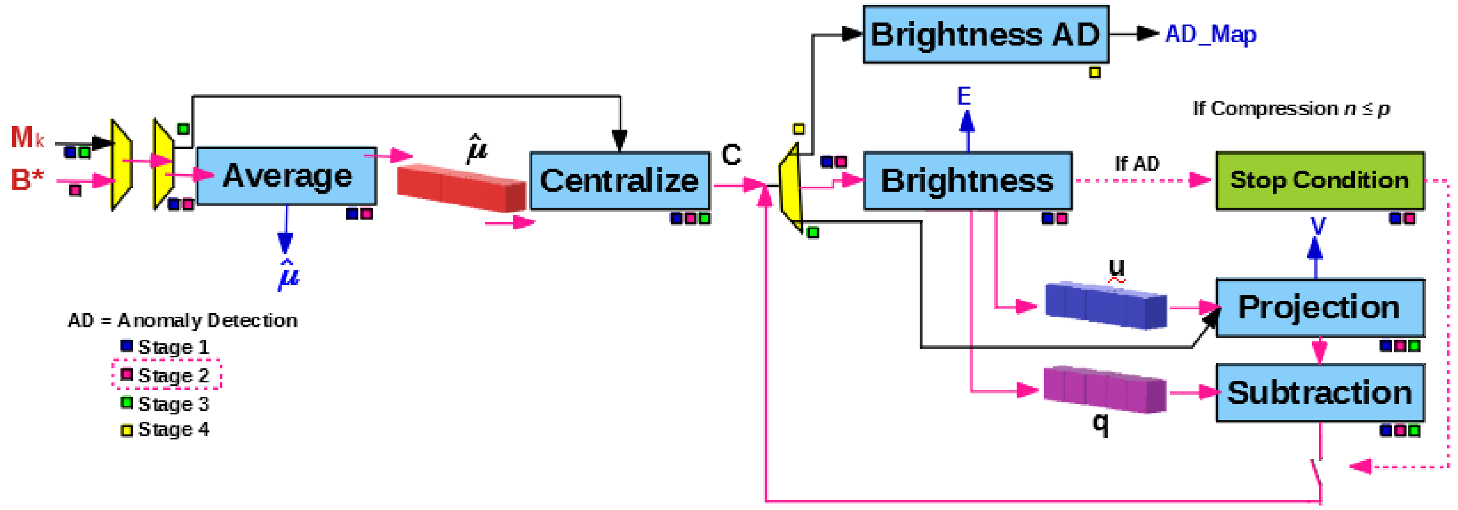
Stage 1: Line-by-line background spectra extraction





Stage 2: Overall background subspace estimation





THE HW-LbL-FAD

Background estimation

- Stage 3: Orthogonal Subspace to the one spanned by the background Samples

$$P = I - \frac{W \cdot W'}{W' \cdot W}; Q = W; U = \frac{W}{W' \cdot W} \longrightarrow P = I - Q \cdot U$$

Anomaly detection

- Stage 4: Detection of anomalies

$$\mathbf{d} = (\mathbf{r}_j - \hat{\mu}_b)' \cdot \mathbf{P} \cdot (\mathbf{r}_j - \hat{\mu}_b) \longrightarrow \textit{Threshold} \rightarrow 1{,}5 \cdot \tau$$

THE HW-LbL-FAD

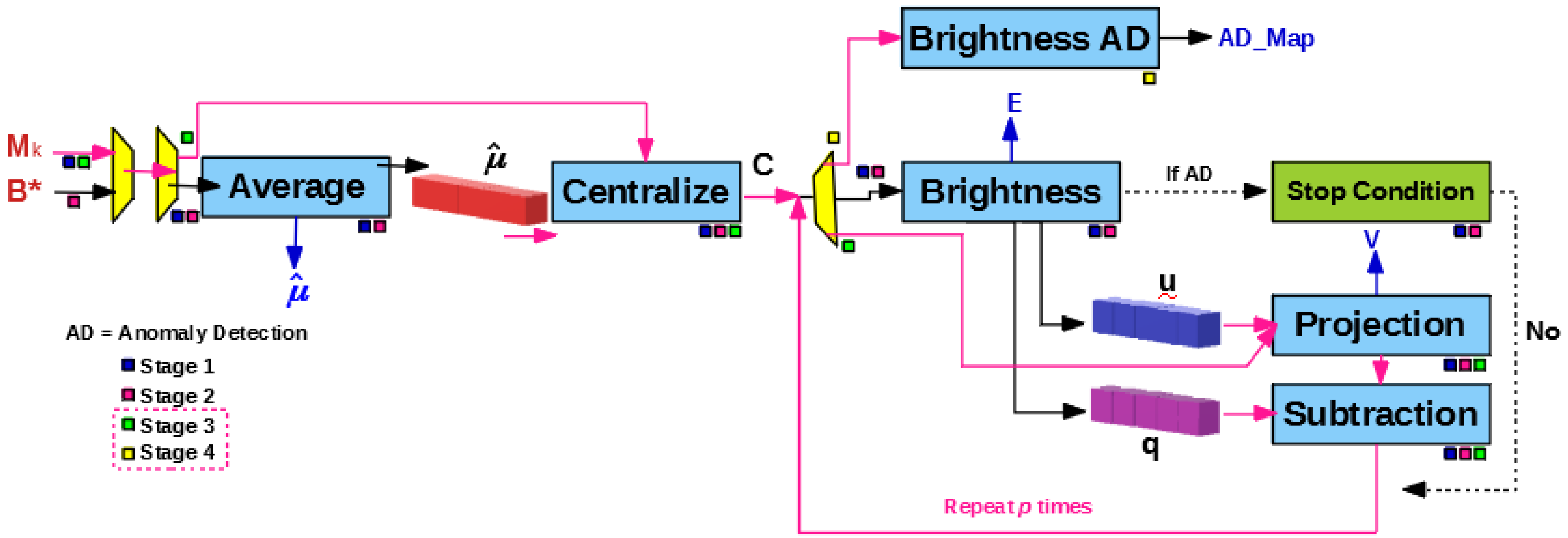
Background estimation

- Stage 3: Orthogonal Subspace to the one spanned by the background Samples

$$P = I - \frac{W \cdot W'}{W' \cdot W}; Q = W; U = \frac{W}{W' \cdot W} \longrightarrow P = I - Q \cdot U$$

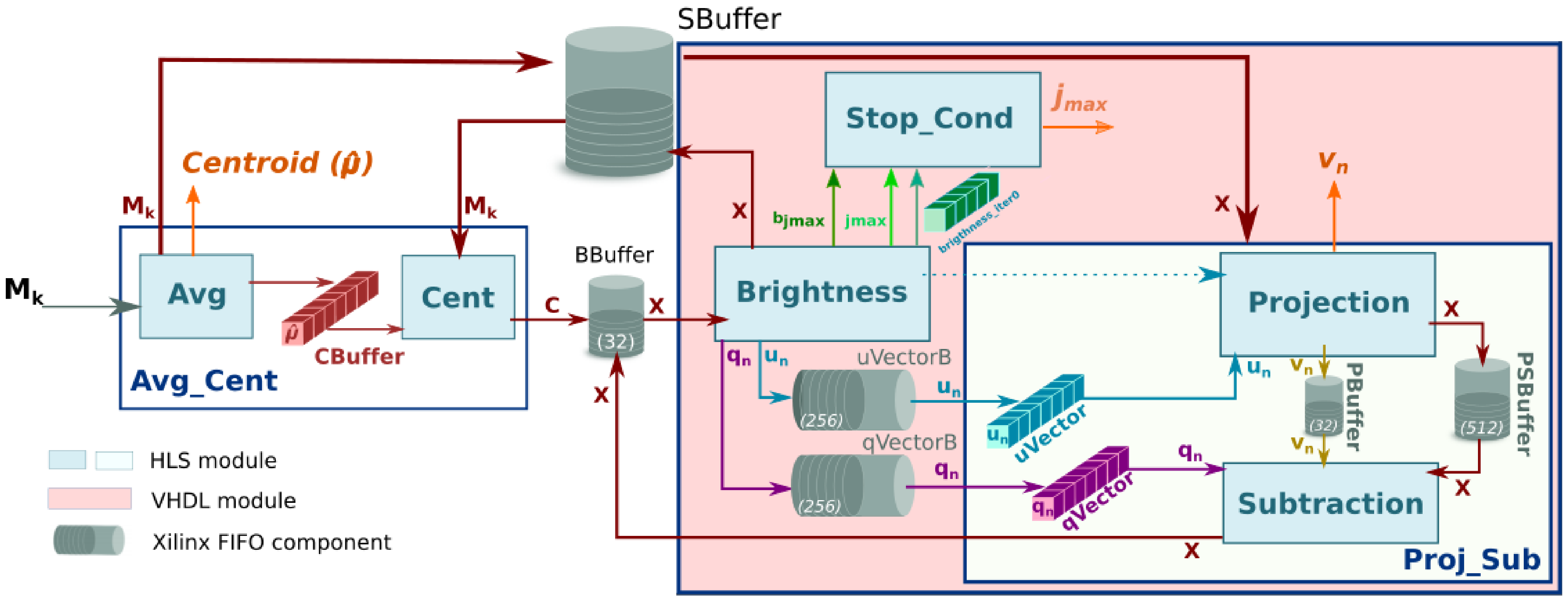
Anomaly detection

- Stage 4: Detection of anomalies



$$\mathbf{d} = (\mathbf{r_j} - \hat{\mu}_b)' \cdot \mathbf{P} \cdot (\mathbf{r_j} - \hat{\mu}_b) \longrightarrow Threshold \rightarrow 1{,}5 \cdot \tau$$

FPGA-BASED IMPLEMENTATION



Set of Core operations

Avg_Cent, Brightness, Proj_Sub, Stop_cond ---> HLS

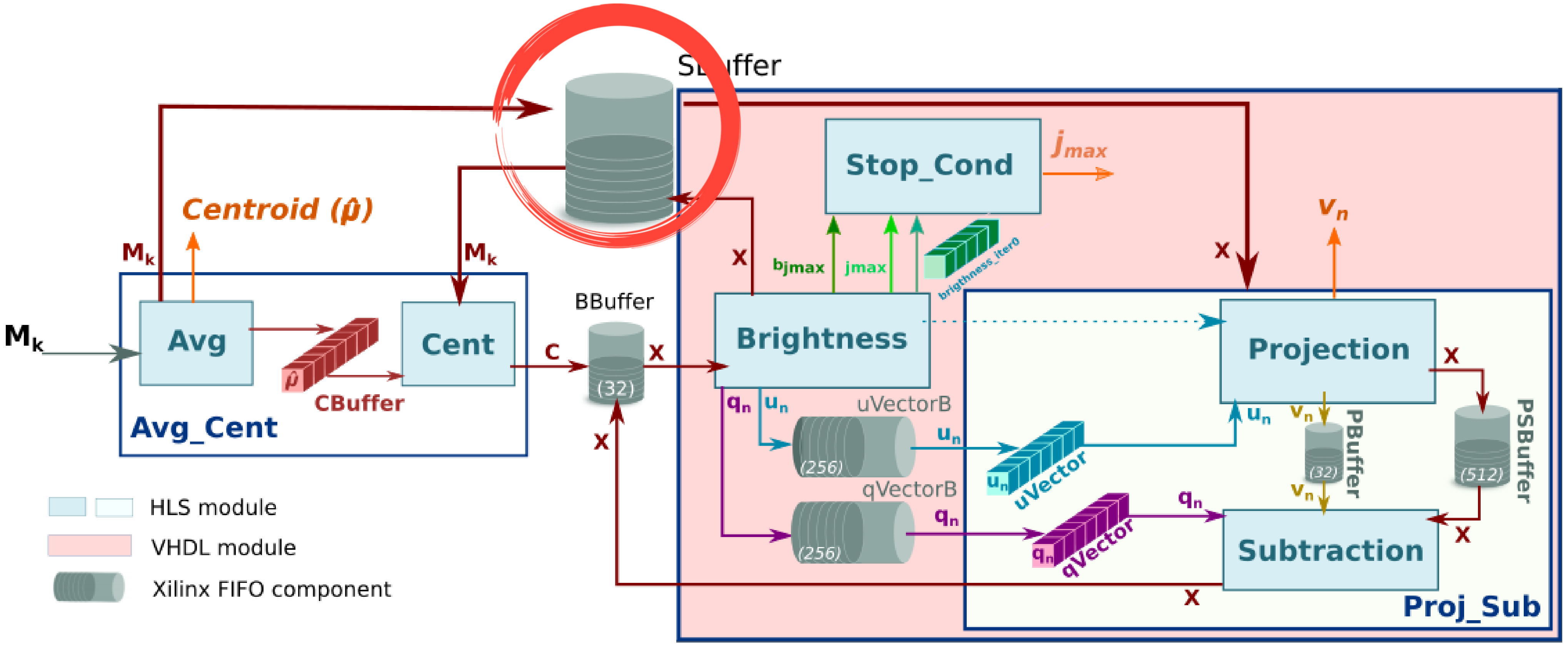
Glue logic + memory elements + FSM ---> VHDL

Hybrid solution HLS + VHDL

FPGA-BASED IMPLEMENTATION

FPGA-BASED IMPLEMENTATION

SET OF CORE OPERATIONS



Set of Core operations

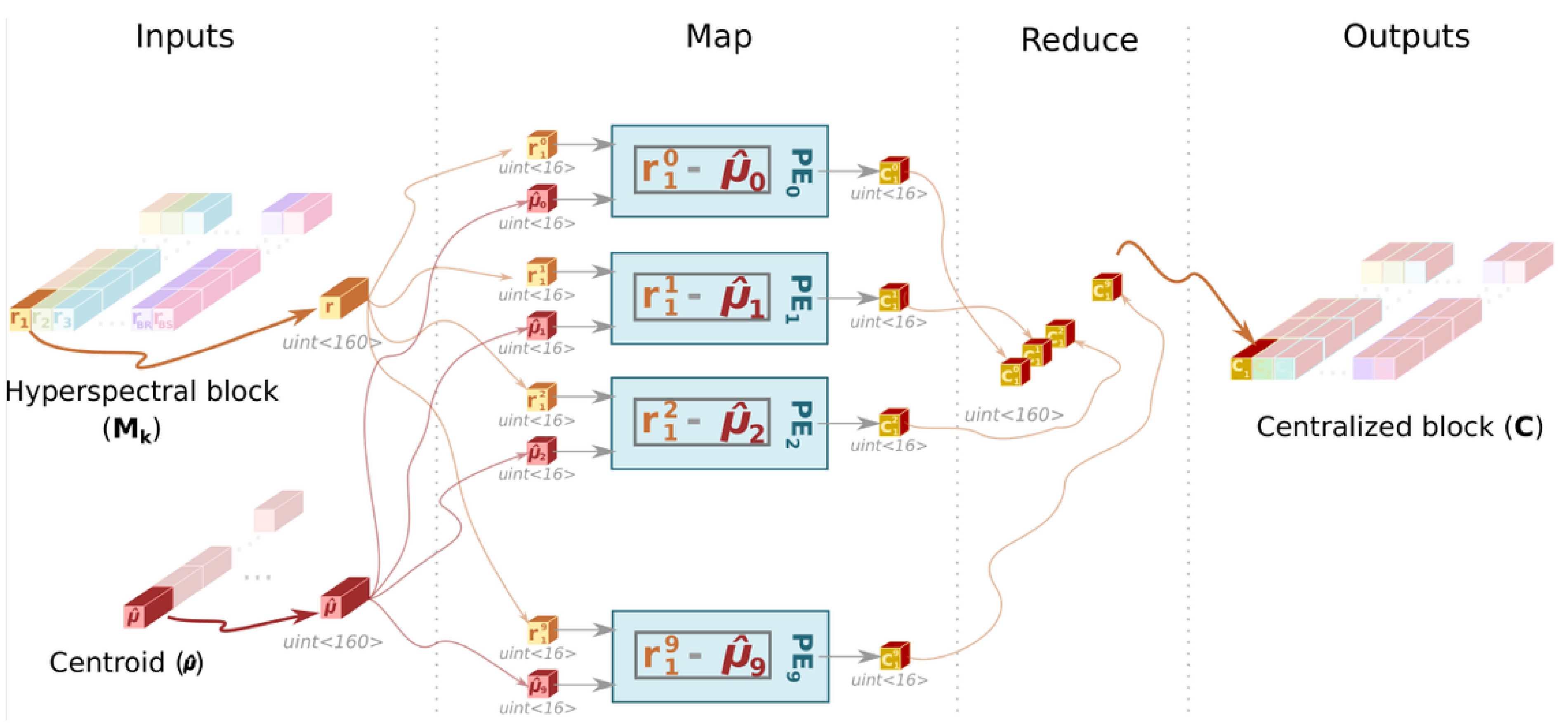
Avg_Cent, Brightness, Proj_Sub, Stop_cond ---> HLS

Glue logic + memory elements + FSM ---> VHDL

Depends on BS and bit-depth ---> Design time

Hybrid solution HLS + VHDL

FPGA-BASED IMPLEMENTATION



Map-reduce programming model --- > PEs

Set of Core operations

Avg_Cent, Brightness, Proj_Sub, Stop_cond ---> HLS

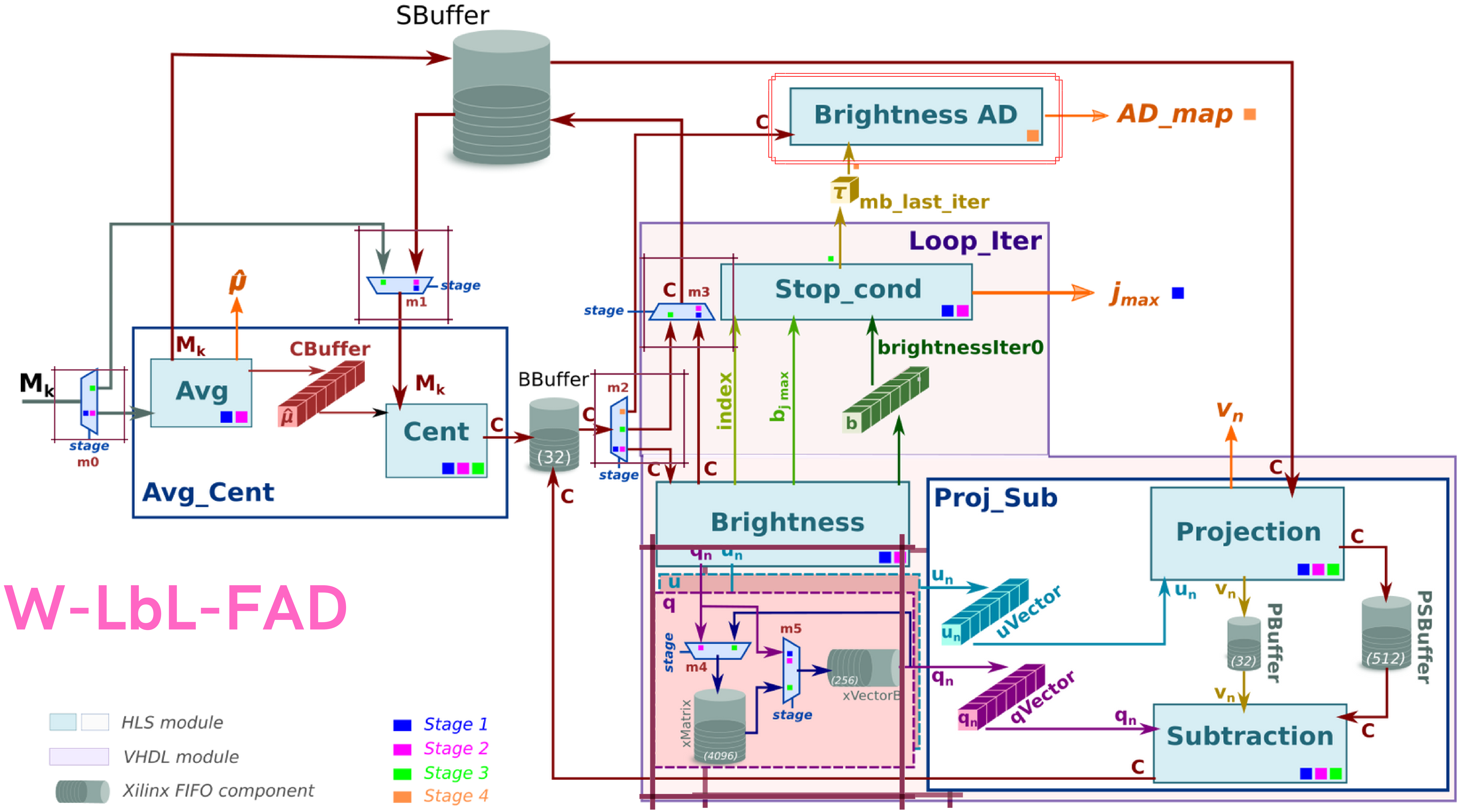
Glue logic + memory elements + FSM ---> VHDL

Hybrid solution HLS + VHDL

FPGA-BASED IMPLEMENTATION

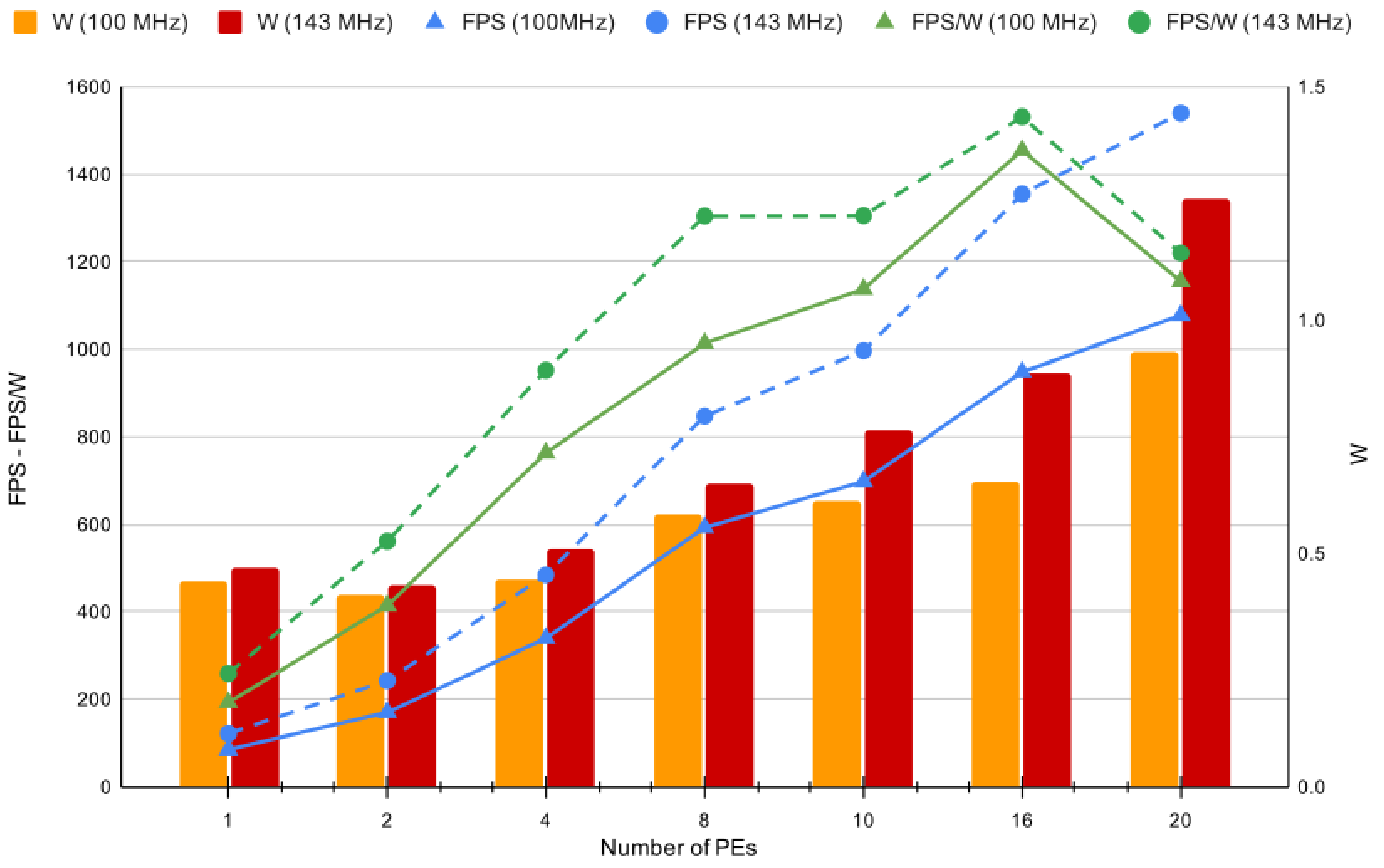
SET OF CORE OPERATIONS

FPGA-BASED IMPLEMENTATION



HW-LbL-FAD

EXPERIMENTAL RESULTS



BS	*PEs*	BRAM18K	DSP48E	FFs	LUTs
	1	107 (76.43%)	14 (6.36%)	8073 (7.59%)	6744 (12.68%)
	2	91.5 (65.36%)	22 (10%)	8624 (8.11%)	7470 (14.08%)
	4	95.5 (68.21%)	38 (17.27%)	9981 (9.38%)	9115 (17.13%)
1024	8	98.5 (70.36%)	70 (31.82%)	12666(11.90%)	12411(23.33%)
	10	102.5 (73.21%)	86 (39.09%)	14787 (13.90%)	14856 (27.92%)
	16	96.5 (68.93%)	134 (60.91%)	18433 (17.32%)	18724 (35.20%)
	20	98.5 (70.36%)	166 (75.45%)	23071 (21.68%)	23493 (44.16%)

Sensor	Pixels per frame	Bands per frame	Frames per second	Data resolution
SPECIM FX-10	1024	240/160	330	12 bits/ 16 bits packaged

THE HW-LbL-FAD

Image	Inputs	LbL-FAD		HW-LbL-FAD							
				Float32		Int32		Int16		Int16-rd	
	n_f	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Synthetic Image	20	100	0,00	100	0	100	0	100	2,79	100	0
WASP RIT	60	93,06	0,09	93,06	0,09	93,06	0,09	79,17	0,06	93,06	0,09
AVIRIS WTC	30	43,37	0,02	43,37	0,02	43,37	0,02	44,58	0,02	43,37	0,02

CONCLUSIONS

Line-by-line anomaly detector

… for power-contrained environments such as onboard scenarios.

LbL-FAD hardware accelerator

- Targeting the XC7Z020-CLG484 version of the Xilinx Zynq-7000SoC due to its low-cost, low-weight and high flexibility.
- Hybrid solution: combination of HLS generated modules and custom glue logic in VHDL.

Real-time performance

… since obtrained FPS are between 130.11 (PE=1) and 1609.90 (PE=20).

Good power-efficiency relation

… critical for applications that use remote sensing platforms powered by batteries.

AN FPGA-BASED IMPLEMENTATION OF A HYPERSPECTRAL ANOMALY DETECTION ALGORITHM FOR REAL-TIME APPLICATIONS

Authors:

María Díaz Martín, Julián Caba, Raúl Guerra Hernández, Jesús Barba, Sebastián López Suárez

International Geoscience and Remote Sensing Symposium (iGARSS) July, 2021