High Performance Embedded System for Real-Time Pattern Matching

C.-L. Sotiropoulou[*,1,2], P. Luciano[2,3], S. Gkaitatzis[4], S. Citraro[1,2], P. Giannetti[2] and M. Dell'Orso[1,2]

[1] University of Pisa, [2]INFN-Pisa, [3]UNICLAM, [4]AUTH

[*]Presenter email: c.sotiropoulou@cern.ch

Abstract

We present an innovative and high performance embedded system for real-time pattern matching. This system is based on the evolution of hardware and algorithms developed for High Energy Physics (HEP) and more specifically for the execution of extremely fast pattern matching for tracking of particles produced by proton-proton collisions in hadron collider experiments, such as the ATLAS FTK [1]. A miniaturized version of this complex system is being developed for pattern matching in generic image processing applications. The design uses the flexibility of Field Programmable Gate Arrays (FPGAs) and the powerful Associative Memory Chip (ASIC) to achieve real-time performance. The system works as a contour identifier able to extract the salient features of an image. It is based on the principles of cognitive image processing, which means that it executes fast pattern matching and data reduction mimicking the operation of the human brain.
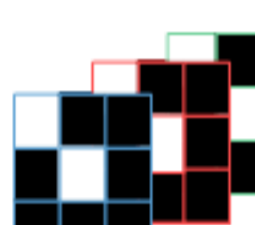
Algorithm Description

Static Images

Black/White — $2^9 = 512$ patterns: 101-010-100, ..., 111-011-001

4 lv of Gray — $2^{18} = 256$ Kpatterns: 00,00,01-00,01,00-11,00,10 ....

Movies

$2^{27} = 128$ Mpatterns: 111,000,000 - 000,111,000 - 000,000,000

Algorithm Sequence [2]:

1° Stage → **Training:** calculate the frequency and entropy of each pattern in the sample images/frames to select the relevant patterns to be used as reference for the matching process
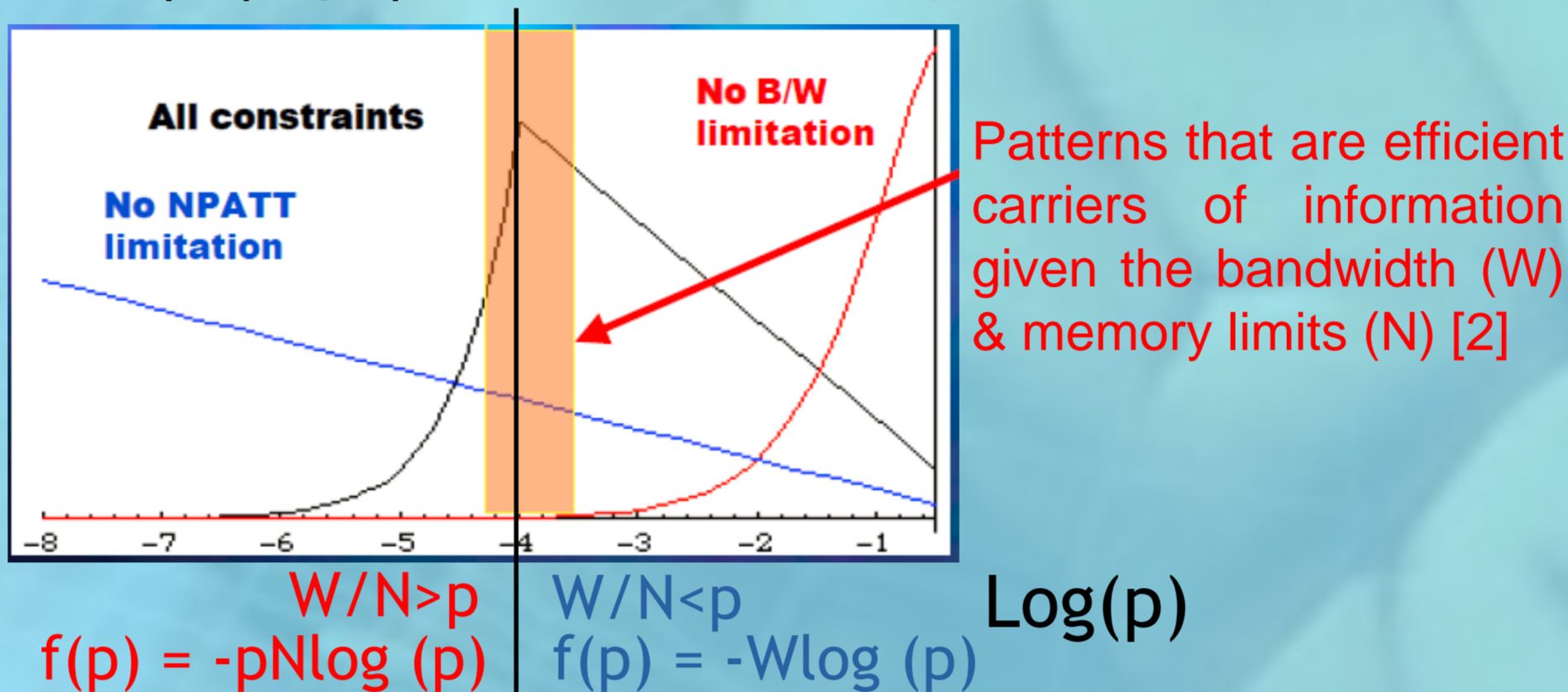
2° Stage → **Run:** send the image patterns to AM chip to be filtered matching the relevant patterns

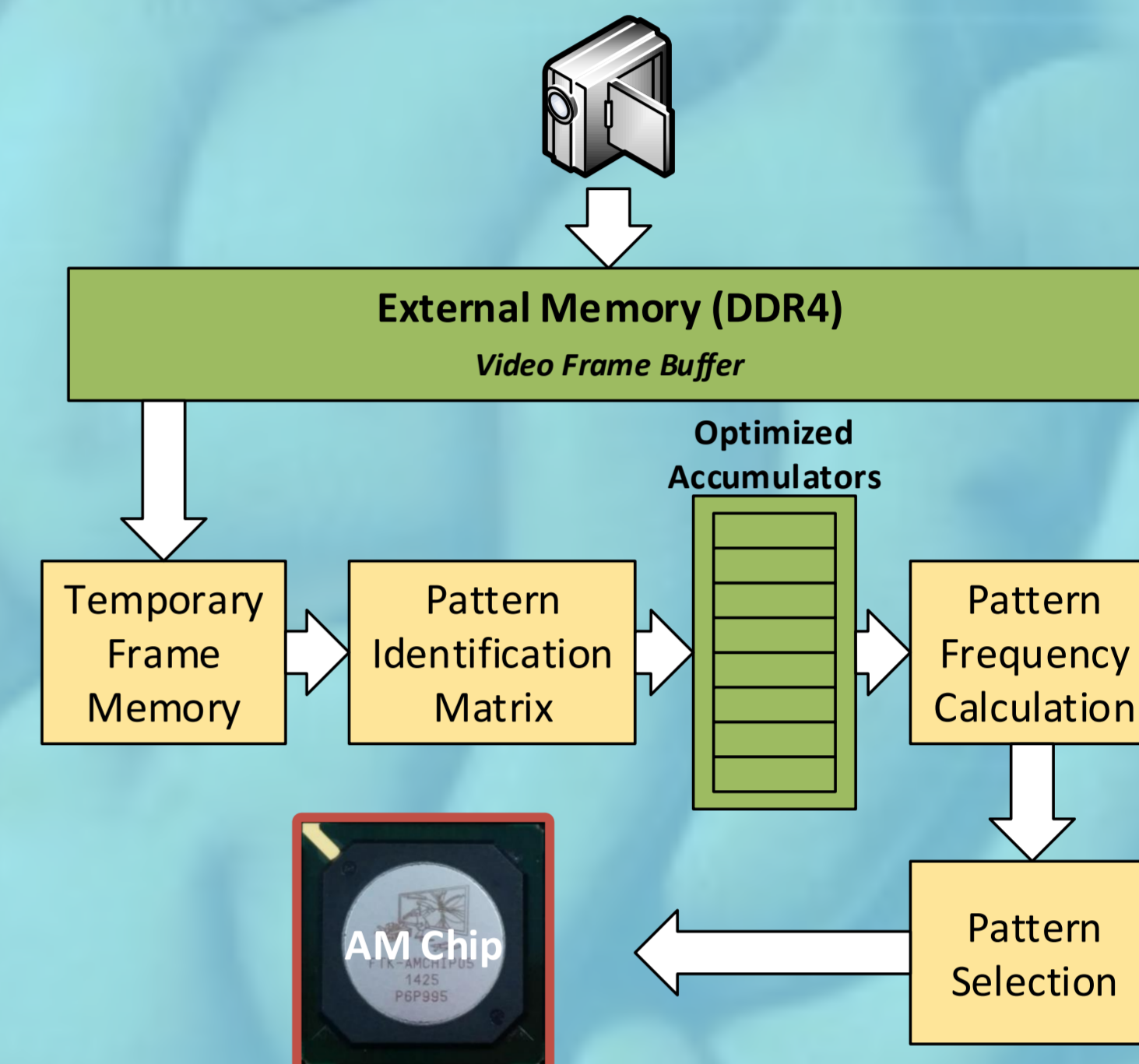The AM chip is a CAM like ASIC that executes pattern matching in parallel [3]

Pattern Selection Definition

p_i : probability pattern "i" in sample image set
N_{tot} : total number of patterns
N : number of stored patterns (available memory space)
W : available bandwidth
H : measure of output information

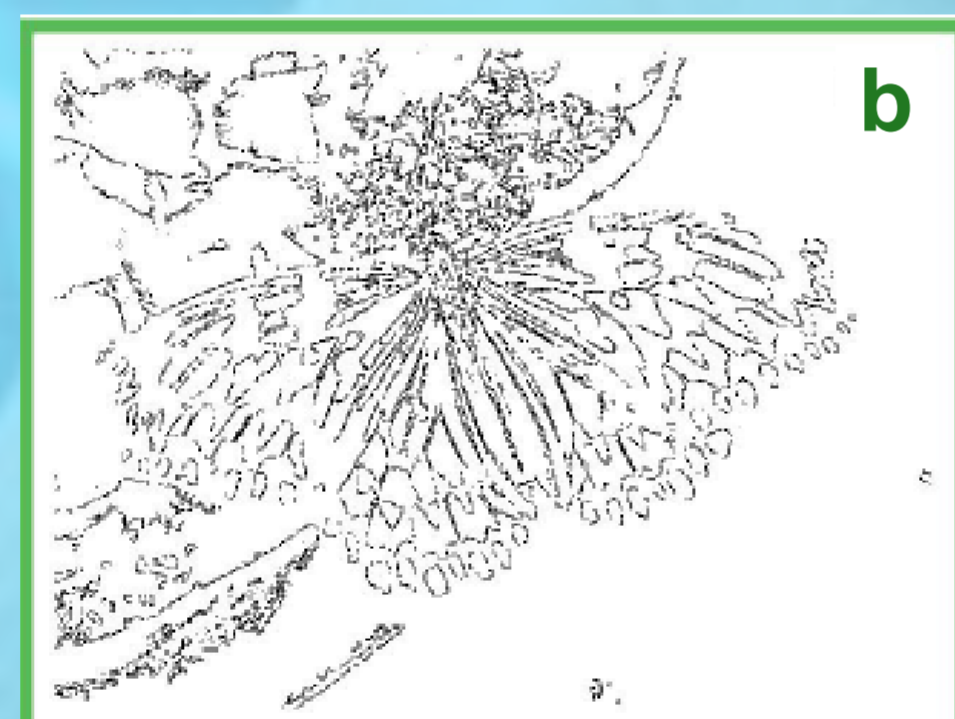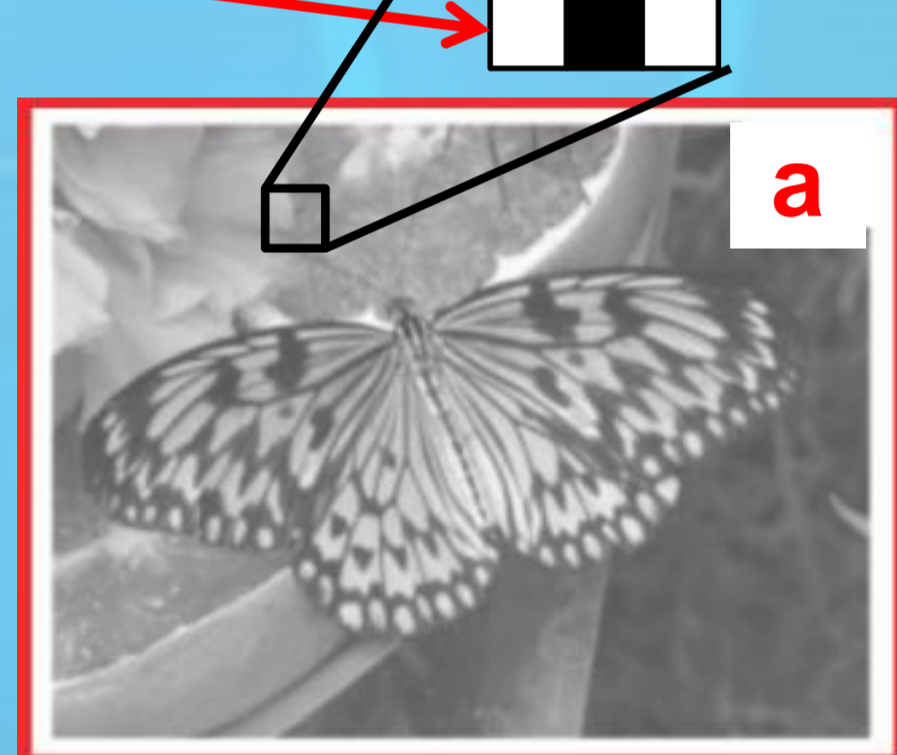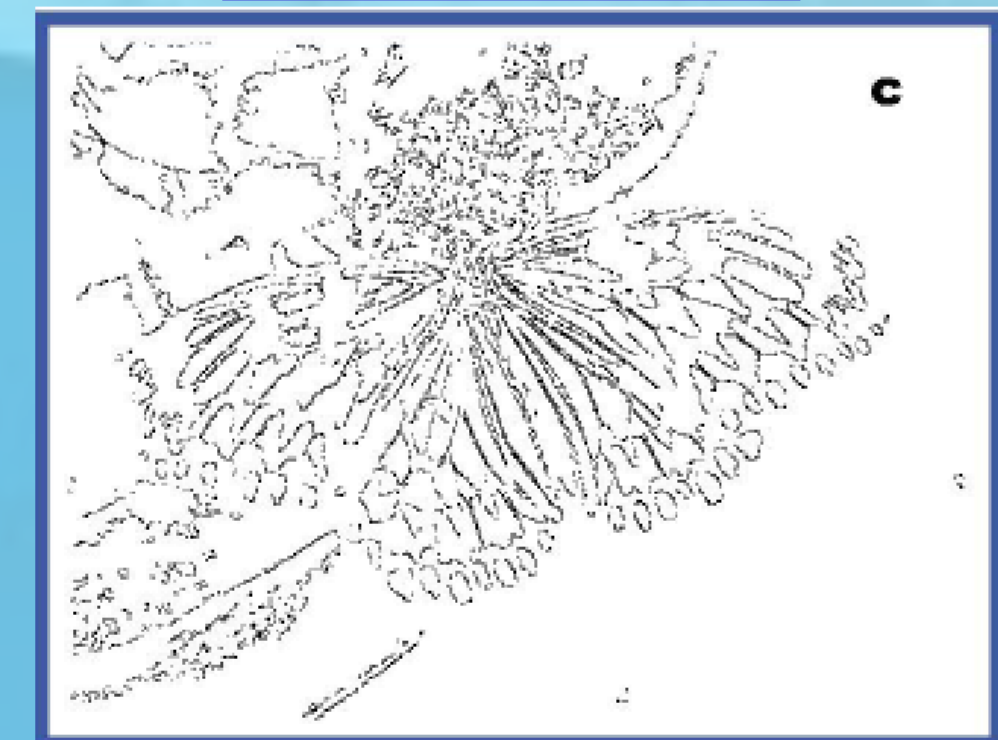$$H = \sum_i -p_i \log(p_i) \quad \text{(Boltzmann entropy)}$$

All constraints
No NPATT limitation
No B/W limitation

Patterns that are efficient carriers of information given the bandwidth (W) & memory limits (N) [2]

$Log(p)$

$W/N > p$ $\quad$ $W/N < p$
$f(p) = -pN\log(p)$ $\quad$ $f(p) = -W\log(p)$

Training Phase Implementation

External Memory (DDR4)
Video Frame Buffer

Optimized Accumulators

Temporary Frame Memory → Pattern Identification Matrix → Pattern Frequency Calculation → Pattern Selection → AM Chip

Pattern Selection and Impact on Output

pixel

Info: 100%

Y-axis: Number of patterns/freq unit (0, 20, 40, 60, 80, 100)
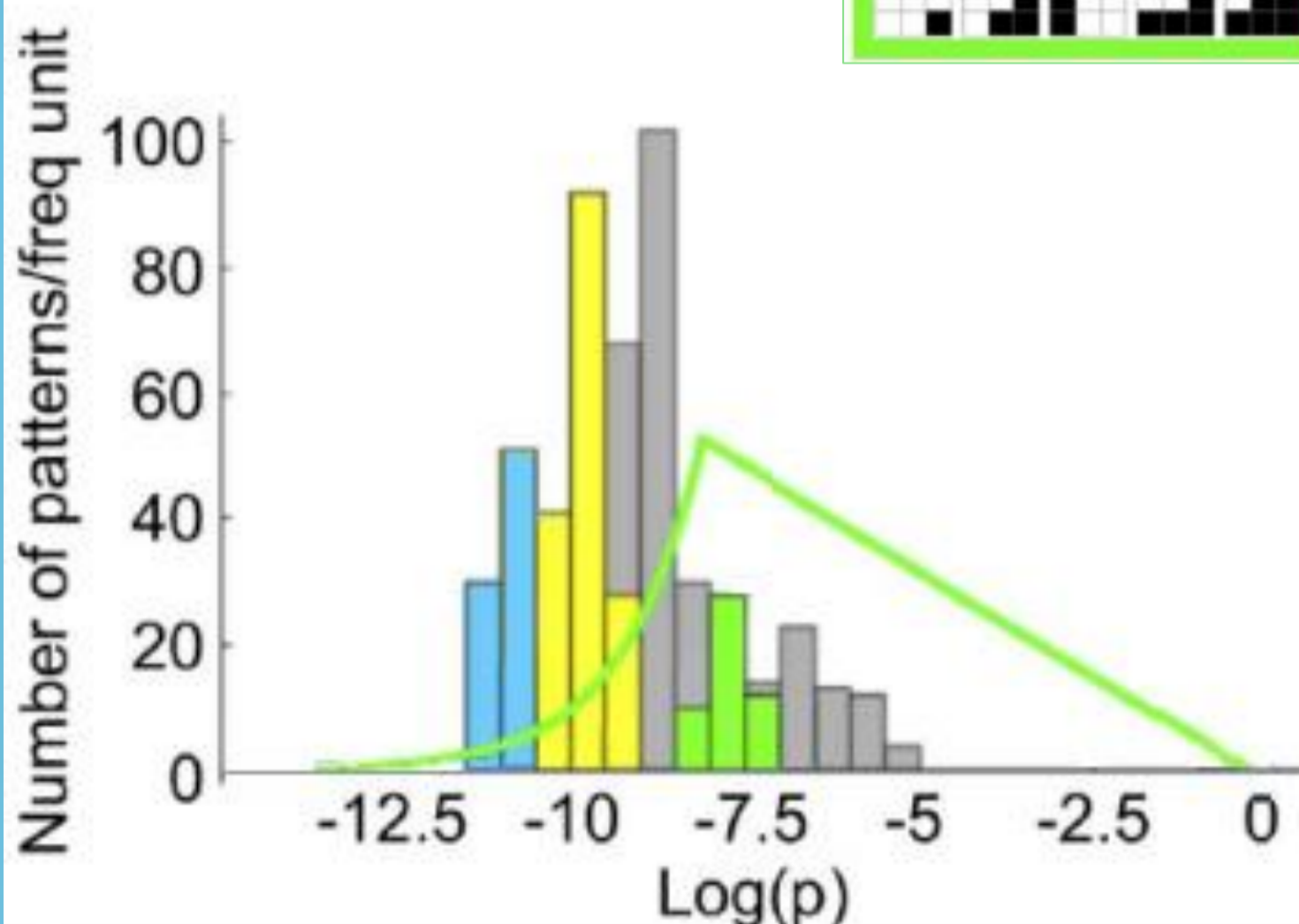*X-axis: Log(p) (-12.5, -10, -7.5, -5, -2.5, 0)*

50 relevant patterns

16 relevant patterns

An example of pattern selection (as described in [2]) performed on a natural digitized image representing a butterfly. It can be seen that the filtered images (b) and (c) obtained with the selection of "relevant patterns" are recognizable, while in image (d) filtered with low probability patterns, the important features are lost.

Hardware Setup

The AM chip: Pattern Matching Execution

FPGA: Training – Pattern Selection and Image Post-Processing
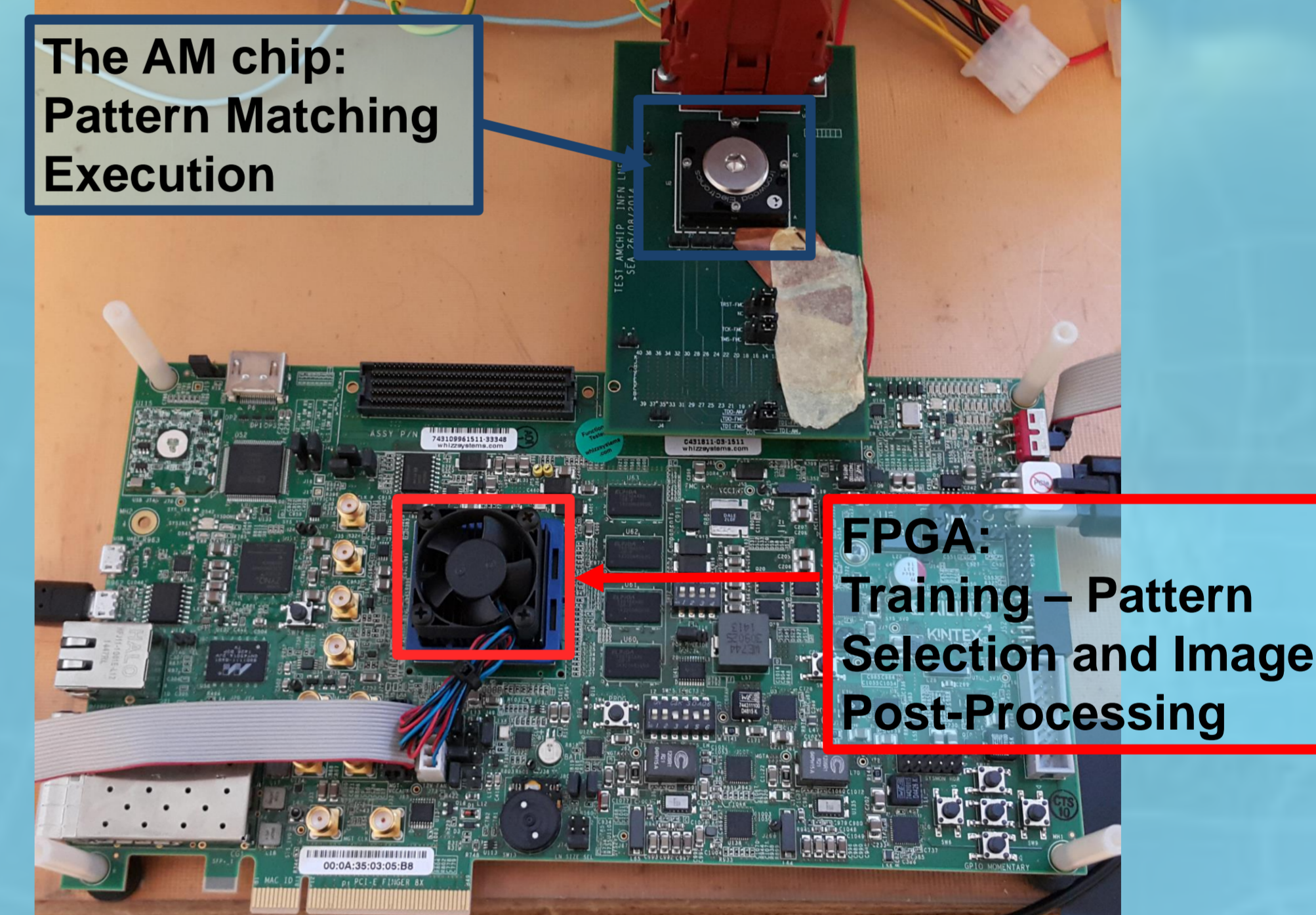
The simplest hardware system we can use is a combination of a Xilinx Kintex Utrascale evaluation board (KCU105) with a small mezzanine card housing the Associative Memory chip AM05, both designed by M. Beretta, F. Crescioli, A. Stabile et. al [3].

The training and post-processing step (e.g. Clustering as a post-processing algorithm in [4]) of the algorithm are implemented on the FPGA to take advantage of the device flexibility. The Associative Memory chip can be used to execute very fast pattern matching.

Simulation and Implementation Results
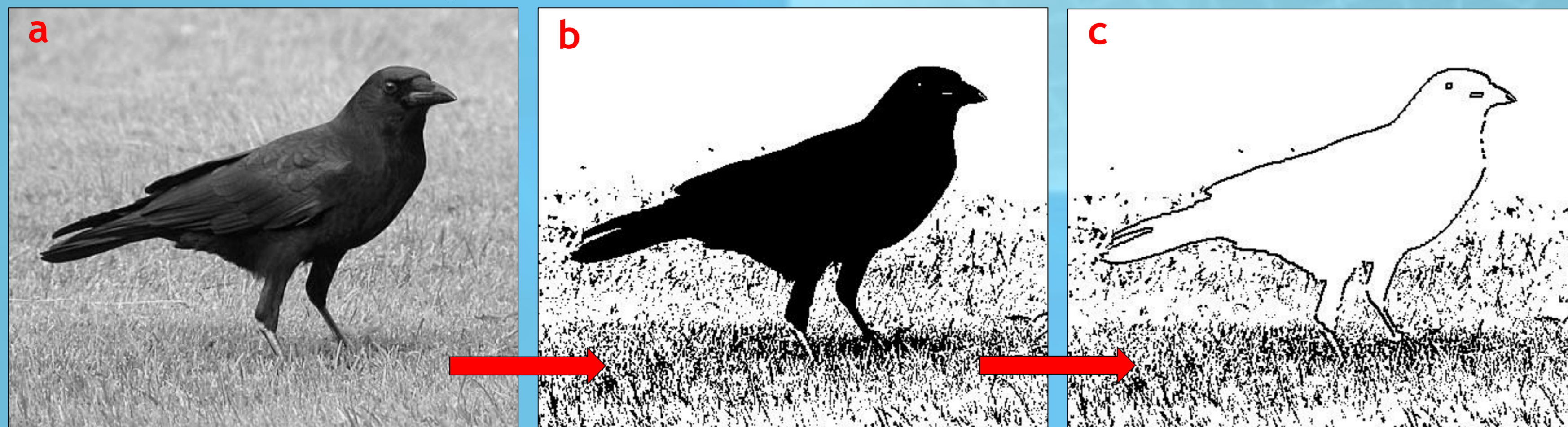
Output example from software simulation framework: a) Input: 8bit grayscale image, b) Black and white transformation using threshold at 150 luminosity, c) Output: Contour identification, pattern matching executing using 50 relative patterns – edge detection operation

Hardware status:

The Training System is implemented and tested on the Kintex Ultrascale FPGA. We use a set of sample images to test and verify the system operation. The complete training implementation requires **less than 2% of the FPGA LUTs** and **2.4% of the available BRAMs** for processing of black and white images. We use a **250 MHz clock**. To process a **512x512 pixel** image **less than 2,5 msecs** are required.

The software simulation instead requires more than 3 secs to execute the training algorithm for the same image on a 3rd generation i5 processor with 4 GB RAM. The results for 50 relative patterns are comparative in quality with a Canny Edge Detector.

Future Targets

- Extend the implementation to be operational for 4 levels of gray and 3D images/movies
- Create a small mezzanine card (FPGA+Amchip) to be used as a hardware accelerator for normal CPUs
- Targeting biomedical, security, space applications

References

[1] A. Andreani, et al, "The FastTracker Real-Time Processor and Its Impact on Muon Isolation, Tau and b-Jet Online Selections at ATLAS," IEEE Trans. on *Nuclear Science*, vol.59, no.2, pp. 348-357, April 2012.

[2] M. Del Viva et. al "Information and Perception of Meaningful Patters," PLoS one 8.7 (2013): e69154.

[3] A. Andreani, et al., "Characterisation of an Associative Memory Chip for high-energy physics experiments," in *Proc. I2MTC*, 2014, Montevideo. pp. 1487 – 1491

[4] C.-L. Sotiropoulou et. al "A Multi-Core FPGA-based 2D-Clustering Implementation for Real-Time Image Processing", in IEEE Trans. on *Nuclear Science*, vol. 61, no. 6, pp. 3599 - 3606, December 2014.

Acknowledgements