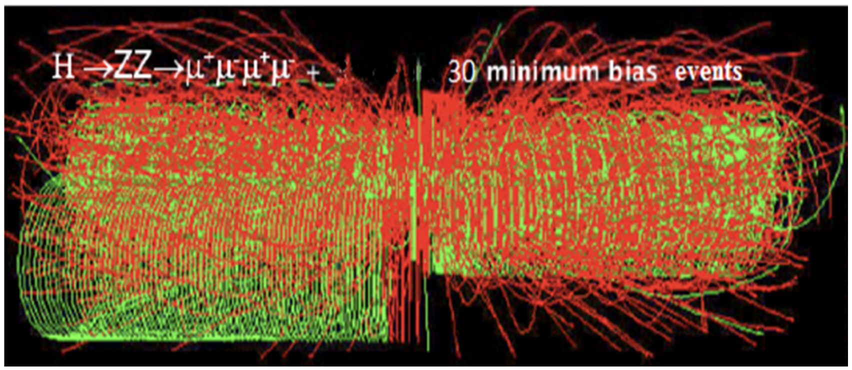
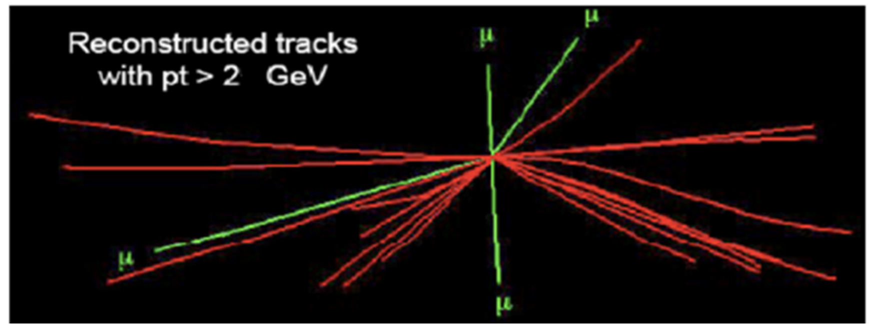
FTK IAPP Project: A New Approach to Real Time Image Processing

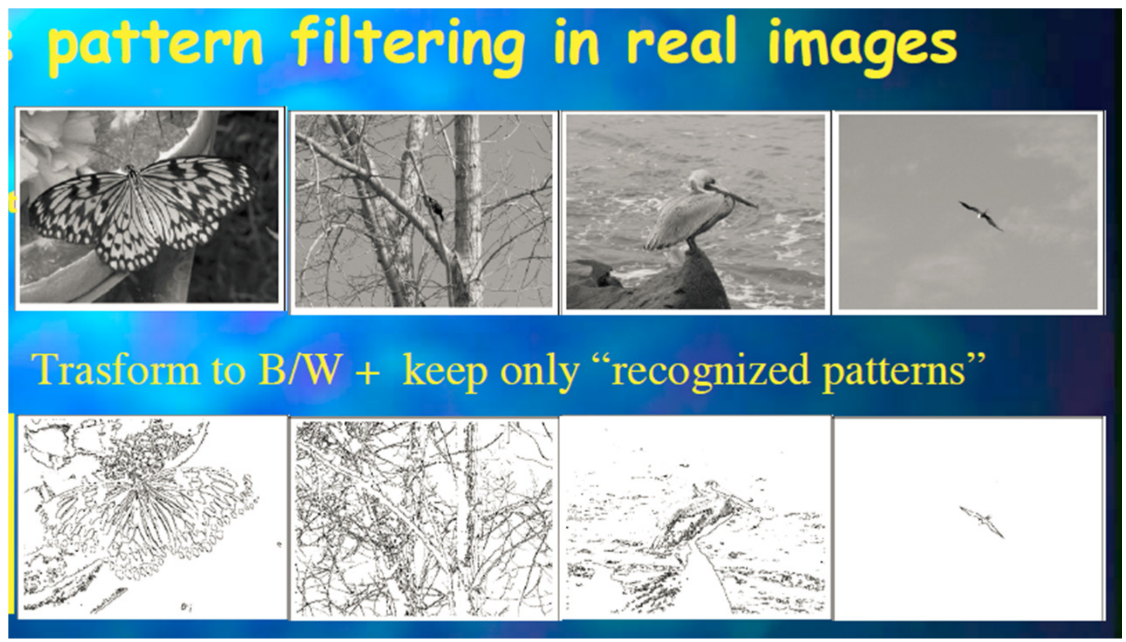
AM: a filter to detect the IMAGE relevant features



HEP



/www.plosone.org/article/info%3Adoi%2F10.1371%2Fjournal.pone.0069154 M. Del Viva, G. Punzi

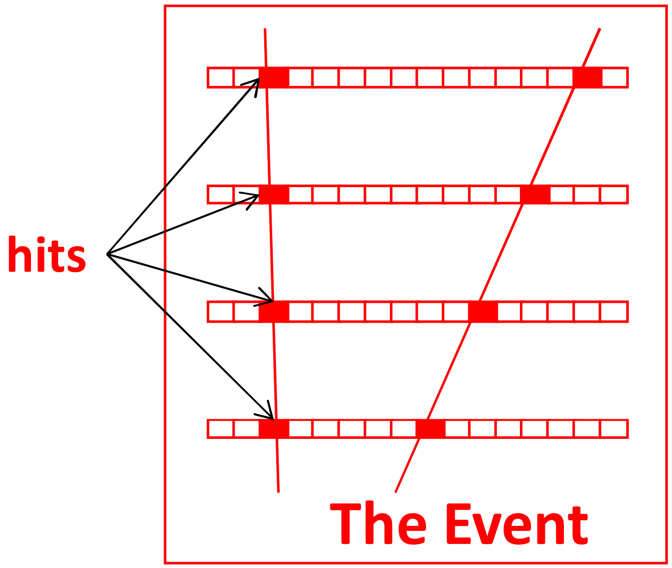


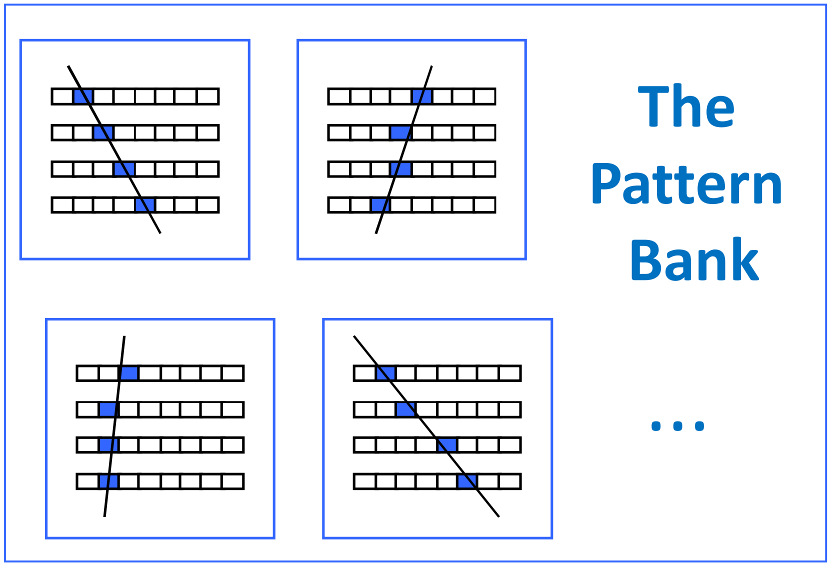
FILTERING NATURAL IMAGES: edge detector

→ AM as neurons?

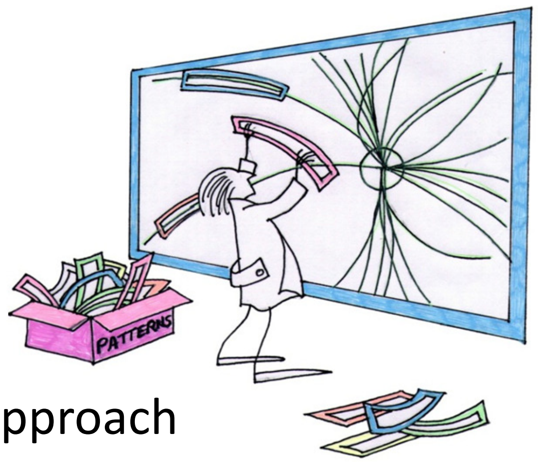
Filtered images are clear to human eyes

Pattern-Matching for Tracking



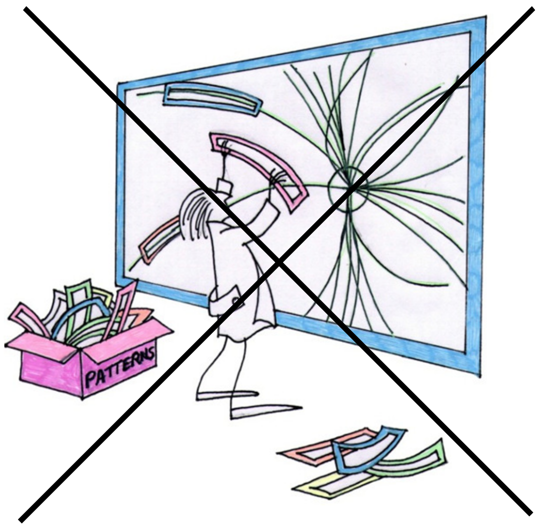


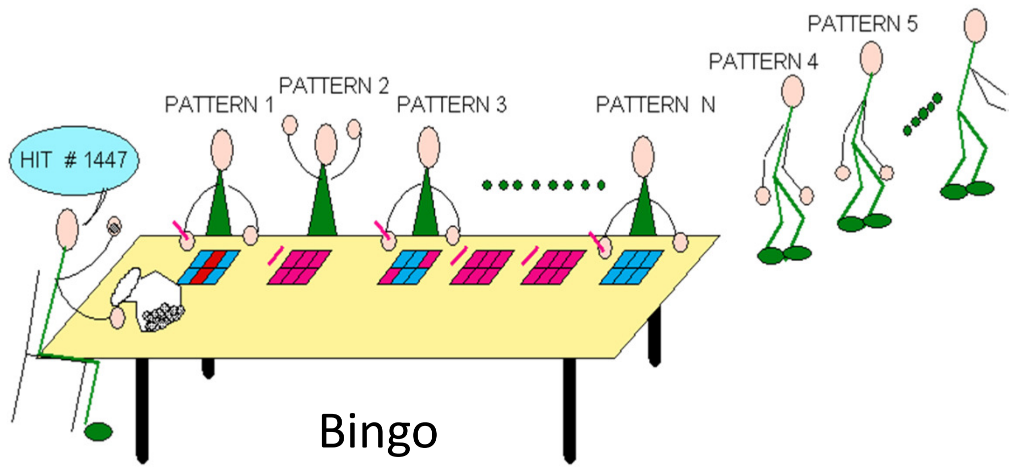
Before the run:
Finite resolution defines bins.
Precalculate bin patterns: roads.
Store roads in a bank.
While running:
Compare roads to event hits.



Serial approach

Associative-Memory Pattern-Matching





AM chip 03

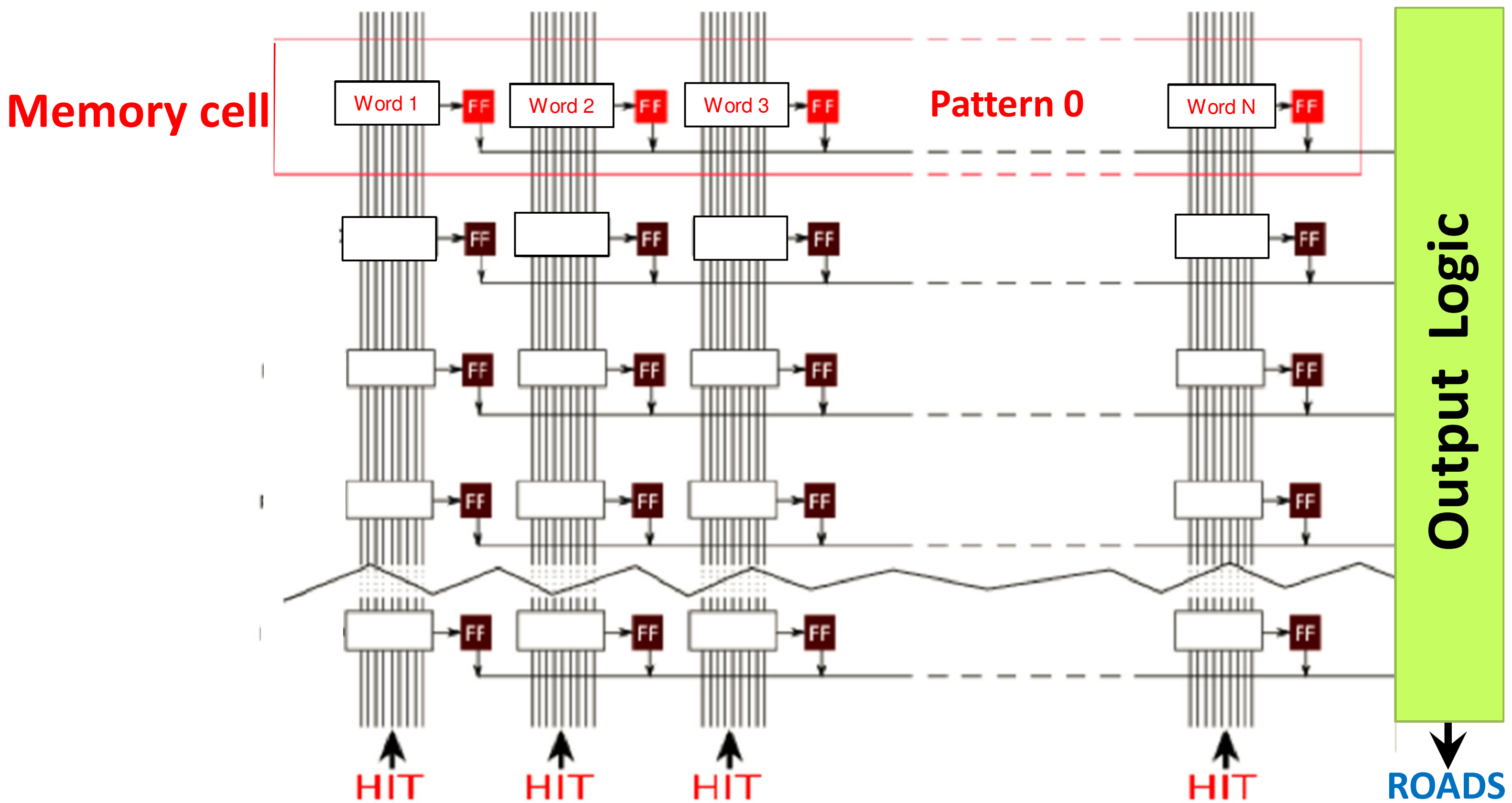
~10^9 patterns for FTK @ ATLAS

All patterns simultaneously see silicon hits coming from detector

Pattern recognition is complete as soon as all data are read out!

But… coarse resolution: ~100s μm

AM-Chip Architecture

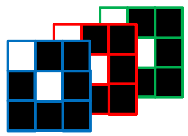


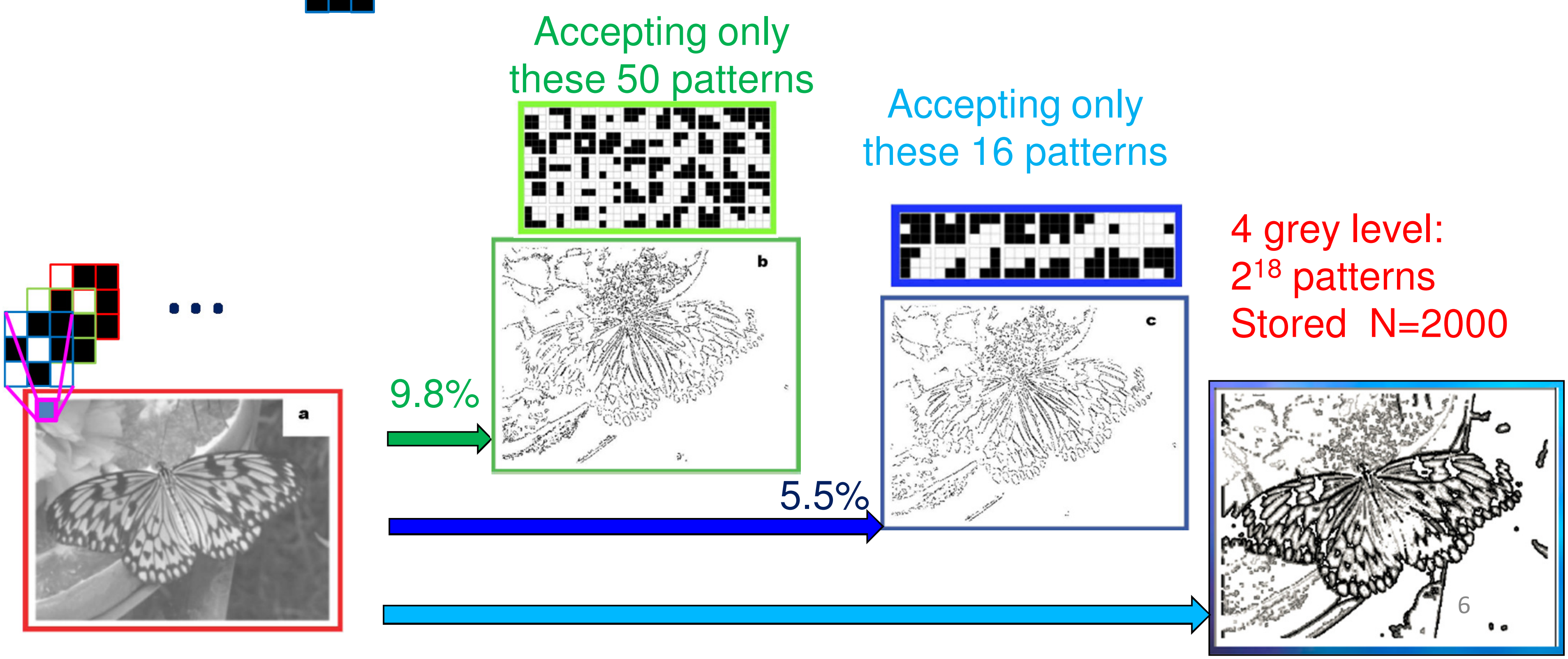
One hit bus per detector layer
N words per memory cell

Each word stores a bin coordinate
Each word has its own comparator

HOW we use the AM to filter images?

We build small arrays of pixels (3x3 for static images or 3x3x3 for movies) that are AM patterns - M. Del Viva, G. Punzi

- B/W  2^9=512 patterns: 101-010-100, ……. , 111-011-001
- 4 gray level  2^{18}= 256 Kpatterns: 00,00,01-00,01,00-11,00,10 …..
- B/W + time  2^{27}= 128 Mpatterns: 111,000,000 - 000,111,000 - 000,000,000 …



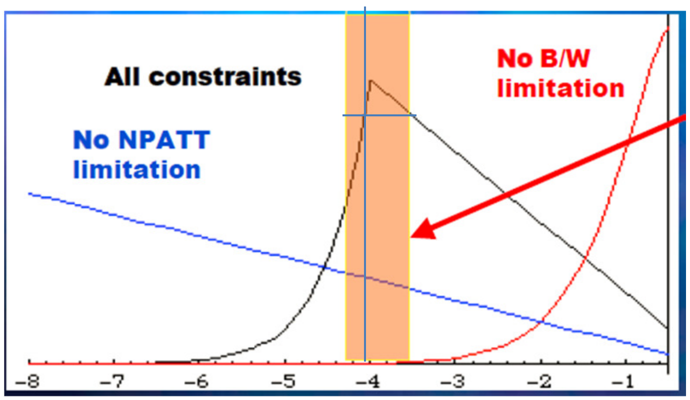
How we select the relevant patterns?

Hypothesis:
The best strategy is to choose the pattern set that maximizes output ENTROPY H under real constraints

information associated to pattern #*i* $I_i = Log_2(p_i)]$

Entropy (H) of the system $H = \sum_i^{NPatt} [-p_i Log_2(p_i)]$

Entropy yield per unit cost and for each pattern: $f(p) = \frac{-p\log(p)}{\max(1/N, p/W)}$

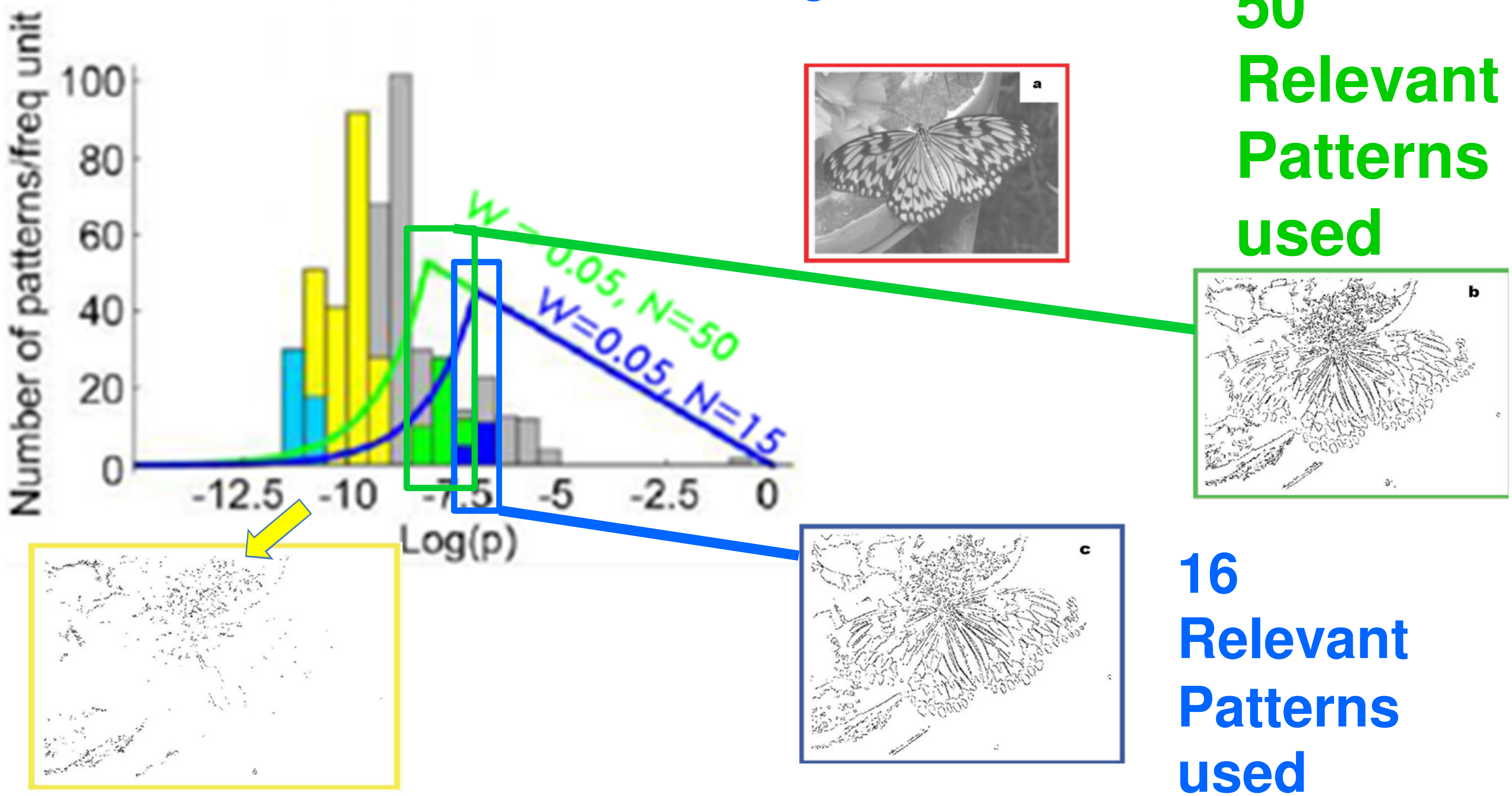


Patterns that are efficient carriers of information given the bandwidth (W) & memory limits (N),

[1] M. Del Viva, G. Punzi, and D. Benedetti. Information and Perception of Meaningful Patterns. PloS one 8.7 (2013): e69154.

Entropy Calculation for “relevant patterns”

Probability distribution of the N_{tot} = 2^9 possible 3x3 square pixel matrices in black/white for natural images

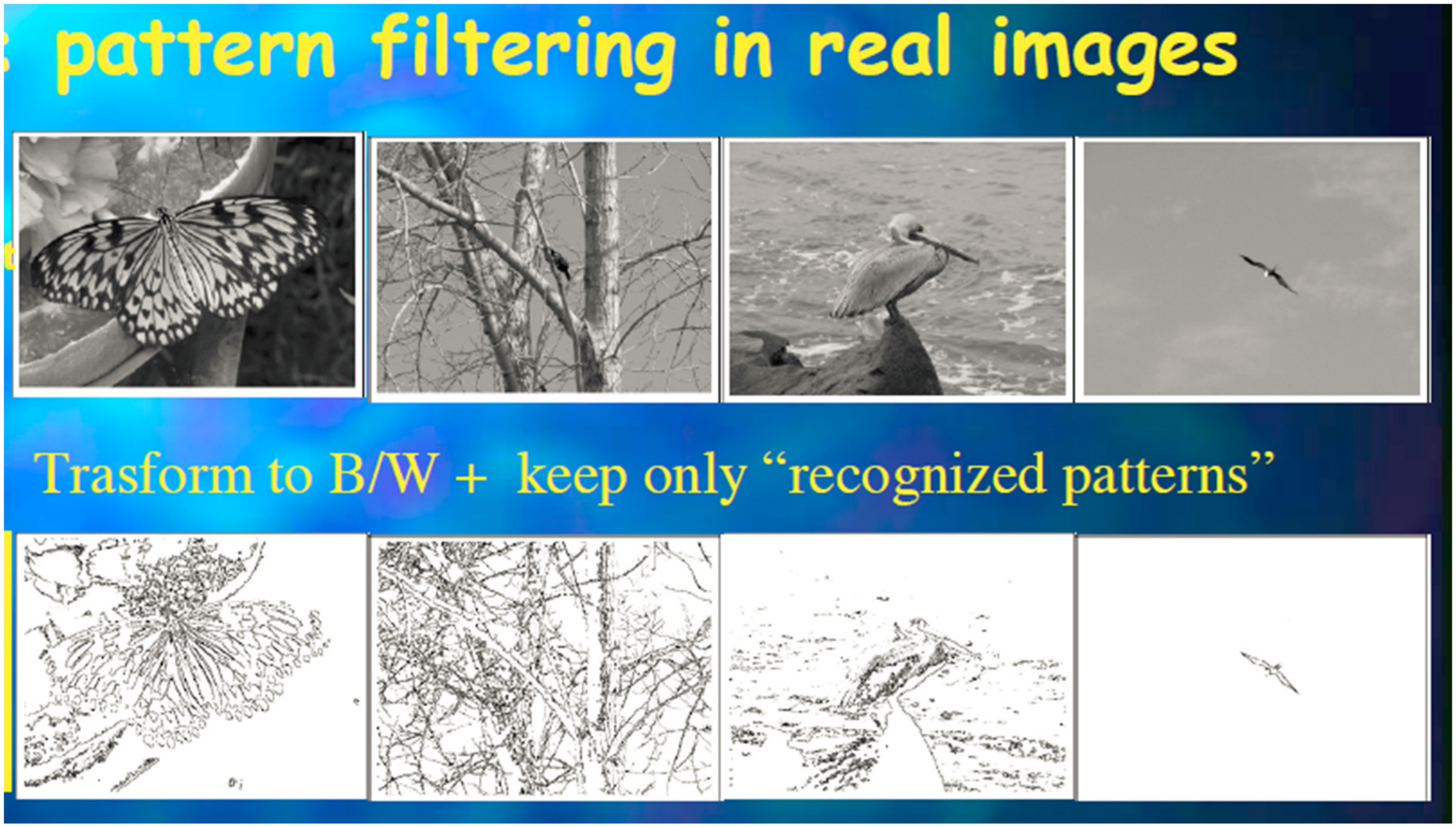


50 Relevant Patterns used

16 Relevant Patterns used

[1]M. Del Viva, G. Punzi, and D. Benedetti. Information and Perception of Meaningful Patterns. PloS one 8.7 (2013): e69

CLEAR to human eyes: tests done with people



Michela Del Viva e Giovanni Punzi (Universita' di Firenze e Universita' di Pisa)

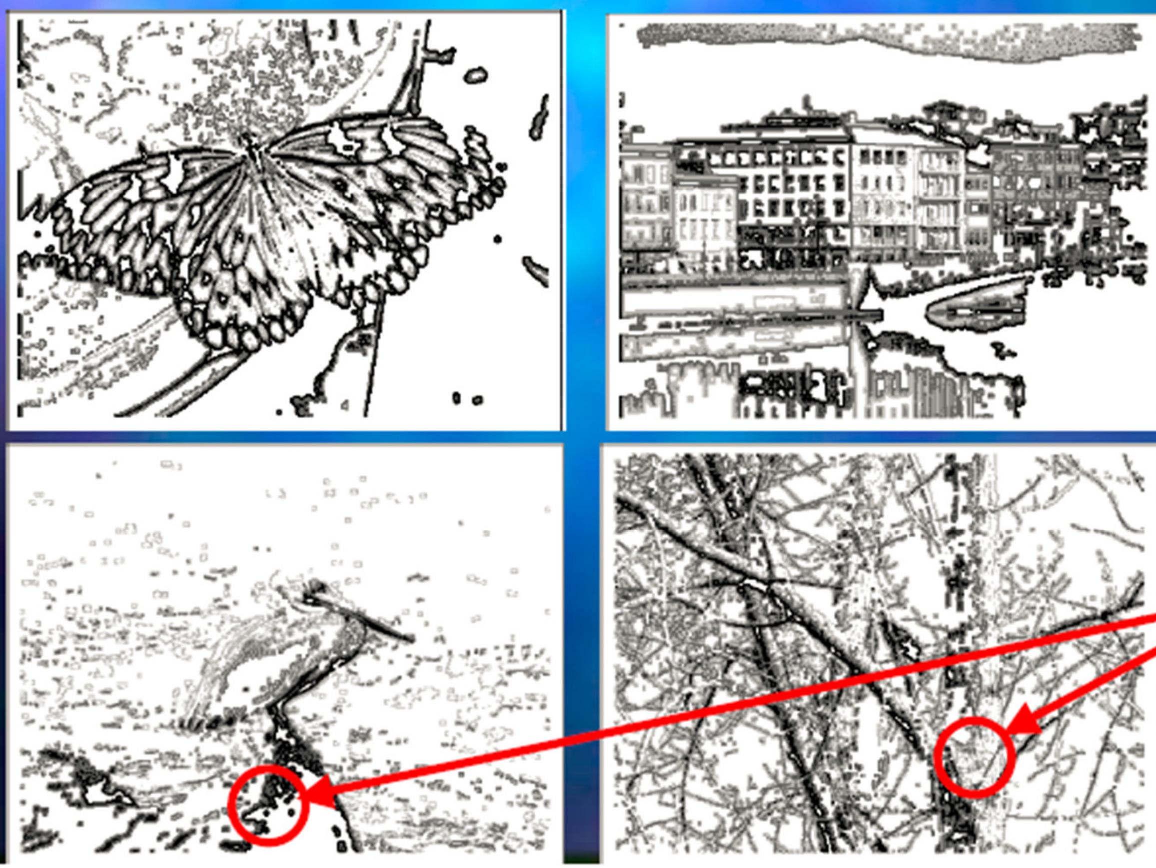
Increasing n° of bits per pixel

The same method can be used with gray-level images, the only difference being the amount of computing effort required.

TEST with 4 gray-levels: number of possible patterns = $2^{18} \approx 262{,}144$. NPATT=2000

MUCH MORE REALISTIC, BUT NEEDS PARALLEL COMPUTING RESOURCES

REAL APPLICATION REQUIRES COLOR + MOTION : LARGE COMPUTING

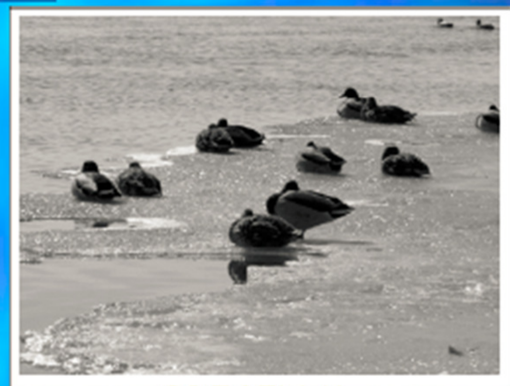


Some patterns represent texture elements rather than edges, and seem to perform useful functions in marking important features

Michela Del Viva e Giovanni Punzi (Universita’ di Firenze e Universita’ di Pisa)

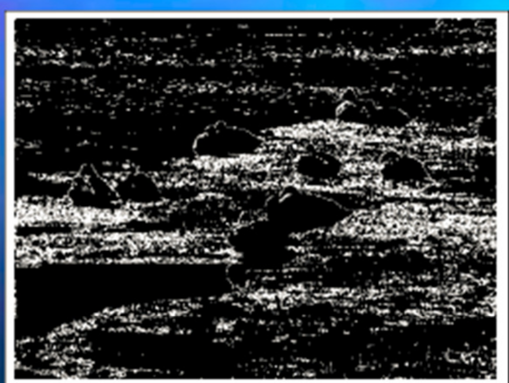
Michela Del Viva e Giovanni Punzi (Universita' di Firenze e Universita' di Pisa)

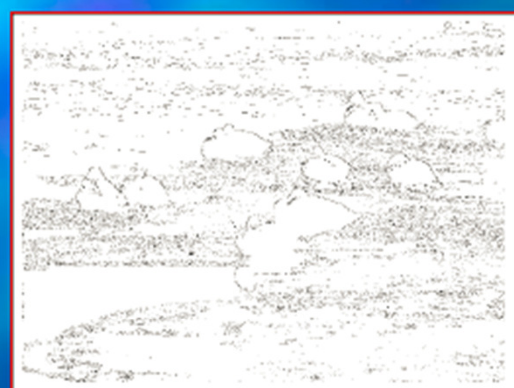
Is it "success by chance"?



ORIGINAL

BLACK DOTS MARK POSITION OF DETECTED PATTERNS





+ FREQUENCY

RECOGNIZED PATTERNS

- FREQUENCY

Only the "right" patterns yield a recognizable image !

Movie Sketches ?

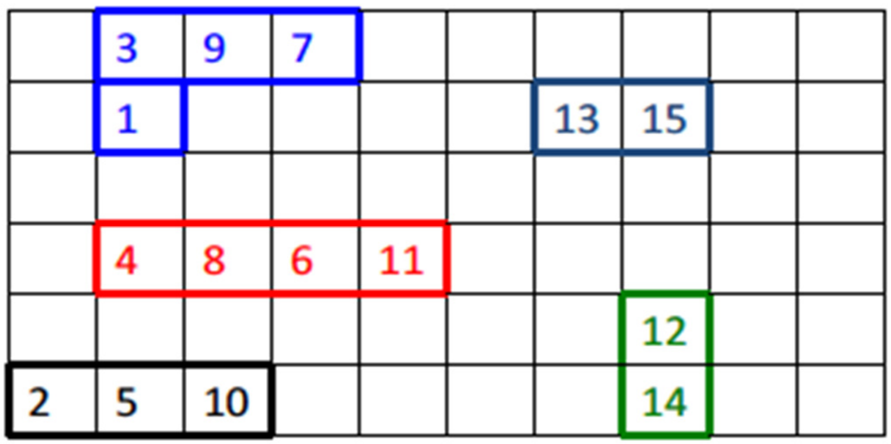
- The filtered images shown above were obtained using Mathematica on a standard CPU and were used to do tests with persons.

- Sketches for movies require a large amount of CPU

- Our goal is to provide them to test if this mechanism works for brain movie processing: is W and N similar in the case of movie? Does the brain use more resources in this case?

Analisys of the image at higher levels: Clustering contiguous pixels over threshold – very suited for FPGAs

The 2-dimensional clustering problem



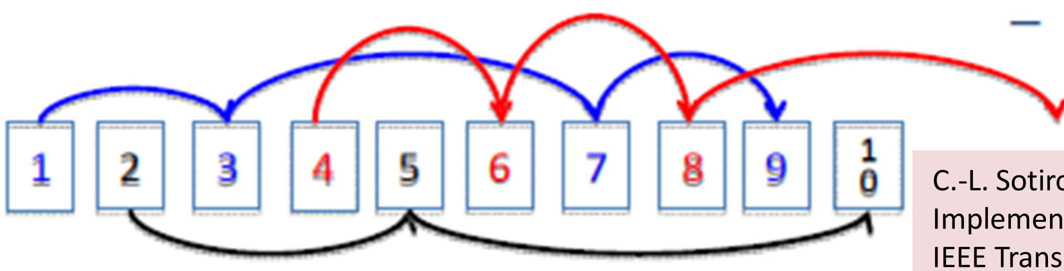
- Clustering is a 2D problem

HIT = 1 image pixel over treshold

1. Associate hits from same cluster
   - Loop over hit list
   - Time increases with occoupancy
   - Non linear execution time
2. Calculate cluster properties
   - e.g. center, size, shape …

- Goal: execution time linear with number of hits (usually it $\propto$ to N_{Hits}^2)
  - Not a limiting factor even at highest occupancies

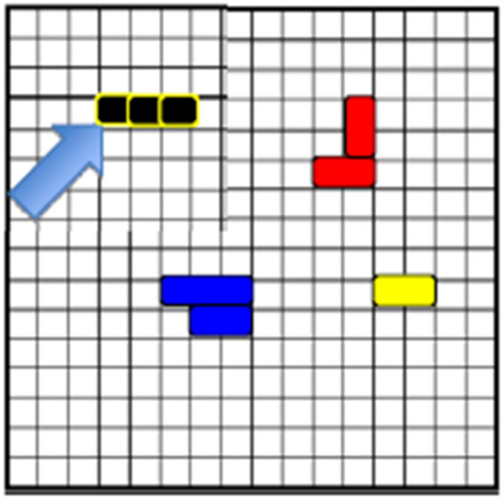
Loop over list of hits



C.-L. Sotiropoulou et al. “A Multi-Core FPGA-based 2D-Clustering Implementation for Real-Time Image Processing”, Nuclear Science, IEEE Transactions on, 61 no. 6, (2014) 3599–3606.

The Algorithm Working Principle

FPGA replica of pixel matrix



Eta direction -->

1st phase:

The image HIT array

Replicate it in a hardware matrix.

The matrix identifies hits in the same cluster (local connections).

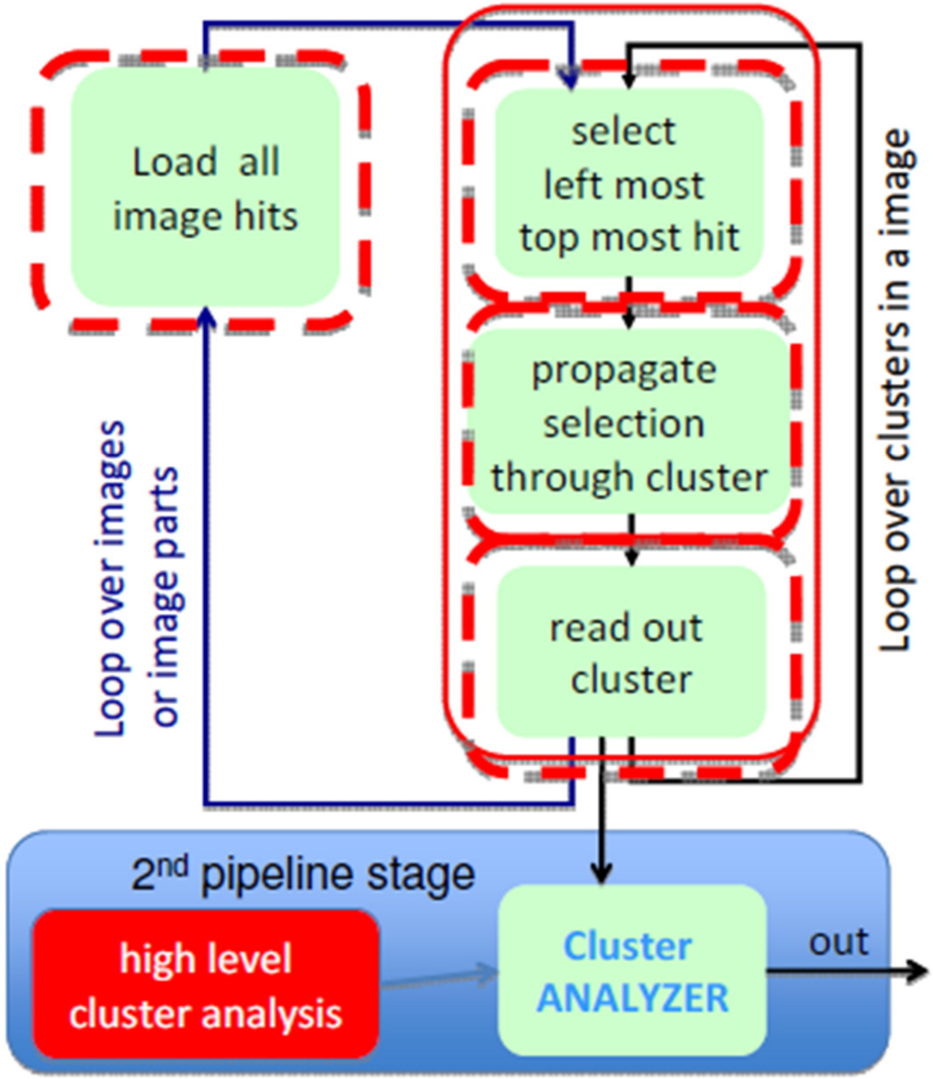
2nd phase:

Hits in cluster are analyzed

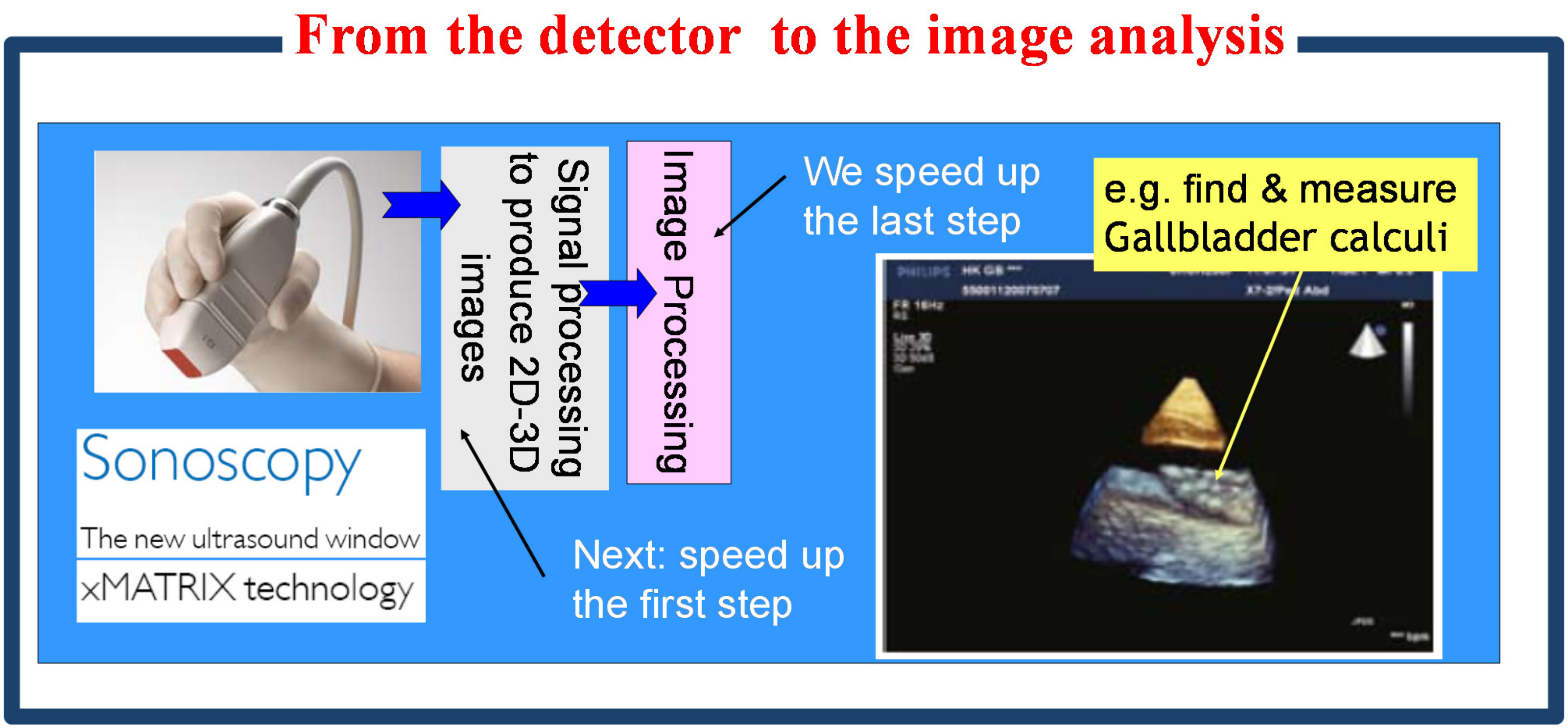
Flexibility to choose algorithm!

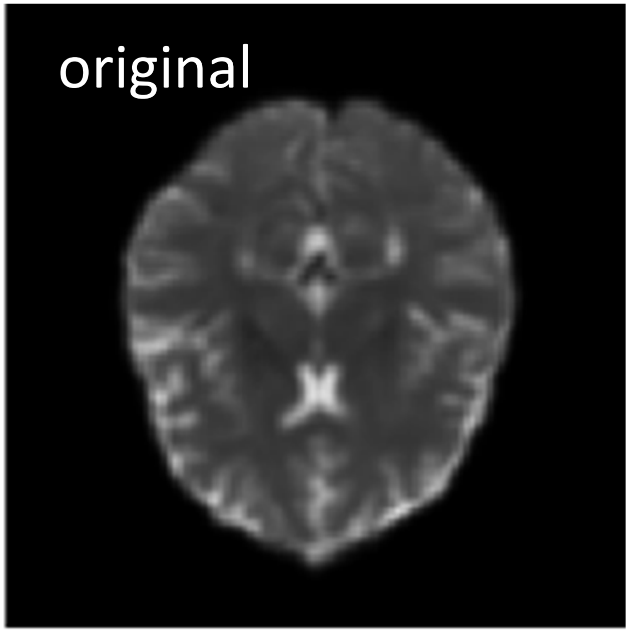
Cluster Finding logic:

Hit associated into clusters

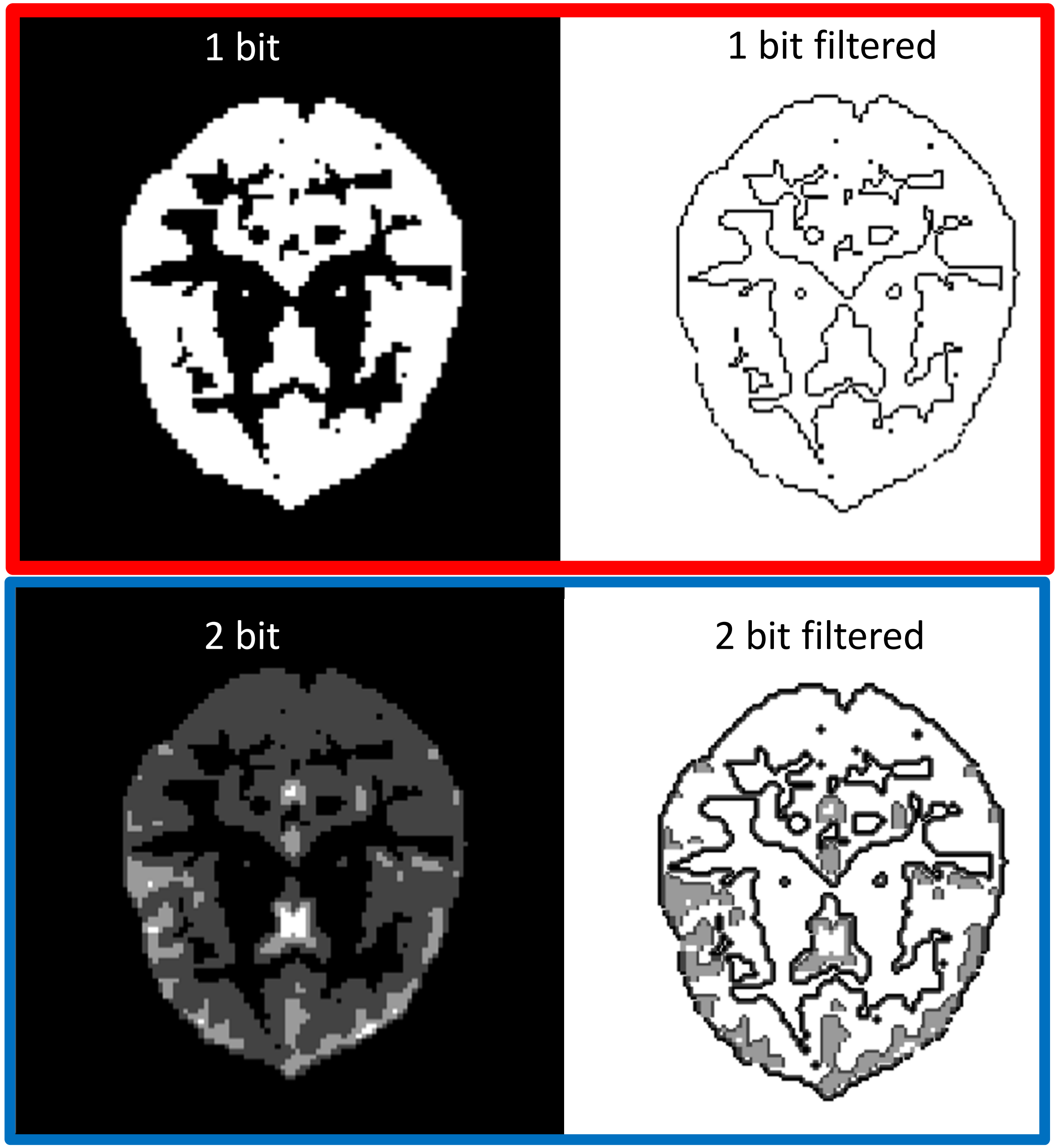


Medical imaging: is there a field where decreasing the info (working on edges) could be an advantage?





EXAMPLES:



Movement of the object under diagnosis could be an issue?

Could be interesting to **look only to contours**, but **reconstruct the movement** of the object?

Conclusion – Collaboration?

If we find something that is interesting: can we collaborate?

- We provide the hardware
- You provide the use case and the images
- We process them
- You guide us judging the results

Grazie .....