Fast Radar Image Change Detection by Matlab Embedded Routines

Hans Hellsten
Saab EDS, April 2016

Overview

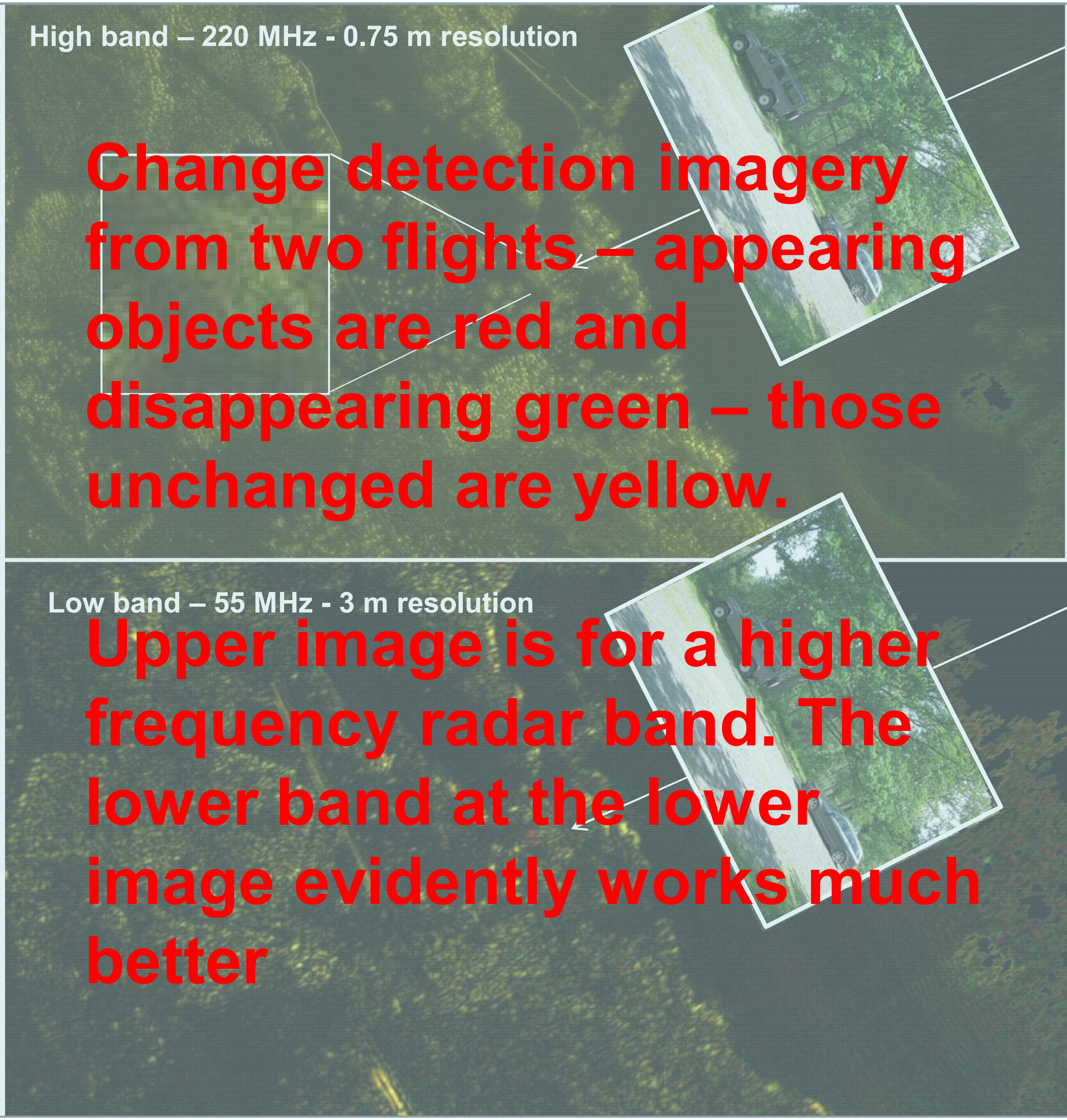
- Saab has developed a novel type of low frequency airborne radar – CARABAS - which detects objects even if camouflaged, e.g. by trees screening.

- The radar images are temporally stable (i.e. do not change from one registration to another) so detection can be achieved by “change”. For instance a vehicle being parked and camouflaged in between flights will be detected.

- This talk addresses the specific probability method for change detection: It assesses the probability of the small natural changes and compares with a change caused by an appearing object. Thereby the probability that a change means that an object has appeared in any particular position is obtained. Targets become found and listed in order of decreasing probability.

- A challenge is the computation time in obtaining the statistics for natural changes since images are large (millions of pixels) and must be compared with each other. Nevertheless using Matlab embedded routines a method were found to reduce the computation time by one to two orders of magnitude, thus making the approach practical.

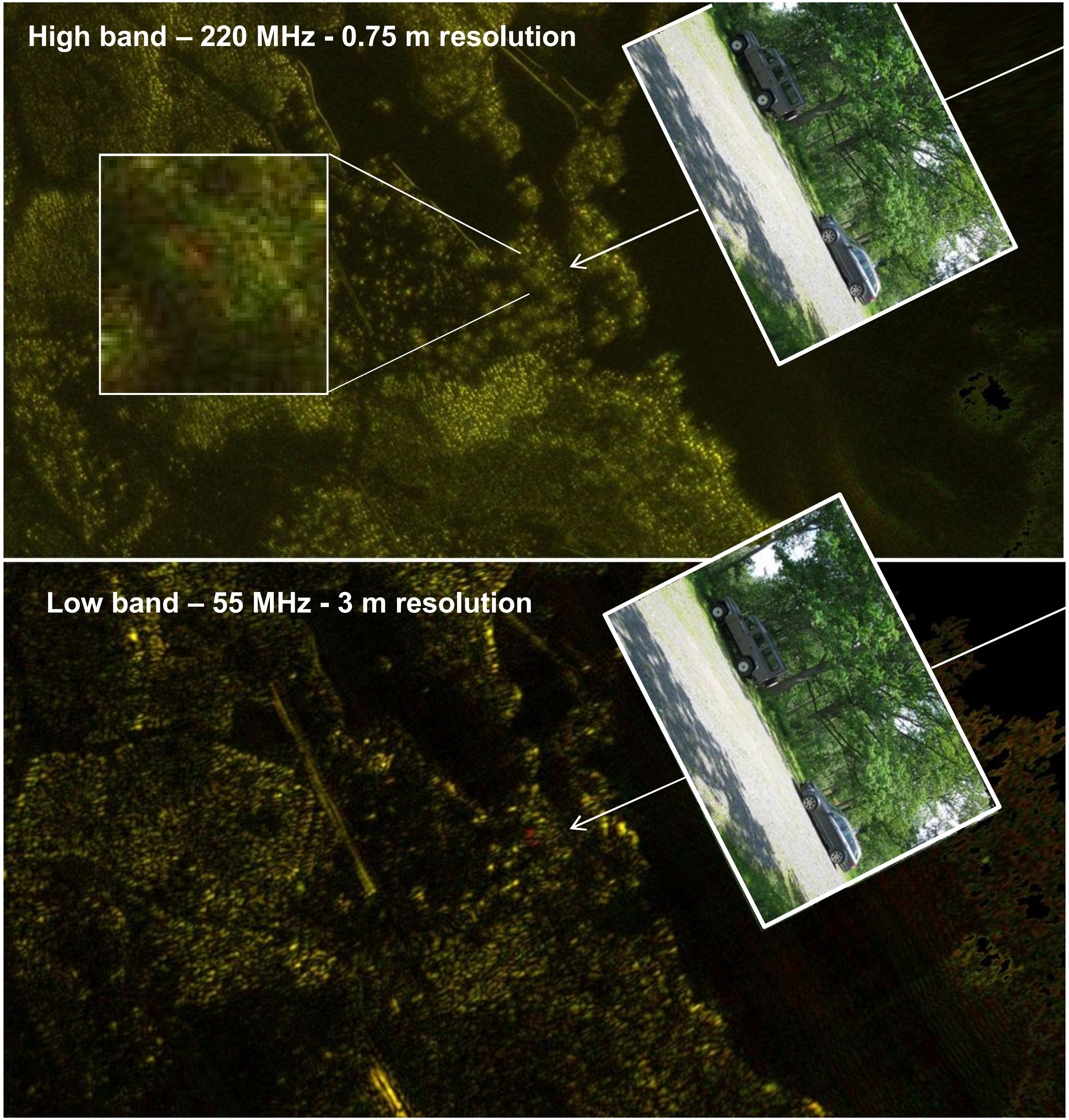
- It is suggested that the method practiced has a wider application speeding up any routine which requires manipulating functions defined over large data sets – typically images.



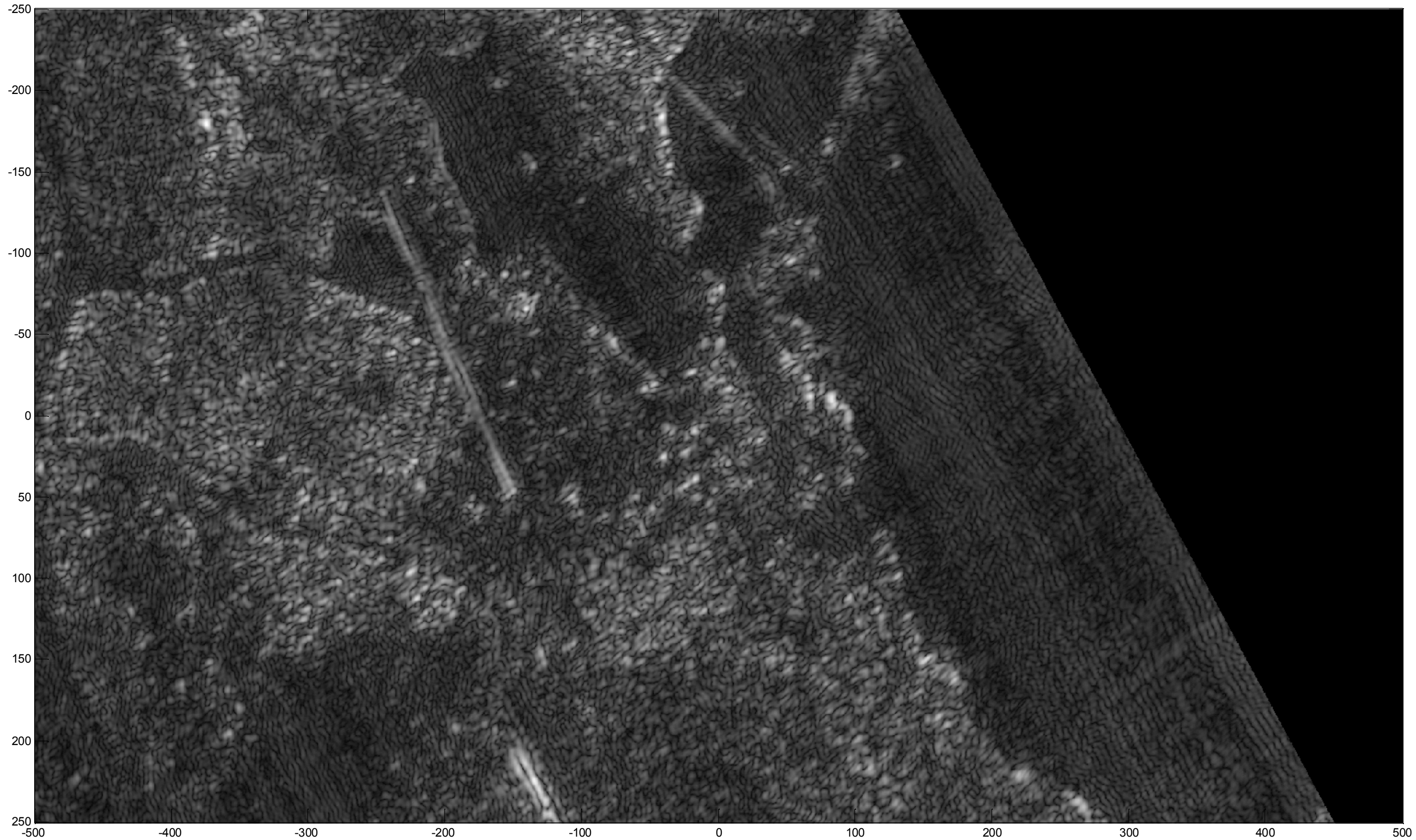
Change detection imagery from two flights – appearing objects are red and disappearing green – those unchanged are yellow.

Upper image is for a higher frequency radar band. The lower band at the lower image evidently works much better

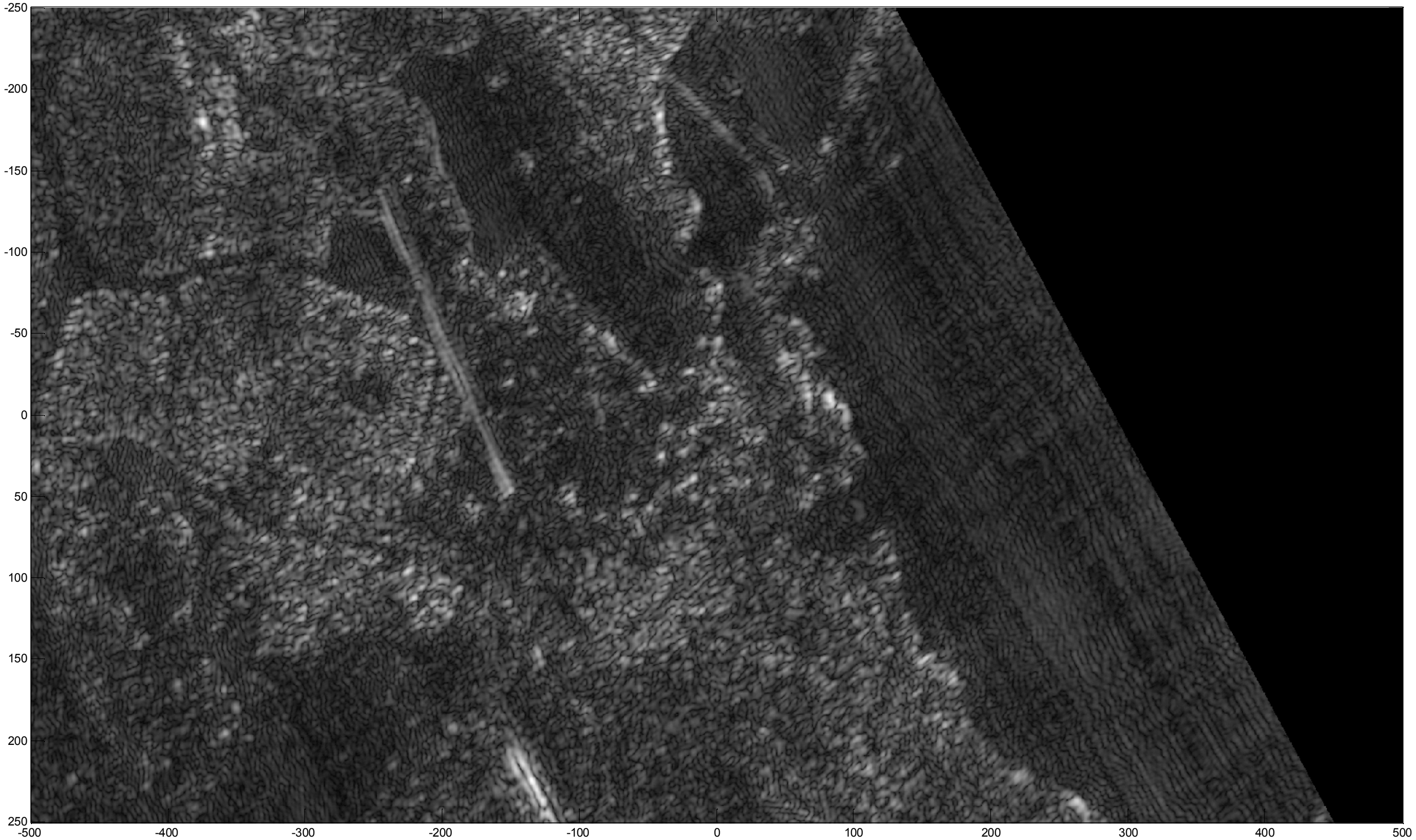
SAAB


High band – 220 MHz - 0.75 m resolution
Low band – 55 MHz - 3 m resolution

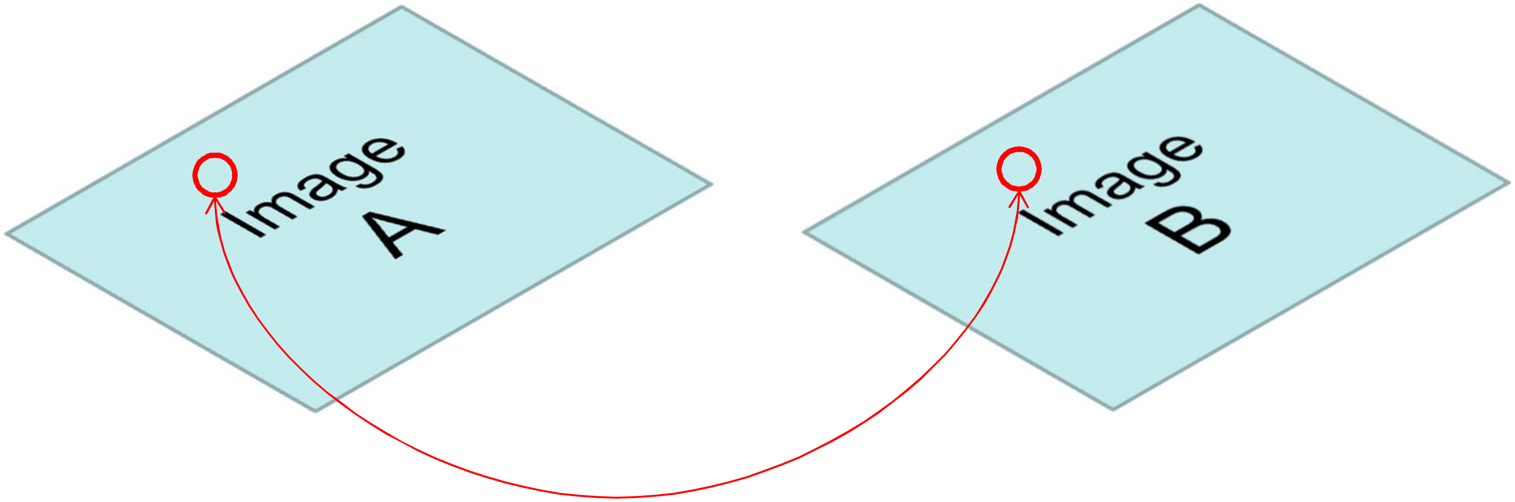
Lower band image from first flight



Lower band image from <u>second</u> flight



Statistical Method

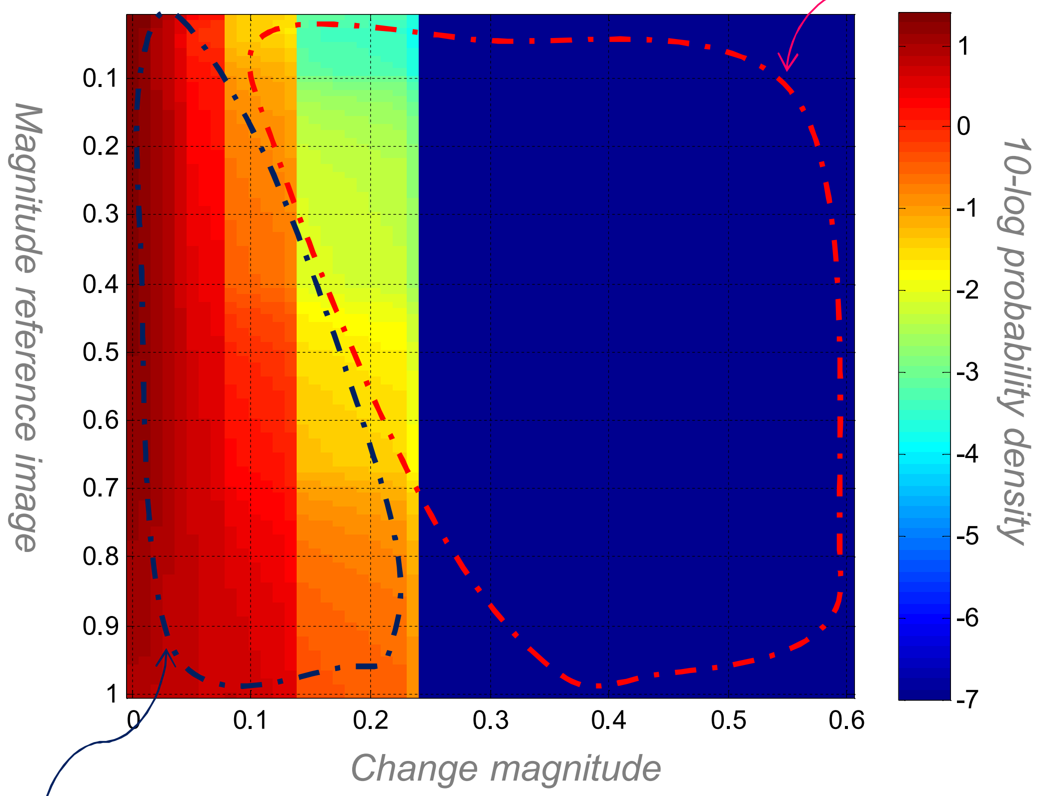


1. Asses change statics by counting number of cases when for a given amplitude in A there is a given amplitude change between A and B
2. Divide by the calculated probability that a target causes such a change thus obtaining probability that for ach image point there is a target
3. Find the most probable target candidate and exclude
4. Redo process until there are no more probable targets

The computational burden in step 1 can be very large since it typically means testing each image pixel whether it has certain image amplitude and change amplitude (within some tolerance). It thus has to assess all image pixels (millions) several hundred times. According to 3 this must be redone for every target in the image.

Change statistics with no appearing or disappearing objects

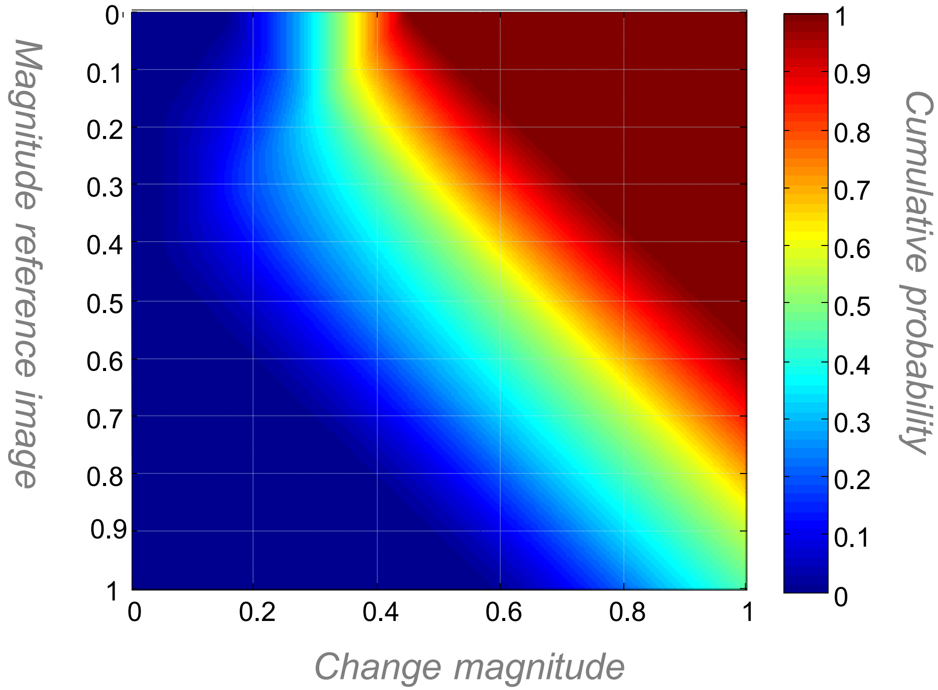
Changes in this area will be detectable



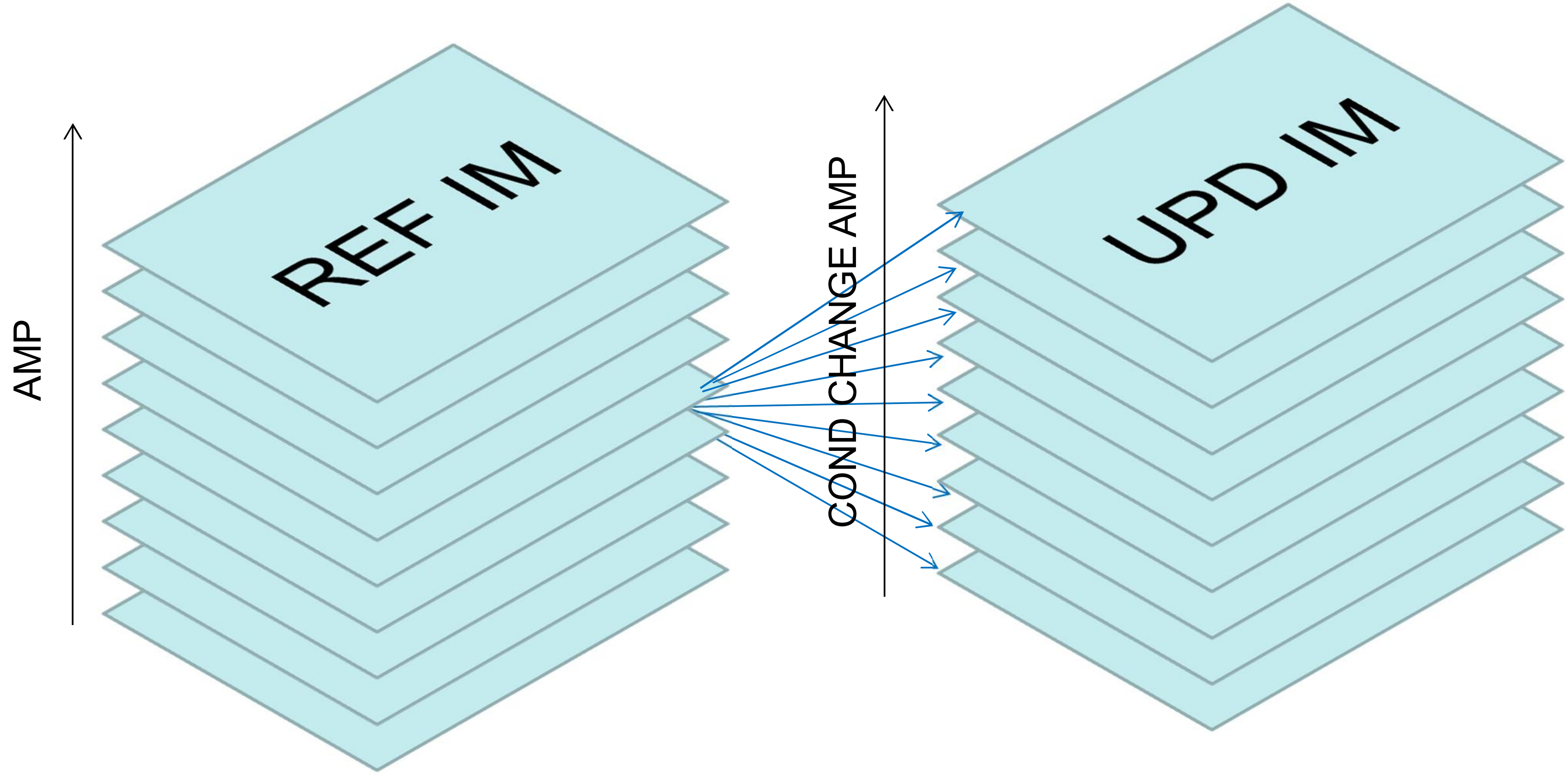
NOTE: The stronger the intensity in the reference image the more uncertain the amplitude in the update

Target model

$$u_{total}(x) = u_{object}(x) + u_{terrain}(x)$$



Fast method of obtaining statistics



Define logical masks: DSUPPOA=AMP>=BILD;
Statistics: sum(DSUPPOA)

GENERALIZATION OF FAST METHOD

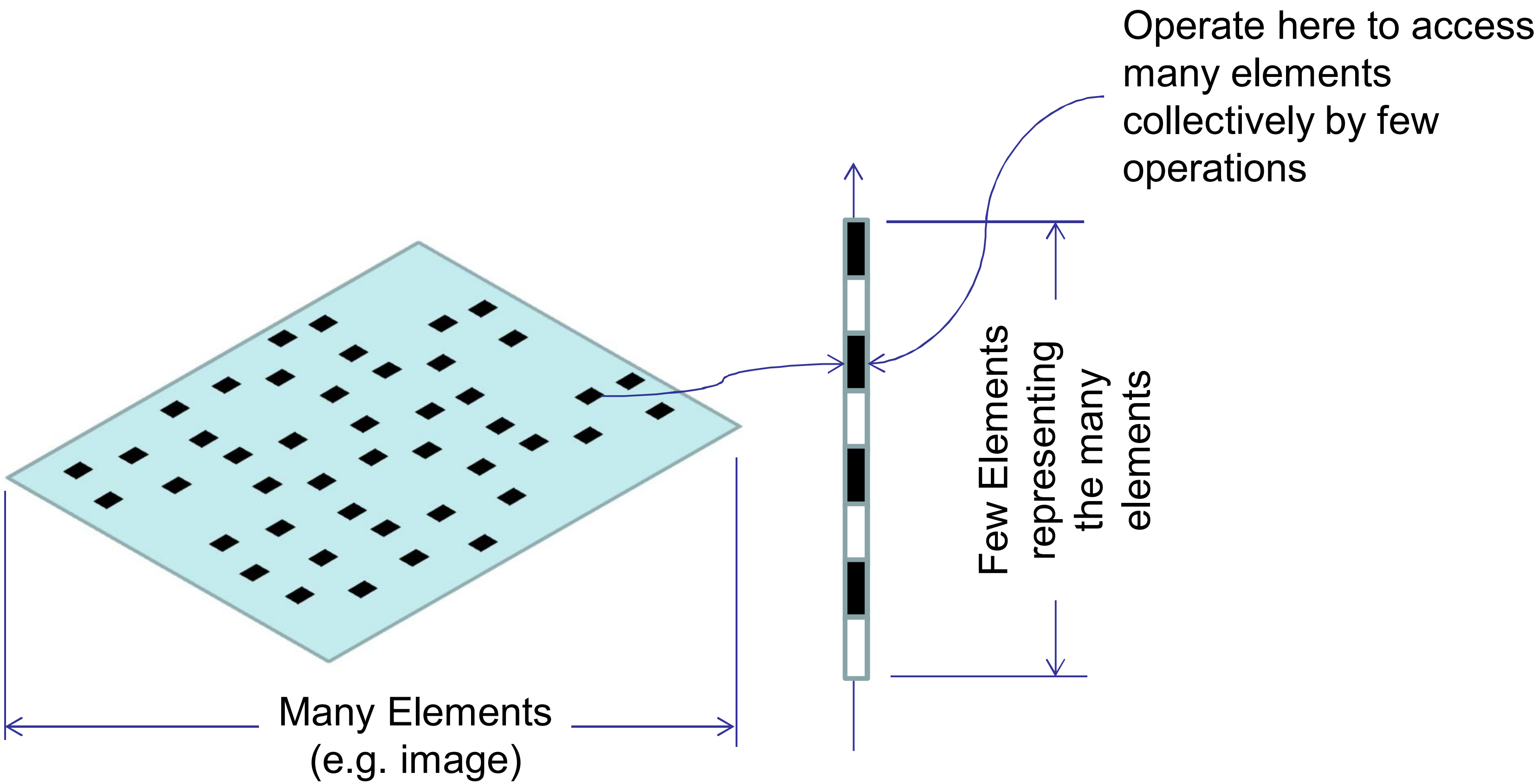
- Consider any function f(x) defined on any set X, e.g. the pixels in an image.

- Consider any operation which for each value f=const of the function perform some task specific to that value. Examples can be counting the number of elements in X holding this value, or it can be to change the value according to some rule (e.g. multiplication), or to sum the value for the subset of points holding the value (integration).

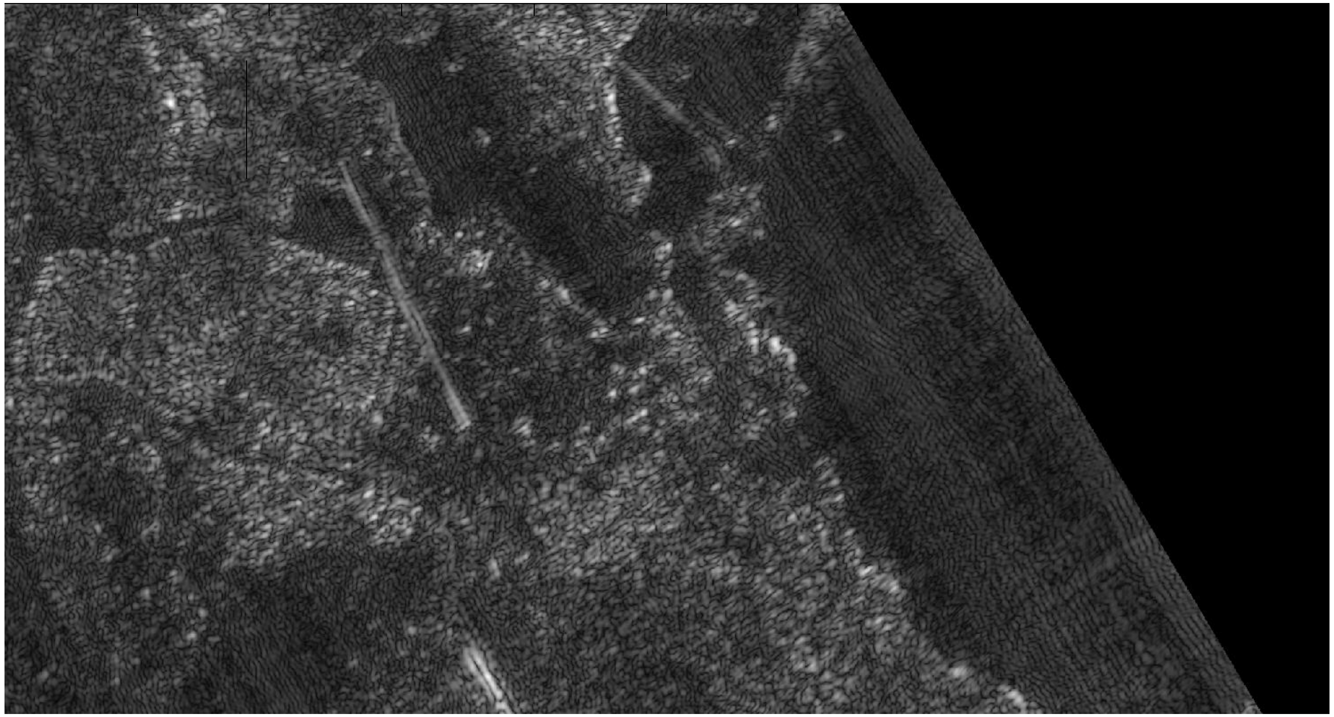
- Common for all these situations is that it is not required to loop through the set but that the set can segmented as f^{-1}(const). The operation can be performed as one single computational step for all elements in f^{-1}(const) while the overall solution is attained by recombining the segments across all possible functional values.

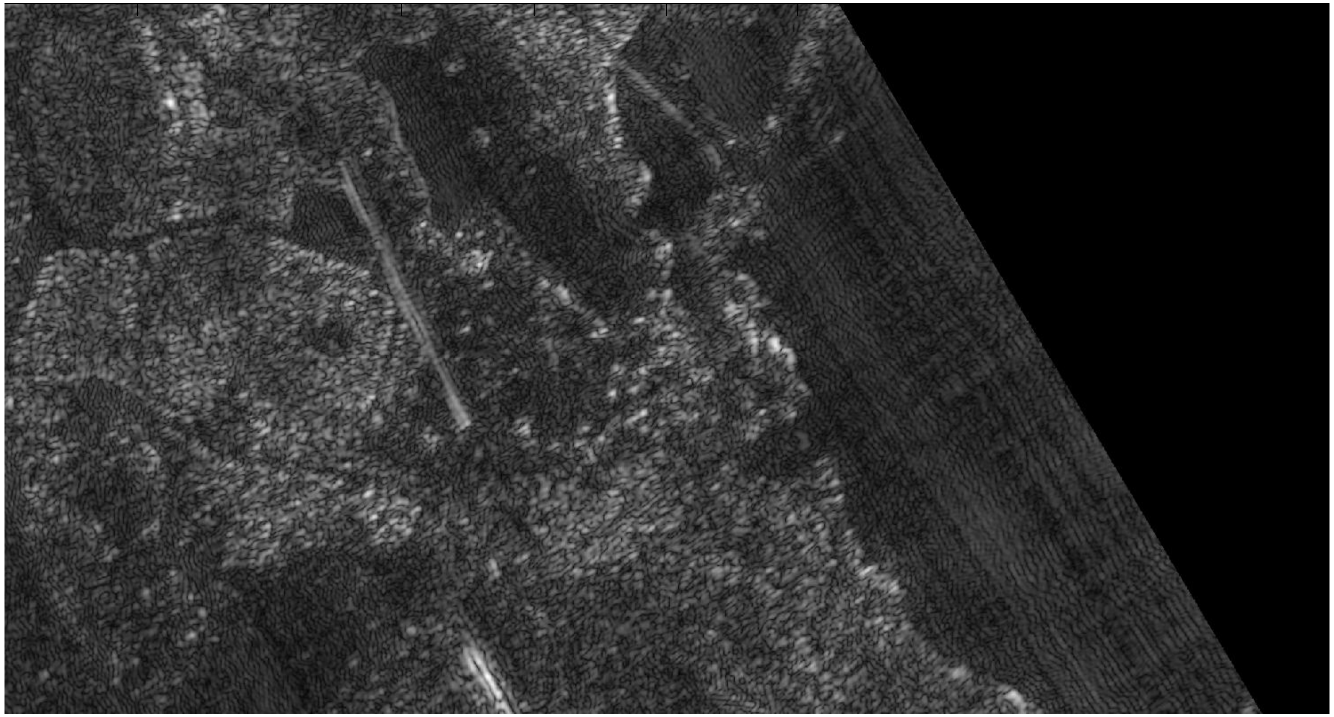
- The gain in computational effort is that functional values (within a pre-defined tolerance) are common to many elements.

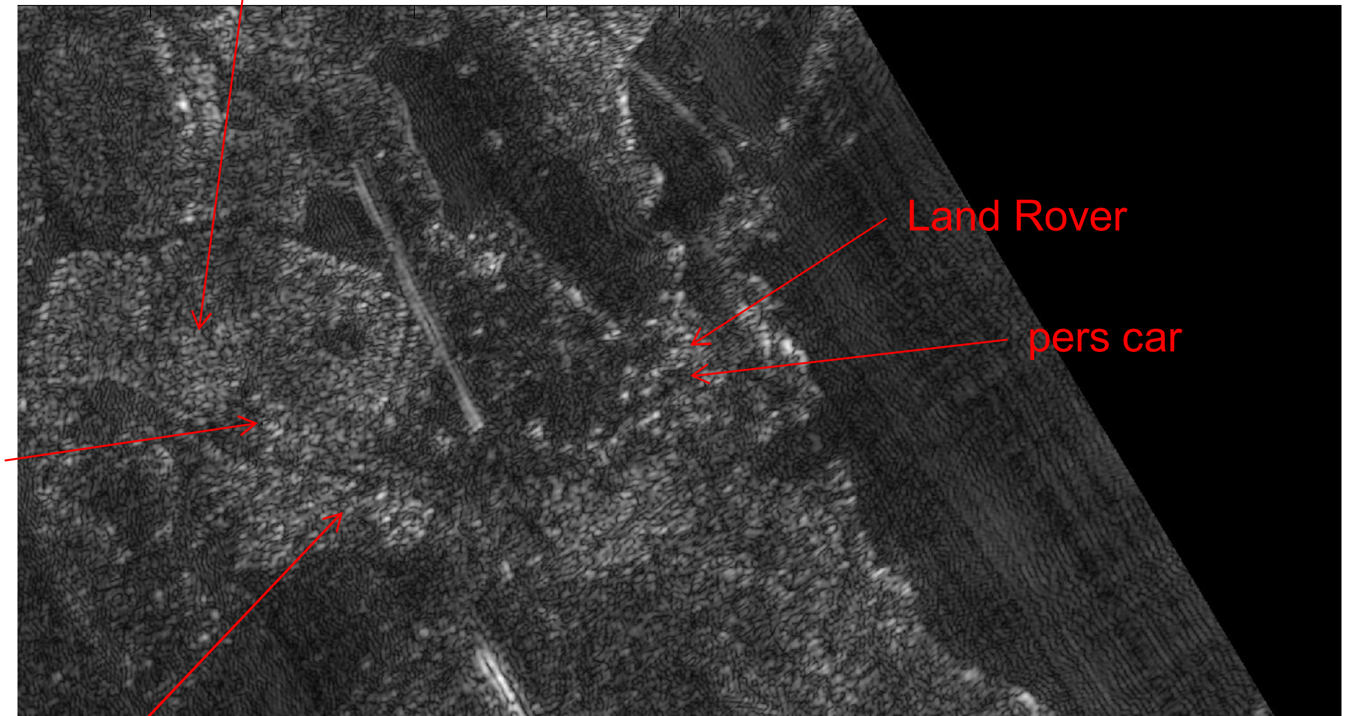
GENERALIZATION OF FAST METHOD



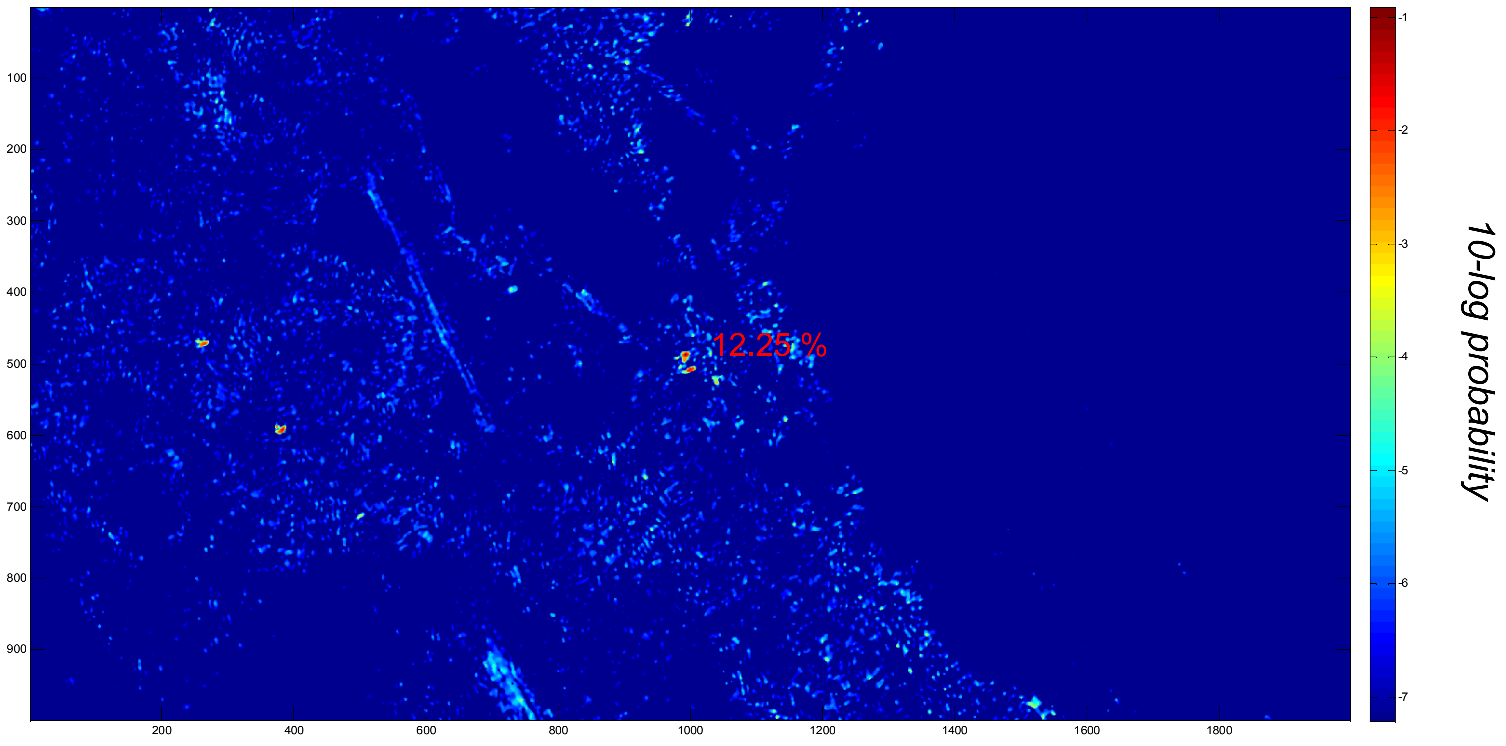
Performance of Method
Experiment with artificial insertion of three additional weak targets



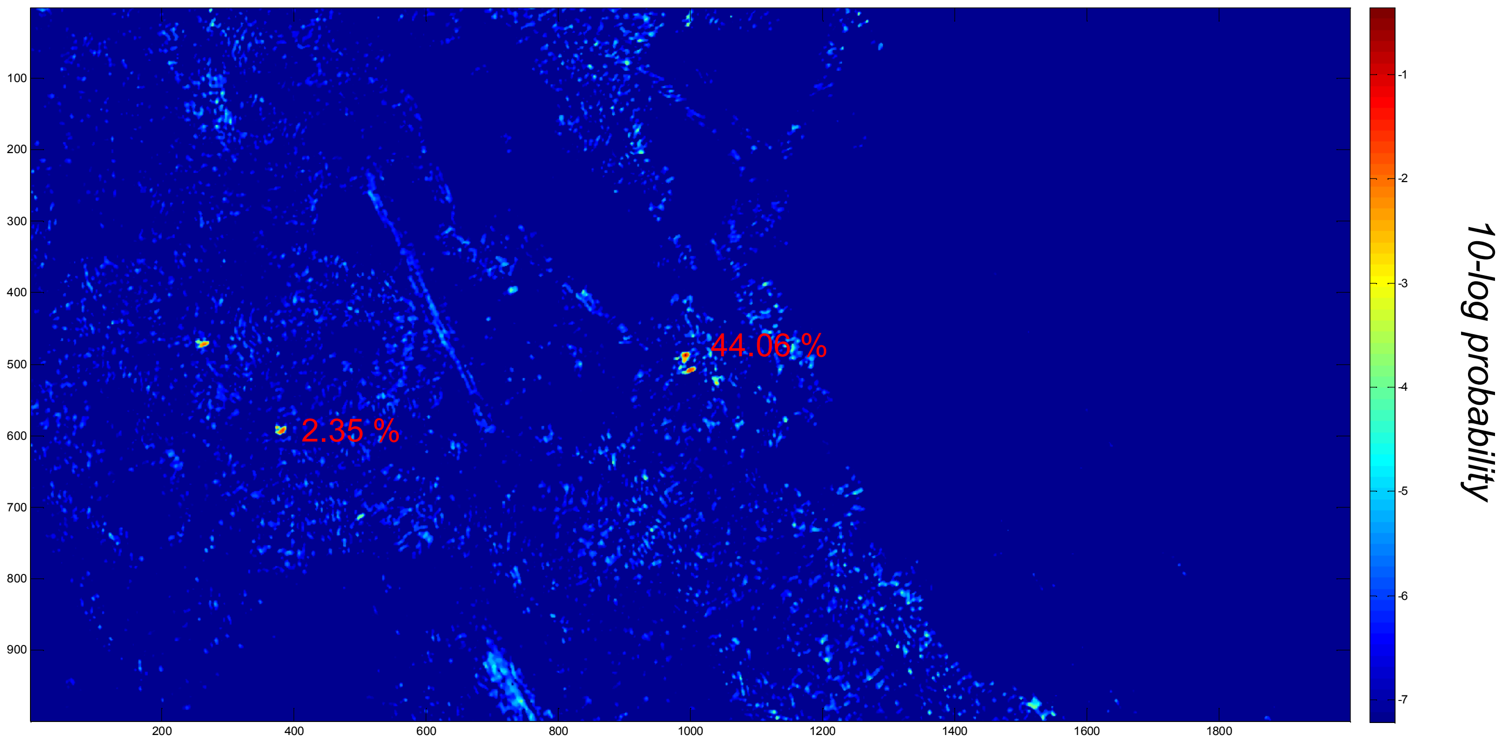



Artificial: 0.75 x pers car
Land Rover
pers car
Artificial:
pers car
Artificial:
0.5 x pers car

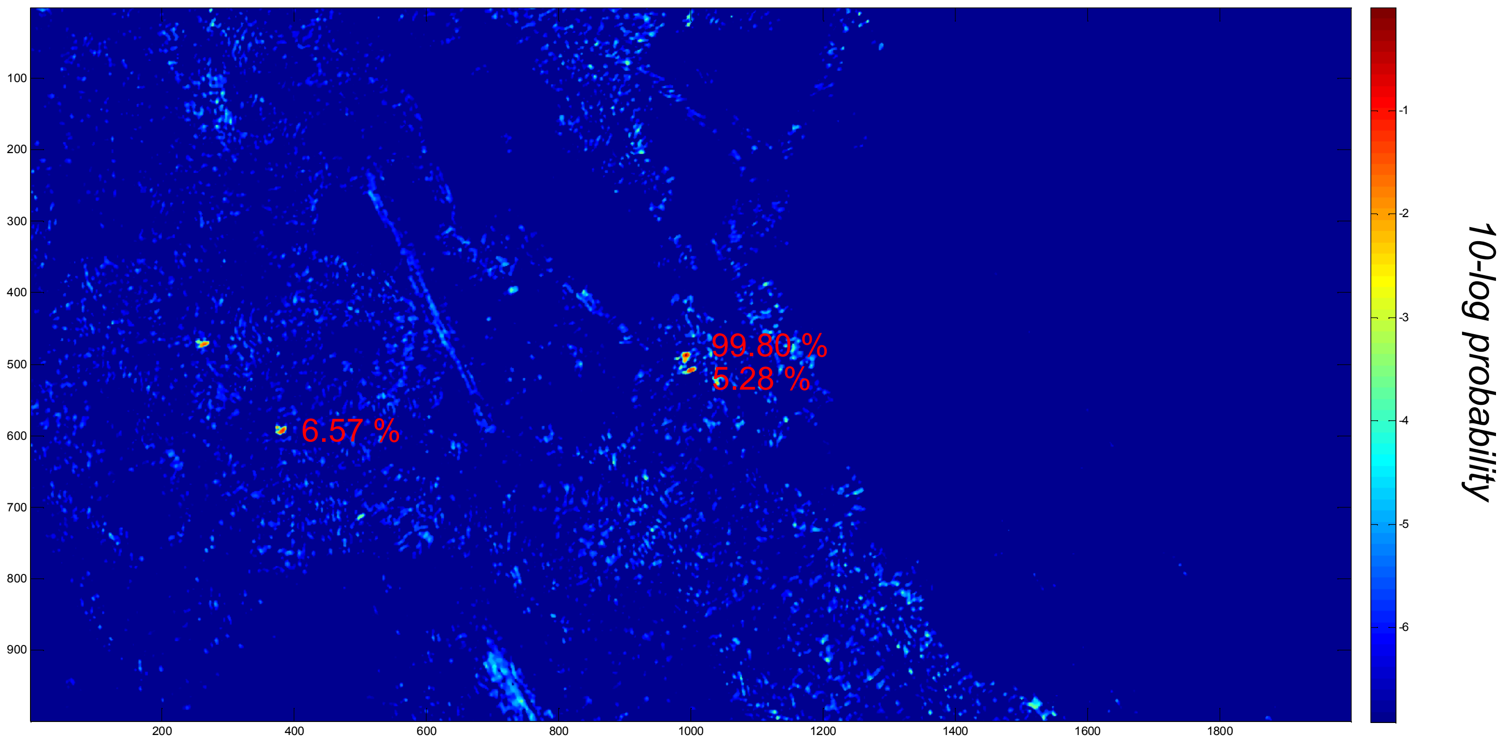
Probability image – 0 target hypothesis



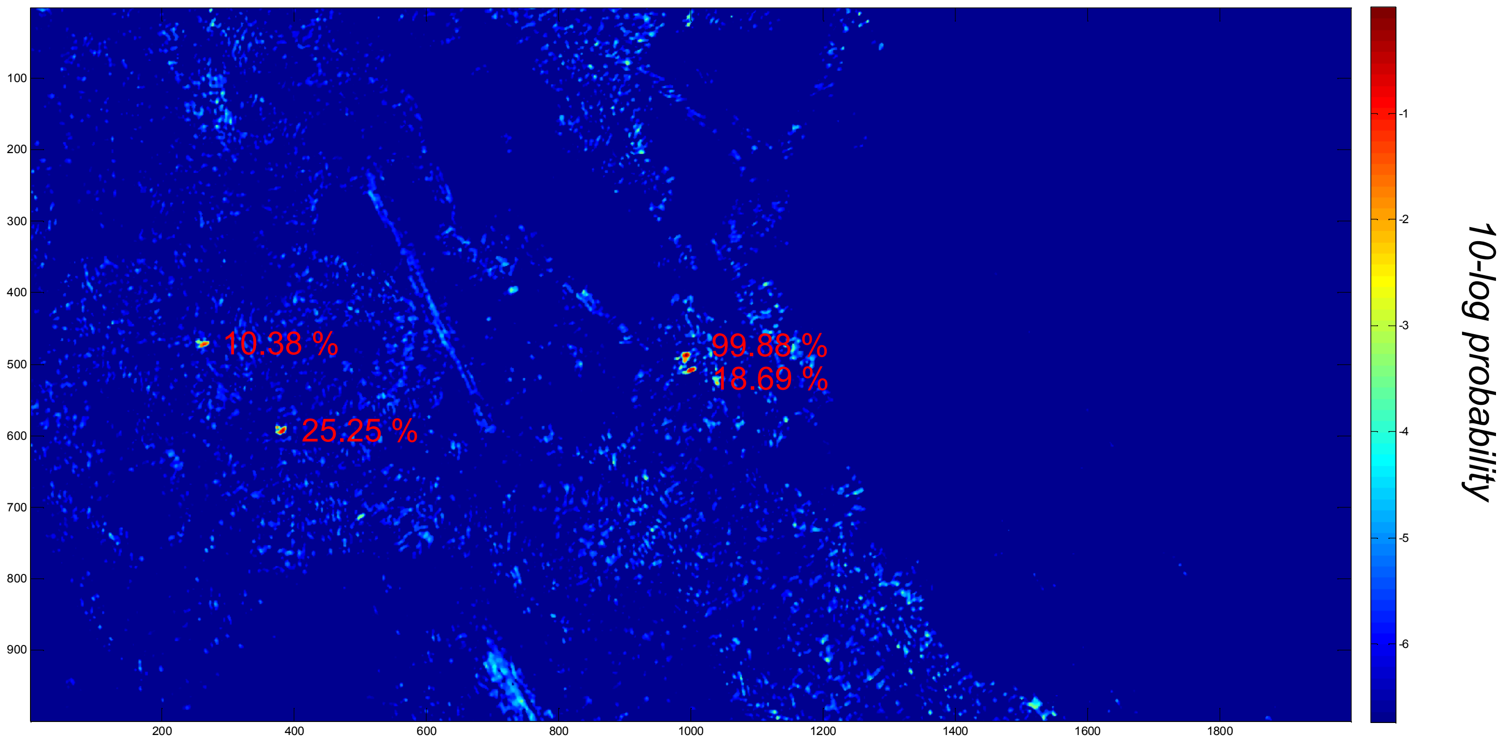
Probability image – 1 target hypothesis



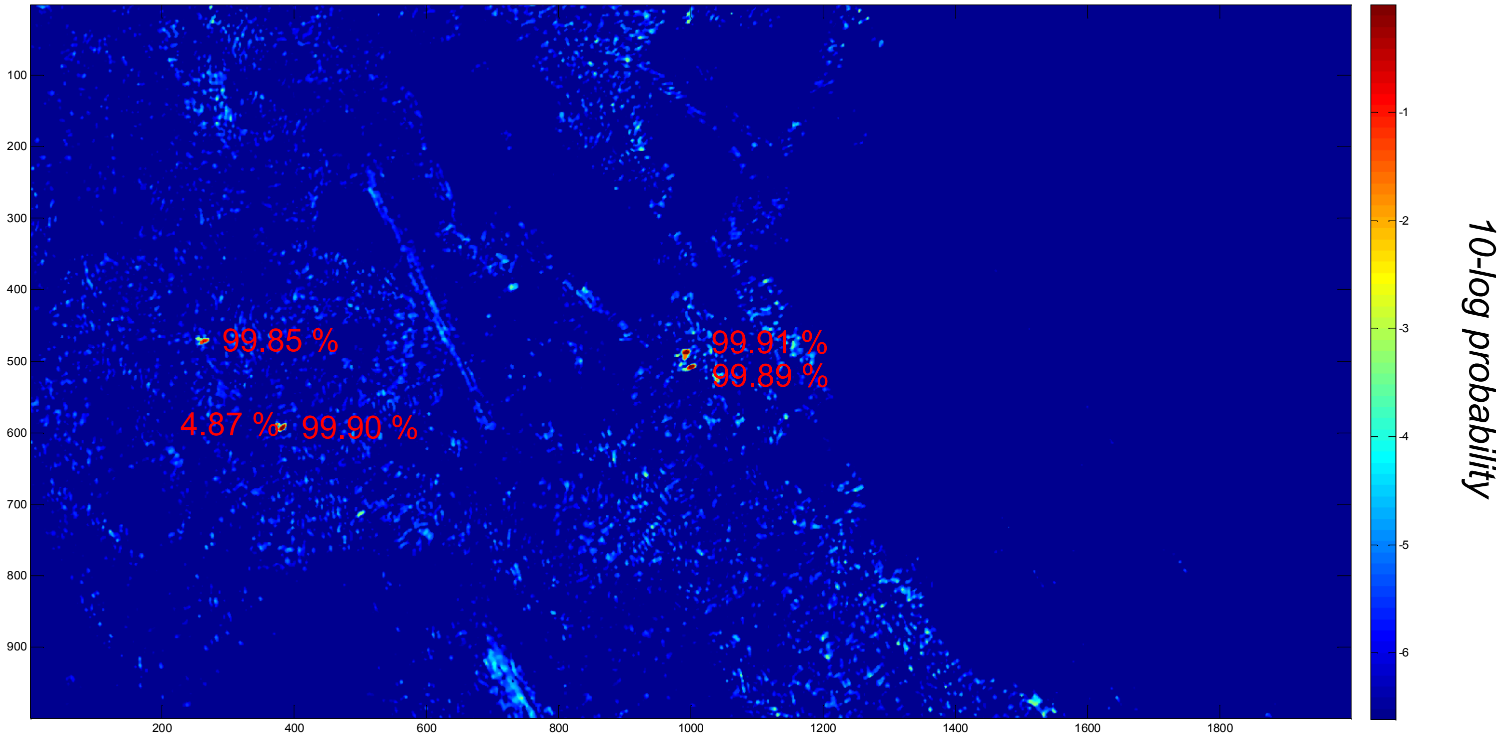
Probability image – 2 target hypothesis



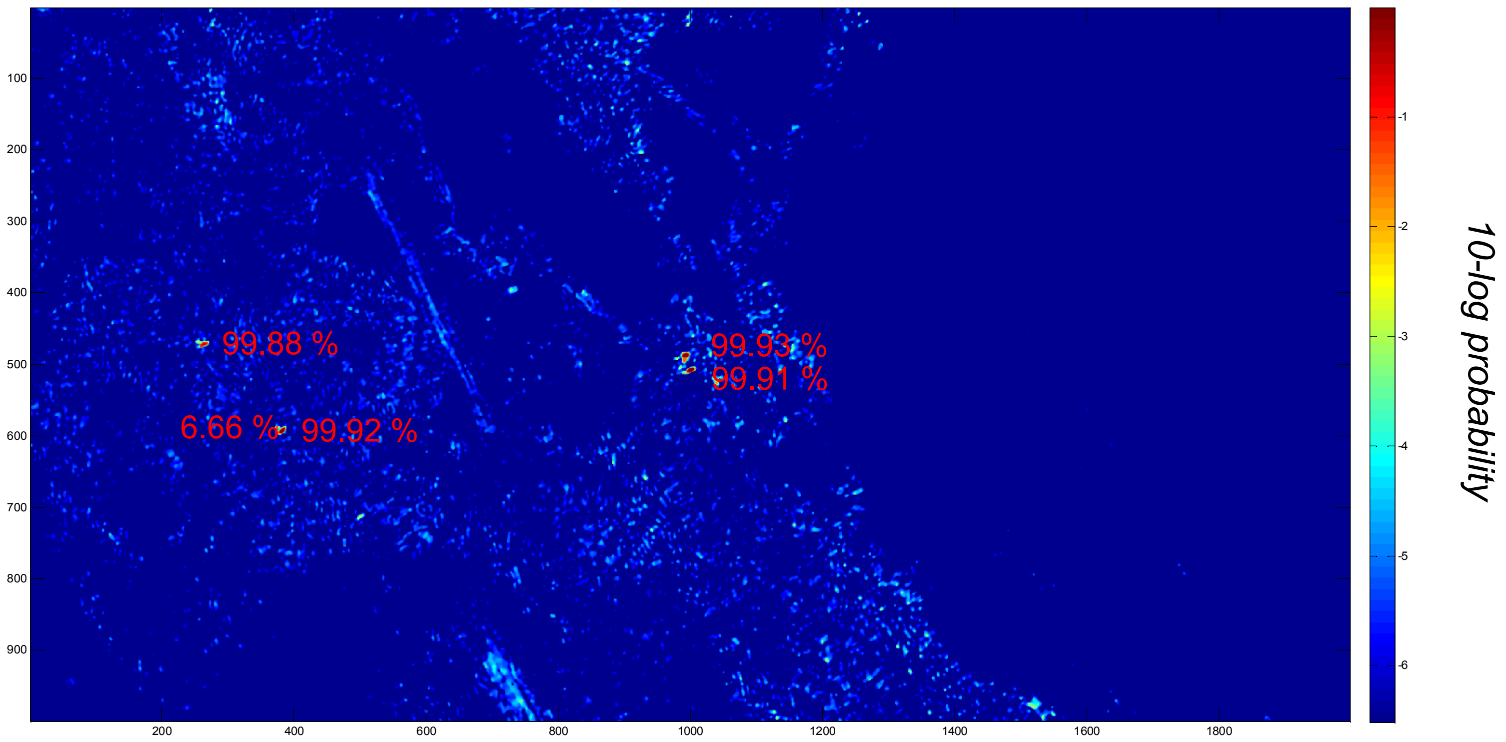
Probability image – 3 target hypothesis



Probability image – 4 target hypothesis



Probability image – 5 target hypothesis



Conclusion: 4 targets found- the weakest not !