Conquering the extremes

Three new algorithms of extreme value computation for NWP data visualization

EGOWS '18 – ECMWF Reading, UK

Sören Kalesse, Deutscher Wetterdienst

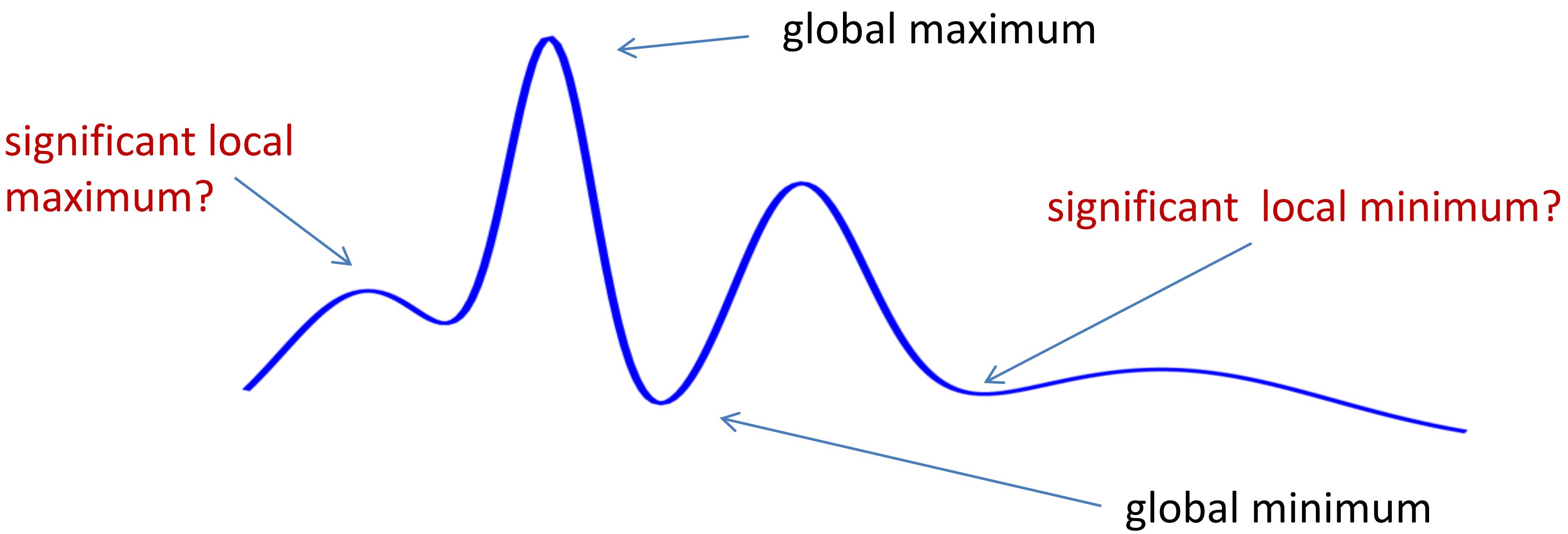
Outline

- Extreme values
  - in general
  - and NWP
  - in NinJo
- Three new approaches
  - commons
  - The “Cowpat” algorithm
  - The “Firefly” algorithm
  - The “Watershed” algorithm
- Conclusions and Future aspects

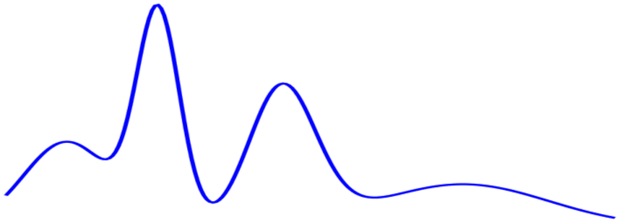
Extreme value computation?

clustering and optimization problem:

- find the *<u>significant</u>* local and global maxima and minima of a (2D) function



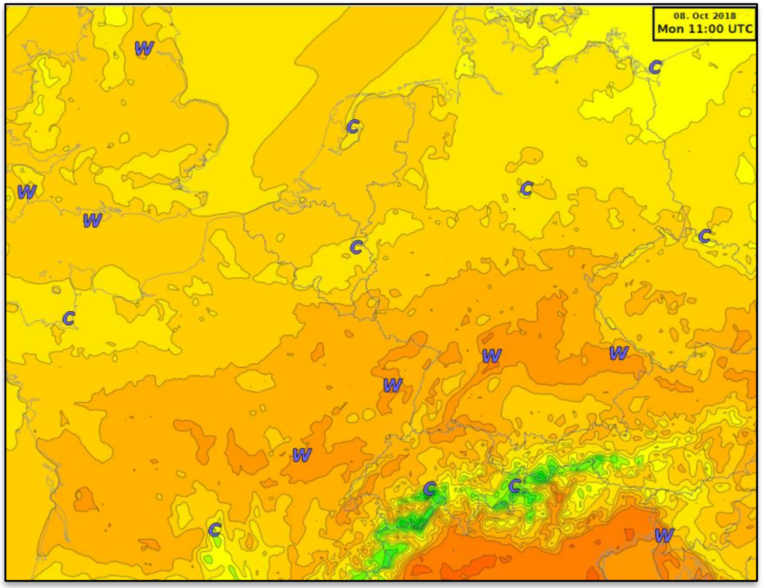
Extremes of NWP data



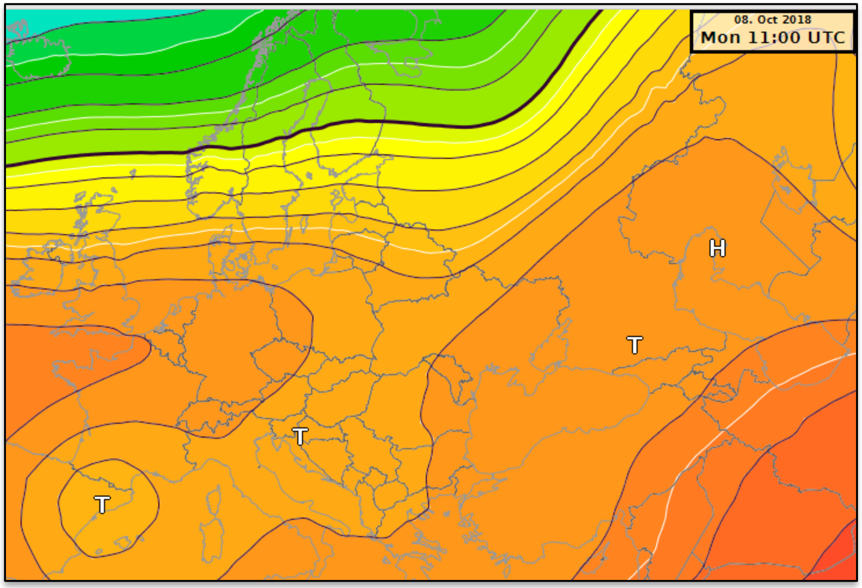
for NWP model data:

- data is assumed continuous, but presented to us as discrete value field
- distribution of extreme values is extremely multi-modal
- depends highly on the type of data
  - e.g.: temperature vs. pressure, near-ground vs. near tropopause, …
- topography as “hidden” variable in near-ground fields carries “clutter”

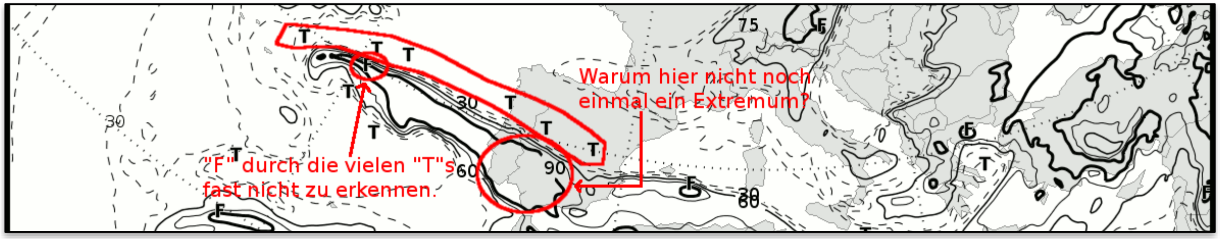
Extreme values in NinJo



Temperature, 2m, ICON Regional



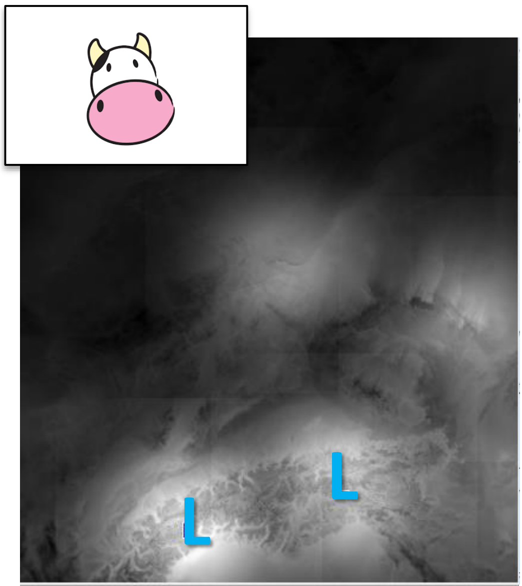
GPDAM, 200 hPa, ICON Globe



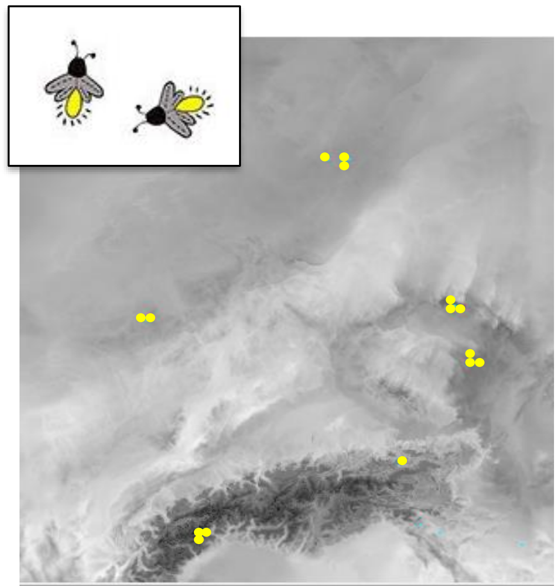
Quality of calculation results has not been satisfying

Three new approaches

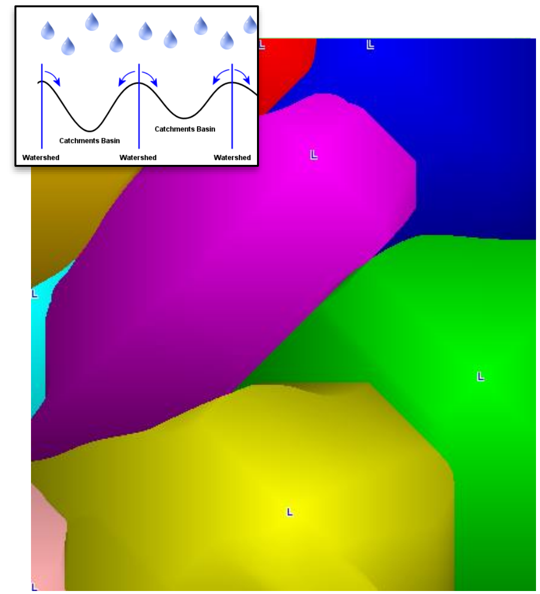
“Cow pat”



“Firefly”

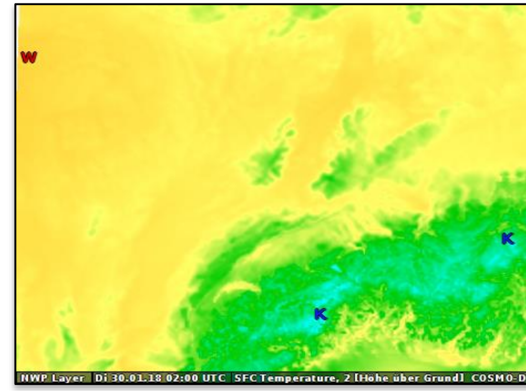


“Watershed”



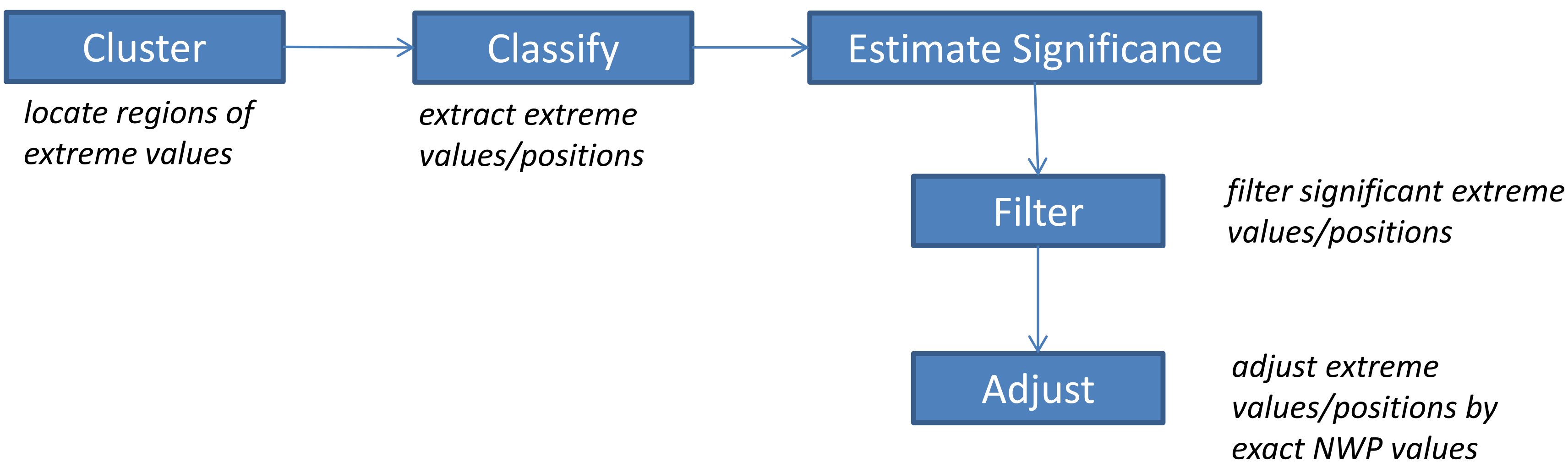
- funded as a NinJo research project as part of NWP visualization re-design.

Commons



requirements:

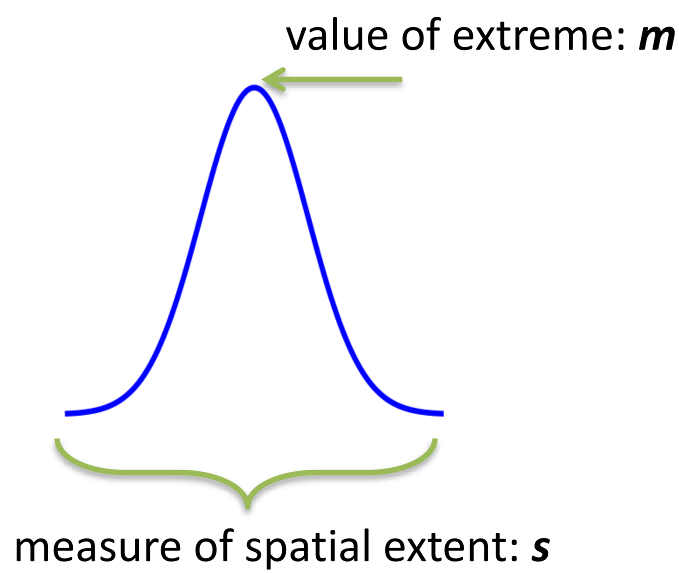
- operate on NWP raw data
- a modular architecture
- fast enough for interactive usage
- minimum possible number of needed configuration parameters



Extrema Significance

What is a significant extreme value?

- common definition for all algorithms
- combination of:
  - field value at extreme position
  - estimation of spatial extent of extreme value curvature

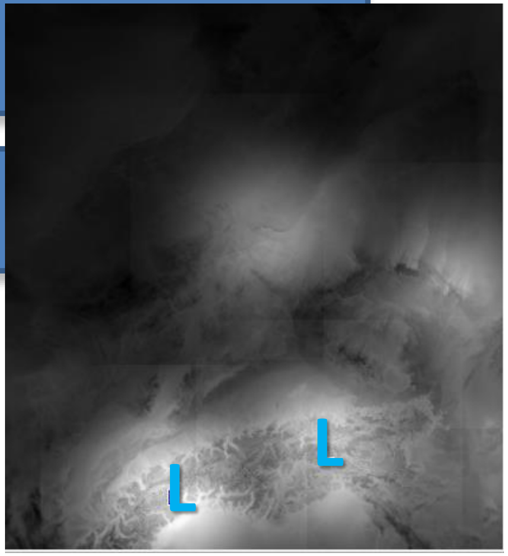


$$sig := m^{(1-\gamma)} * s^{\gamma}$$

– γ defines weight between value and extent

The “cow pat” algorithm

1. Find global extremum
2. Adapt surrounding using 2D Gaussian
3. Repeat (until break condition)
4. Filter for significance

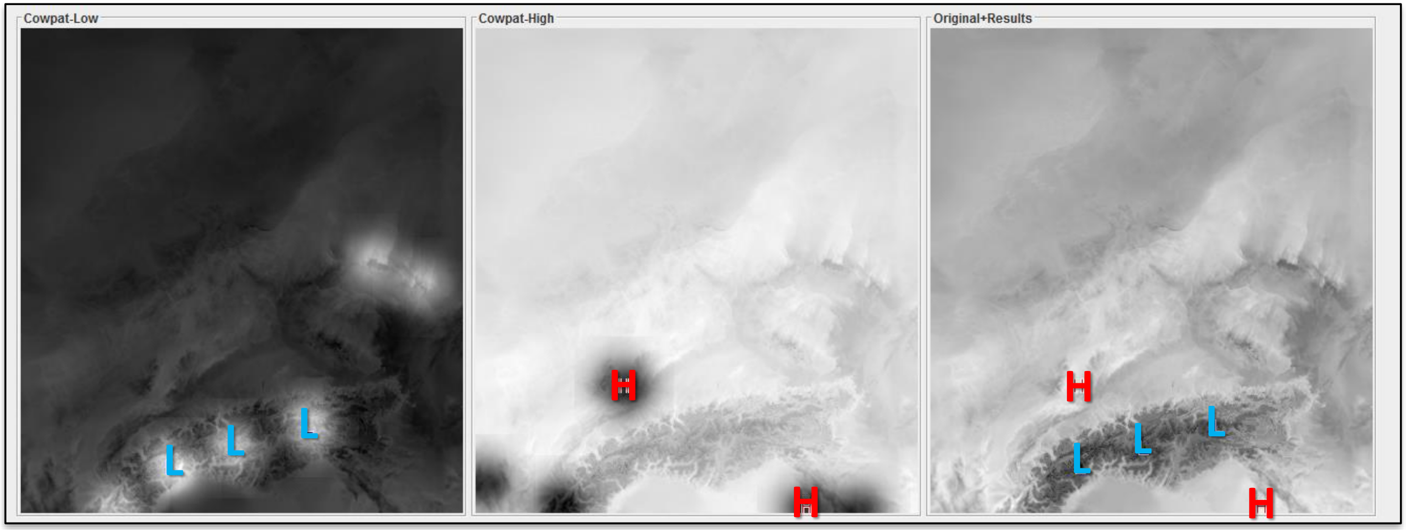


The “cow pat” algorithm

- extremely simple and fast
- intuitive and easy to understand
- one core configuration option: width of the Gaussian

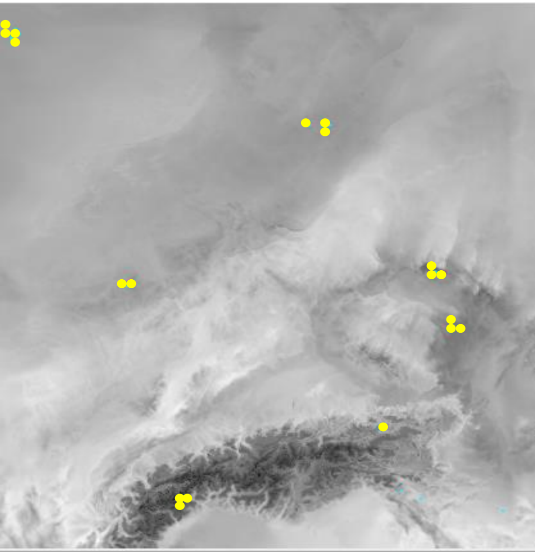
disadvantages:

- width defines globallly static search radius



The “Firefly” algorithm

1. Place “fireflies” on map
2. Let them attract each other and move towards a local extrema
3. Repeat (until break condition)
4. Cluster and filter for significance



The "Firefly" algorithm

- particle swarm optimization for multi-modal distributions

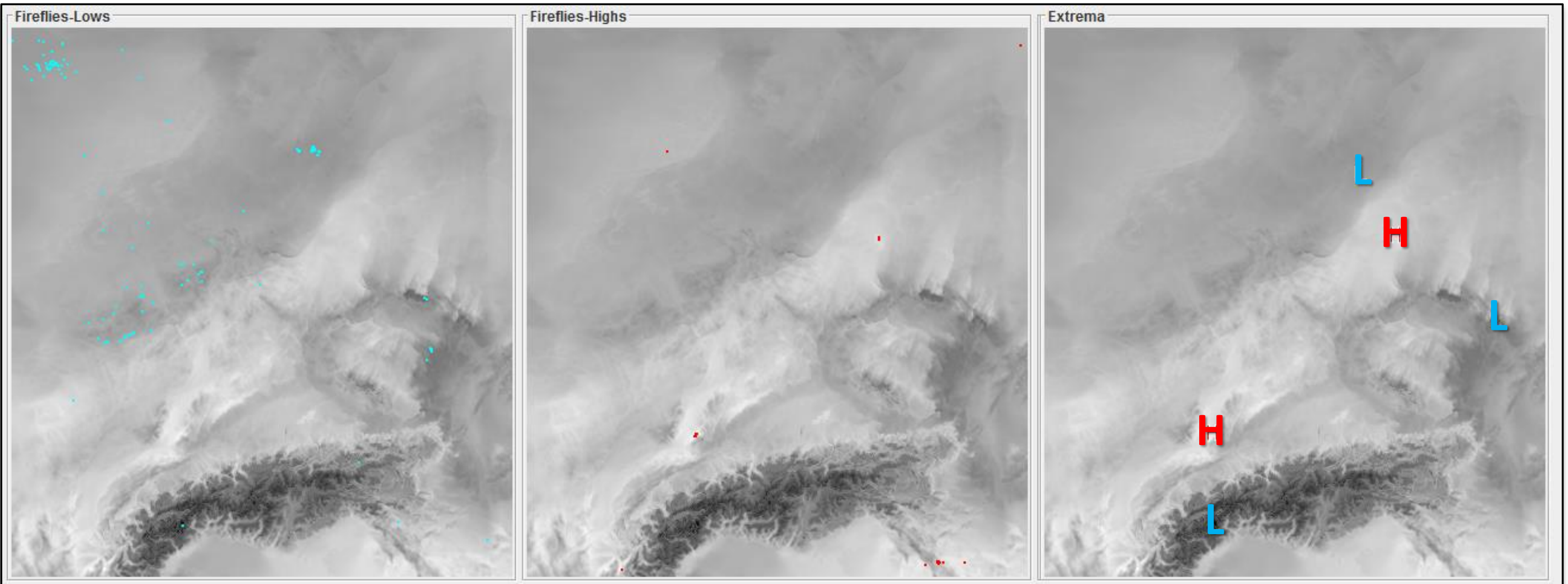
$$x_i := x_i + \beta\, q_i\, q_j\, e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(rand - 1/2)$$

- here: q-values denote particle charge to cluster at local extreme value positions (coulomb constants)
- q_i := field value at particle position *i*

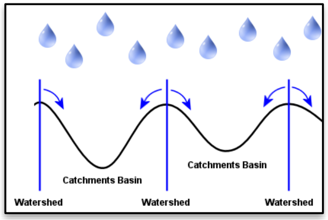
"A Multimodal Firefly Optimization Algorithm Based on Coulomb's Law",
Taymaz Rahkar-Farshi & Sara Behjat-Jamal, IJACSA, Vol7, No.5, 2016

The “Firefly” algorithm

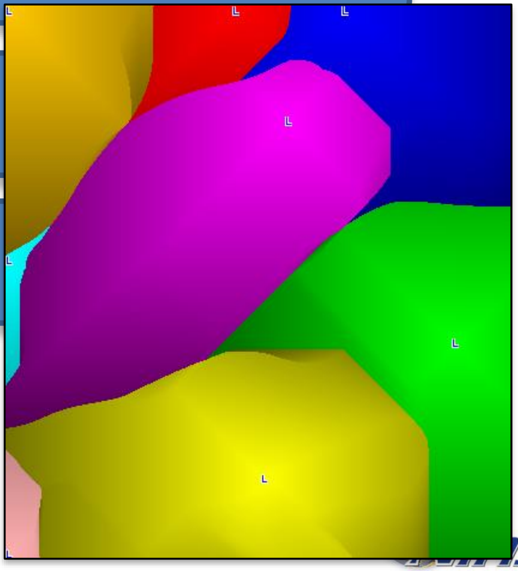
- several improvements to the proposed algorithm made, to:
  - reduce amount of particles needed
  - reduce number of required iterations → faster convergence
- post-processing steps:
  - extract single local extreme positions/values from cluster of particles (by NWP field lookup)
  - calculate extreme value significance and filter



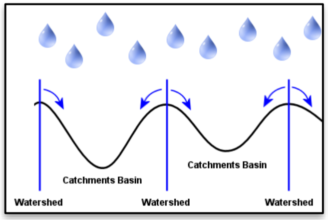
The "Watershed" algorithm



1. Place "raindrops" on blurred map
2. Let them follow the strongest gradient towards a local extrema
3. Identify extrema in "catchment basins"
4. Merge basins if applicable
5. Filter for significance



The “Watershed” algorithm



well known segmentation algorithm in image processing

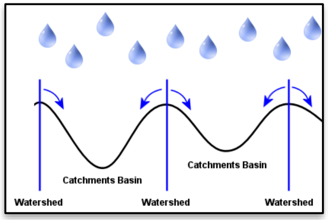
- let “rain-drops” fall on each pixel
- trace the path along the strongest gradient
- non-interconnected resulting graphs form “catchment basin”s → area surrounding a local extreme value

adaptations

- merging based on the energy required to overflow into an adjacent basin
- extraction of local extreme values/positions (by NWP field lookup)

https://en.wikipedia.org/wiki/Watershed_(image_processing)#Watershed_by_the_drop_of_water_principle

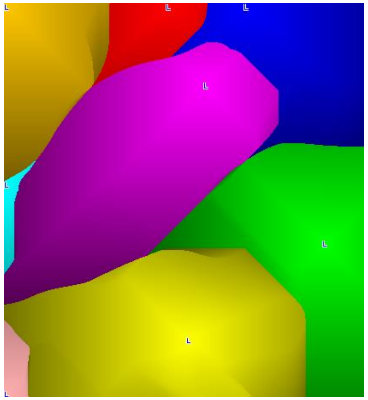
The "Watershed" algorithm



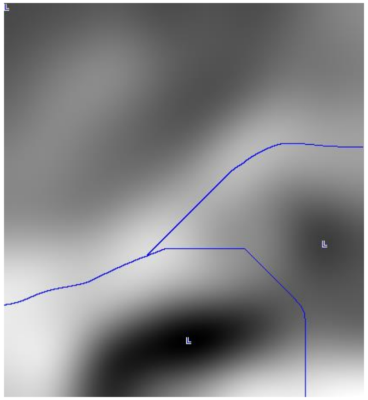
- steps of the algorithm

blur input field

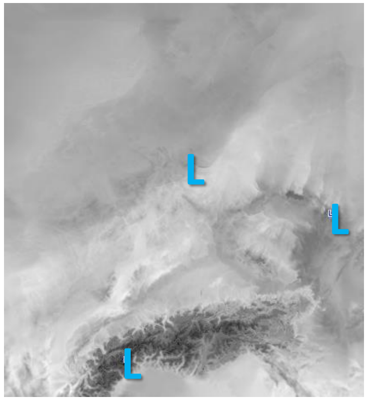
calculate watersheds



merge basins



extract and re-locate

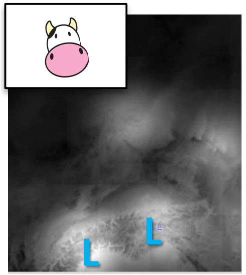


- fast and easy to configure
  - core options: blur factor, overflow energy for the merge
- shortcomings
  - tends to flow out towards data field boundaries
  - blur factor is a global option and might diminish local effects

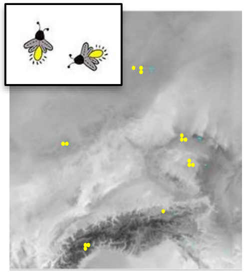
Conclusions

three new algorithms in NinJo for detection and visualization of NWP extreme values

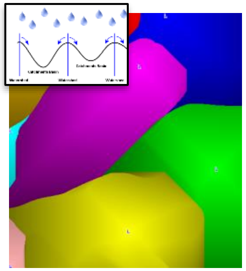
- operating on raw data (therefore applicable to a variety of use-cases)
- different speed, complexity and configurability
- now being evaluated by NinJo super users
  - estimate performance for operational usage (quality and speed)
  - fine-tune to find optimal parameter sets



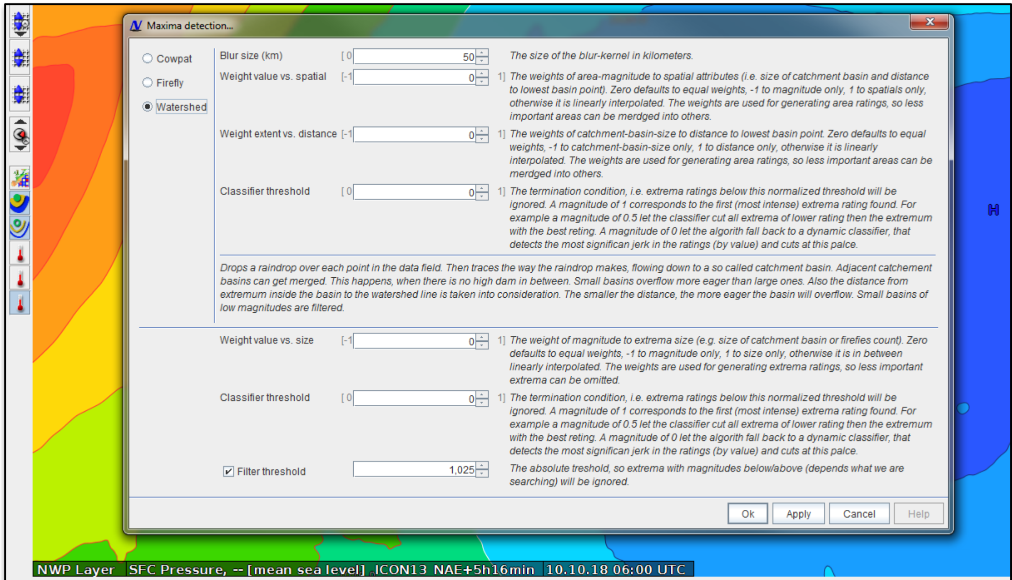
“Cow pat”



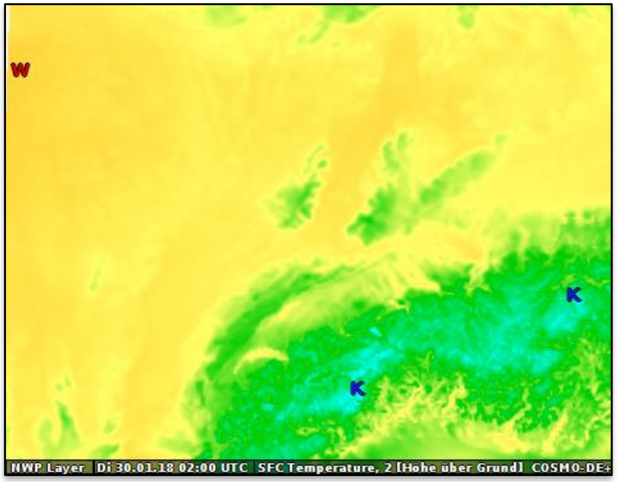
“Firefly”



“Watershed”



Future aspects



to be considered:

- topography
  - more extreme values expected in rough terrain (if applicable)
- type of data?
  - pressure vs. relative humidity
- temporal evolution
  - extreme values usually do not (dis)-appear from one time-step to the next
- closed iso-lines must include an extreme value
  - currently not tackled as this limits use-cases for other types of data

Thanks for your attention!

Credits:

- *Waldemar Busiakiewicz*
- *Jan Schröter*

EGOWS '18 – ECMWF Reading, UK

Sören Kalesse, Deutscher Wetterdienst