BeaulieuJM.ca/publi/Bea2019b

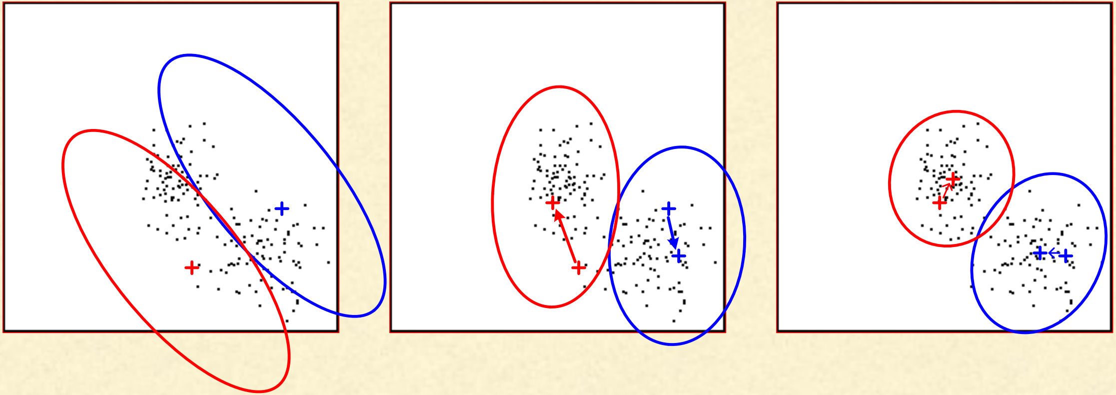
EFFICIENT HIERARCHICAL CLUSTERING FOR POLSAR IMAGE ANALYSIS

Jean-Marie Beaulieu

Hierarchical clustering is hard
large computing time

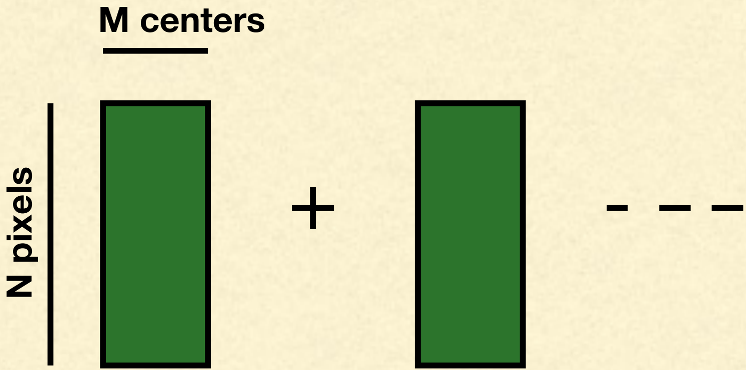
Could be a useful tool
with a fast implementation

- **Iterative Clustering**
- **Move group centers (K-means algorithm)**
- **Fixed number of groups**



K-Means: Iterative Clustering

Calculate the distances between pixels and centers

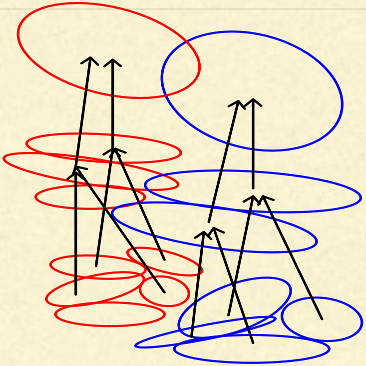


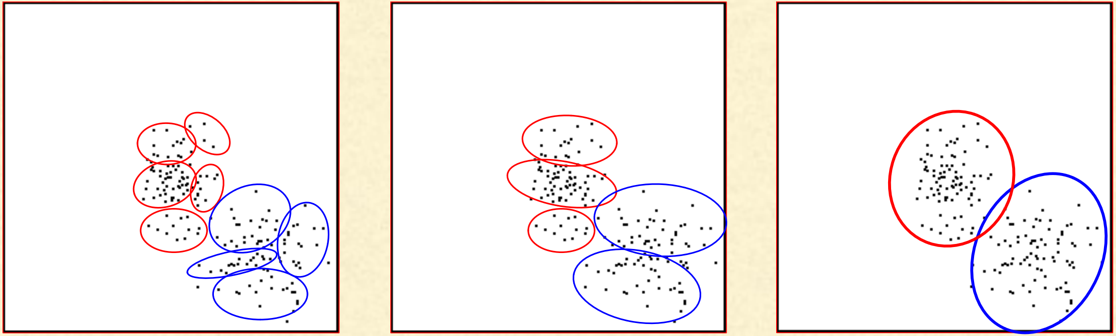
Number of operations

$M \times N$ (by iterations) $\times$ L iterations $\approx$ $L\,M\,N$

10 centers 1,000,000 pixels 10 iterations → 10^8 → 100 M op

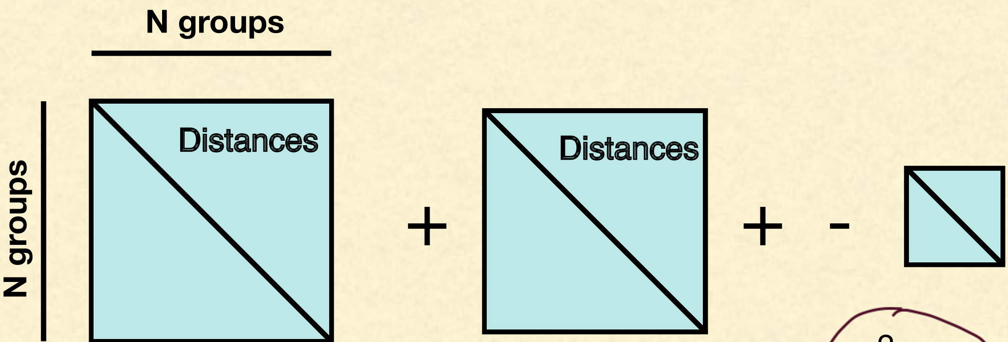
- **Hierarchical clustering**
- **Sequential merging of clusters**
- **Merge the best pair**
- **Represented by a tree**
- **Step Wise Optimization**





Hierarchical Clustering

Calculate the distances between groups



$$\sum_{n=N}^{2} n^2$$

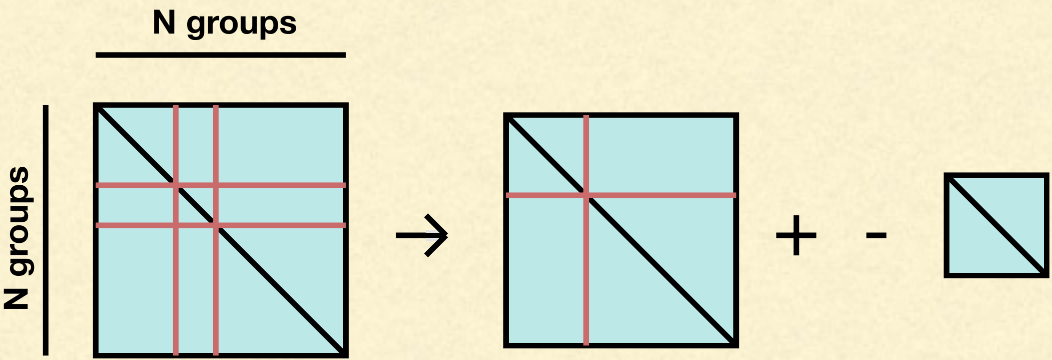
Number of operations

$N \times N$ (by iterations) $\times$ (N-1) iterations $\approx \alpha N^3$

$10^6 \times 10^6 \times 10^6 \rightarrow 10^{18} \rightarrow$ 1,000,000 Tera op

Hierarchical Clustering

Update of Distances



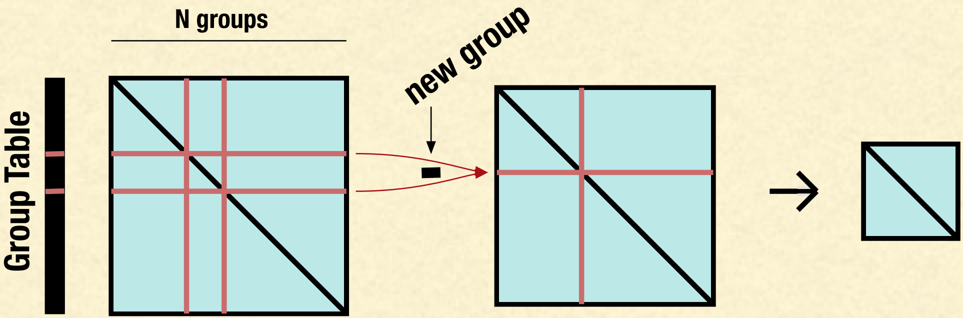
Number of operations

$$N \times N \ \text{(initialization)} + \sum_{n=N}^{2} n \approx \alpha N^2$$

$$10^6 \times 10^6 \rightarrow 10^{12} \rightarrow \text{1 Tera op}$$

Hierarchical Clustering

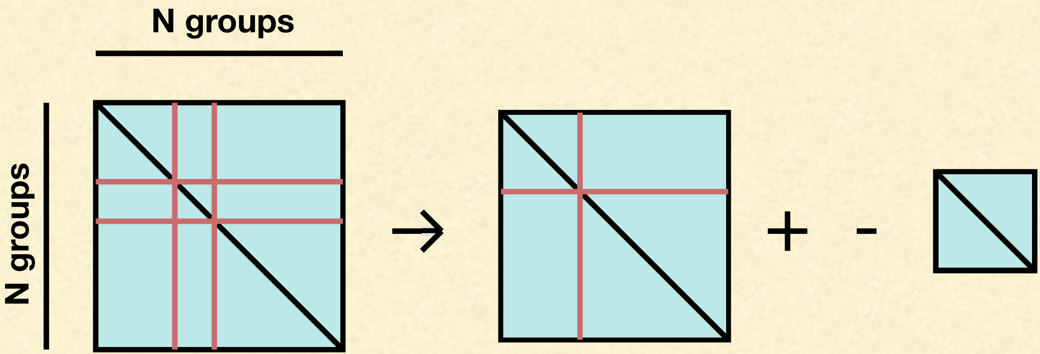
Update of Distances



Number of operations

$$N \times N \;\; \text{(initialization)} + \sum_{n=N}^{2} n \approx \alpha N^2$$

$$10^6 \times 10^6 \rightarrow 10^{12} \rightarrow 1 \text{ Tera op}$$



Number of operations

$$N \times N \text{ (initialization)} + \sum_{n=N}^{2} n \approx \alpha N^2$$

Memory space

$N \times N \rightarrow 10^6 \times 10^6 \rightarrow 10^{12} \rightarrow$ 1 Tera values

- **Non textured multi-look Polsar image**
- **Z_k follows a complex Wishart distribution**

$$p(Z_k \mid \Sigma) = \frac{L^{3L} \left|Z_k\right|^{L-3} \quad \exp\left\{-L\, tr\left(\Sigma^{-1} Z_k\right)\right\}}{\pi^3\ \Gamma(L)\Gamma(L-1)\Gamma(L-2) \left|\Sigma\right|^{L}}$$

- **Distance between groups D(Gi,Gj)**
- **Log of the Likelihood Ratio**

$$D(G_i, G_j) = (n_i + n_j) \ln\left|\hat{\Sigma}_{Gi \cup Gj}\right| - n_i \ln\left|\hat{\Sigma}_{Gi}\right| - n_j \ln\left|\hat{\Sigma}_{Gj}\right|$$

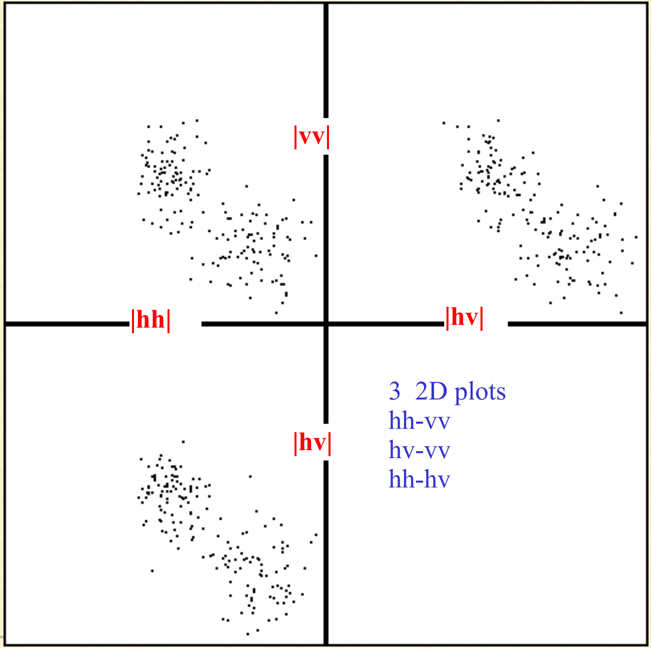
- **Attributes or feature space** **(many dimensions)**
- **Radiometric information** **(or color/spectral)**

Radar 1-look

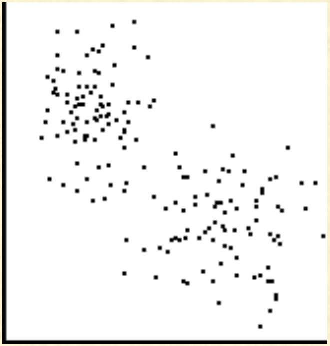
$$x = \begin{bmatrix} hh \\ hv \\ vv \end{bmatrix}$$

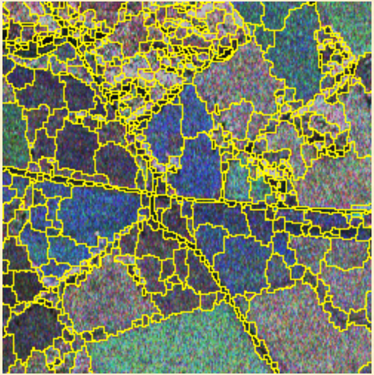
Radar multi-look

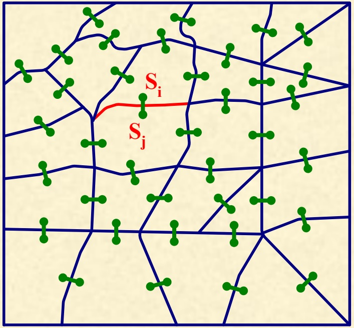
$$Z = \begin{bmatrix} \overline{hh\,hh^*} & \overline{hh\,hv^*} & \overline{hh\,vv^*} \\ \overline{hv\,hh^*} & \overline{hv\,hv^*} & \overline{hv\,vv^*} \\ \overline{vv\,hh^*} & \overline{vv\,hv^*} & \overline{vv\,vv^*} \end{bmatrix}$$



- **Spatial information - position in the image**
- **Clustering -- distance between points D(Gi,Gj)**
- **Segmentation -- only adjacent regions**

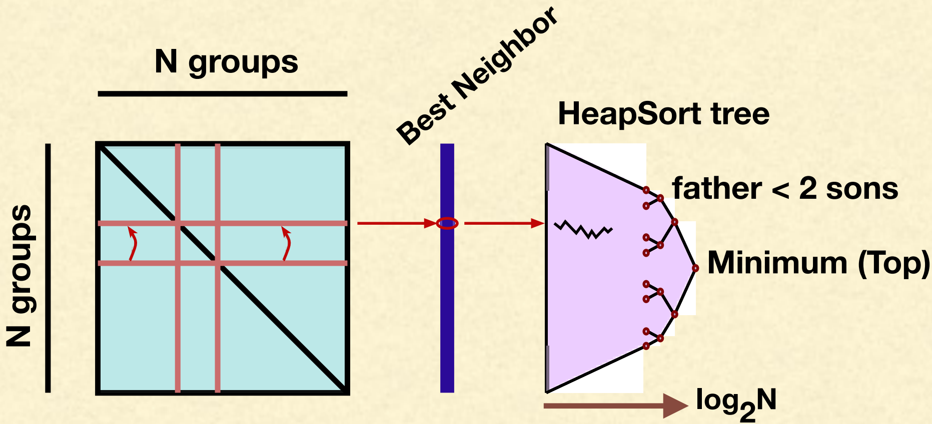






Finding the Minimum

HeapSort sort tree — $\propto$ N $\log_2 N$ 20 Mega



The minimum indicate which group to merge and it is merged with its best neighbor

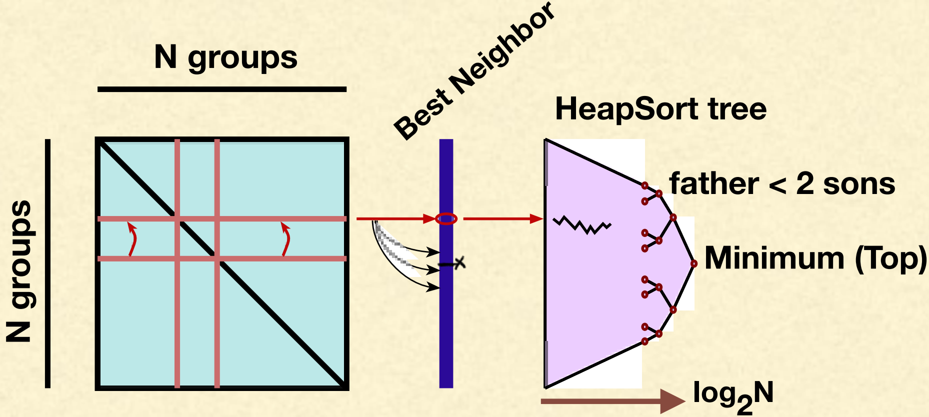
Do not need to keep and store the distance matrix 1 Tera

Delayed Update of the neighbors

when a group is selected as the minimum

➜ check if already merged

➜ remove or update (2 to 5 factor)



Fast Pre-Selection

Fast Testing ➜ remove 90% to 95% of distance calculations

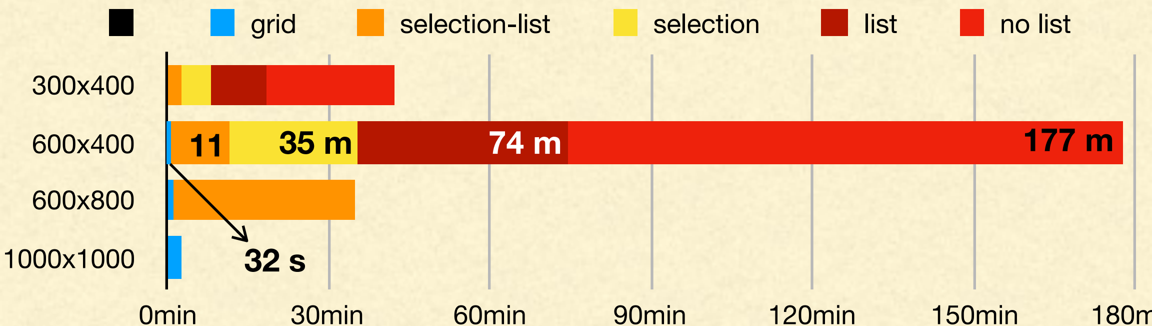
Thresholding ➜ ratio $|hh|_\mathbf{i}/|hh|_\mathbf{j}$, $|hv|_\mathbf{i}/|hv|_\mathbf{j}$ et $|vv|_\mathbf{i}/|vv|_\mathbf{j}$

Lists of the Closer Neighbors

for the merging steps only

Grid for Group Selection

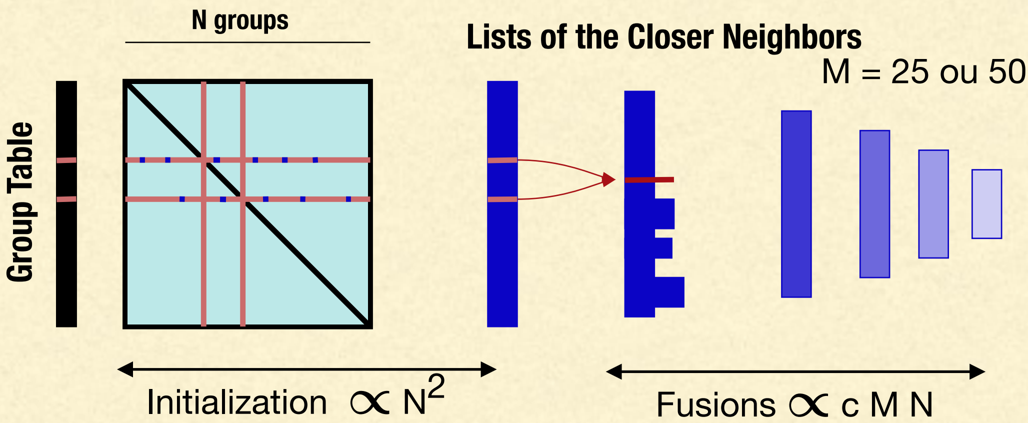
speedup list initialization



Lists of the Closer Neighbors

Calculation of distances only for closer neighbors
during the merging steps $\propto$ M N

large initialization time $\propto N^2$



c=5, delayed update

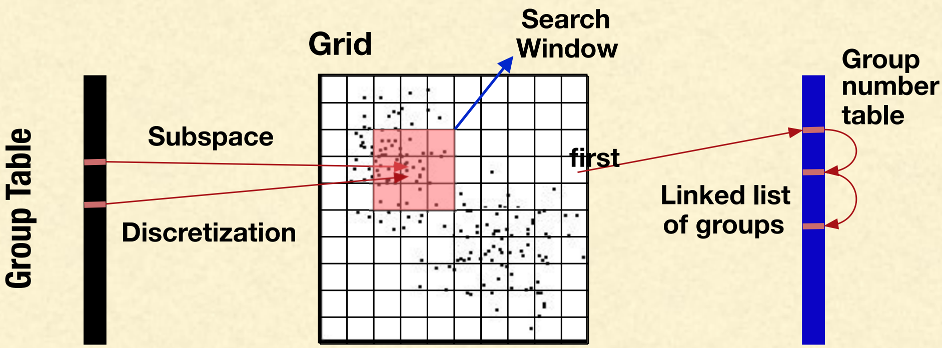
Grid for Group Selection

Subspace of attributes divided into cells (5D, 8×8×8×8×8)

- Discretization ➜ index (no) of the cell
- Cell ➜ hold a linked list of groups (pointer)

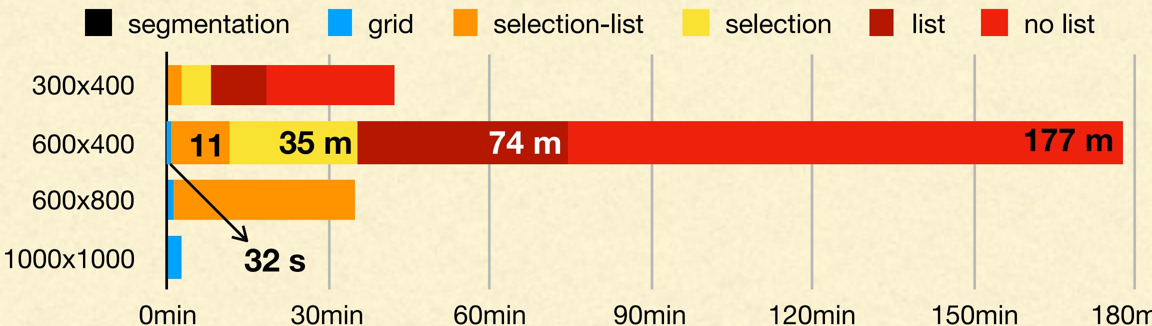
Search for the closer neighbors

- Inspect cells inside a window (3×3×3×3×3)
- Inspect less groups, but better candidates



Computing Time (CPU)

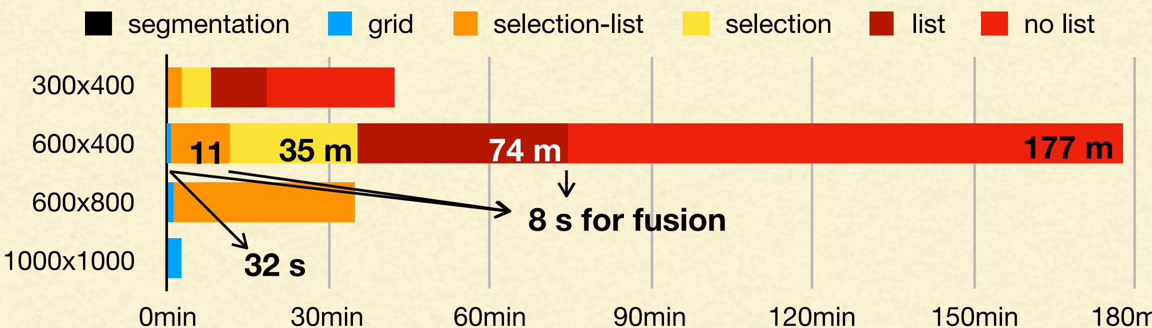
Image size	segmen-tation	grid	selection + list	selection	list	no list
300x400	0s 400ms	15s 200ms	2min 50s	8min 21s	18min 36s	42min 20s
600x400	0s 830ms	31s 500ms	11min 31s	35min 18s	74min 42s	177min 29s
600x800	1s 780ms	1min 7s 400ms	35min 3s			
1000x1000	3s 860ms	2min 31s 300ms				



Computing Time (CPU)

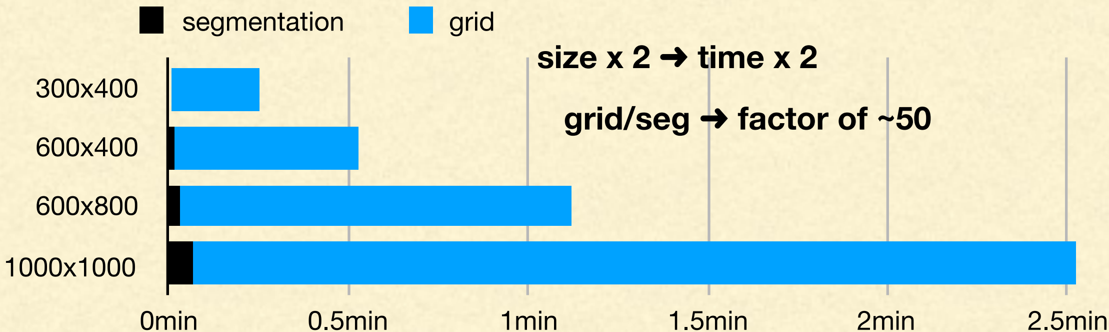
Image size	segmen-tation	grid	selection + list	selection	list	no list
300x400	0s 400ms	15s 200ms	2min 50s	8min 21s	18min 36s	42min 20s
600x400	0s 830ms	31s 500ms	11min 31s	35min 18s	74min 42s	177min 29s
600x800	1s 780ms	1min 7s 400ms	35min 3s			
1000x1000	3s 860ms	2min 31s 300ms				

size x 2 ➜ time x 4



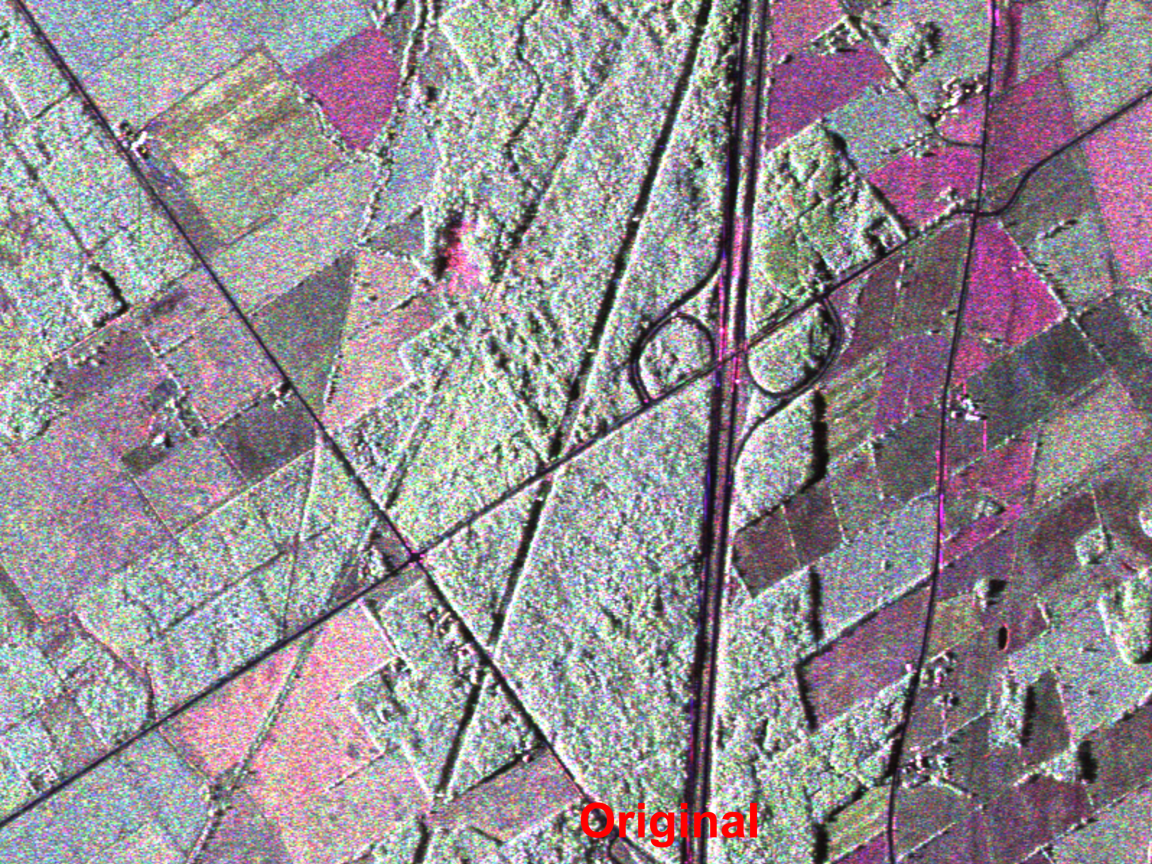
Computing Time (CPU)

Image size	segmen-tation	grid	selection + list	selection	list	no list
300x400	0s 400ms	15s 200ms	2min 50s	8min 21s	18min 36s	42min 20s
600x400	0s 830ms	31s 500ms	11min 31s	35min 18s	74min 42s	177min 29s
600x800	1s 780ms	1min 7s 400ms	35min 3s			
1000x1000	3s 860ms	2min 31s 300ms				

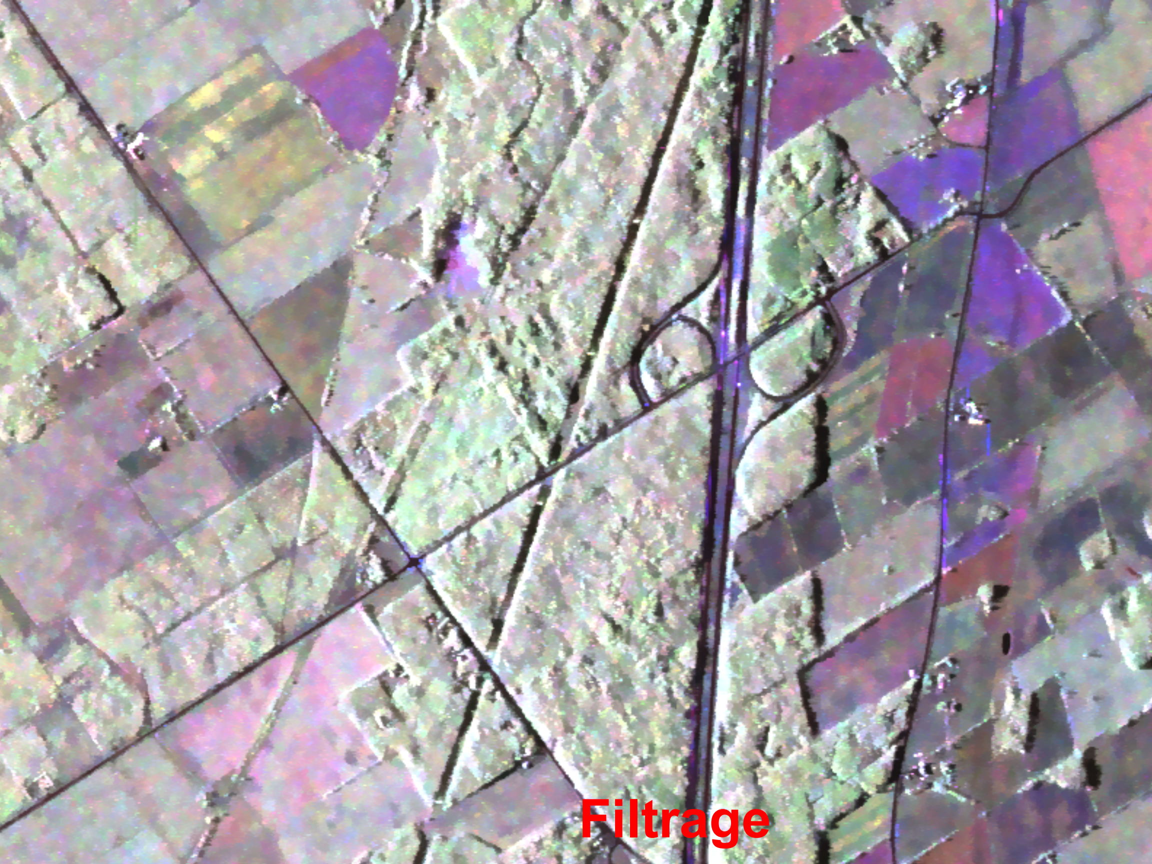


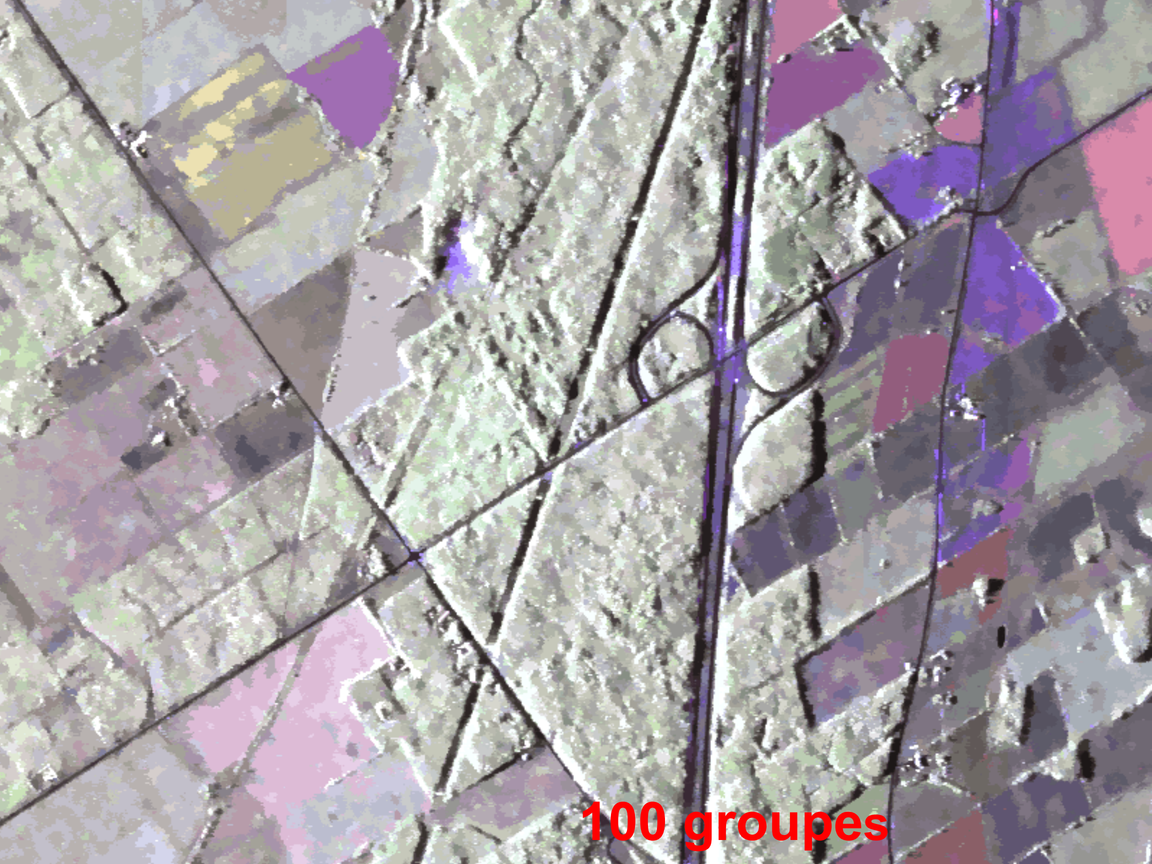
size x 2 ➜ time x 2

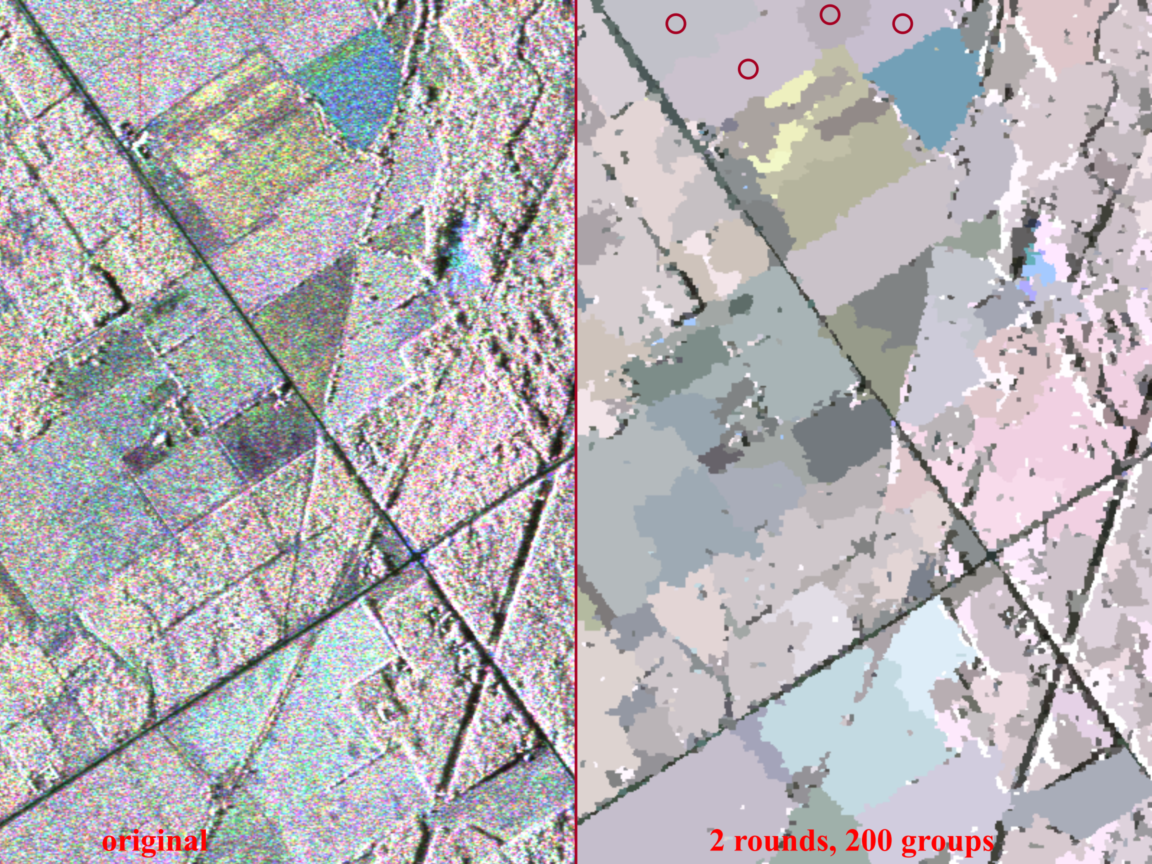
grid/seg ➜ factor of ~50


Original


100 groupes


Filtrage


100 groupes

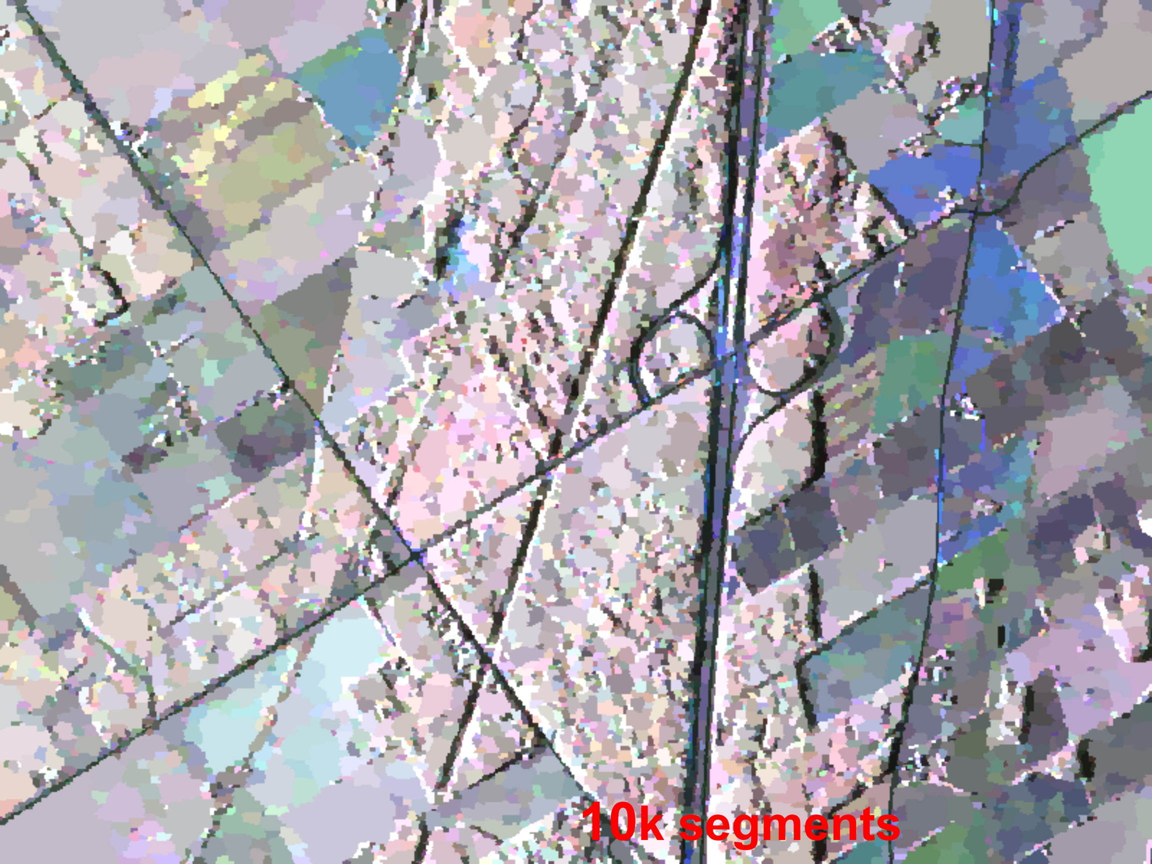

original
2 rounds, 200 groups

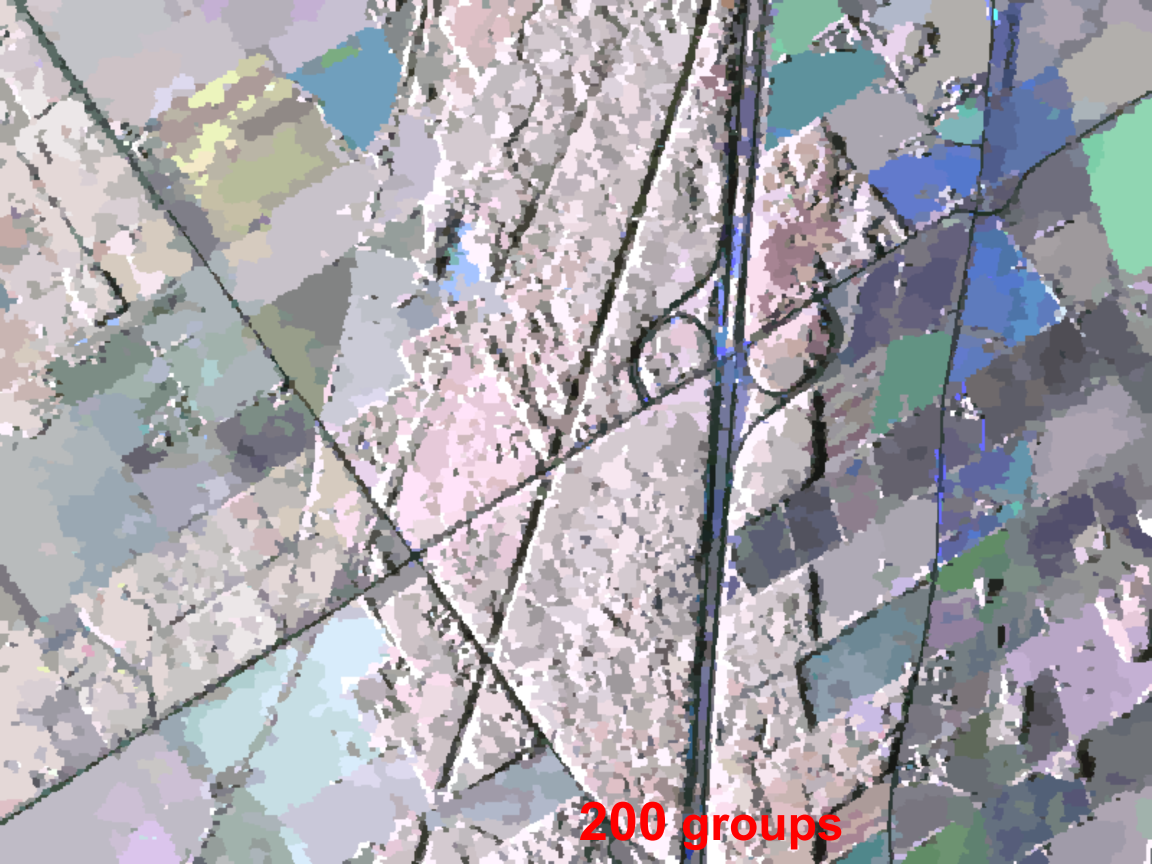
Conclusion

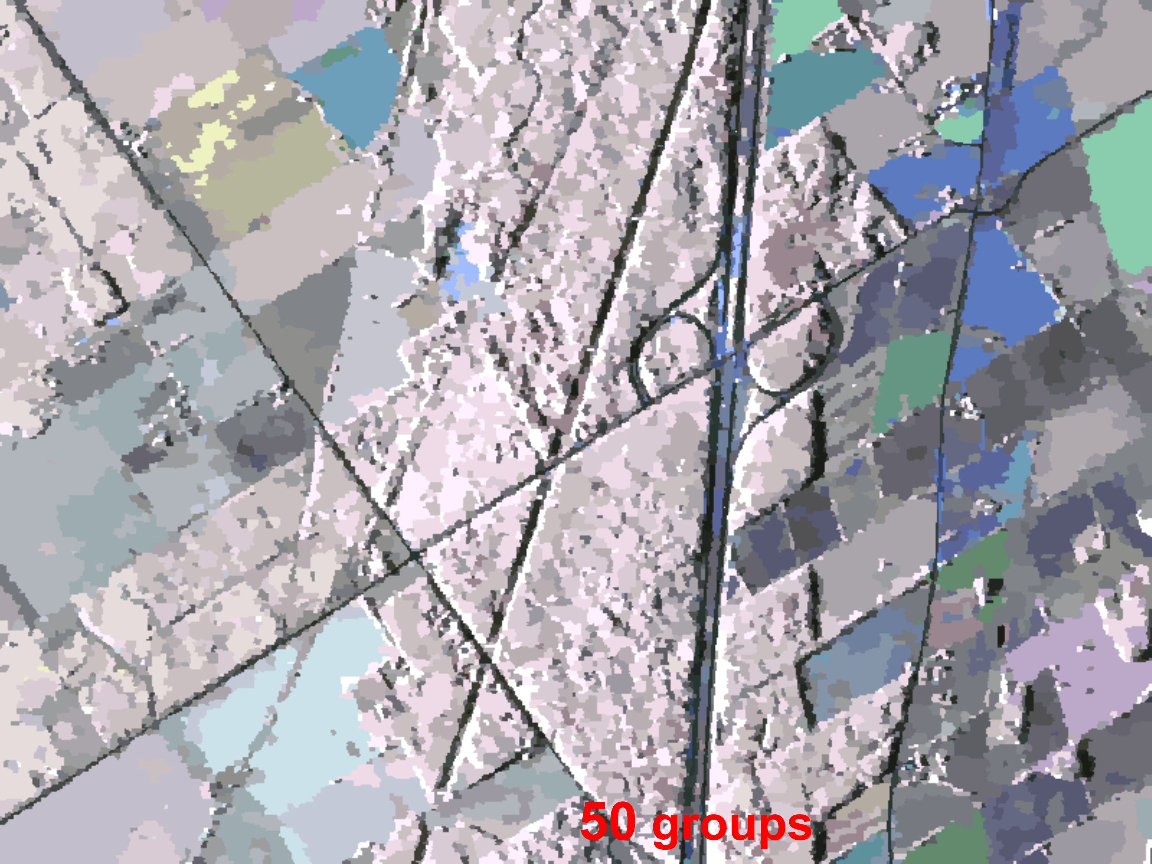
Hierarchical clustering is hard
large computing time

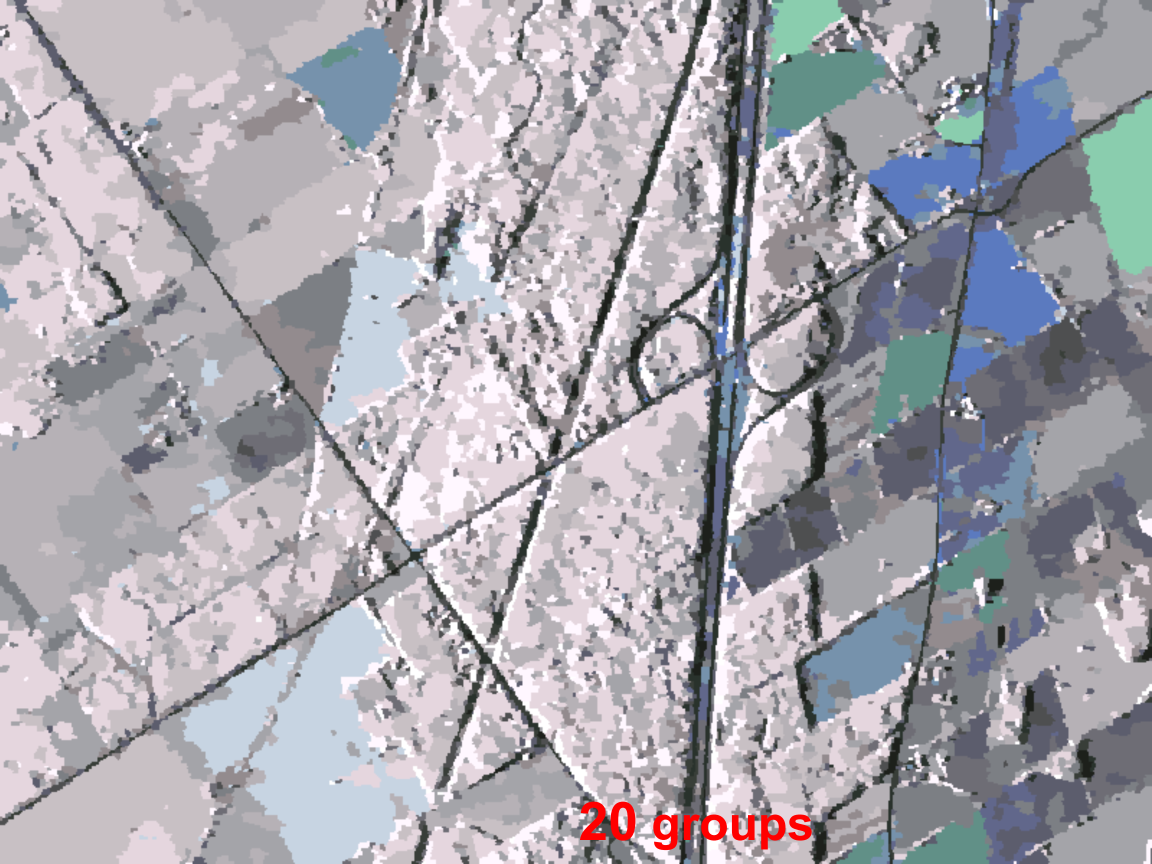
Become a useful tool
with a fast implementation

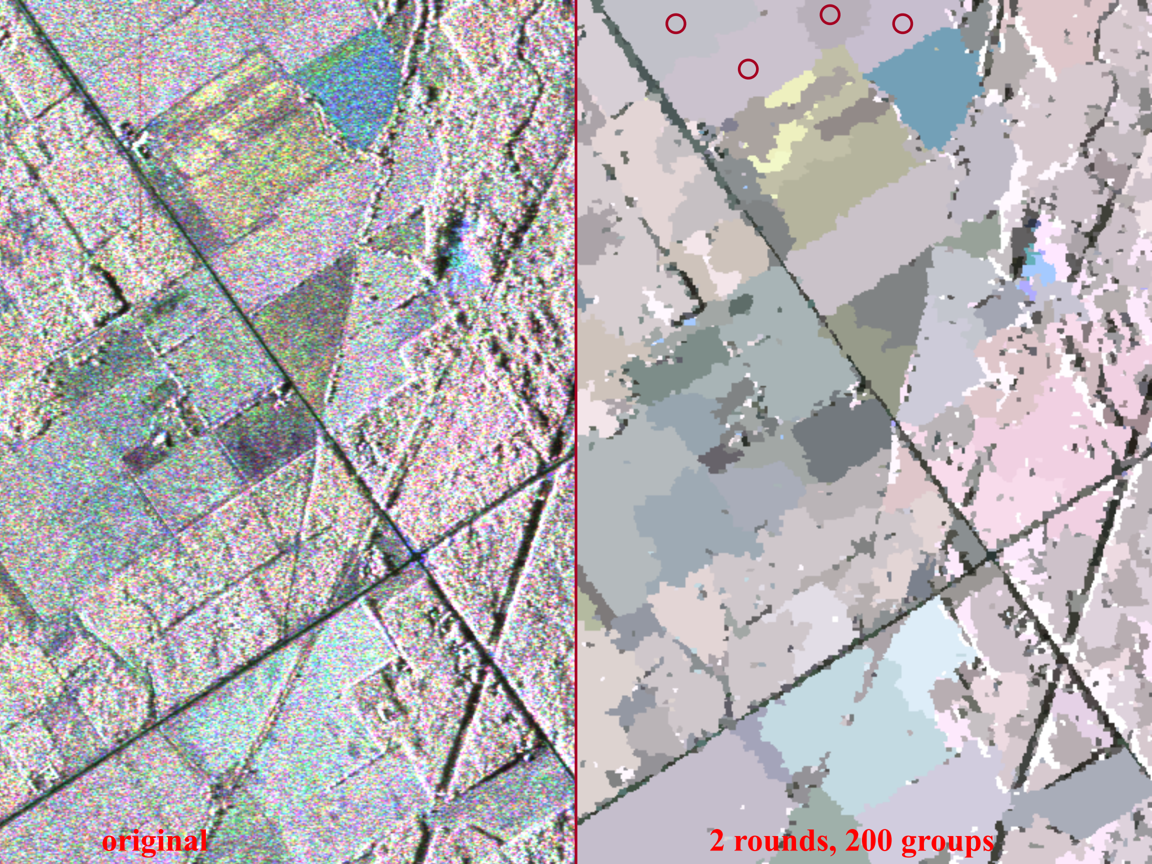
Should know when and how to use it


10k segments


200 groups


50 groups


20 groups


original
2 rounds, 200 groups

Conclusion

Hierarchical clustering is hard
large computing time

Become a useful tool
with a fast implementation

Should know when and how to use it