Calorimeter clustering
with minimal spanning trees

G.Mavromanolakis, University of Cambridge

Outline

- **Introduction**
- **Clustering with MST**
- **Clustering with MST + extensions**
- **Technicalities – Performance**
- **Summary**

Introduction – theory

▶ **minimal spanning tree**

: a tree which contains all nodes with no circuits and of which the sum of weights of its edges is minimum

▶ **properties**

: unique for the given set of nodes and the chosen metric

: deterministic, no dependence on random choices of nodes

: invariant under similarity transformations that preserve the monotony of the metric

Introduction – theory

▶ MST and clustering

: *theorem 1:* any MST contains at least one edge from each link-set between P and Q partitions

: *theorem 2:* all MST edges are links of some partition of graph

: *theorem 3:* if S denotes the nodes of graph and C is a non-empty subset of S with the property that ρ(P,Q) $<$ ρ(C,S-C) for all partitions P, Q of C, then the restriction of any MST to the nodes of C forms a connected subtree of the MST

: *theorem 4:* if T is an MST for graph G and X, Y are two nodes of G then the unique path in T from X to Y is a minimax path from X to Y

References

[1] C.T.Zahn, *IEEE Trans.Comput. C20(1971)68*

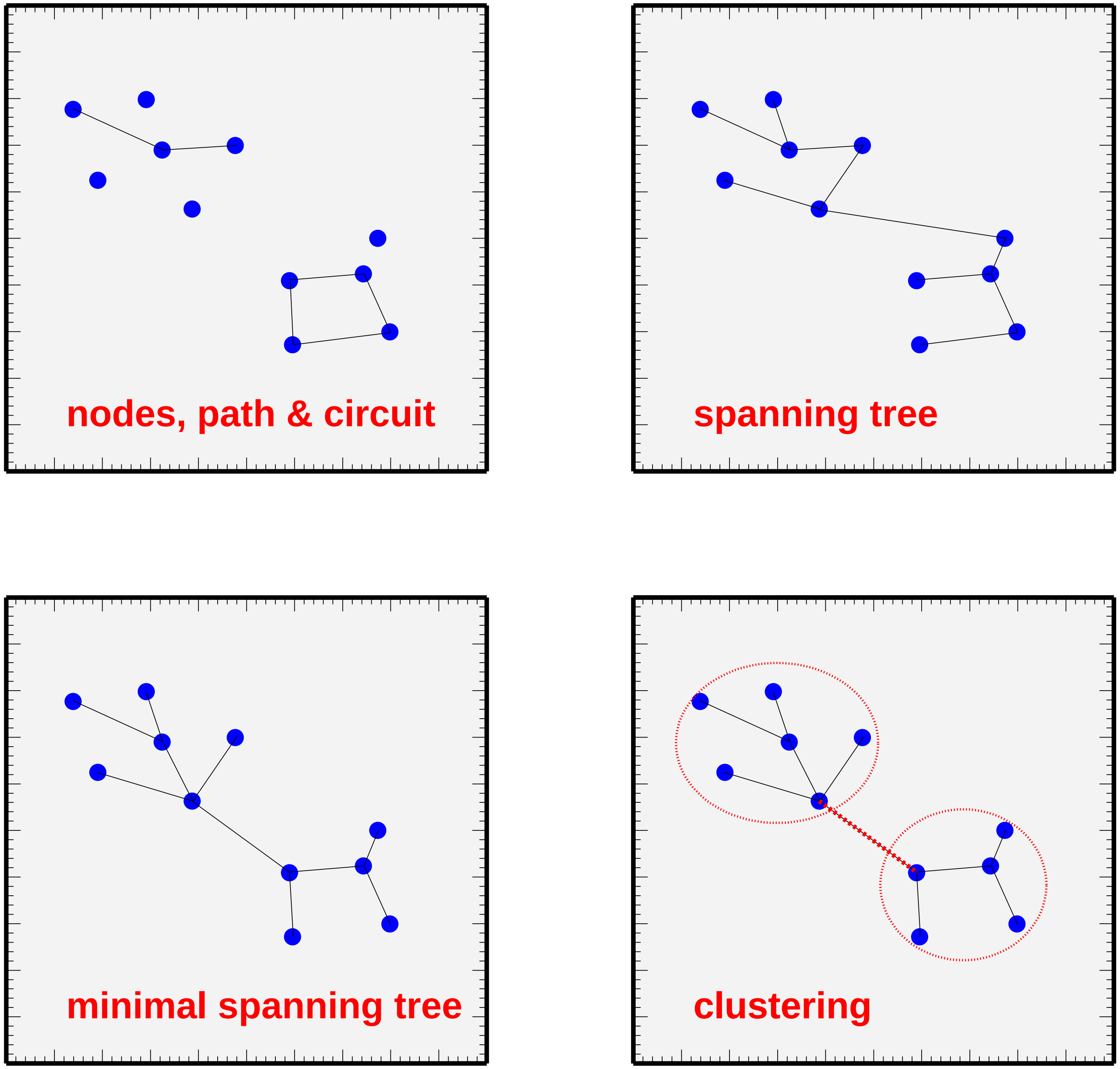
[2] J.C.Gower, G.J.S.Ross, *Appl.Statis. 18(1969)54*

[3] G.J.S.Ross, *Appl.Statis. 18(1969)103*

[4] R.C.Prim, *Bell System Techn.Jour. 36(1957)1389*

[5] K.DeWinter etal. (CHARM II Collaboration) *Nucl.Instr.Meth. A277(1989)170*

[6] N.Saoulidou *Ph.D. thesis, Univ. of Athens 2003*


nodes, path & circuit
spanning tree
minimal spanning tree
clustering

MST Clustering – Main steps

►. **metric**

define a metric : metric defines configuration space
: not necessarily euclidean

fill distance matrix : lower triangular N^2 matrix

►. **MST**

construct the MST : apply Prim’s algorithm

►. **clustering**

set cut : "proximity" bound between nodes belonging to the same cluster

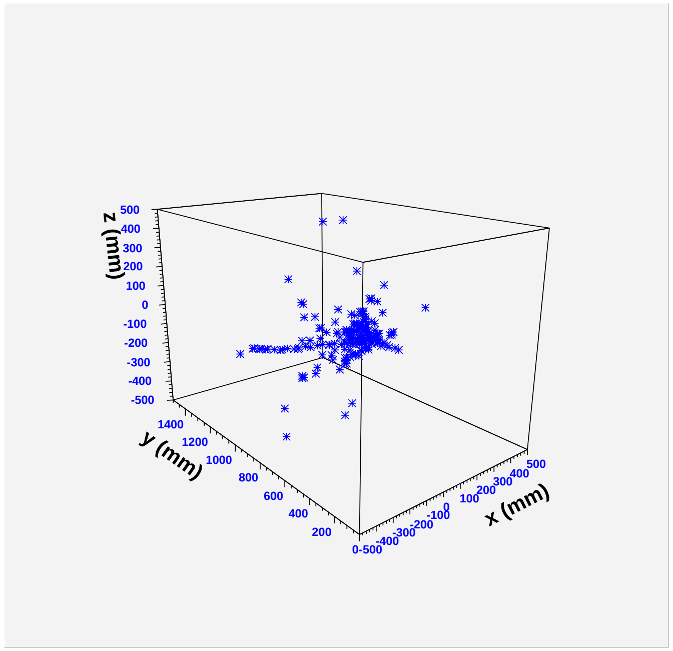
find clusters : single linkage cluster analysis
: i.e. go through MST and cut branches with length above cut

MST Clustering – pros and cons

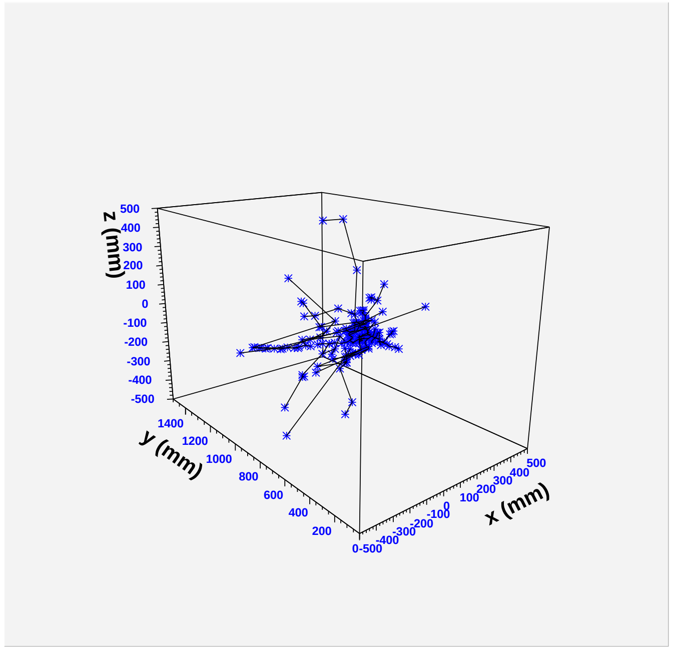
- **pros**
  - after defining an "appropriate" metric to fill the distance matrix, the rest of the algorithm has no dependency on data format and detector geometry since only the metric deals with these
  - so very easy to switch to different geometries/detectors
  - algorithm is a $\mathcal{O}(N^2)$ loop

- **cons**
  - chaining effect due to single linkage method,
    problem can be partially solved by implementing average linkage

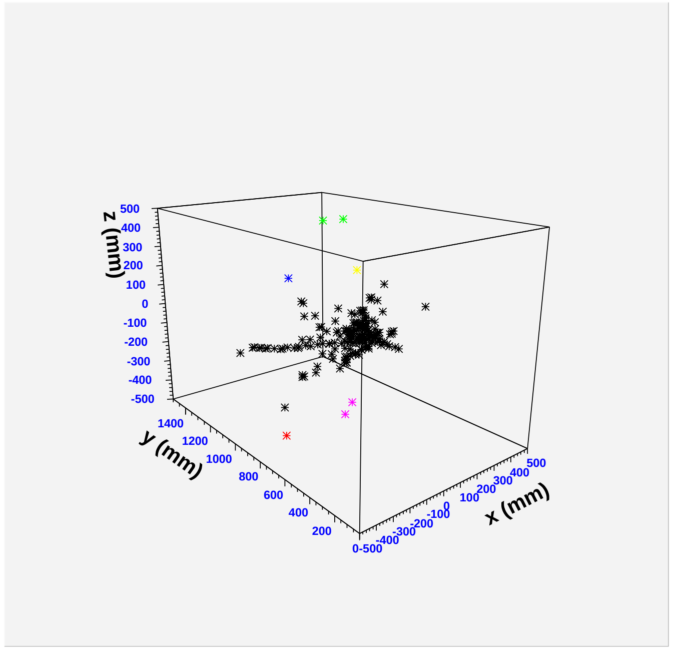
nodes

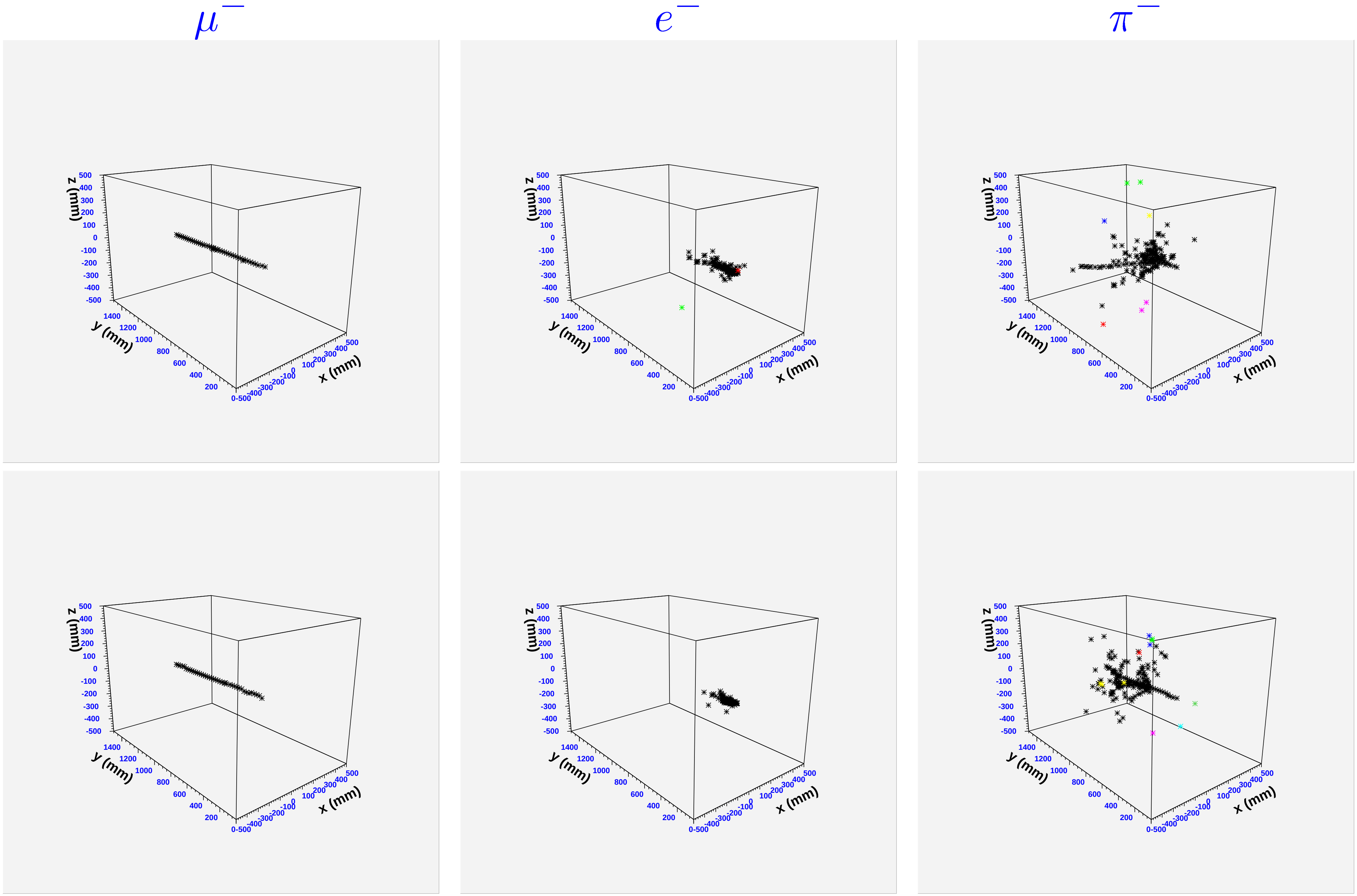


minimal spanning tree



clustering





μ^-

e^-

π^-

z (mm)
500
400
300
200
100
0
-100
-200
-300
-400
-500
y (mm)
1400
1200
1000
800
600
400
200
0
x (mm)
-500
-400
-300
-200
-100
0
100
200
300
400
500

Top-down and then bottom-up clustering

▶ **in brief**

: use MST clustering algorithm with loose cut to perform coarse clustering

: go through MST clusters found in previous step and refine using a cone like clustering algorithm

▶ **advantages of top-down and then bottom-up approach**

: (definitely) speed because of preclustering,
vital for a very granular calorimeter even if its occupancy is low

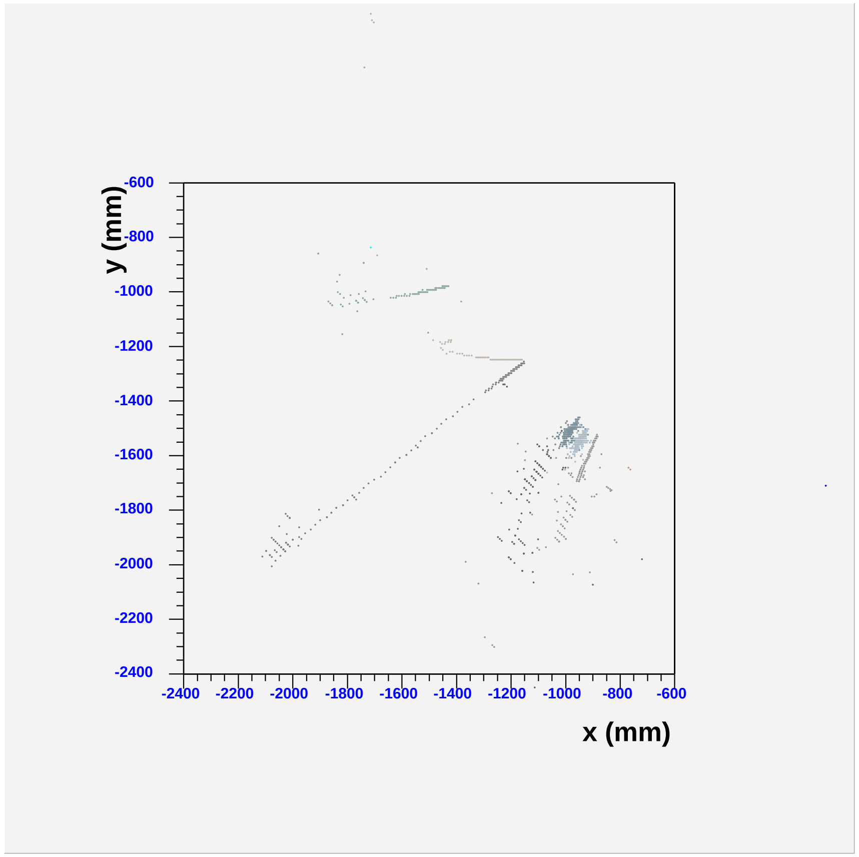
: geometry independence (or at least no strict bindings)

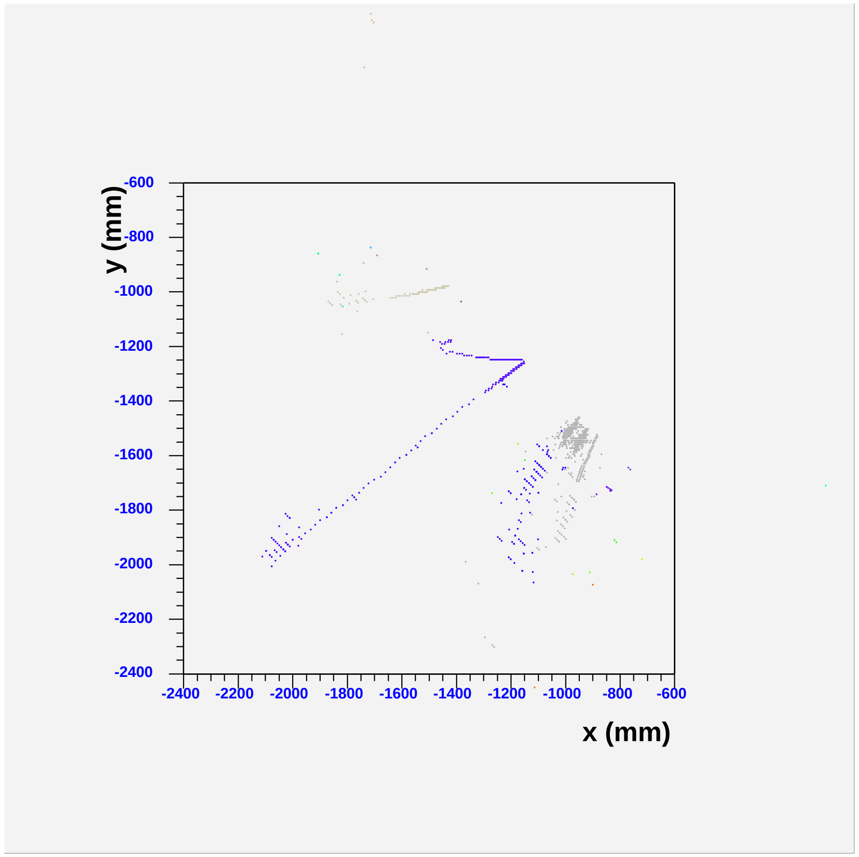
: efficiency (hopefully)

Cone algorithm – General description

▶. per MST cluster do

```
: order points in ascending distance from origin

: assign layer index according to range and pitch

: for a point $j$ at distance $R_j$ with layer index $l$
  and a point $i$ at $R_i < R_j$, layer $m$, belonging to cluster $k$,
  find point $i'$ = projection $\vec{u}(k,m)$ of $i$ to distance $R_j$

: check $angle\ (j, centroid(k,m), i')$
    if $angle < cut$ then
        points $j$, $i$ belong to cluster $k$
        update $centroid(k,l)$ and projection vector $\vec{u}(k,l)$
    else, repeat with point $i-1$

: if end of list and point $j$ is still single then it belongs
  to a new cluster
```

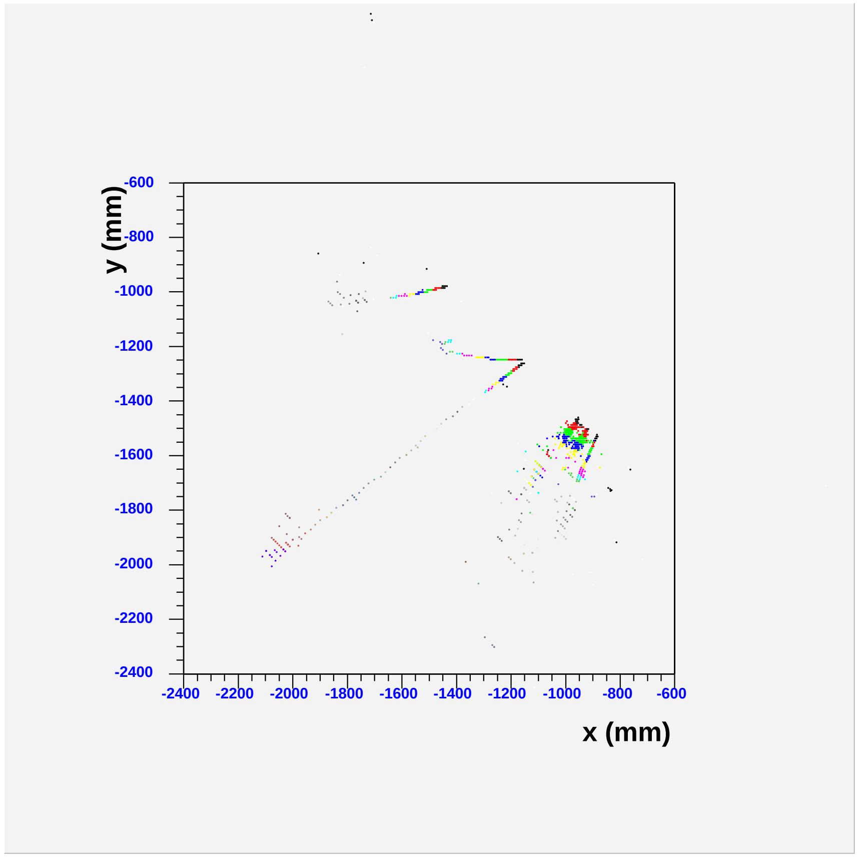




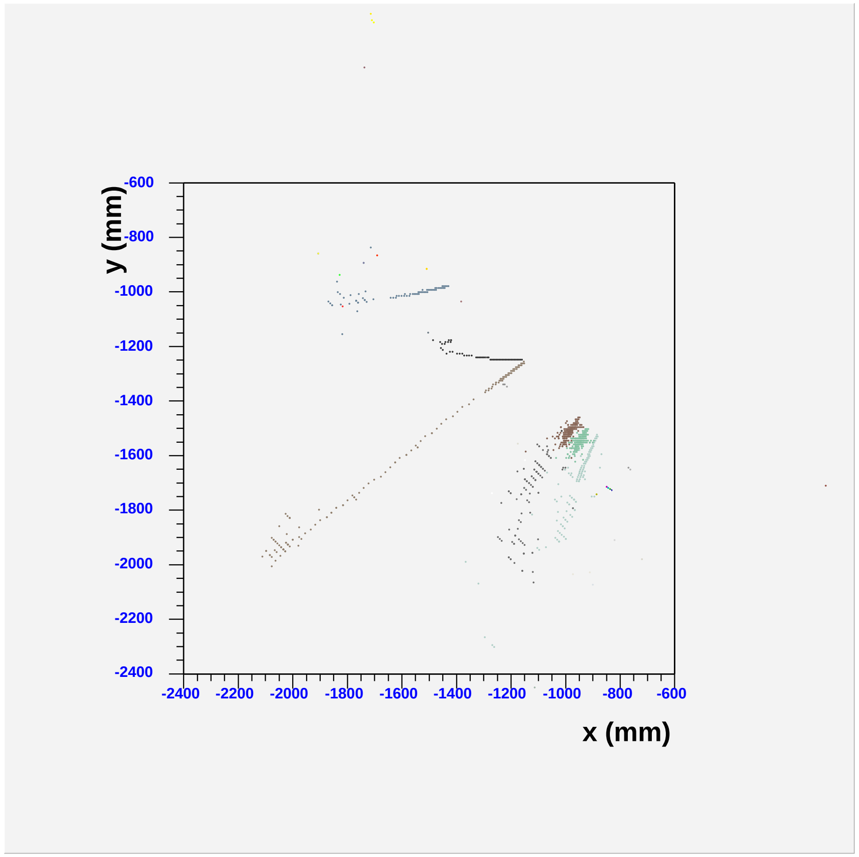
true clusters

MST clusters

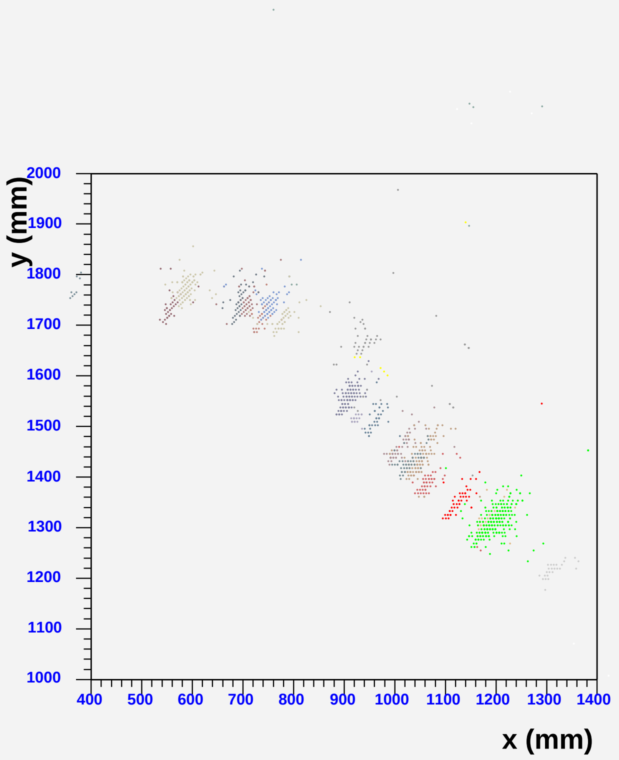
2 or 3 close by true clusters are merged together



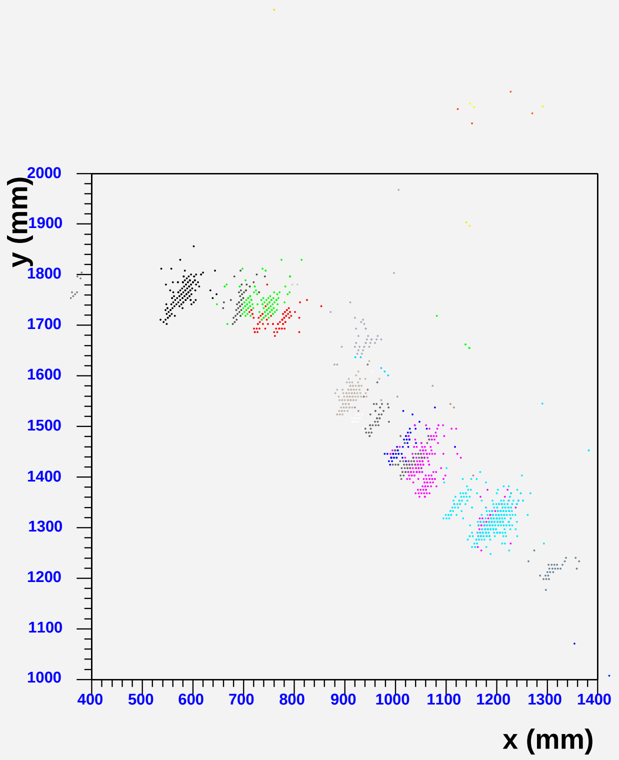
layer indexing per MST cluster



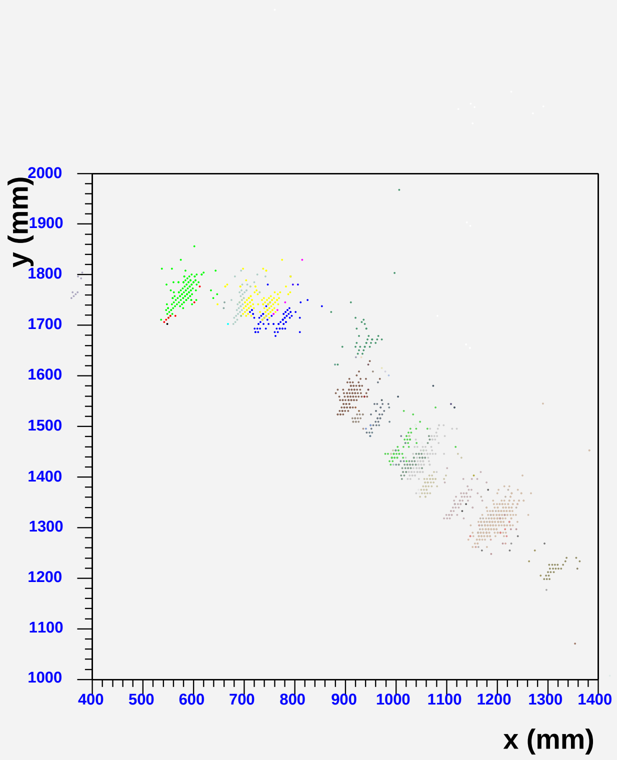
final reconstructed clusters



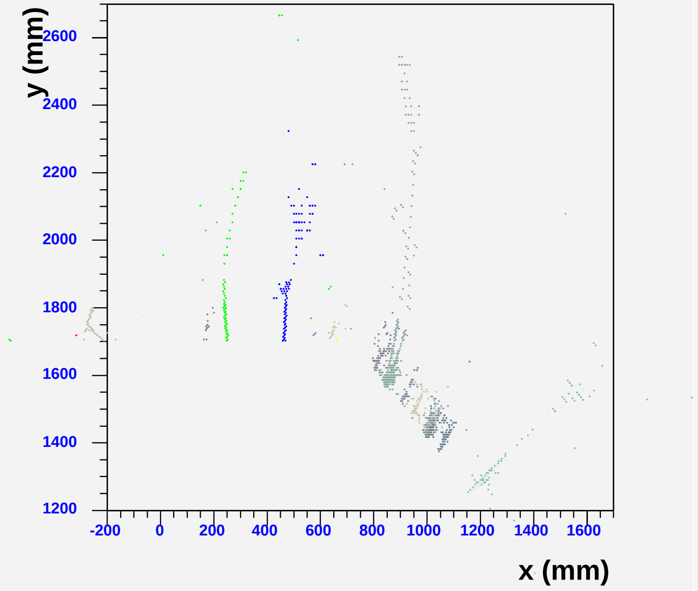
true clusters



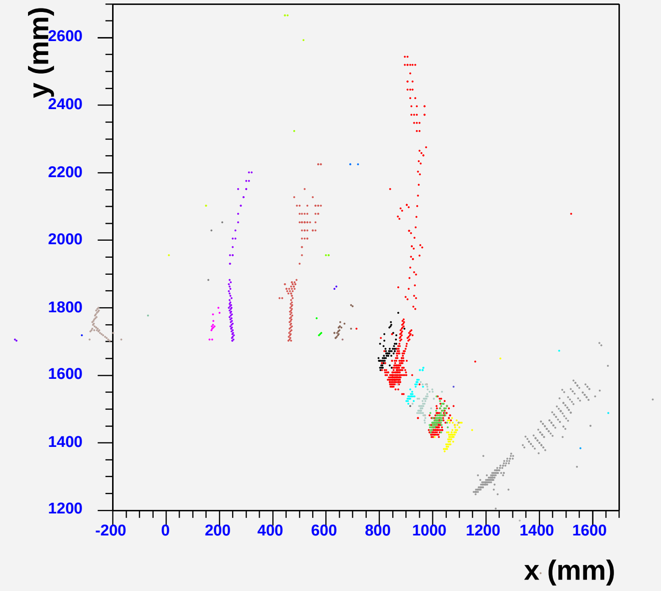
after mst clustering



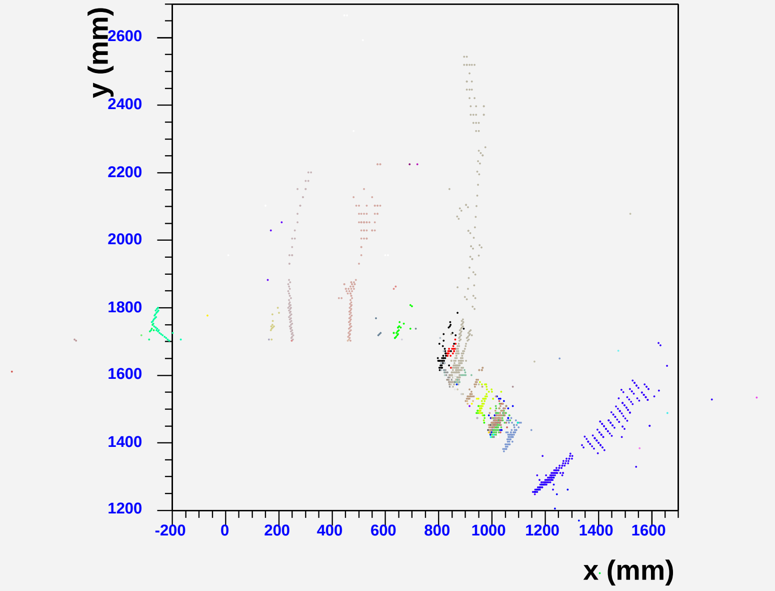
final reconstructed clusters



true clusters



after mst clustering



final reconstructed clusters

Technicalities

▶ **so far, the code consists of**

- : 6 classes, about 60 methods
- : still in early stages of development, expect to evolve and grow fast

▶ **design issues**

- : to be a flexible working base
- : try to avoid over-specialization, it should be easy to implement different clustering algorithms
- : optimise design to improve robustness

Performance evaluation

▶ **first attempt to quantify clustering efficiency**

: use V.Morgunov, A.Raspereza concept of "separation quality"

: shoot pairs of particles with fixed energy and at given distance from each other, how well the algorithm reconstructs the showers

: "separation quality = fraction of events in which reconstructed energy of the shower lies in the range $E_{true} \pm 3\sigma$, where σ is the nominal energy resolution of the shower without a close by shower"

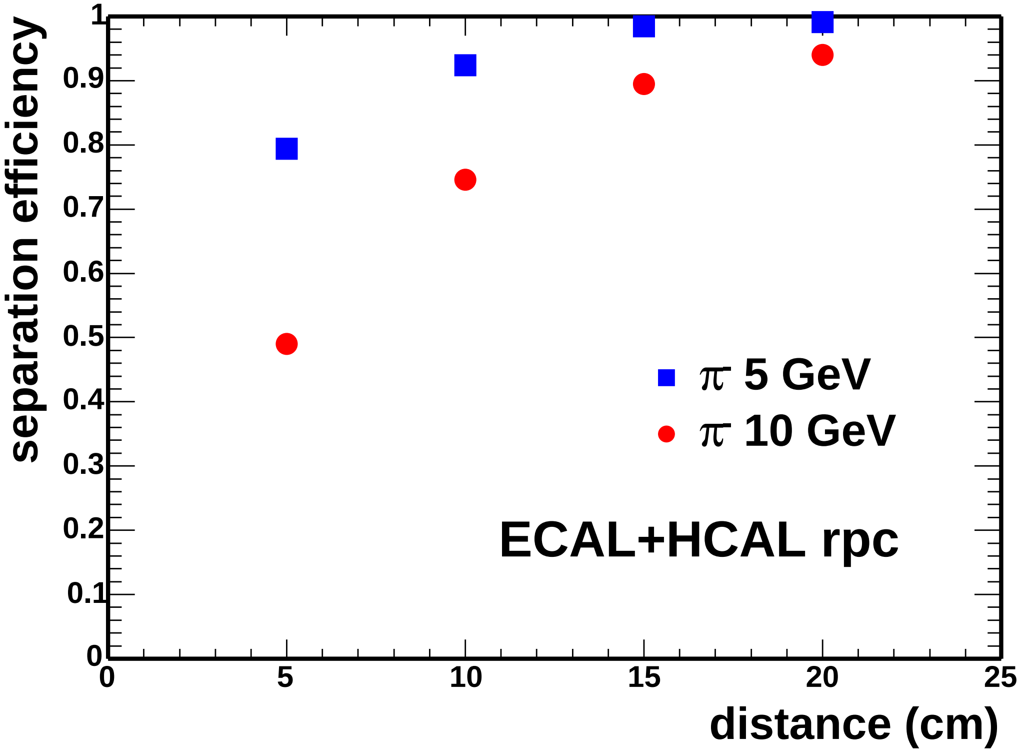
Performance evaluation

► .

: π^- pairs at 10 GeV or 5 GeV on CALICE ECAL+HCAL RPC prototypes

: ECAL, HCAL cellsize 1 × 1 cm^2, cell threshold = 0.5 mip

: satisfactory performance given the fact that the algorithm is seedless and both ECAL and HCAL hits are treated as digital in clustering



Summary

- **clustering with MST + extensions**
  - : clustering algorithm evolves to combine and exploit the advantages of top-down (MST clustering) and bottom-up approach to the problem
  - : simple first efficiency test satisfactory
  - : some design issues must be worked out to avoid problems later when complexity grows
- **to do ...**
  - : do refinements – check efficiency with realistic jets/events
  - : implement particle identification
  - : evaluate overall reconstruction performance