OPTIMAL JET FINDER

solution of the problem of jet definition

Ernest Jankowski

University of Alberta

in collaboration with

D. Grigoriev • F. Tkachov

edited and presented by F.T.

Plan of the talk

- I explain what the Optimal Jet Definition is
- and how its implementation works
- I compare the Optimal Jet Definition to the cone and k_T algorithms
- I present the results of a benchmark MC test based on fully hadronic decays of W-boson pairs

Introduction

- most common jet finding algorithms used in data analysis are cone and k_T binary recombination algorithm
- I will present so called Optimal Jet Definition (OJD) proposed by F. Tkachov
- a short introduction to the subject is hep-ph/0301185 (Phys. Rev. Lett., in print)
- FORTRAN 77 implementation of OJD called Optimal Jet Finder (OJF) is described in hep-ph/0301226 (Comp. Phys. Commun., in print)
- OJF has many advantages in comparison with the cone and k_T

Recombination matrix z_{aj}

HEP event: list of **particles** $p_a, \quad a = 1, 2, \ldots, n_{\text{parts}}$
(partons • hadrons • calorimeter cells • towers • preclusters)

recombination matrix $\{z_{aj}\}_{n_{\text{parts}} \times n_{\text{jets}}}$

$$q_j = \sum_{a=1}^{n_{\text{parts}}} z_{aj} \, p_a$$

the 4-momentum q_j of the j-th jet expressed by 4-momenta p_a of the particles

result: list of **jets** $q_j, \quad j = 1, 2, \ldots, n_{\text{jets}}$

Recombination matrix z_{aj}

$$q_j = \sum_{a=1}^{n_{\text{parts}}} z_{aj}\, p_a$$

the 4-momentum q_j of the j-th jet expressed by 4-momenta p_a of the particles (a=1,2,...,n_{parts})

$$z_{aj} \geq 0$$

the fraction of the energy of the a-th particle can positive only

$$\bar{z}_a \equiv 1 - \sum_{j=1}^{n_{\text{jets}}} z_{aj}$$

the fraction of the energy of the a-th particle that does not go into any jet

$$\bar{z}_a \geq 0$$

i.e. no more than 100% of each particle is assigned to jets

Optimal Jet Definition

- any allowed value of the recombination matrix $\{z_{aj}\}$ describes some jet configuration

- the desired optimal jet configuration is the one that **minimizes** some function $\Omega(\{z_{aj}\})$

- the details of Ω are different for CM lepton-lepton collisions (spherical kinematics) and collisions involving hadrons (cylindrical kinematics) where the role of transverse energy $E^{\perp}$ is emphasized

Optimal Jet Definition – spherical kinematics

$$\Omega(\{z_{aj}\}) = \frac{4}{R^2} \sum_{j=1}^{n_{\text{jets}}} \underbrace{\sum_{a=1}^{n_{\text{parts}}} z_{aj} E_a \sin^2 \frac{\theta_{aj}}{2}}_{\text{width of the } j\text{-th jet}} + \underbrace{\sum_{a=1}^{n_{\text{parts}}} \bar{z}_a E_a}_{\text{energy outside jets}}$$

$$q_j \equiv (E_j, \mathbf{q}_j) \equiv \sum_{a=1}^{n_{\text{parts}}} z_{aj} p_a$$

θ_{aj} is the angle between the a-th particle and j-th jet

E_a is the energy of the a-th particle

R>0 is a parameter with a similar meaning as the cone radius

Optimal Jet Definition - cylindrical kinematics

$$\Omega(\{z_{aj}\}) = \frac{4}{R^2}\sum_{j=1}^{n_{\text{jets}}}\sum_{a=1}^{n_{\text{parts}}} z_{aj}E_a^{\perp}\left(\sinh^2\frac{\eta_a-\eta_j}{2}+\sinh^2\frac{\varphi_a-\varphi_j}{2}\right)+\sum_{a=1}^{n_{\text{parts}}}\bar{z}_a E_a^{\perp}$$

$$\eta_j \equiv \frac{\sum_{a=1}^{n_{\text{parts}}} z_{aj}E_a^{\perp}\eta_a}{\sum_{a=1}^{n_{\text{parts}}} z_{aj}E_a^{\perp}}$$

$$E_a^{\perp} \equiv \sqrt{\left(p_a^{\mathrm{x}}\right)^2+\left(p_a^{\mathrm{y}}\right)^2}$$

$$q_j \equiv \left(E_j, \mathbf{q}_j\right) \equiv \sum_{a=1}^{n_{\text{parts}}} z_{aj}p_a$$

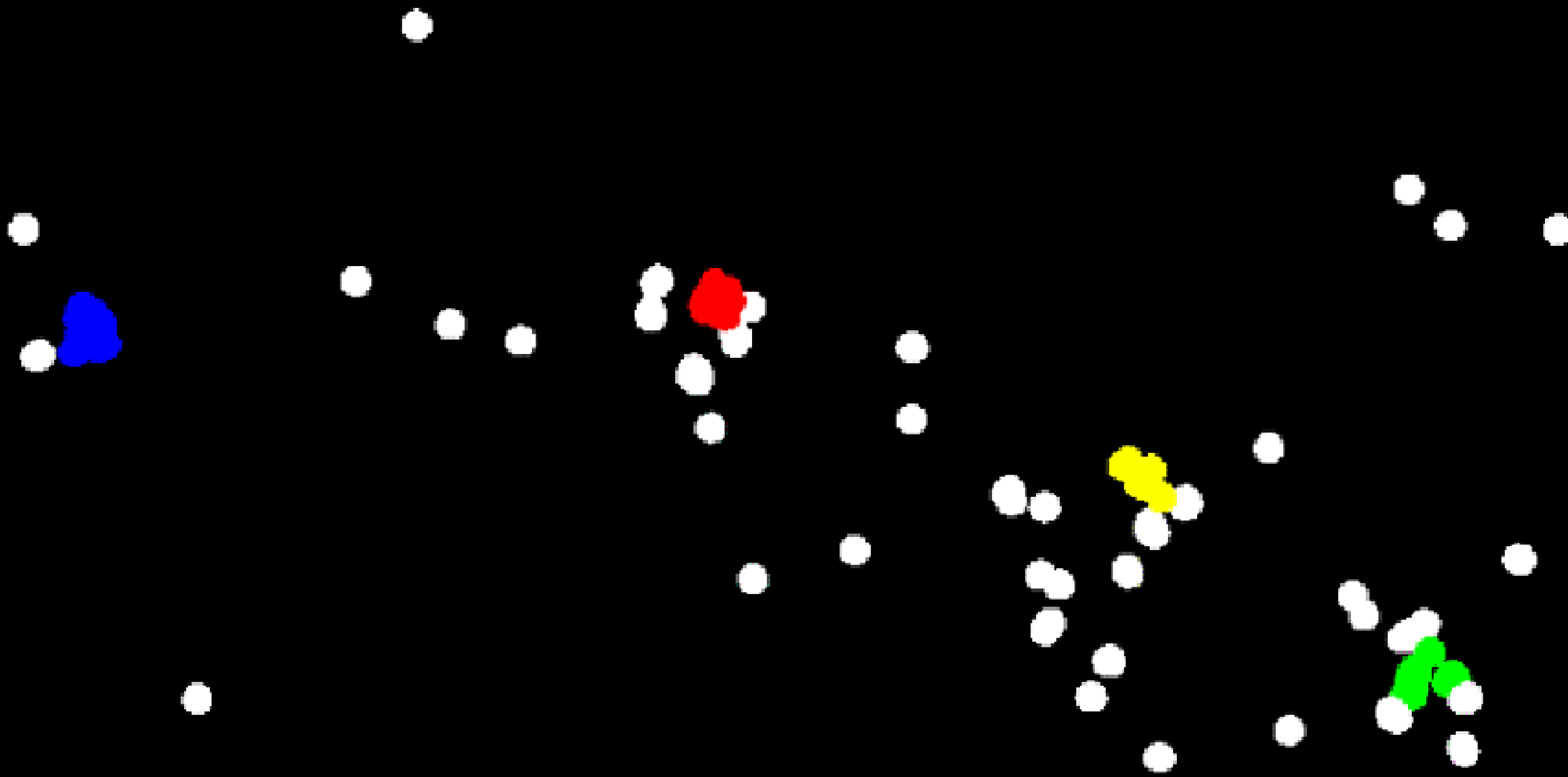
$$\frac{\mathbf{q}_j^{\perp}}{\left|\mathbf{q}_j^{\perp}\right|} \equiv \left(\cos\varphi_j, \sin\varphi_j\right)$$

Optimal Jet Finder – fixed number of jets

- desired optimal jet configuration corresponds to minimum of $\Omega(\{z_{aj}\})$
- the program finds a minimum iteratively using a ~~simple~~ gradient-based method
- facilitated by analytical formulas for gradient
- start with some candidate minimum (the initial jet configuration) and descend into a local minimum in subsequent iterations
- the initial jet configuration may be completely random

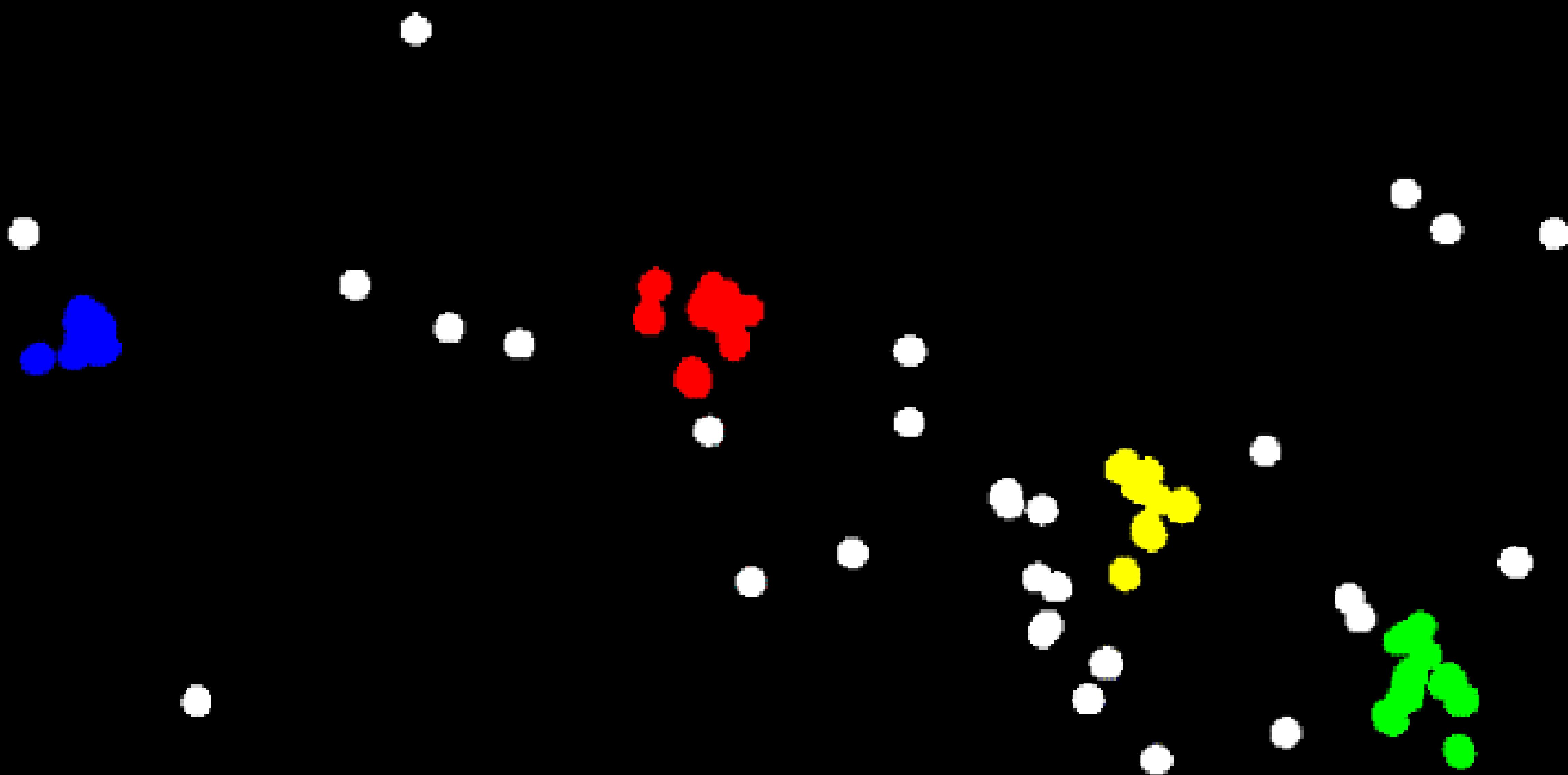
$R = 0.1$

algorithm finds the most energetic particles



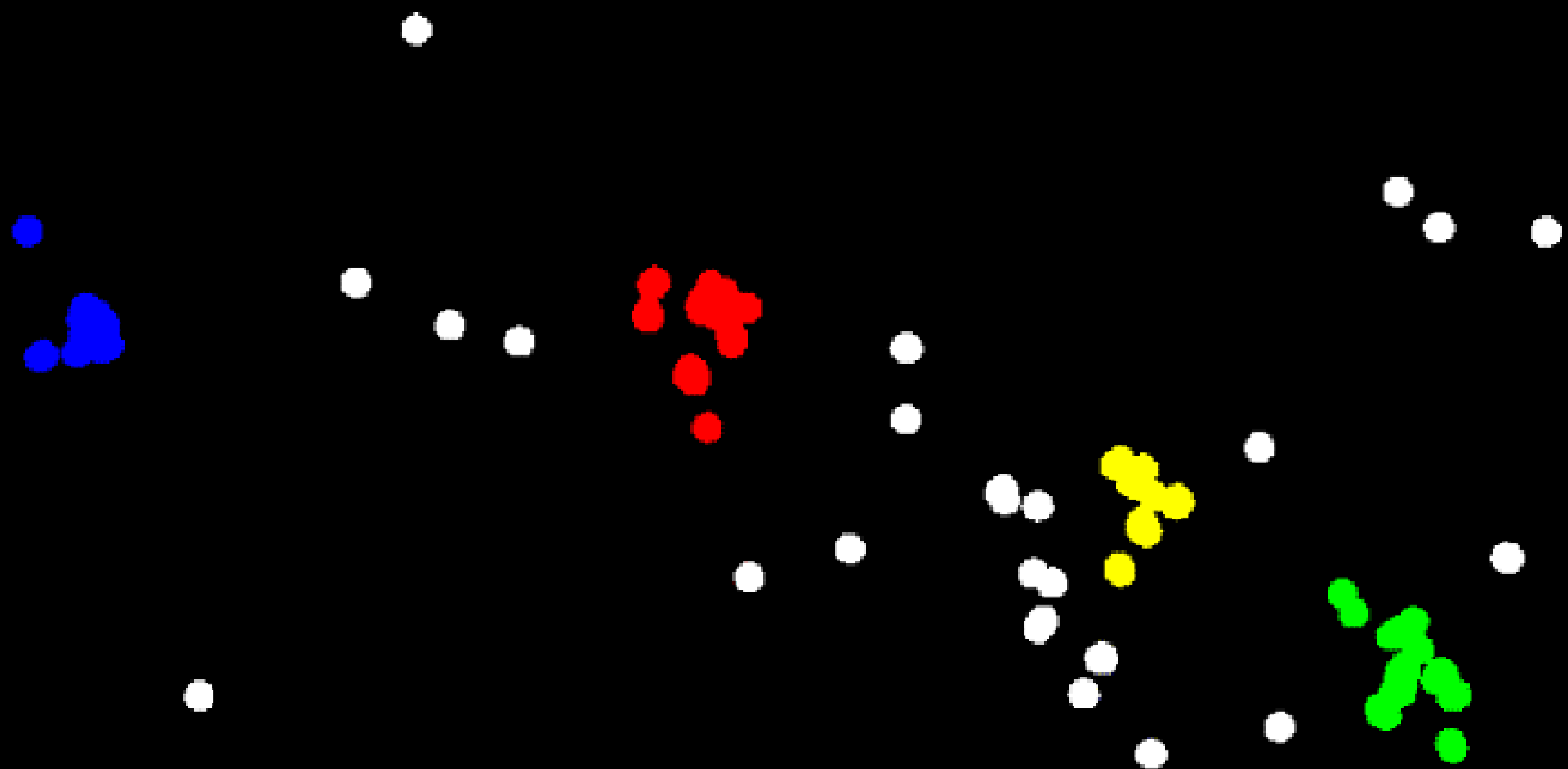
$\rightarrow \varphi\,[0^\circ - 360^\circ]$ $\uparrow \theta\,[0^\circ - 180^\circ]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

$R = 0.2$



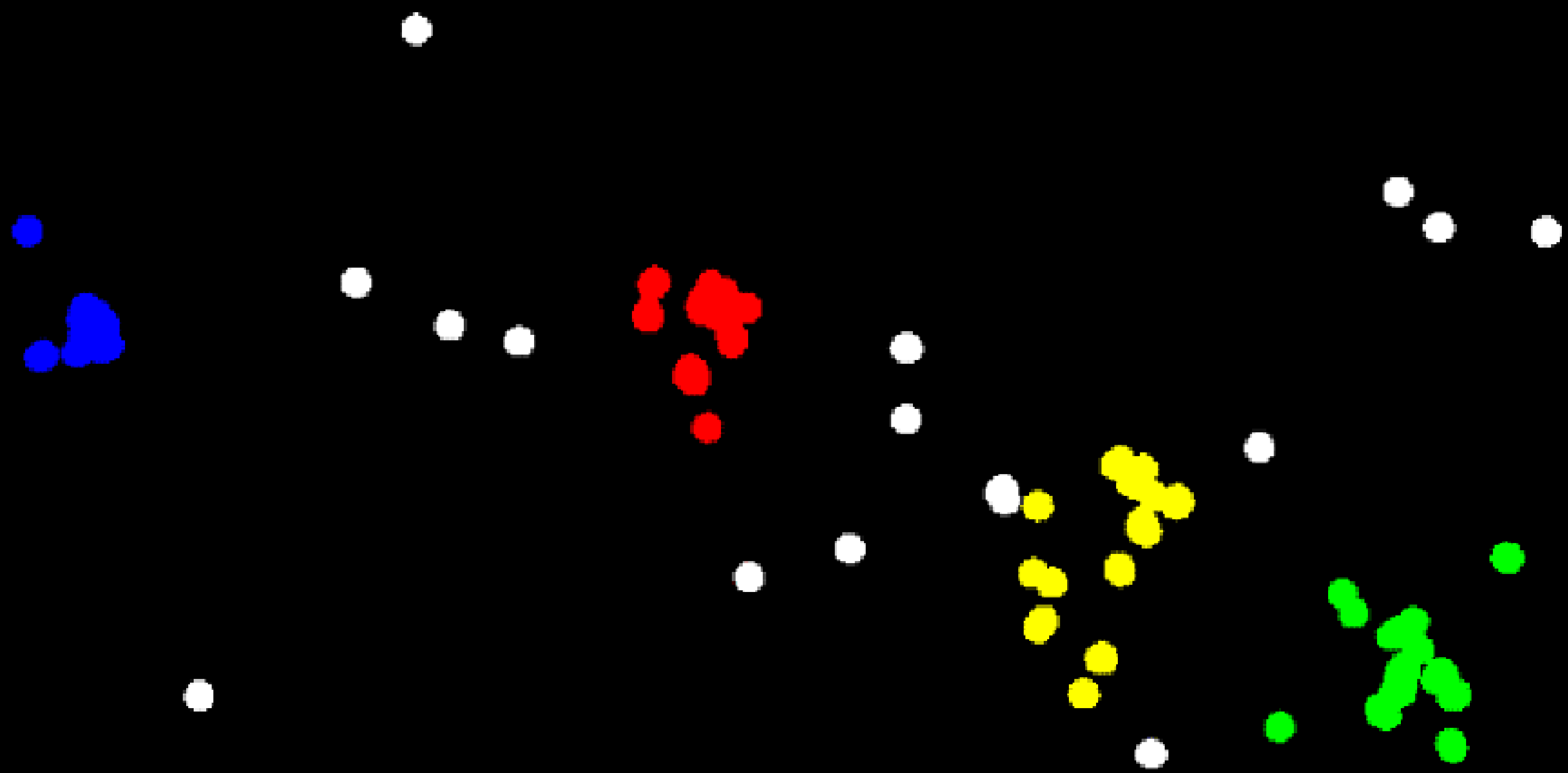
$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

$R = 0.3$



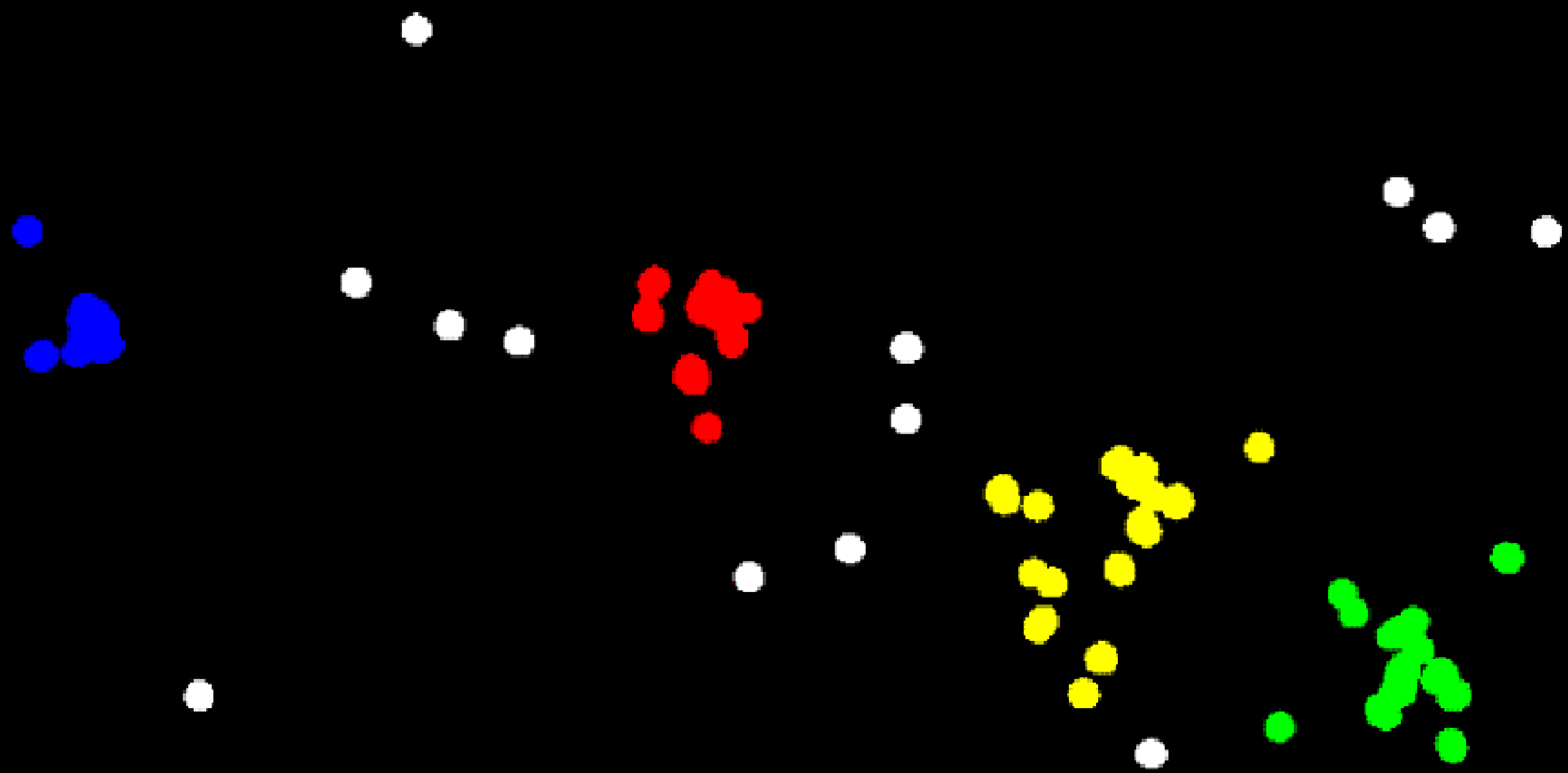
$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

$R = 0.4$



$\rightarrow \varphi\,[0^\circ - 360^\circ]$ $\uparrow \theta\,[0^\circ - 180^\circ]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

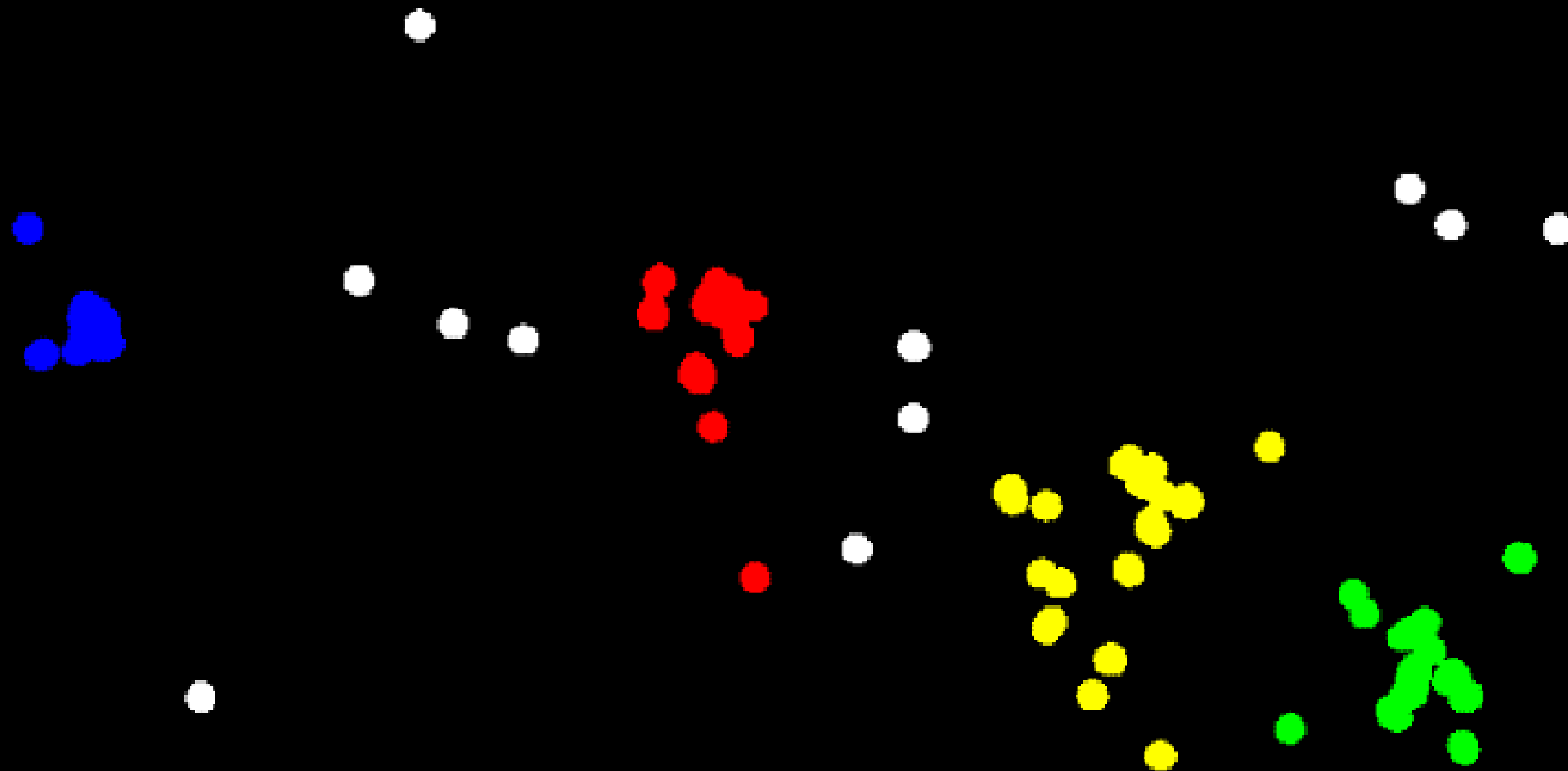
$R = 0.5$



$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

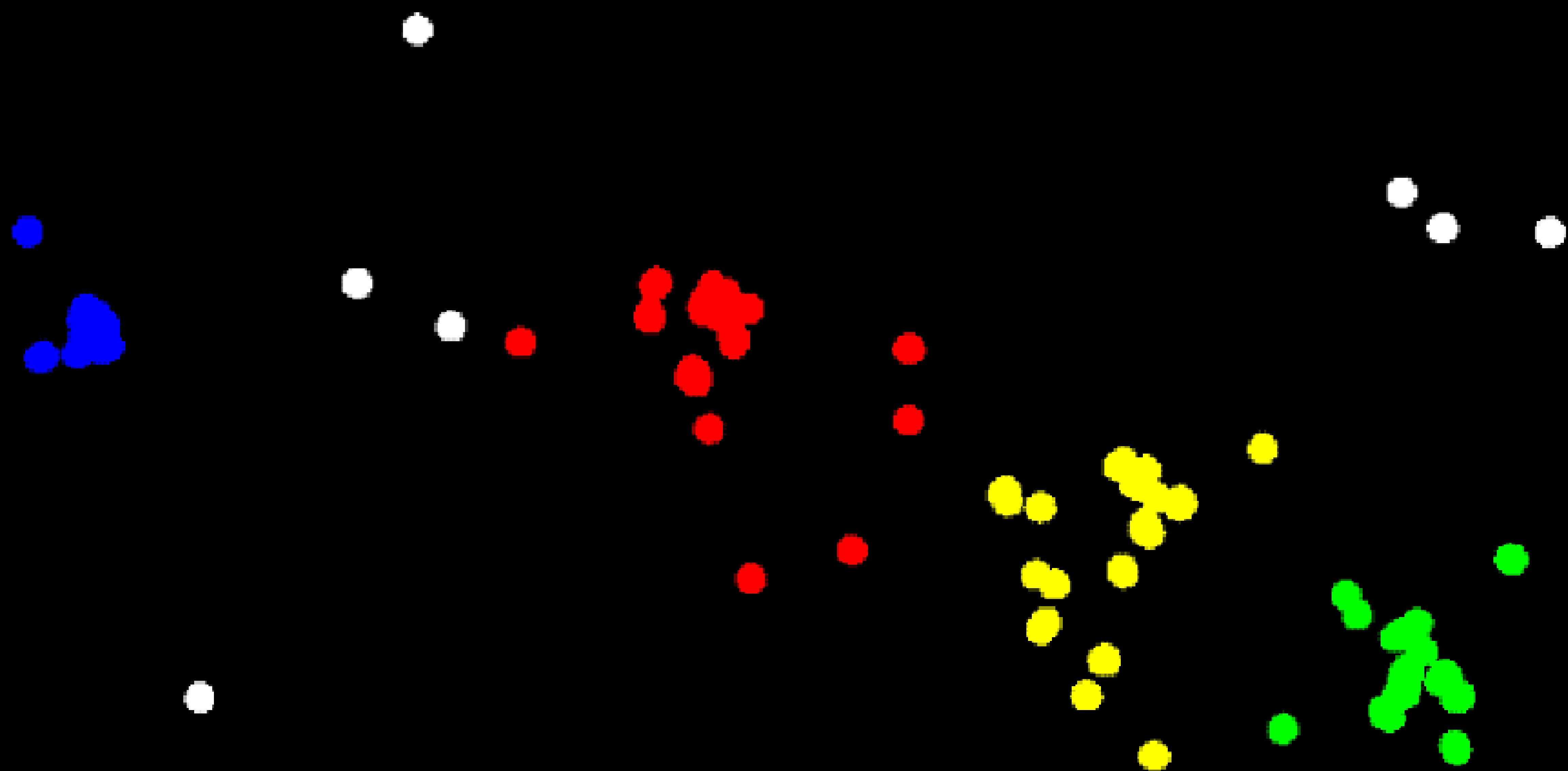
$R = 0.6$

notice convex jet shapes



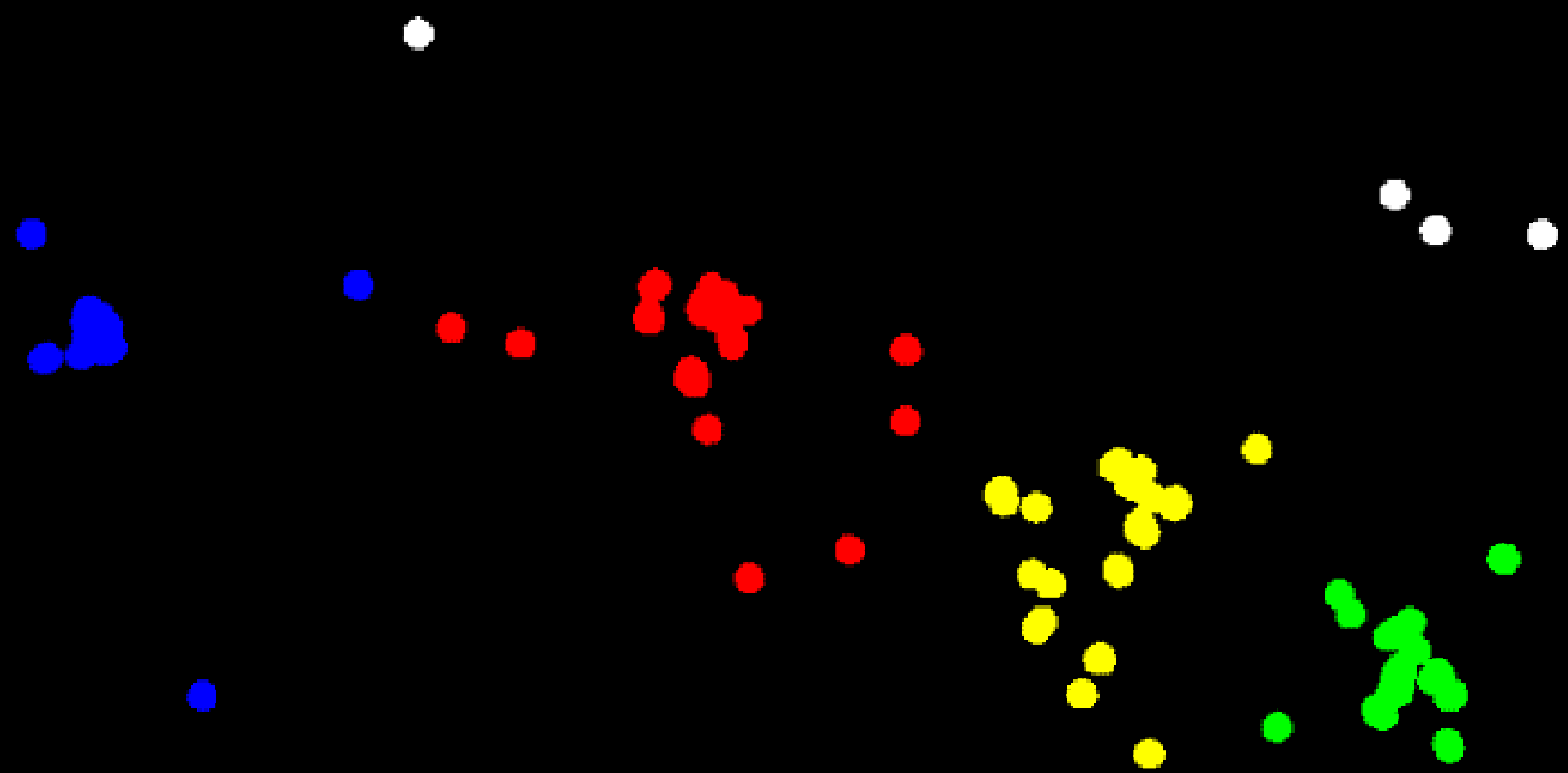
$\rightarrow \varphi\,[0^\circ\text{-}360^\circ]$ $\uparrow \theta\,[0^\circ\text{-}180^\circ]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

$R = 0.7$



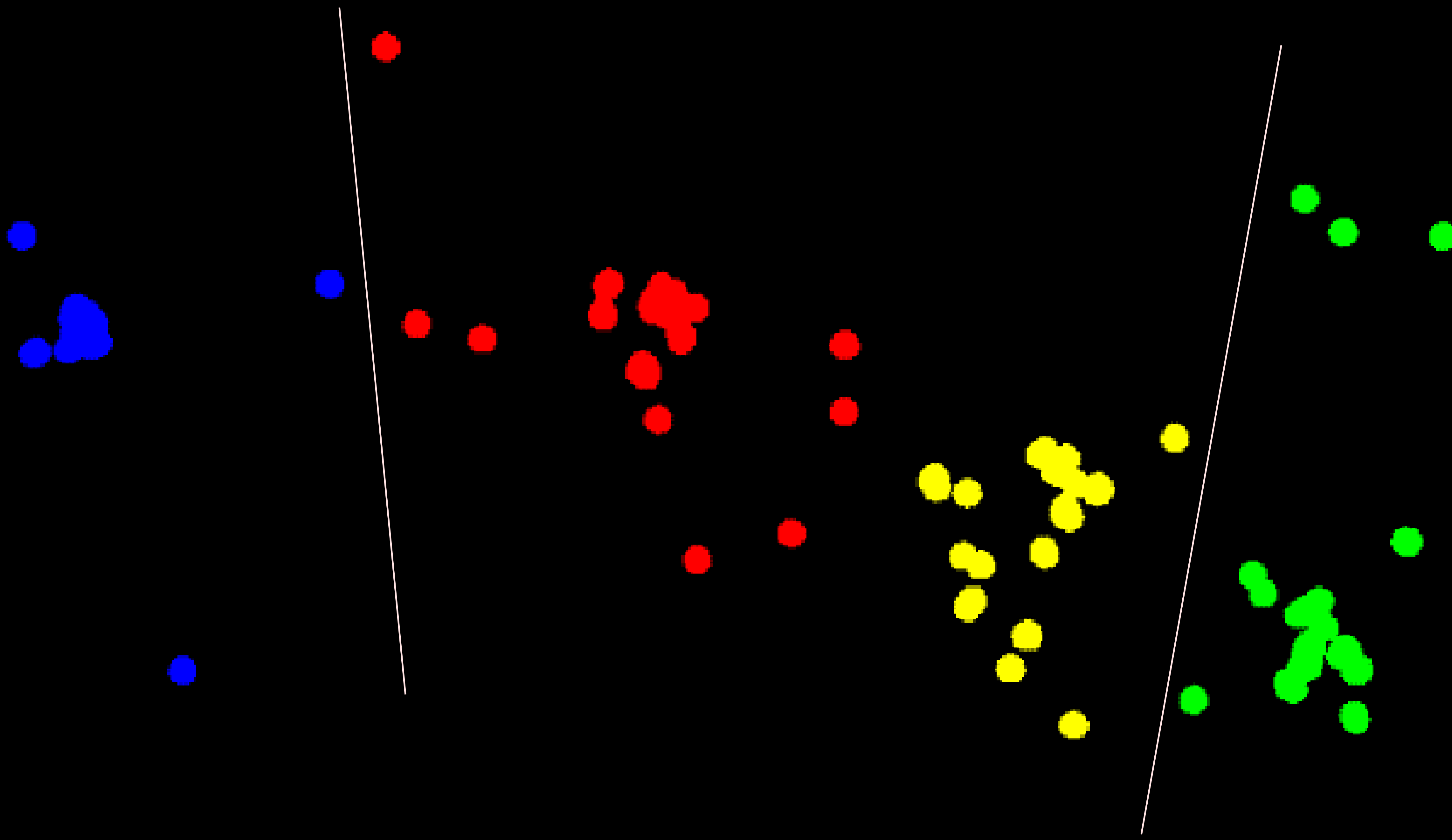
$\rightarrow \varphi$ [0° - 360°] $\uparrow \theta$ [0° - 180°] $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

$R = 0.9$



$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

$R = 1.1$

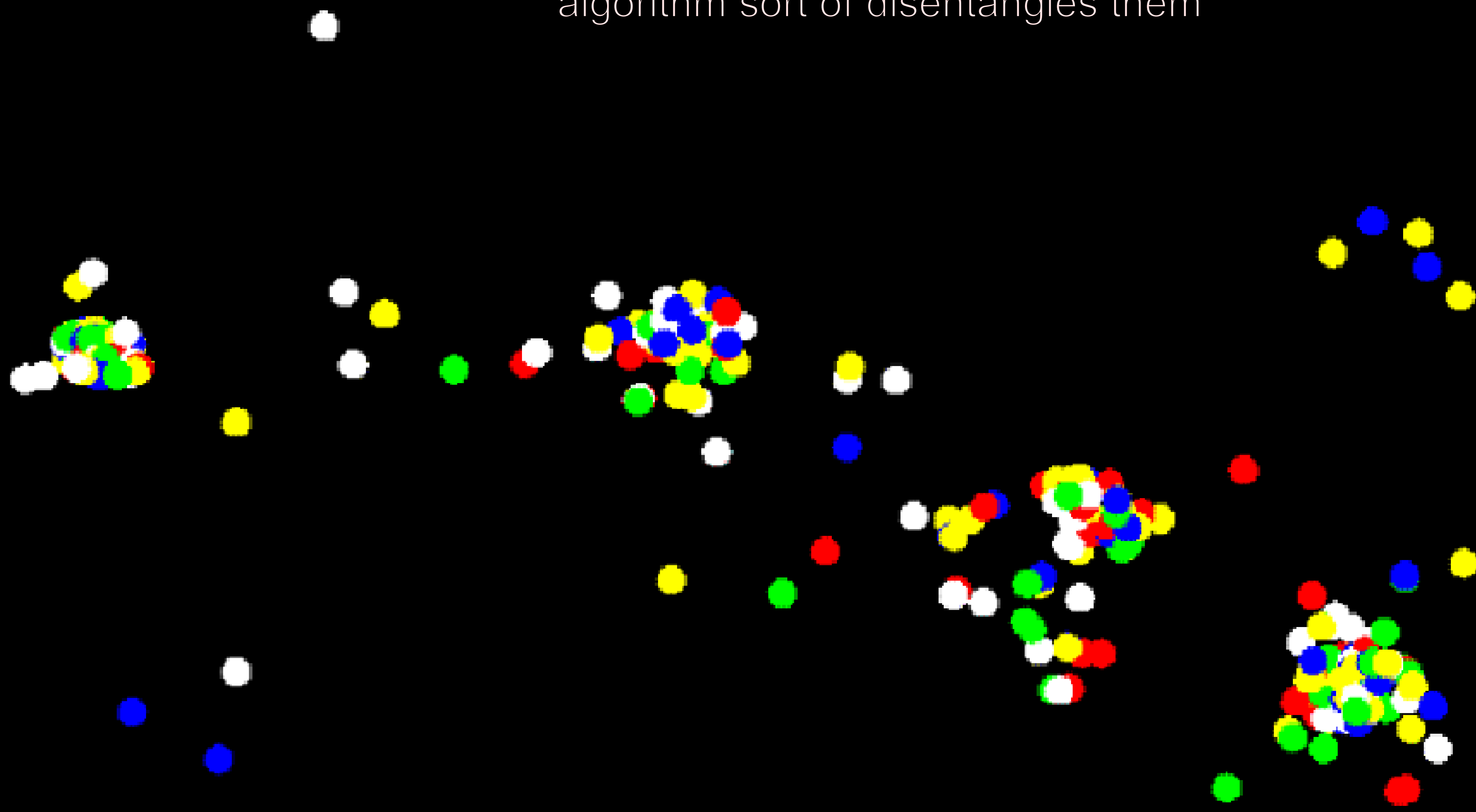


$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

iteration = 0

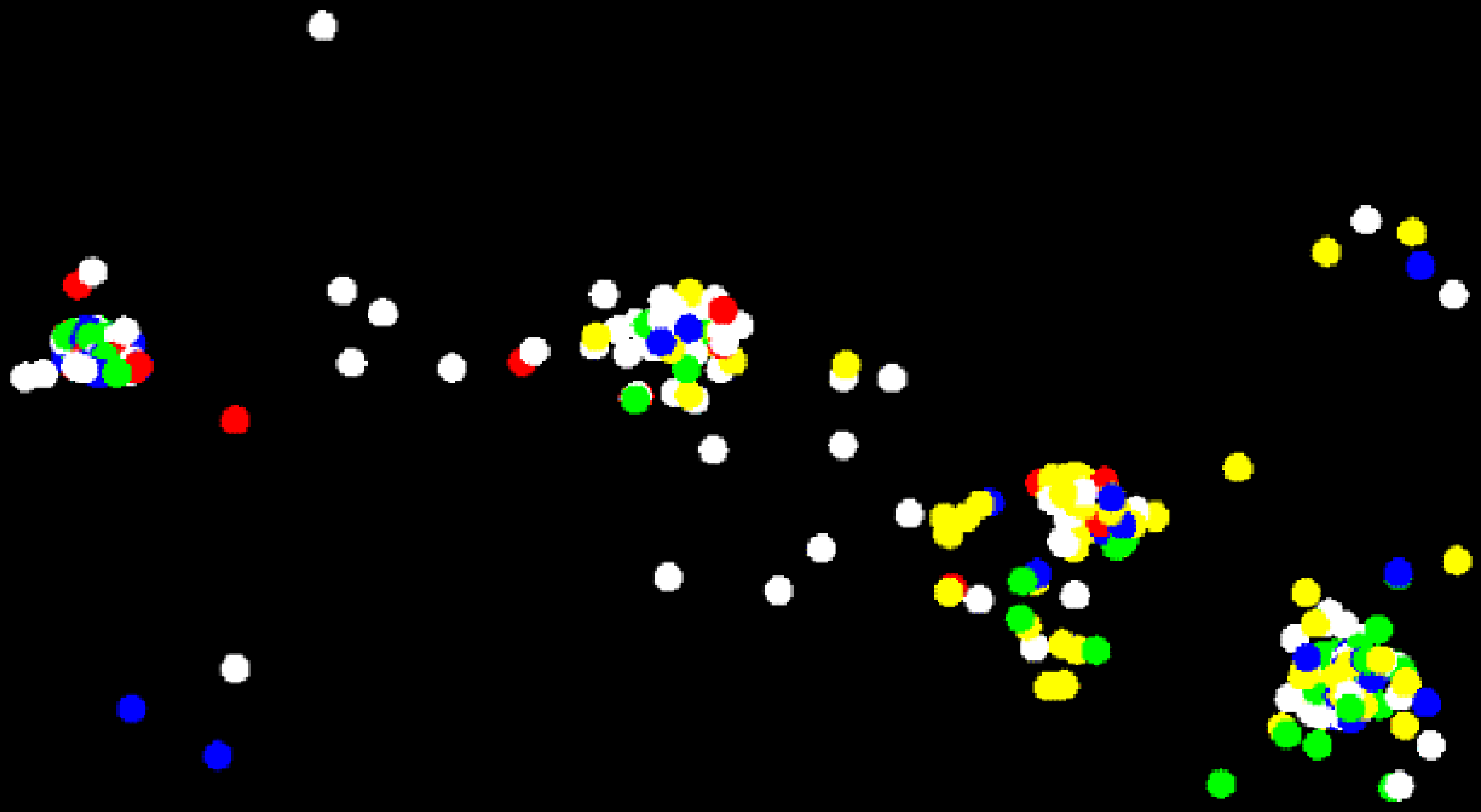
jets are entangled

algorithm sort of disentangles them



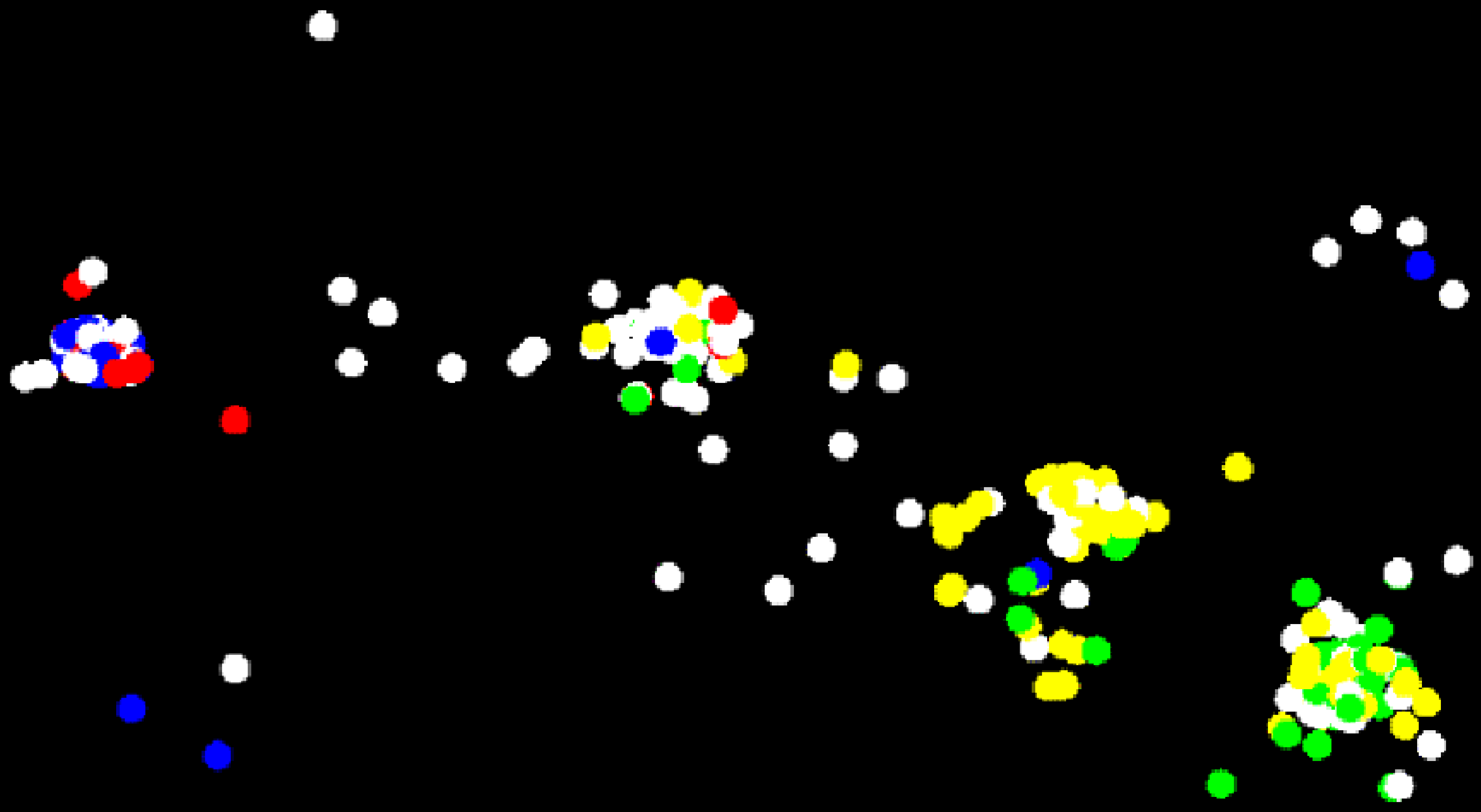
$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

iteration = 1



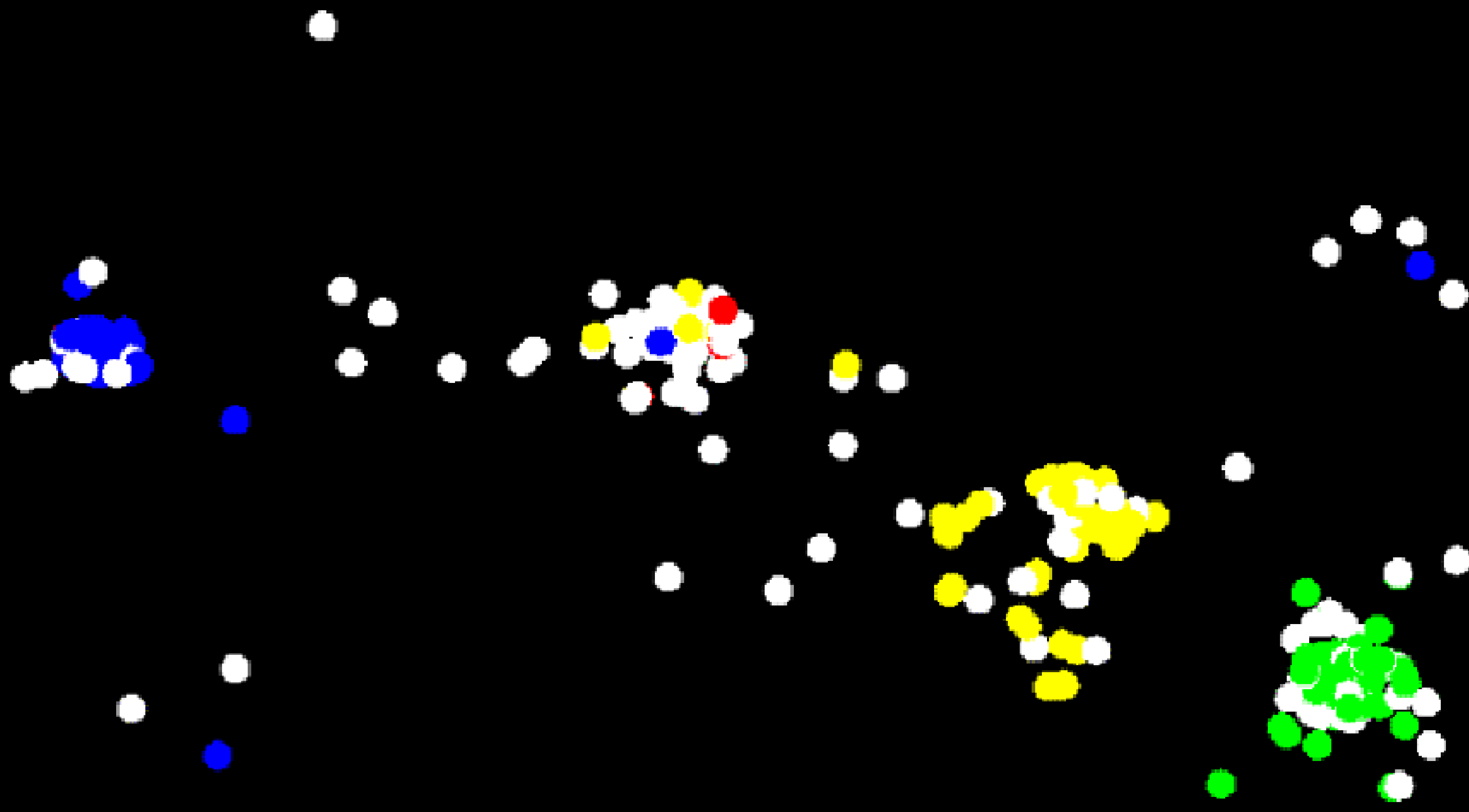
→ φ [0° - 360°]   ↑ θ [0° - 180°] $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets   180 GeV

iteration = 2



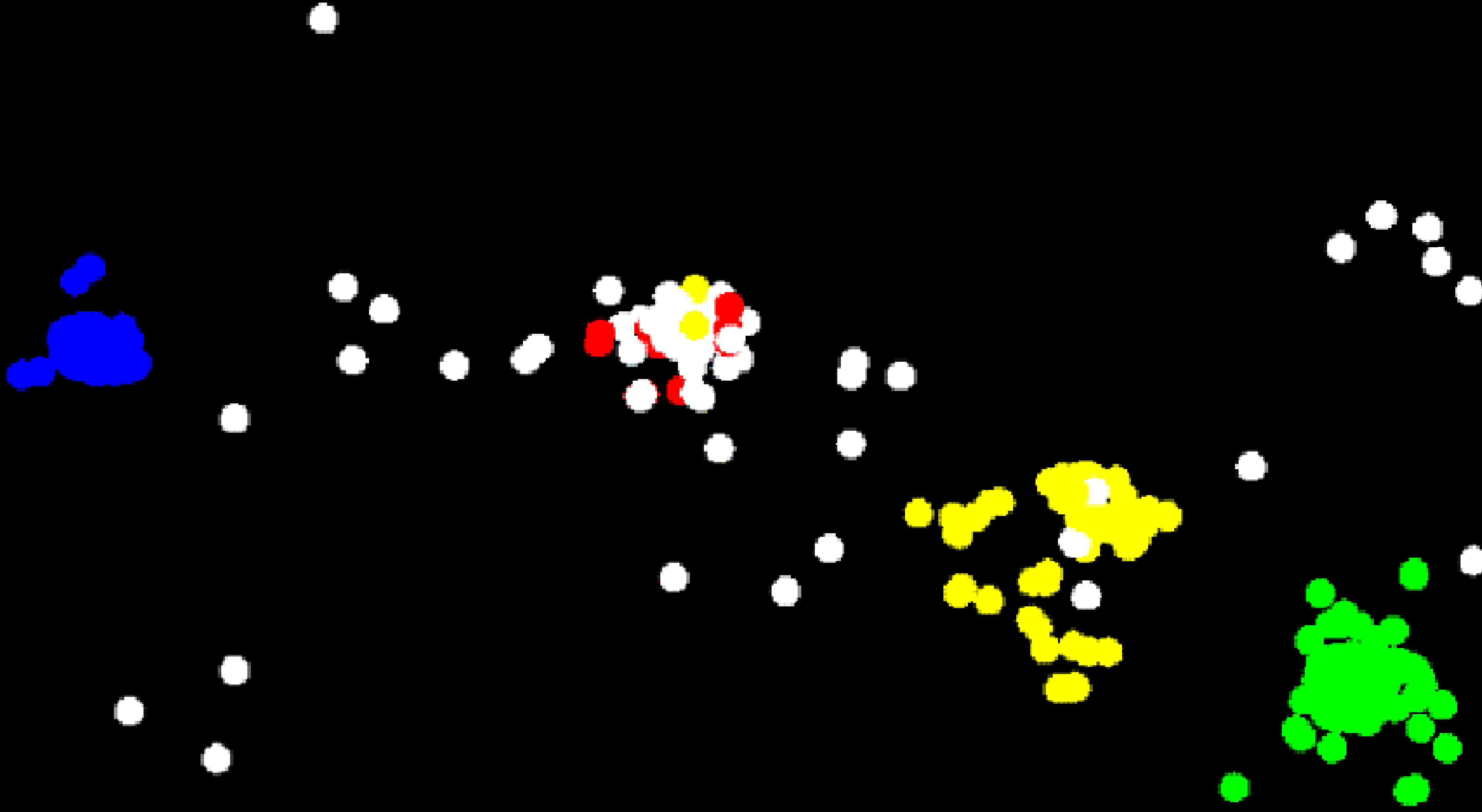
$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

iteration = 3



$\rightarrow \varphi$ [0° - 360°] $\uparrow \theta$ [0° - 180°] $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

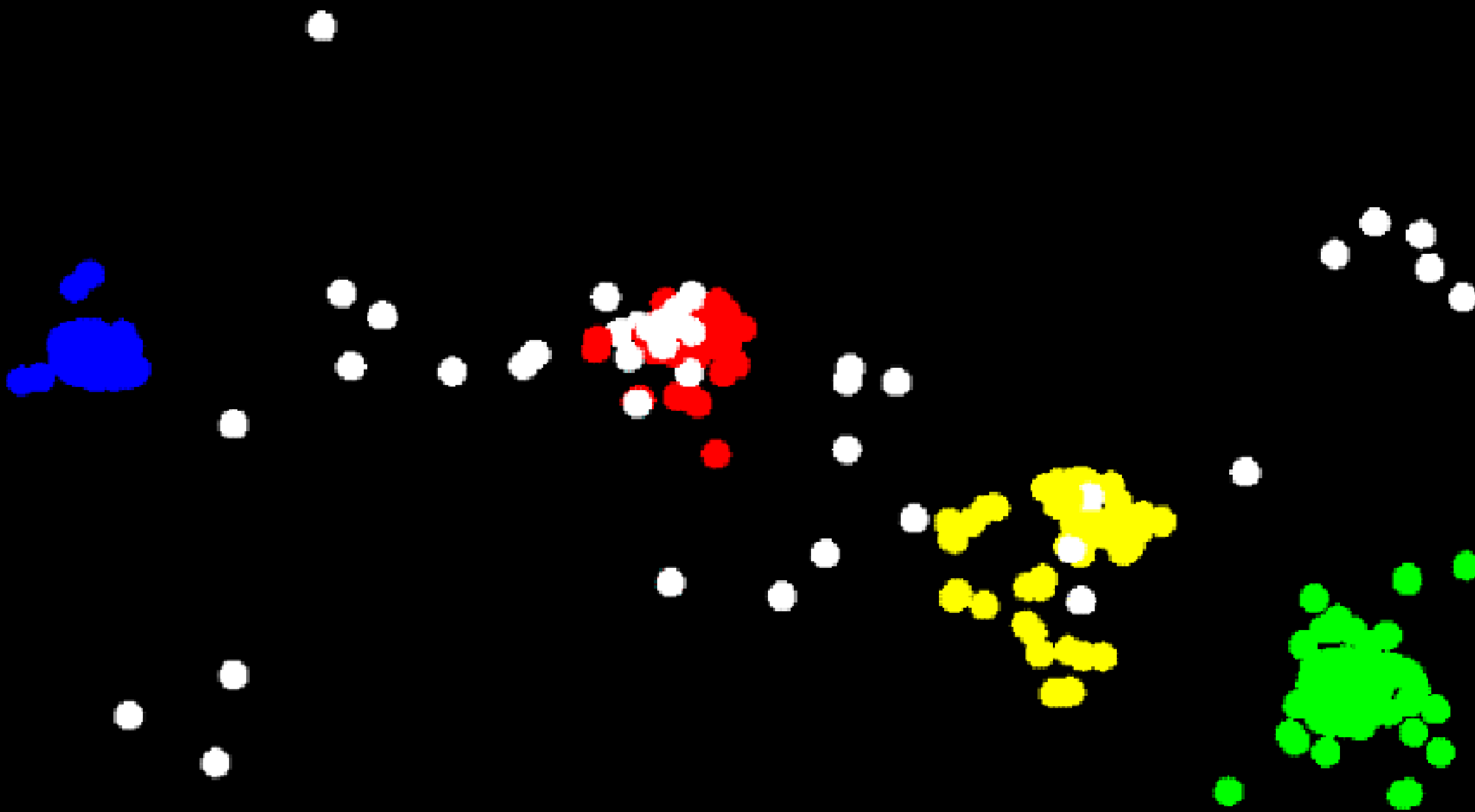
iteration = 4



$\rightarrow \varphi\,[0° - 360°]$ $\uparrow \theta\,[0° - 180°]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

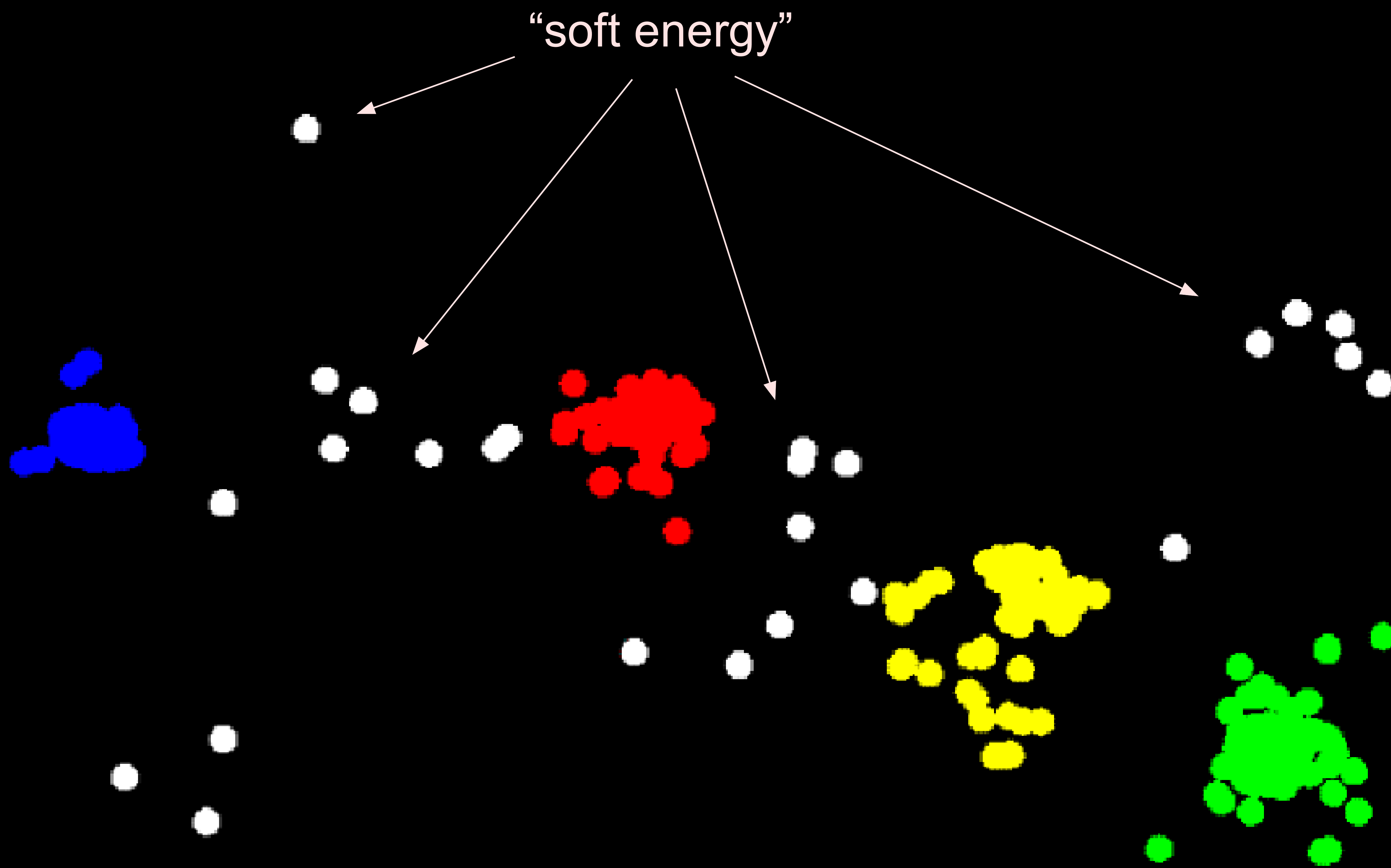
iteration = 5

jets sort of condense out of the event



$\rightarrow \varphi$ [0° - 360°] $\uparrow \theta$ [0° - 180°] $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets   180 GeV

iteration = 6



$\rightarrow \varphi\,[0^\circ - 360^\circ]$ $\uparrow \theta\,[0^\circ - 180^\circ]$ $e^+e^- \rightarrow W^+W^- \rightarrow$ 4 jets 180 GeV

Optimal Jet Finder – number of tries

- each time the programs starts with a random configuration, it finds a local minimum which does not need to be the global minimum of $\Omega(\{z_{aj}\})$
- this is similar to how the result of cone algorithm depends on the initial position for the cone iterations
- but here in contrast with the cone algorithm, we know which local minimum we should choose: the one that gives the smallest value of Ω

smallest loss of physical information

Optimal Jet Finder – number of tries

- in order to find the global minimum or increase the probability of finding it, the program tries 3-10 different random initial configurations

- OJF parameter: number of tries n_{tries}

- it was sufficient to take $n_{\text{tries}} \leq 10$ (even $n_{\text{tries}} \sim 3$) in the cases we studied

- compromise between quality of jets found and computing time used, fully controlled

OJF – number of jets to be determined

1. assume some small positive parameter ω_{cut}, which is analogous to the jet resolution parameter y_{cut} in binary recombination algorithms

2. start with (for example) $n_{jets}=1$

3. find the best configuration with the number of jets equal to n_{jets} as described previously (starting from n_{tries} different initial configurations and choosing the best configuration)

4. check if $\Omega < \omega_{cut}$

5. if **so**, this is the **final** jet configuration and the final number of jets is n_{jets}

6. if **not**, increase n_{jets} by 1 and go to point 3

OJF and cone algorithm

- in OJF shape of jets is determined dynamically based on the energy flow in the event

  -- no problem of jet overlaps!

- jets are not regular cones as they are represented in the cone algorithm

  but still nice convex shapes,
  good for studies of detector effects (energy corrections etc.)
  -- unlike kT jets

OJF and k_T

- OJF is much faster than k_T if a large number of calorimeter cells has to be analyzed
- average time per event ~ N_{cells} (number of cells in the event) for OJF whereas it is ~ N^3_{cells} for k_T
- it finds more regular jets that k_T
- k_T merges only 2 particles at a time whereas OJF takes into account the global structure of the energy flow in the event, i.e.
- jet configuration is found from the momenta of **all** particles in the event

Benchmark test: W-boson mass extraction

- benchmark test based on the W-boson mass extraction from the process

$$e^+e^- \rightarrow W^+W^- \rightarrow q_1\bar{q}_2q_3\bar{q}_4 \rightarrow 4 \text{ jets}$$

- modeled on the OPAL analysis (CERN-EP-2000-099)
- we compared OJF with JADE and Durham (k_T) algorithm (the best algorithm used by the OPAL collaboration)
- we obtained the same accuracy as Durham (still we did not explore all possibilities)
- we studied the speed of OJF and we found that it is much faster then Durham when large number of calorimeter cells needs to be analysed

Quality of jets

the first truly scientific comparison

ALGORITHM	statistical error of W-boson mass (corresponding to 1000 experimental events) based on Fisher's information [MeV] (±3)
Durham (k_T)	105
JADE	118
OJF	106

powerful concept of optimal observables -- throw away your NN for this kind of problems

FT arxiv.org/physics 2000

Speed of OJF and k_T

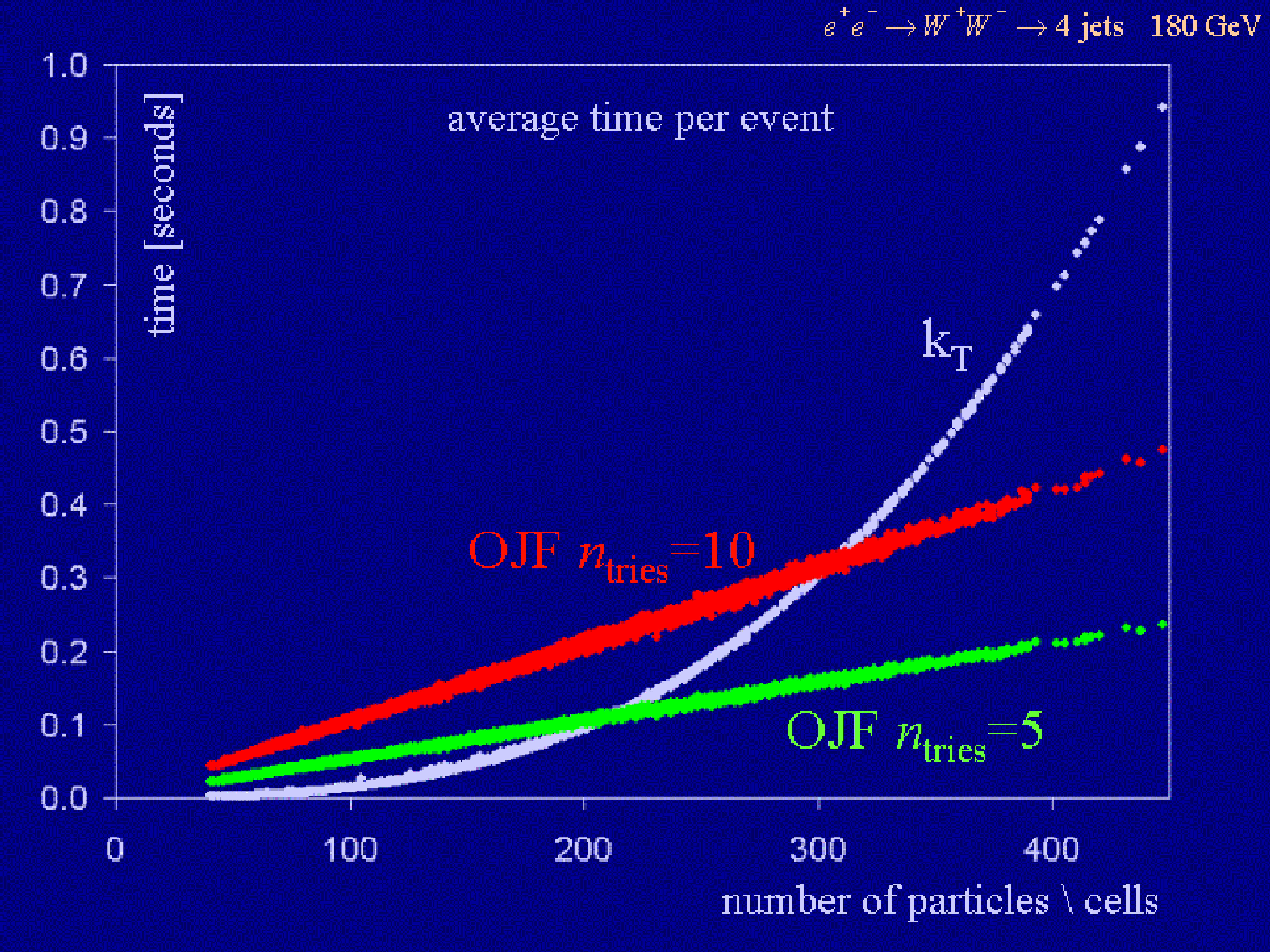
- average analysis time per event:

  $\approx 1.0 \cdot 10^{-4} \times N_{cells} \times n_{tries}$ for OJF

  $\approx 1.2 \cdot 10^{-8} \times N^3_{cells}$ for k_T

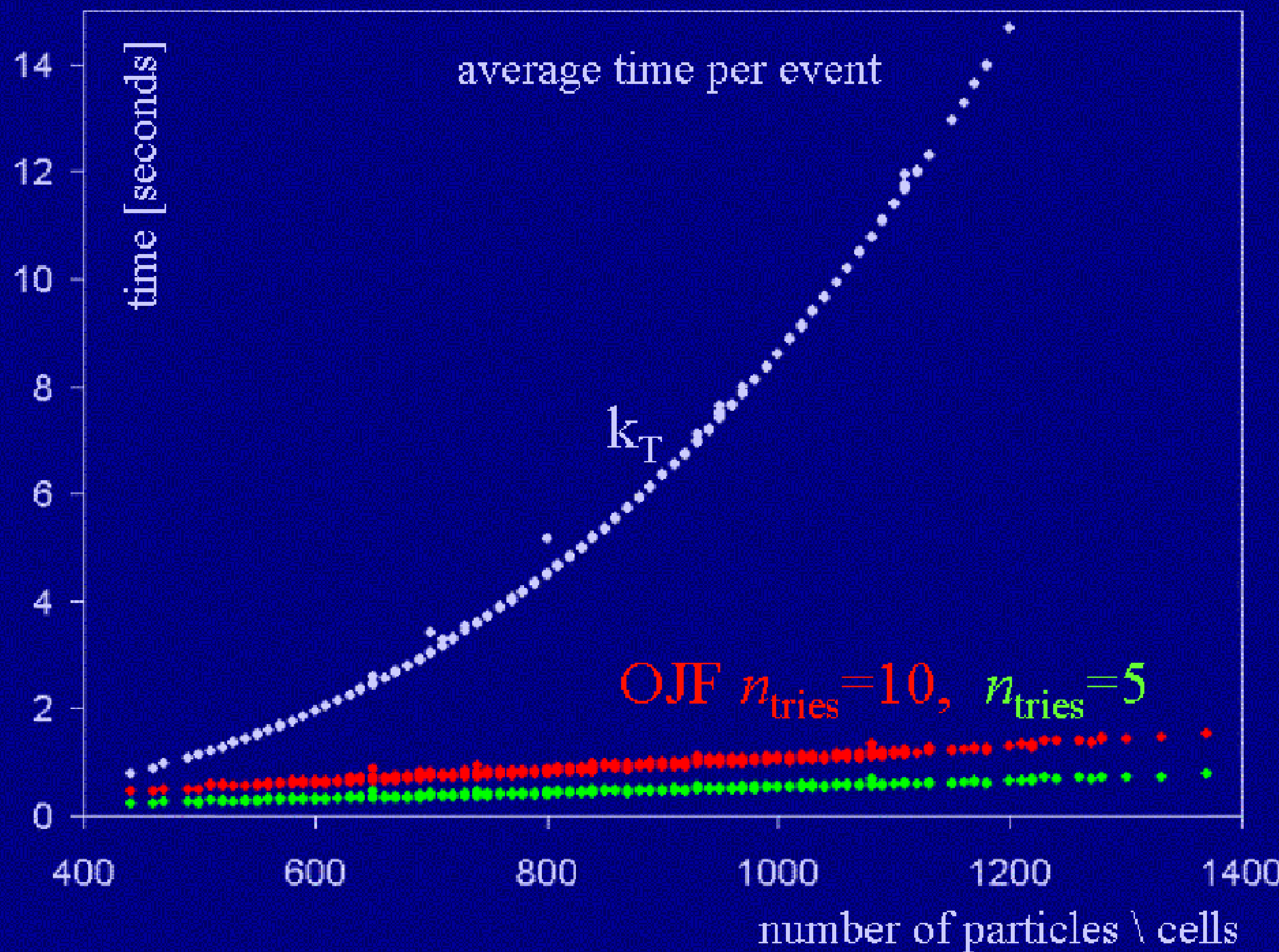
- k_T can not be applied directly at the level of calorimeter cells or even towers (D0 ~ 45000 cells, ATLAS ~ 200 000 cells)
- preclustering step is needed for k_T to reduce the initial data to ~ 200 preclusters
- how the preclustering affects measurements ?
- it may be possible to apply OJF directly at the level of cells or towers
- or study how the preclustering step affects measurements

  everything with OJF is under scientific control, no woodoo...



$e^+e^- \to W^+W^- \to$ 4 jets 180 GeV
average time per event
time [seconds]
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
100
200
300
400
k_T
OJF n_{tries}=10
OJF n_{tries}=5
number of particles \ cells

$e^+e^- \to W^+W^- \to 4$ jets 180 GeV



Summary

- I presented the Optimal Jet Finder
- based on the global energy flow in the event
- infra-red and collinear safe: no seed-related problems
- no overlapping jets-related problems
- returns additional numerical characteristics of the jet configuration found (so called dynamical width and soft energy) which may be helpful in construction of (quasi-) optimal observables in statistical problems
- much faster than k_T for a large number of input cells

F77 source: www.inr.ac.ru/~ftkachov/projects/jets/

Future

- comprehensive testing is necessary
- extend to pp collisions: Z/W + jets; multi-jet channels, such as tt→(6+) jets
- noise and pile-up
- further optimizations possible
- ATLAS note in preparation
- C++ version of the code to use in the ATHENA framework

  royal pain ... coming any day now ...

Workshop on OJF
May, 2004
U. of Alberta, Canada
contact:
Ernest Jankowski
ejankows@phys.ualberta.ca

software = a huge hidden cost in physics (as elsewhere);
something needs to be done...
but that is another story (workshop at CERN on March 10, 2004).

Two closely related developments:

Quasi-optimal observables see FT @ ACAT`02 and refs therein

POUZYRY see FT @ ACAT`03

a novel scheme to model arbitrary function from a random sample
based on the same mathematical and algorithmic ideas as OJF
= OJF-like pre-clustering + kernels

$$\frac{1}{N}\sum_n f_n \delta(x - x_n) \to \frac{1}{P}\sum_n \tilde{f}_p \delta(x - x_p) \quad \text{with } P << N$$

$$\to \frac{1}{P}\sum_n \tilde{f}_p \frac{1}{R^{\dim}} H\left(\frac{x - x_p}{R}\right)$$

pretty fast, works in any DIM,
does essentially the same as the popular variant of NN

$$NN \to g\left(\sum_k C_k g\left(A_{ki} x_i + B_k\right)\right)$$

but can handle *arbitrary* functions

C++ code will be made available (what`s possible without GC)