A New Jet Clustering Algorithm Using Vertex Information

Hiroki Kawahara
(The Univ. of Tokyo)

T.Suehara(Tokyo), K. Fujii(KEK),
A. Miyamoto(KEK)
& all ILD optimization, ILC-Asia physics group

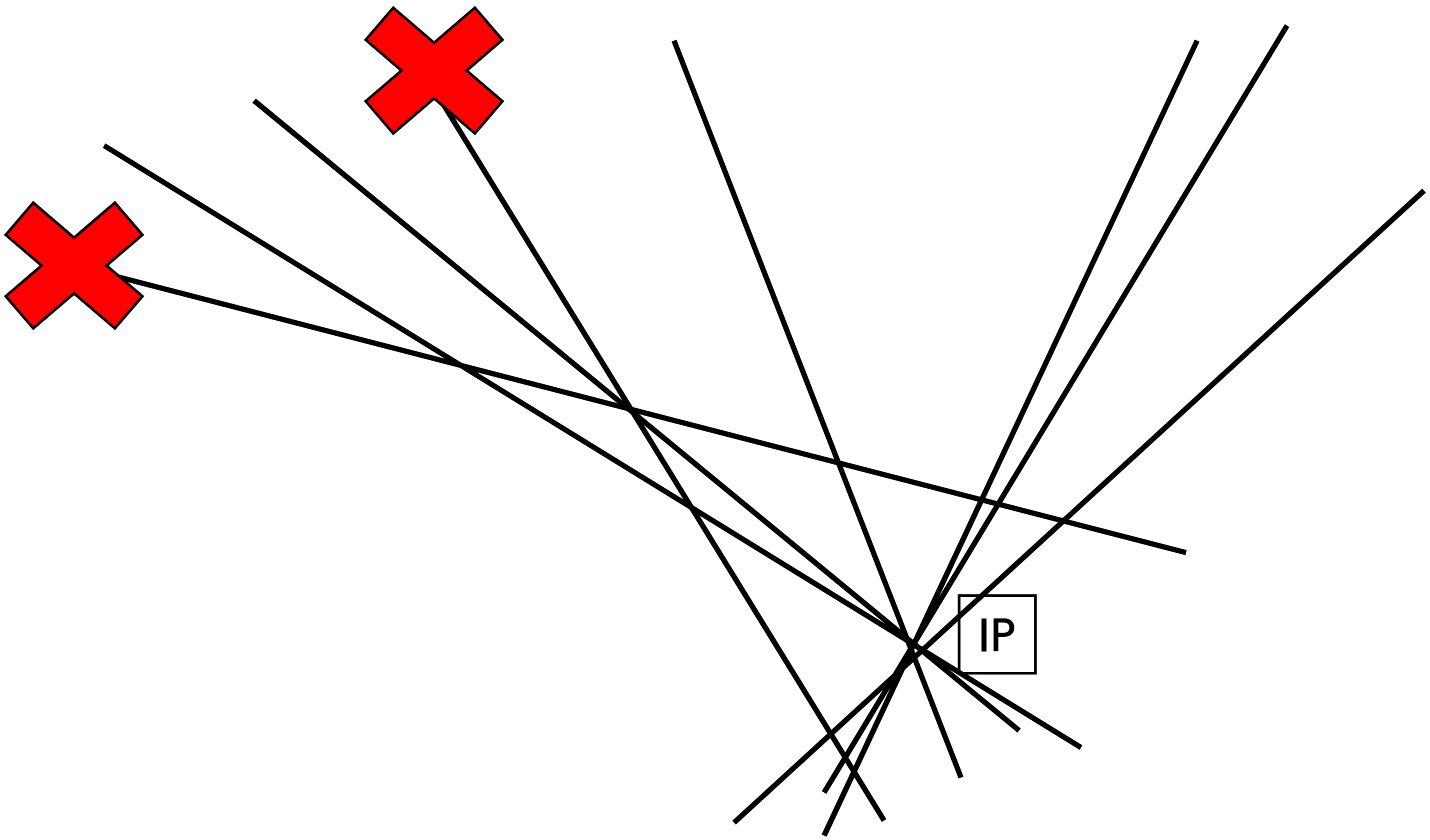
Motivation

- Most of important physics processes in the ILC have six or more jets.
  - E.g. ZHH: 6 jets with 4 b-jets
- Jet clustering algorithm with high performances for b-tagging is needed.

Existing Clustering Algorithm

1. List all reconstructed particles.
2. Calculate 'y' value of every pair of reconstructed particles using their energies and momentum.
   - Durham: $$y = \frac{2\min(E_1, E_2)^2(1-\cos\theta_{ij})}{Q^2}$$
3. Pairs of 'y' less than threshold value are associated into one jet.
   - Association order is least-order of 'y' or opening angles
4. Repeat clustering with associated particles treated as a single 'particle'.

Traditional Jet Clustering
(without vertex information)



Problems of the Existing Algorithms

- They take account only of the momentum of the particles and the opening angles between them.
- Particles from secondary vertices sometimes lead misclustering.

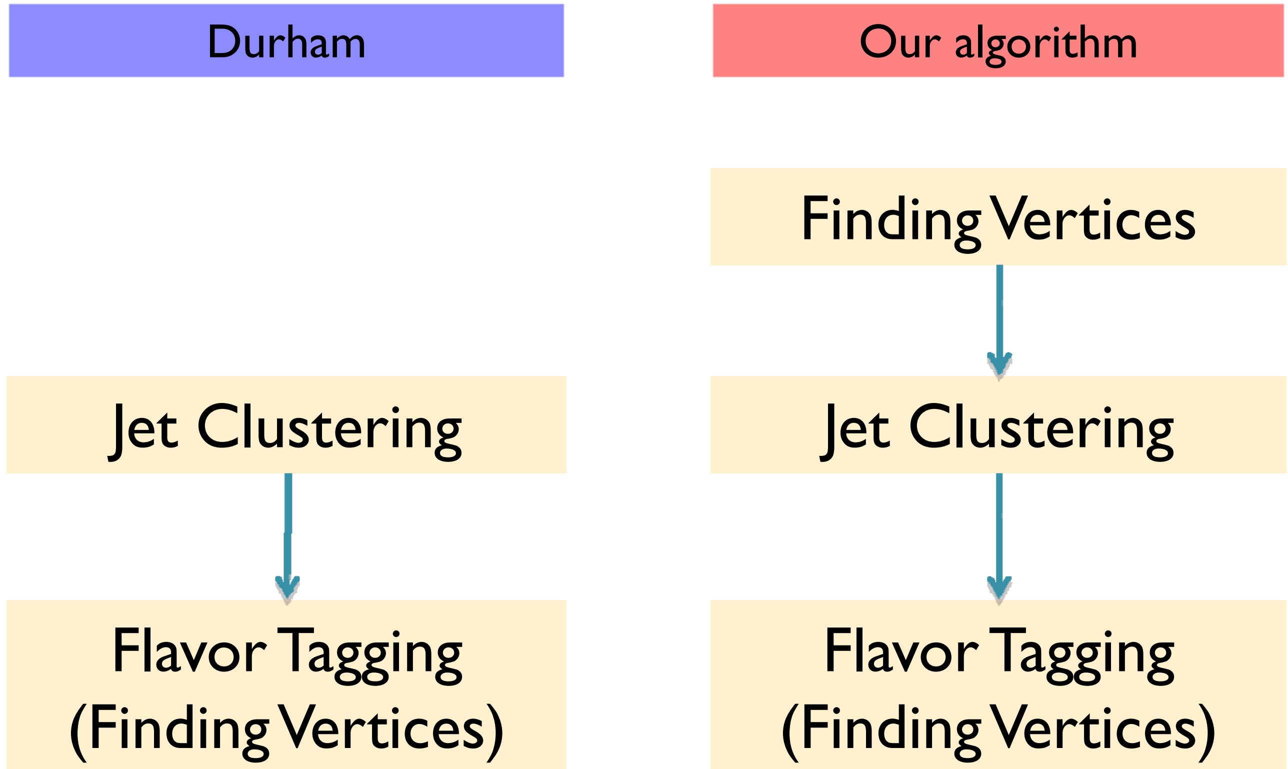
A new idea for jet clustering

We would like to use the vertex information (e.g. Its position and the particles associated with it) because…

- The vertex direction can be identified as the jet direction.

Vertex information can be used to improve performances of jet clustering.

Flow Chart


Durham
Our algorithm
Finding Vertices
Jet Clustering
Jet Clustering
Flavor Tagging (Finding Vertices)
Flavor Tagging (Finding Vertices)

How to find vertex

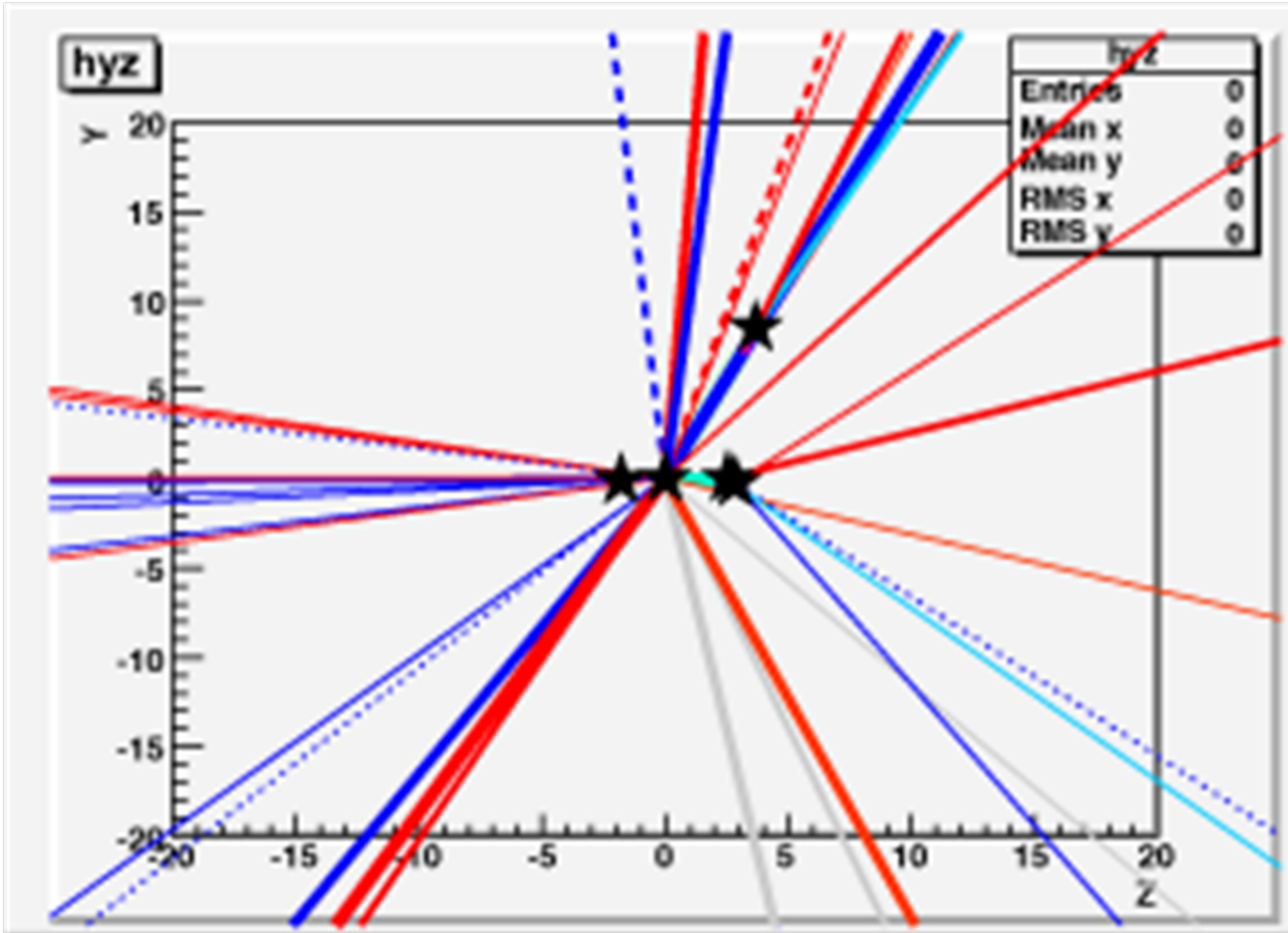
- We used the ZVTOP vertex finding algorithm, which was developed for finding secondary vertex after jet clustering has been done.
- We adjusted it not to use the direction of jets.

$$V(r) = \begin{cases} V(r) & (D \leq 50\mu m) \\ V(r)\exp(-K_\alpha \alpha^2) & (D > 50\mu m) \end{cases}$$

V(r) : a kind of jet density
α: opening angle between the jet and the particle

So… $K_\alpha \rightarrow 0$

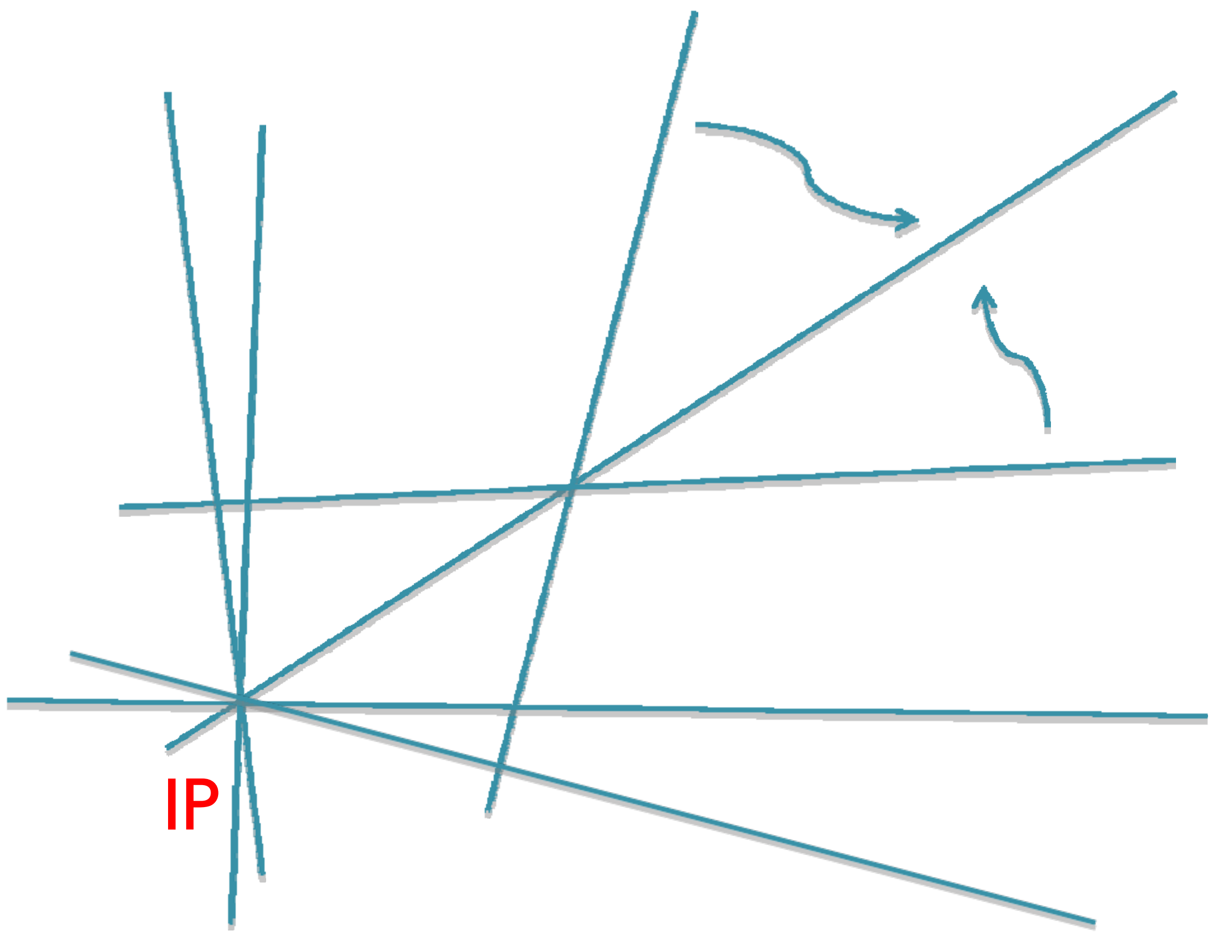
Secondary vertices found in this way



★: found vertices

Jet Clustering with found vertices

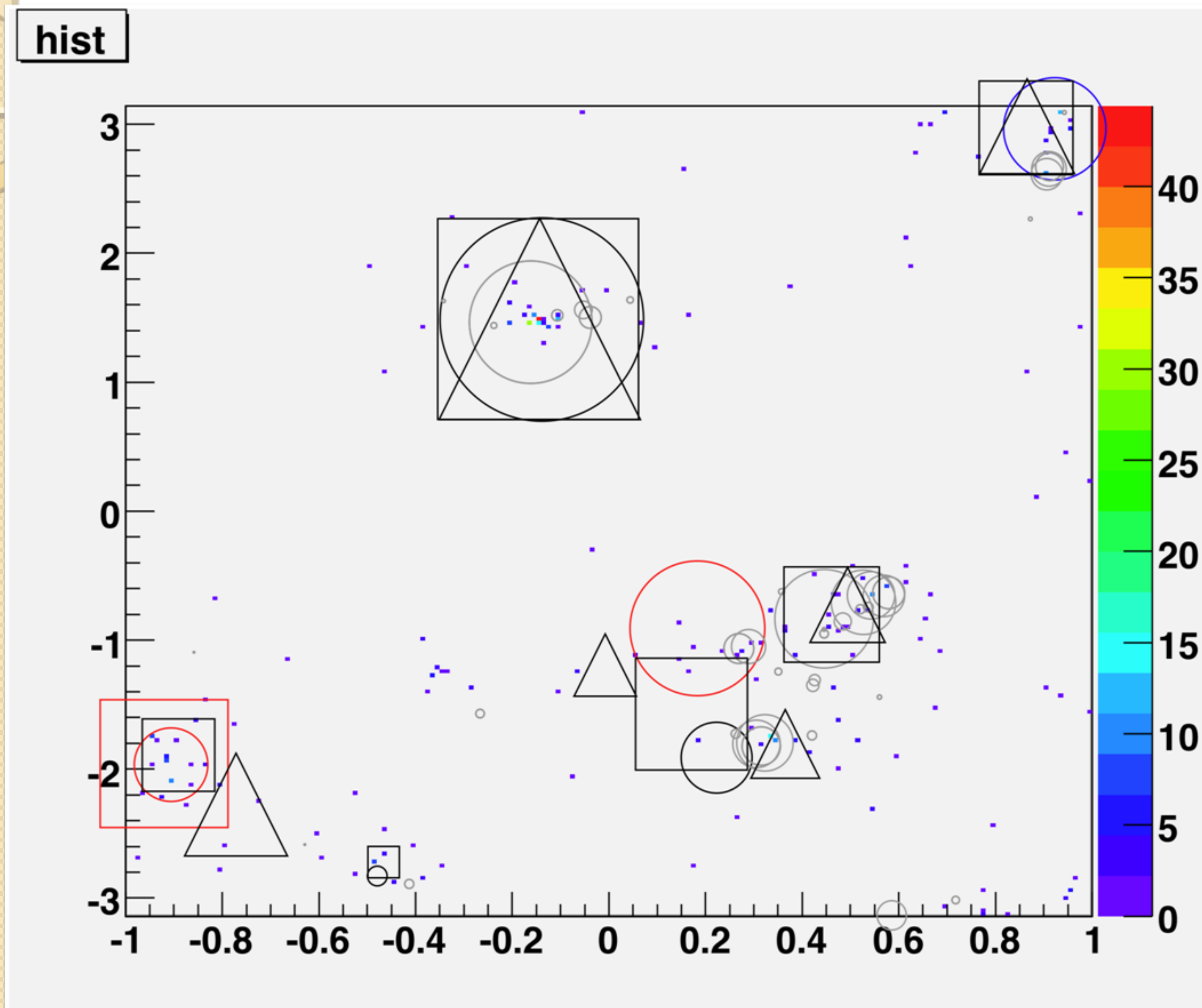
1: Consider a found vertex as a "particle".



2: Define “Jet cores”. The reconstructed secondary vertices (heavy flavor) + other isolated particles (light flavor) are chosen for the cores.

3: Rest particles are associated to a core which gives least ‘y’ value with it.

Event Sample



O: jet (MC)

O: b quark(MC)

□: vertex (our algorithm)

□: jet (our algorithm)

△: jet (Durham)

Flavor Tagging

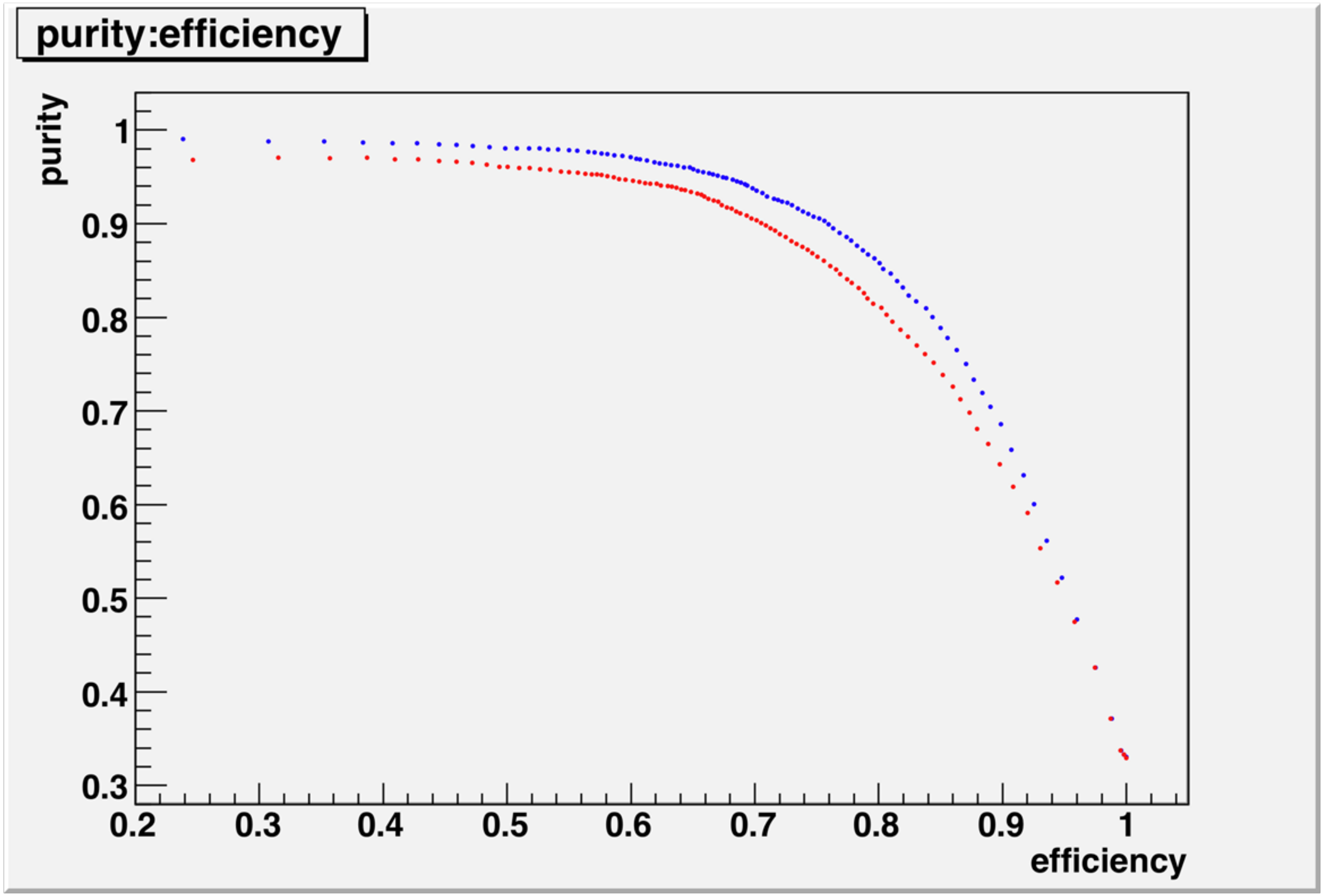
- LCFIVertex
- ILD_00 standard training sample (Z pole qqbar, q=uds,c,b)
- Used for performance estimation of the clustering

Event Samples

- bbcssc (mainly from tt): ~3500 events
- ZHH (qqhh): ~23000 events
  - m_H=120GeV, H->bb (~60% br.)

- Slac SM sample
- ILD_00

Performance of b-tagging (bbcssc)



- : Durham
- : KSJ (Our algorithm)

Analysis

- Our algorithm itself is currently
  not better than Durham algorithm.
  → Misidentification of vertices is critical.
- If the vertices are not mis-reconstructed,
  our algorithm uses more information than Durham,
  so there should be events in which our algorithm gives better results than Durham.
  → Combining the two algorithms can lead better result if we can find good criteria.

Select which algorithm should be used

- The b-likelihood value, given by the LCFIVertex, approximately represents the probability in which the initial quark is really b.

→ From this b-likelihood value, we can estimate the expectation value of the efficiency and purity in an event when threshold is determined.

Sample

b-likelihood value of jets:

0.9 0.8 0.7 0.5 0.3 0.1

Identify these jets as ones with b quark

If we set the threshold value = 0.6

→

$$\text{Pseudo Efficiency} = \frac{0.9+0.8+0.7}{0.9+0.8+0.7+0.5+0.3+0.1}$$

$$\text{Pseudo Purity} = \frac{0.9+0.8+0.7}{3}$$

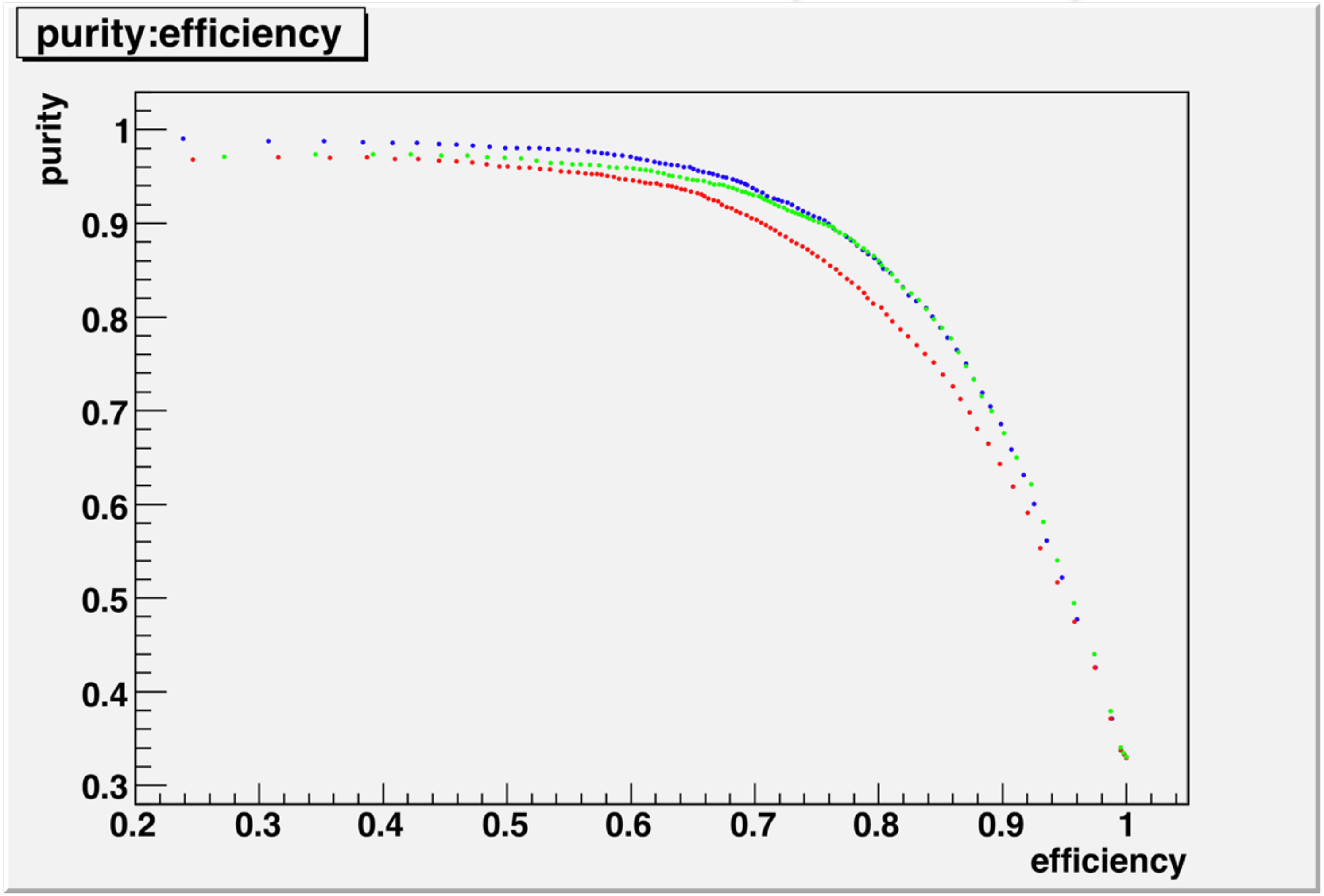
# of jet which have btag-likelihood value larger than the threshold

	Durham	KSJ (Our algorithm)
Pseudo Efficiency	0.7	0.9
Pseudo Purity	0.7	0.6
Sum	1.4 <	1.5

Use this one!!

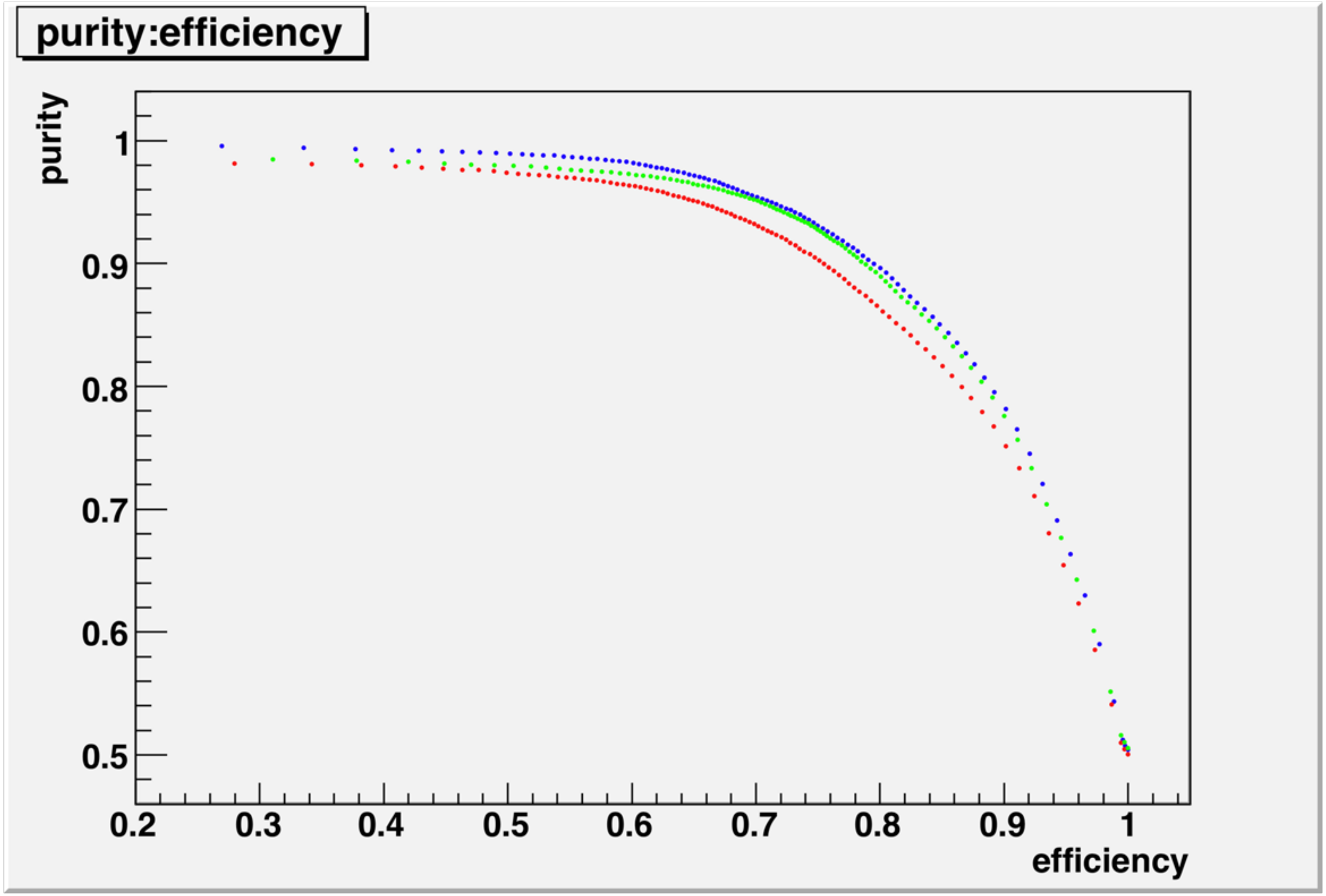
We use an algorithm with higher sum of the pseudo efficiency and the pseudo purity, event by event.

Performances (bbcssc)



- : Durham
- : KSJ (Our algorithm)
- : KSJ'(Durham + KSJ)

Performances (ZHH)



: Durham
: KSJ (Our algorithm)
: KSJ' (Durham + KSJ)

Summary and prospect

- Our clustering algorithm (KSJ') gives comparable performance to Durham in high efficiency b-tagging.
  - For high-purity b-tagging, improvements of algorithm selection must be needed to obtain better results.
- The performance degradation of the KSJ algorithm is mainly from mis-reconstruction of vertex positions.
  - More efficient vertex finder is inevitable for further improvements.