Theoretical aspects of jet-finding

Gavin P. Salam

LPTHE, UPMC Paris 6 & CNRS

4th Atlas Hadronic Calibration Workshop
Tucson, Arizona
14–16 March 2008

Based on work with Cacciari (LPTHE) & Soyez (BNL), and also with Butterworth (ATLAS UCL), Dasgupta (Manchester), Davison (ATLAS UCL), Magnea (Turin), Rojo (LPTHE), Rubin (LPTHE)

This lecture

Aim: to provide a reminder/intro to the "basics" of jet-finding, with a couple of bleeding-edge subjects at the end.

- General considerations

  Some points that deserve to be at the back of your mind as you carry out the challenging technical work to prepare ATLAS to measure jets.

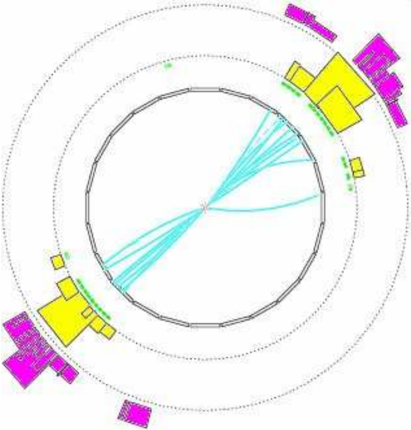
- Modern jet definitions

  Cone algorithms (and SISCone)
  Sequential recombination algorithms (k_t, etc.)
  Learning to compare them

- Jets at work

  Pileup characterisation and subtraction
  A new jet-based Higgs search-channel

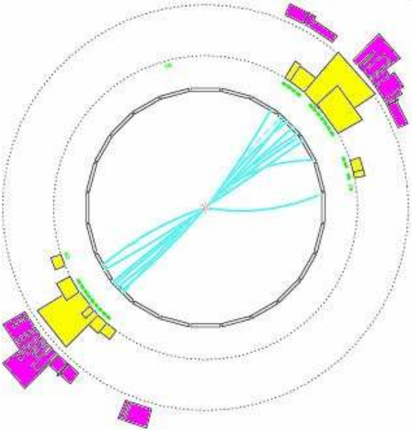
Seeing v. defining jets

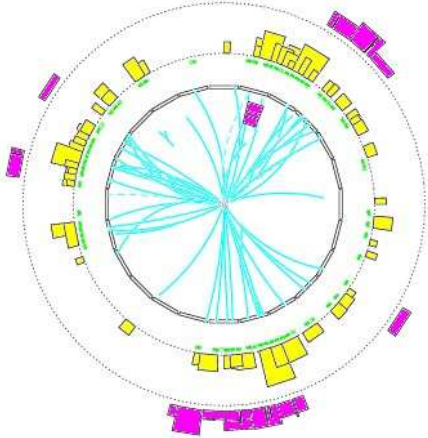


Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Seeing v. defining jets

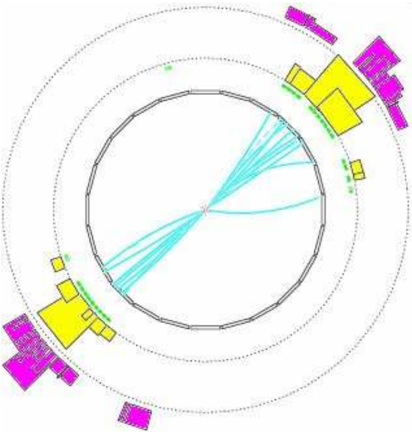




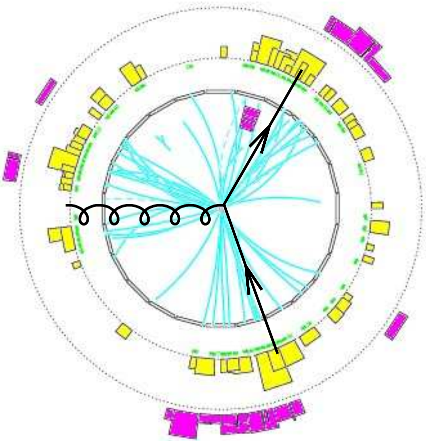
Jets are what we see.
Clearly(?) 2 jets here

How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Seeing v. defining jets

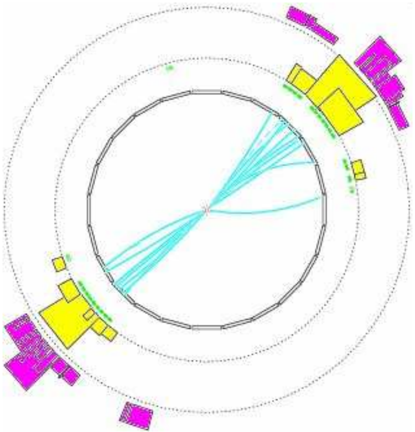


Jets are what we see.
Clearly(?) 2 jets here

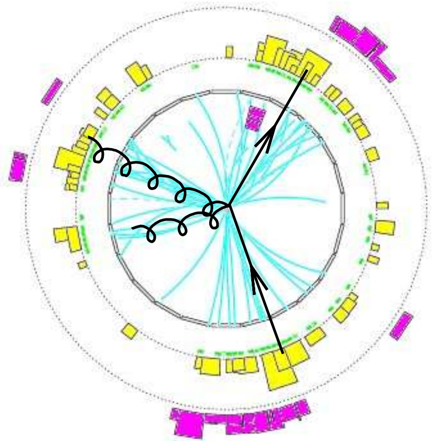


How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Seeing v. defining jets

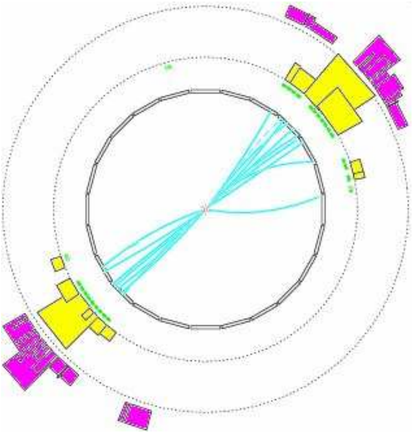


Jets are what we see.
Clearly(?) 2 jets here

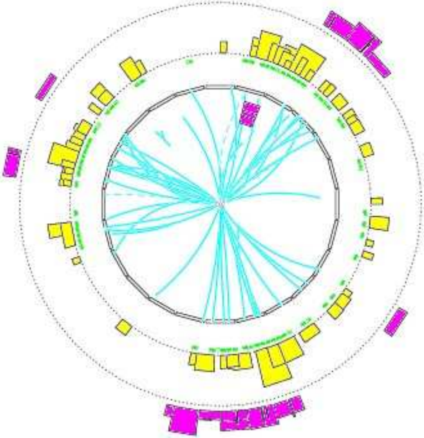


How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Seeing v. defining jets



Jets are what we see.
Clearly(?) 2 jets here



How many jets do you see?
Do you really want to ask yourself this question for 10^8 events?

Jet definitions

- *A jet definition is a fully specified set of rules for projecting information from 100's of hadrons, onto a handful of parton-like objects:*
  - or project 1000's of calorimeter towers
  - or project dozens of (showered) partons
  - or project a handful of (unshowered) partons

Idea of such a projection: Sterman & Weinberg '77

- Resulting objects (jets) used for many things, e.g. :
  - reconstructing decaying massive particles — e.g. top $\longrightarrow$ 3 jets
  - constraining proton structure
  - as a theoretical tool to attribute structure to an event

E.g. in CKKW matching

- You *lose much information* in projecting event onto jet-like structure:
  - Sometimes information you had no idea how to use
  - Sometimes information you may not trust, or of no relevance

There is no unique jet definition

The construction of a jet is unavoidably ambiguous. On at least two fronts:

1. which particles get put together into a common jet? Jet algorithm + parameters
2. how do you combine their momenta? Recombination scheme
   Most commonly used: direct 4-vector sums (E-scheme)

Taken together, these different elements specify a choice of jet definition

cf. Les Houches '07 nomenclature accord

There is no unique jet definition

The construction of a jet is unavoidably ambiguous. On at least two fronts:

1. which particles get put together into a common jet? — Jet algorithm + parameters
2. how do you combine their momenta? — Recombination scheme

Most commonly used: direct 4-vector sums (E-scheme)

Taken together, these different elements specify a choice of jet definition

cf. Les Houches '07 nomenclature accord

The power of ambiguity

- Physical results (particle discovery, masses, PDFs, coupling) should be independent of your choice of jet definition

  a bit like renormalisation scale/scheme invariance

  Tests independence on modelling of radiation, hadronisation, etc.

- Sometimes there may be a good reason why one jet definition is more optimal than another

  In such cases you should understand why

- But in general there is no single universal 'best' jet definition

  How best to look at an event depends on what you're trying to see

Flexibility in jet-finding is crucial

The power of ambiguity

- Physical results (particle discovery, masses, PDFs, coupling) should be independent of your choice of jet definition

  a bit like renormalisation scale/scheme invariance

  Tests independence on modelling of radiation, hadronisation, etc.

- Sometimes there may be a good reason why one jet definition is more optimal than another

  In such cases you should understand why

- But in general there is no single universal 'best' jet definition

  How best to look at an event depends on what you're trying to see

Flexibility in jet-finding is crucial

"Jetography" like photography

Much fine detail on boarding pass, photograph it from close up, focus=40cm.

keep focus at 40cm

reset focus to 8m

"Jetography" like photography

Much fine detail on boarding pass, photograph it from close up, focus=40cm.

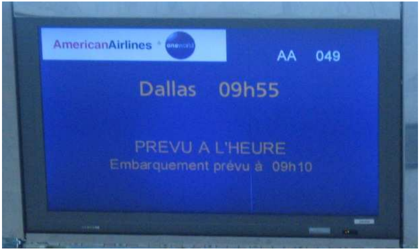
keep focus at 40cm

reset focus to 8m

"Jetography" like photography

Much fine detail on boarding pass, photograph it from close up, focus=40cm.

keep focus at 40cm



reset focus to 8m

Not all ambiguity is "allowed"

Jets should be **invariant** with respect to certain modifications of the event:

- infrared (IR) emission — e.g. 1 GeV particles wrt 1 TeV jets
- collinear (C) splitting

This was one of the key ideas introduced by Sterman & Weinberg in "77.

Why?

- They happen randomly, quantum-mechanically, all the time in QCD.
- IR/C sensitivity ↔ lose real-virtual cancellation in NLO/NNLO QCD calculations → meaningless, divergent results
- Hadron-level IR/C modifications are fundamentally non-perturbative
- Detectors resolve neither full collinear nor full infrared event structure

Requirement known as: **infrared and collinear (IRC) safety**

Not all ambiguity is "allowed"

Jets should be **invariant** with respect to certain modifications of the event:

- infrared (IR) emission — e.g. 1 GeV particles wrt 1 TeV jets
- collinear (C) splitting

This was one of the key ideas introduced by Sterman & Weinberg in "77.

Why?

- They happen randomly, quantum-mechanically, all the time in QCD.
- IR/C sensitivity $\leftrightarrow$ lose real-virtual cancellation in NLO/NNLO QCD calculations $\rightarrow$ meaningless, divergent results
- Hadron-level IR/C modifications are fundamentally non-perturbative
- Detectors resolve neither full collinear nor full infrared event structure

Requirement known as: **infrared and collinear (IRC) safety**

Not all ambiguity is “allowed”

Jets should be **invariant** with respect to certain modifications of the event:

- infrared (IR) emission — e.g. 1 GeV particles wrt 1 TeV jets
- collinear (C) splitting

This was one of the key ideas introduced by Sterman & Weinberg in ”77.

Why?

- They happen randomly, quantum-mechanically, all the time in QCD.
- IR/C sensitivity $\leftrightarrow$ lose real-virtual cancellation in NLO/NNLO QCD calculations $\rightarrow$ meaningless, divergent results
- Hadron-level IR/C modifications are fundamentally non-perturbative
- Detectors resolve neither full collinear nor full infrared event structure

Requirement known as: **infrared and collinear (IRC) safety**

Two main classes of jet alg.

The majority of algorithms map each particle into (at most) one jet.

Cone

- top-down
- centred around idea of an 'invariant', directed energy flow

Sequential pairwise recombination (k_t, etc.)

- bottom-up
- successively undoes QCD branching

Others

- "Optimal" Jet Finder (OJF): weight w_{iJ} for each particle i to be in jet J
- ARCLUS: seq. rec., but with $3 \to 2$ recombination

Two main classes of jet alg.

The majority of algorithms map each particle into (at most) one jet.

Cone

- top-down
- centred around idea of an ‘invariant’, directed energy flow

Sequential pairwise recombination (k_t, etc.)

- bottom-up
- successively undoes QCD branching

Others

- “Optimal” Jet Finder (OJF): weight w_{iJ} for each particle i to be in jet J
- ARCLUS: seq. rec., but with $3 \to 2$ recombination

Cone basics I: IC-SM

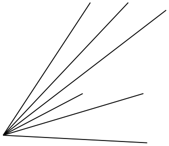
Many cone algs have two main steps:

- Find some/all stable cones

  $\equiv$ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

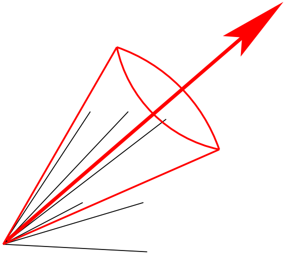
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

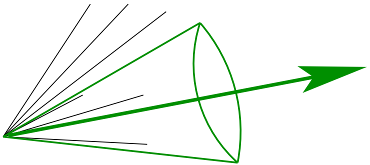
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

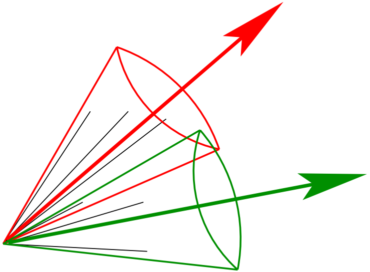
Many cone algs have two main steps:

- Find some/all stable cones

  $\equiv$ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

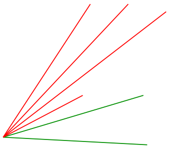
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Cone basics I: IC-SM

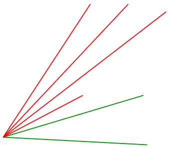
Many cone algs have two main steps:

- Find some/all stable cones

  ≡ cone pointing in same direction as the momentum of its contents

- Resolve cases of overlapping stable cones

  By running a 'split–merge' procedure [Blazey et al. '00 (Run II jet physics)]



Qu: How do you find the stable cones?

Until recently used iterative methods:

- use each particle as a starting direction for cone; use sum of contents as new starting direction; repeat.

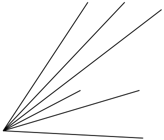
Iterative Cone with Split Merge (IC-SM)

e.g. Tevatron cones (JetClu, midpoint)
ATLAS cone

Cone basics II: IC-PR

Other cones avoid split-merge:

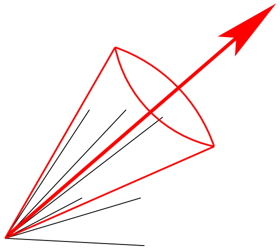
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

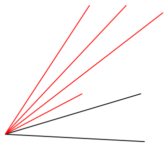
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

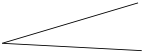
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet; remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

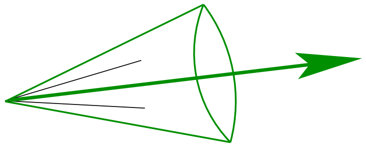
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

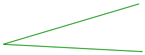
- Find one stable cone E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

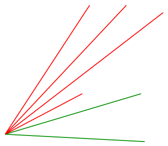
- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



Cone basics II: IC-PR

Other cones avoid split-merge:

- Find one stable cone — E.g. by iterating from hardest seed particle
- Call it a jet;remove its particles from the event; repeat



- This is not the same algorithm
- Many physics aspects differ

Iterative Cone with Progressive Removal (IC-PR)

e.g. CMS cone, Pythia Cone, [GetJet], ...

Iterative cone problems

- What are the starting points for iteration?
- Start with hardest particle as seed (IC-PR): collinear unsafe
- Use all particles (IC-SM): extra soft one $\longrightarrow$ new solution

Iterative cone finding **plagued by IR and collinear unsafety problems**

Iterative cone problems

- What are the starting points for iteration?
- Start with hardest particle as seed (IC-PR): collinear unsafe
- Use all particles (IC-SM): extra soft one → new solution

Iterative cone finding **plagued by IR and collinear unsafety problems**

Among consequences of IR unsafety:

	Last meaningful order		
	ATLAS cone [IC-SM]	MidPoint [IC_{mp}-SM]	CMS it. cone [IC-PR]
Inclusive jets	LO	NLO	NLO
$W/Z + 1$ jet	LO	NLO	NLO
3 jets	**none**	LO	LO
$W/Z + 2$ jets	**none**	LO	LO
$m_{\rm jet}$ in $2j + X$	**none**	**none**	**none**

NB: \$30 – 50M investment in NLO

Some common cone algs (LH '07)

Algorithm	Type	IRC status	Notes
CDF JetClu	IC_r-SM	IR_{2+1}	
CDF MidPoint cone	IC_{mp}-SM	IR_{3+1}	
CDF searchcone	$IC_{se,mp}$-SM	IR_{2+1}	
D0 Run II cone	IC_{mp}-SM	IR_{3+1}	no seed threshold, but c
ATLAS Cone	IC-SM	IR_{2+1}	
PxCone	IC_{mp}-SD	IR_{3+1}	no seed threshold, but c
CMS Iterative Cone	IC-PR	$Coll_{3+1}$	
PyCell/CellJet (from Pythia)	FC-PR	$Coll_{3+1}$	
GetJet (from ISAJET)	FC-PR	$Coll_{3+1}$	

Too many cones, too many problems, needs consolidation.

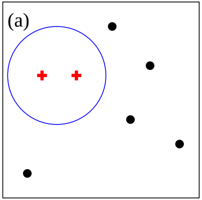
IC-SM $\rightarrow$ SISCone (finds all stable cones)
IC-PR $\rightarrow$ anti-k_t *[NEW!! More in a few minutes]*

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

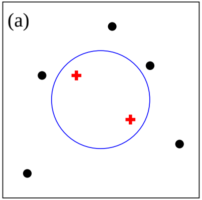
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

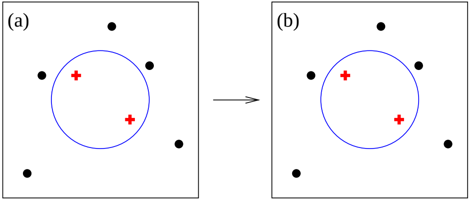
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

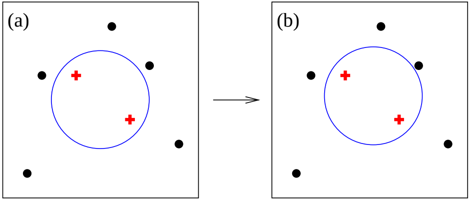
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

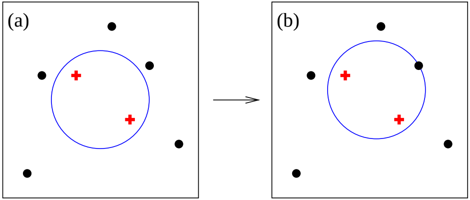
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

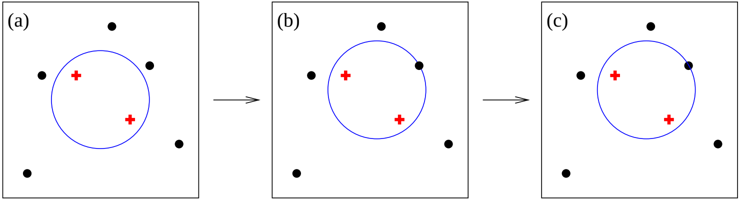
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

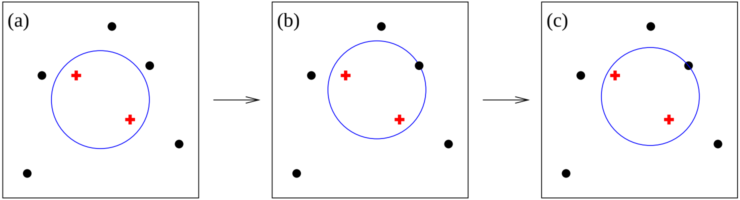
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

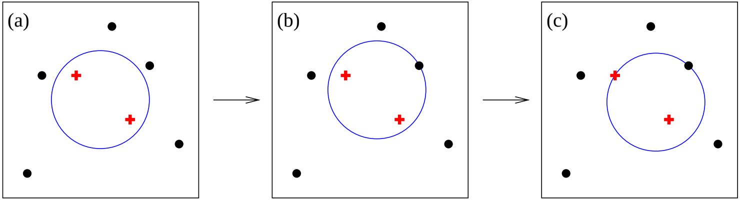
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

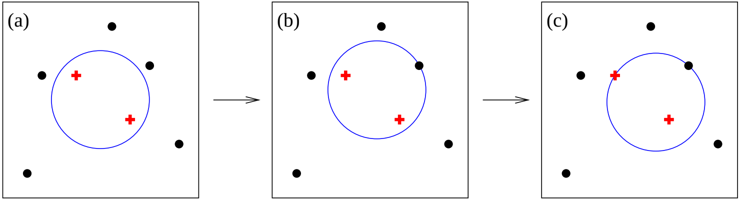
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

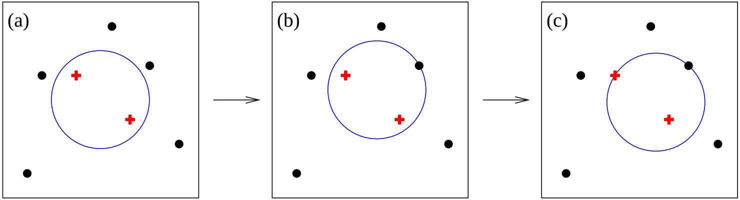
GPS & Soyez '07

Solve IR issue: find all stable cones

Cones are just *circles* in the $y - \phi$ plane. To find all stable cones:

1. Find all distinct ways of enclosing a subset of particles in a $y - \phi$ circle
2. Check, for each enclosure, if it corresponds to a stable cone

Finding all distinct circular enclosures of a set of points is **geometry:**



Any enclosure can be moved until a pair of points lies on its edge.

Result: Seedless Infrared Safe Cone algorithm (SISCone)

Runs in $N^2 \ln N$ time ($\simeq$ midpoint's N^3)

GPS & Soyez '07

SISCone algorithm as a whole

1: Put the set of current particles equal to the set of all particles in the event.
2: **repeat**
3: Find *all* stable cones of radius $\boldsymbol{R}$ for the current set of particles, e.g. using algorithm 2.
4: For each stable cone, create a protojet from the current particles contained in the cone, and add it to the list of protojets.
5: Remove all particles that are in stable cones from the list of current particles.
6: **until** No new stable cones are found, or one has gone around the loop N_{pass} times.
7: Run a Tevatron Run-II type split–merge procedure, algorithm 3, on the full list of protojets, with overlap parameter $\boldsymbol{f}$ and transverse momentum threshold $p_{t,\mathrm{min}}$.

SISCone part 2: finding stable cones

For any group of collinear particles, merge them into a single particle.

for particle $i = 1 \ldots N$ **do**

Find all particles j within a distance $2R$ of i. If there are no such particles, i forms a stable cone of its own.

Otherwise for each j identify the two circles for which i and j lie on the circumference. For each circle, compute the angle of its centre C relative to i, $\zeta = \arctan \frac{\Delta\phi_{iC}}{\Delta y_{iC}}$.

Sort the circles into increasing angle ζ.

Take the first circle in this order, and call it the current circle. Calculate the total momentum and checkxor for the cones that it defines. Consider all 4 permutations of edge points being included or excluded. Call these the "current cones".

repeat

for each of the 4 current cones **do**

If this cone has not yet been found, add it to the list of distinct cones.

If this cone has not yet been labelled as unstable, establish if the in/out status of the edge particles (with respect to the cone momentum axis) is the same as when defining the cone; if it is not, label the cone as unstable.

end for

Move to the next circle in order. It differs from the previous one either by a particle entering the circle, or one leaving the circle. Calculate the momentum for the new circle and corresponding new current cones by adding (or removing) the momentum of the particle that has entered (left); the checkxor can be updated by XORing with the label of that particle.

until all circles considered.

end for

for each of the cones not labelled as unstable **do**

Explicitly check its stability, and if it is stable, add it to the list of stable cones (protojets).

end for

SISCone part 3: split–merge

1: **repeat**

Remove all protojets with $p_t < p_{t,\mathrm{min}}$.

Identify the protojet (i) with the highest $\tilde{p}_t$ ($\tilde{p}_{t,\mathrm{jet}} = \sum_{i\in\mathrm{jet}} |p_{t,i}|$).

Among the remaining protojets identify the one (j) with highest $\tilde{p}_t$ that shares particles (overlaps) with i.

5: **if** there is such an overlapping jet **then**

6: Determine the total $\tilde{p}_{t,\mathrm{shared}} = \sum_{k\in i\&j} |p_{t,k}|$ of the particles shared between i and j.

7: **if** $\tilde{p}_{t,\mathrm{shared}} < f\,\tilde{p}_{t,j}$ **then**

Each particle that is shared between the two protojets is assigned to the one to whose axis it is closest. The protojet momenta are then recalculated.

9: **else**

Merge the two protojets into a single new protojet (added to the list of protojets, while the two original ones are removed).

11: **end if**

12: If steps 7–11 produced a protojet that coincides with an existing one, maintain the new protojet as distinct from the existing copy(ies).

13: **else**

Add i to the list of final jets, and remove it from the list of protojets.

15: **end if**

16: **until** no protojets are left.

Is it truly IR safe?

- Generate event with $2 < N < 10$ hard particles, find jets
- Add $1 < N_{soft} < 5$ soft particles, find jets again [repeatedly]
- If the jets are different, algorithm is IR unsafe.

Unsafety level	failure rate
2 hard + 1 soft	$\sim 50\%$
3 hard + 1 soft	$\sim 15\%$
SISCone	IR safe !

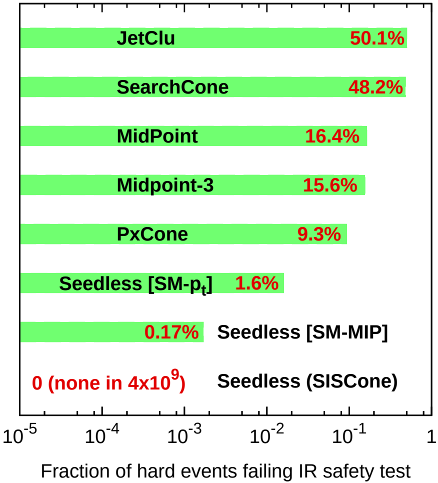
Be careful with split–merge too

Is it truly IR safe?

- Generate event with $2 < N < 10$ hard particles, find jets
- Add $1 < N_{soft} < 5$ soft particles, find jets again [repeatedly]
- If the jets are different, algorithm is IR unsafe.

Unsafety level	failure rate
2 hard + 1 soft	$\sim$ 50%
3 hard + 1 soft	$\sim$ 15%
SISCone	IR safe !

Be careful with split–merge too



k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}}\,, \qquad (E_j \ll E_i\,,\ \theta_{ij} \ll 1)\,.$$

Invert branching process: take pair with strongest divergence between them — they're the most *likely* to belong together.

→ **k_t/Durham algorithm** (e^+e^-)

1. Calculate (or update) distances between all particles i and j:

$$y_{ij} = \frac{2\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})}{Q^2}$$

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}}\,, \qquad (E_j \ll E_i\,,\ \theta_{ij} \ll 1)\,.$$

Invert branching process: take pair with strongest divergence between them — they're the most *likely* to belong together.

→ **k_t/Durham algorithm** (e^+e^-)

1. Calculate (or update) distances between all particles i and j:

$$y_{ij} = \frac{2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})}{Q^2}$$

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t/Durham algorithm

Majority of QCD branching is soft & collinear, with following divergences:

$$[dk_j]|M^2_{g\to g_i g_j}(k_j)| \simeq \frac{2\alpha_s C_A}{\pi} \frac{dE_j}{\min(E_i, E_j)} \frac{d\theta_{ij}}{\theta_{ij}}\,, \qquad (E_j \ll E_i\,,\ \theta_{ij} \ll 1)\,.$$

Invert branching process: take pair with strongest divergence between them — they're the most *likely* to belong together.

→ **$\mathbf{k_t}$/Durham algorithm** (e^+e^-)

1. Calculate (or update) distances between all particles i and j:

$$y_{ij} = \frac{2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})}{Q^2}$$

NB: relative k_t between particles

2. Find smallest of y_{ij}
   - If $> y_{cut}$, stop clustering
   - Otherwise recombine i and j, and repeat from step 1

Catani, Dokshitzer, Olsson, Turnock & Webber '91

k_t alg. at hadron colliders

Exclusive long. inv. k_t algorithm:

- Drop normalisation to Q^2 (not fixed in pp): $y_{ij} \to d_{ij}$, $y_{cut} \to d_{cut}$
- Make it longitudinally boost invariant

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2\,, \qquad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

- Introduce clustering with “beam jets”, distance: $d_{iB} = p_{ti}^2$

Catani, Dokshitzer, Seymour & Webber ’93

Inclusive long. inv. k_t: introduce R, drop d_{cut}

1. Find smallest of d_{ij}, d_{iB}: $d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2/R^2\,, \qquad d_{iB} = p_{ti}^2$
2. if ij, recombine them; *if iB, call i a jet* and remove from list of particles
3. repeat from step 1 until no particles left.

S.D. Ellis & Soper, ’93; most “cone-like”

Jets all separated by at least R on y, ϕ cylinder

Often just called “k_t” in pp, $p\bar{p}$

k_t alg. at hadron colliders

Exclusive long. inv. k_t algorithm:

- Drop normalisation to Q^2 (not fixed in pp): $y_{ij} \longrightarrow d_{ij}$, $y_{cut} \longrightarrow d_{cut}$
- Make it longitudinally boost invariant

$$d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2\,, \qquad \Delta R_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$$

- Introduce clustering with "beam jets", distance: $d_{iB} = p_{ti}^2$

Catani, Dokshitzer, Seymour & Webber '93

Inclusive long. inv. k_t: introduce R, drop d_{cut}

1. Find smallest of d_{ij}, d_{iB}: $d_{ij} = \min(p_{ti}^2, p_{tj}^2)\Delta R_{ij}^2/R^2\,, \qquad d_{iB} = p_{ti}^2$
2. if ij, recombine them; *if iB, call i a jet* and remove from list of particles
3. repeat from step 1 until no particles left.

S.D. Ellis & Soper, '93; most "cone-like"

Jets all separated by at least R on y, ϕ cylinder

Often just called "k_t" in pp, $p\bar{p}$

k_t is a form of Hierarchical Clustering

Fast Hierarchical Clustering and Other Applications of Dynamic Closest Pairs

David Eppstein
UC Irvine

We develop data structures for dynamic closest pair problems with arbitrary distance functions, that do not necessarily come from any geometric structure on the objects. Based on a technique previously used by the author for Euclidean closest pairs, we show how to insert and delete objects from an n-object set, maintaining the closest pair, in $O(n \log^2 n)$ time per update and $O(n)$ space. With quadratic space, we can instead use a quadtree-like structure to achieve an optimal time bound, $O(n)$ per update. We apply these data structures to hierarchical clustering, greedy matching, and TSP heuristics, and discuss other potential applications in machine learning, Gröbner bases, and local improvement algorithms for partition and placement problems. Experiments show our new methods to be faster in practice than previously used heuristics.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms**]: Nonnumeric Algorithms

General Terms: Closest Pair, Agglomerative Clustering

Additional Key Words and Phrases: TSP, matching, conga line data structure, quadtree, nearest neighbor heuristic

1. INTRODUCTION

Hierarchical clustering has long been a mainstay of statistical analysis, and clustering based methods have attracted attention in other fields: computational biology (reconstruction of evolutionary trees; tree-based multiple sequence alignment), scientific simulation (n-body problems), theoretical computer science (network design and nearest neighbor searching) and of course the web (hierarchical indices such as Yahoo). Many clustering methods have been devised and used in these applications, but less effort has gone into algorithmic speedups of these methods.

In this paper we identify and demonstrate speedups for a key subroutine used in several clustering algorithms, that of maintaining closest pairs in a dynamic set of objects. We also describe several other applications or potential applications of the

Idea behind k_t alg. is to be found over and over in many areas of (computer) science.

Sequential recombination of cacti

$\mathbf{k_t}$ alg.: Find smallest of

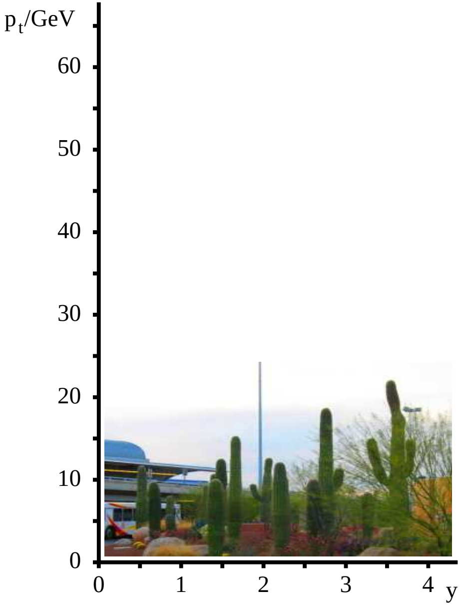
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



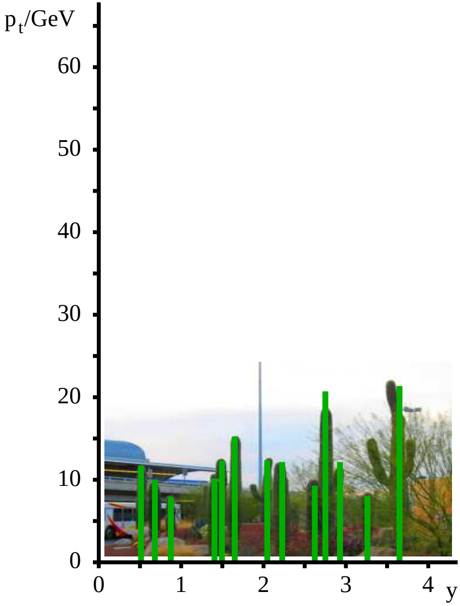
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

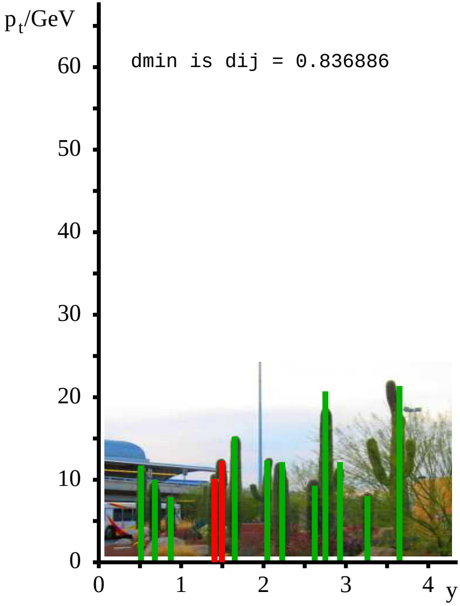
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



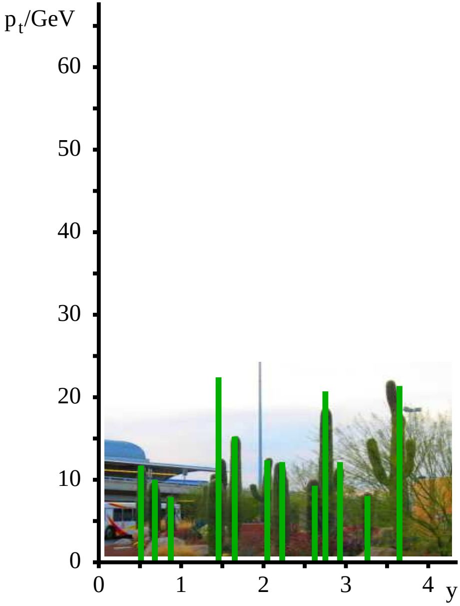
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



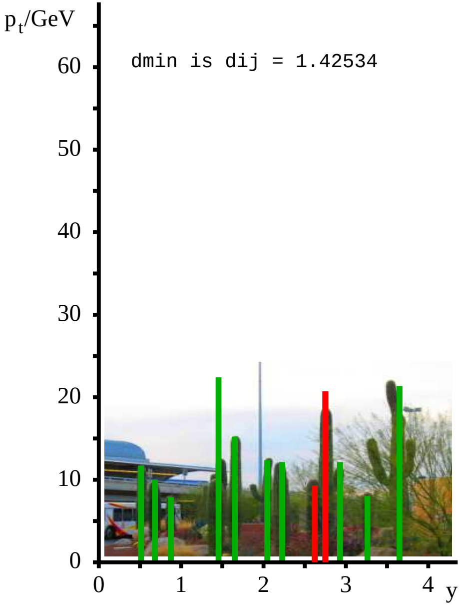
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

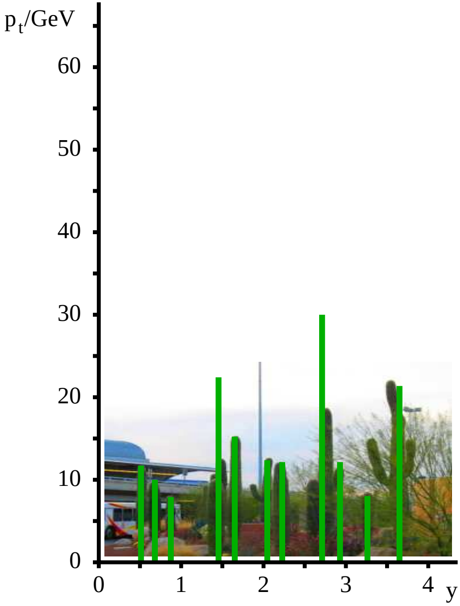
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

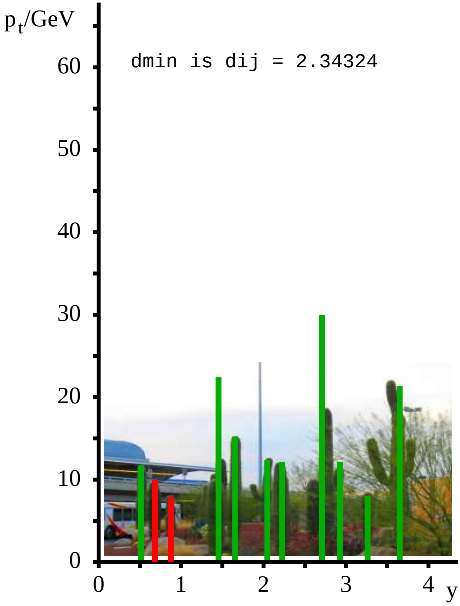
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



$\mathbf{k_t}$ alg.: Find smallest of

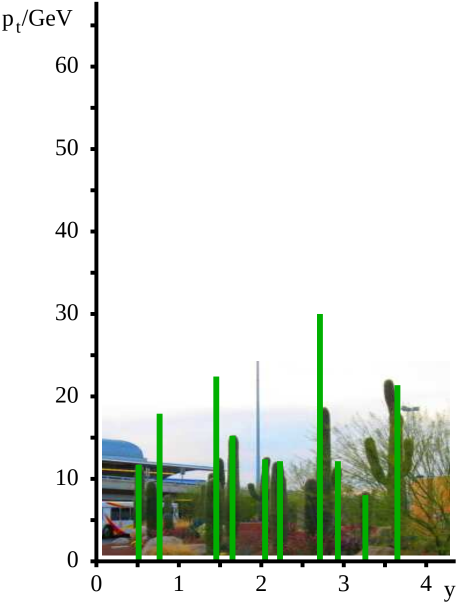
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



$\mathbf{k_t}$ alg.: Find smallest of

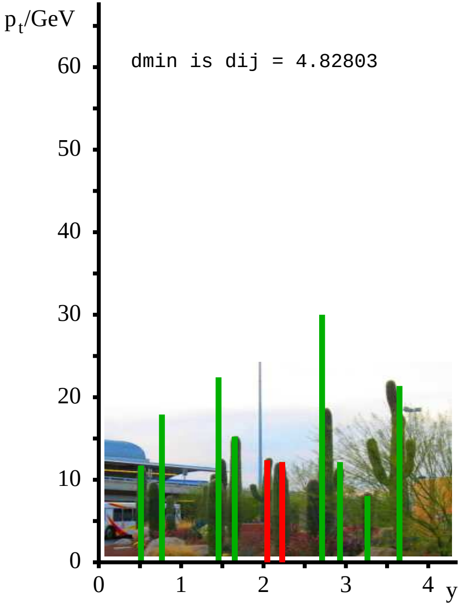
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

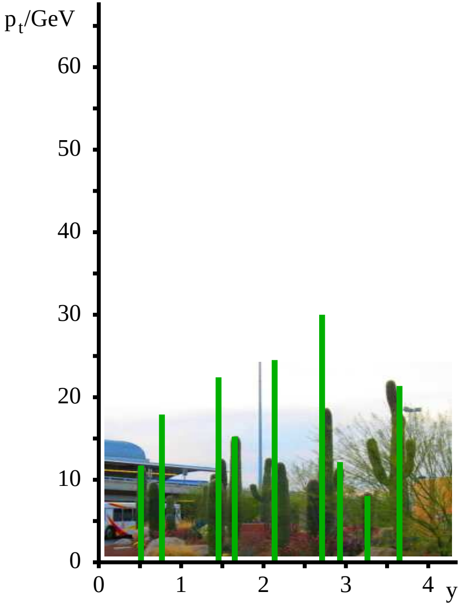
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



$\mathbf{k_t}$ alg.: Find smallest of

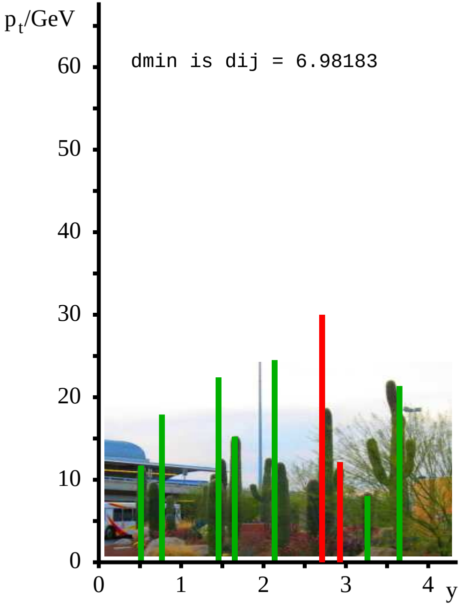
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

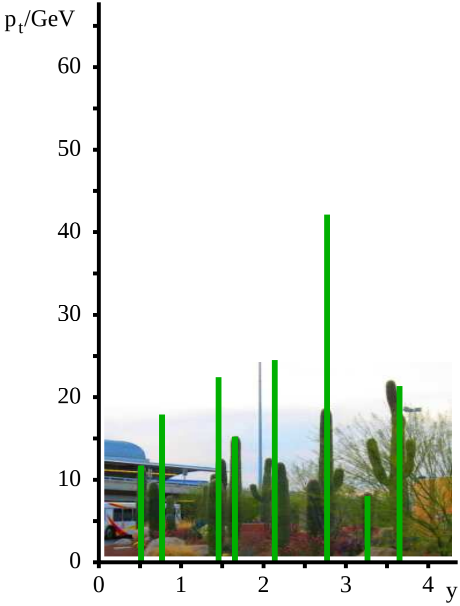
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

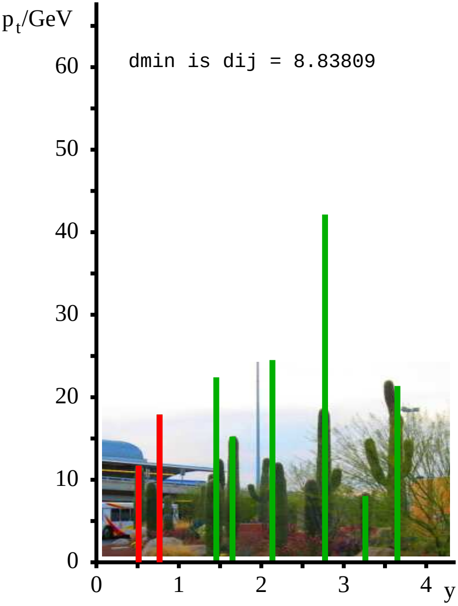
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



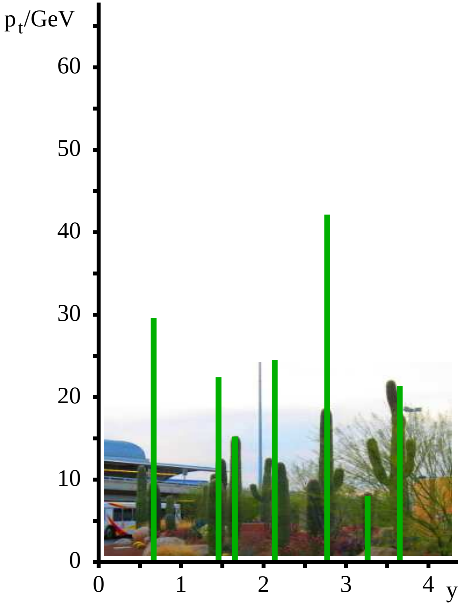
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



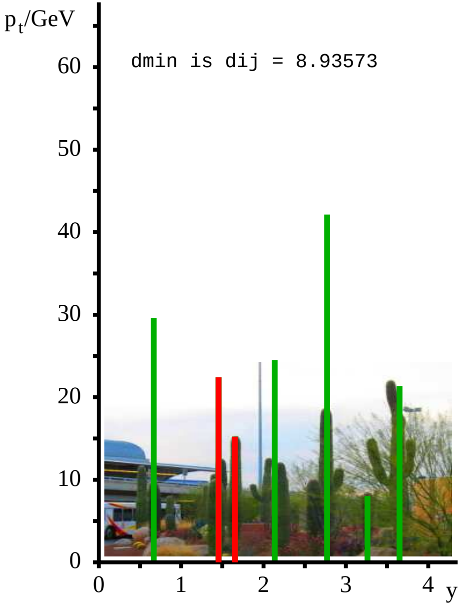
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

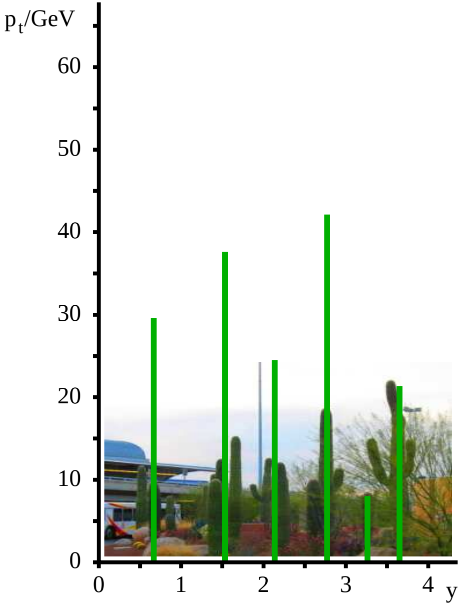
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



$\mathbf{k_t}$ alg.: Find smallest of

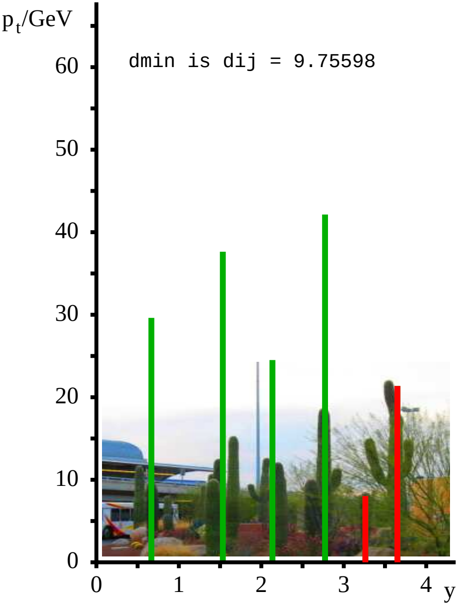
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



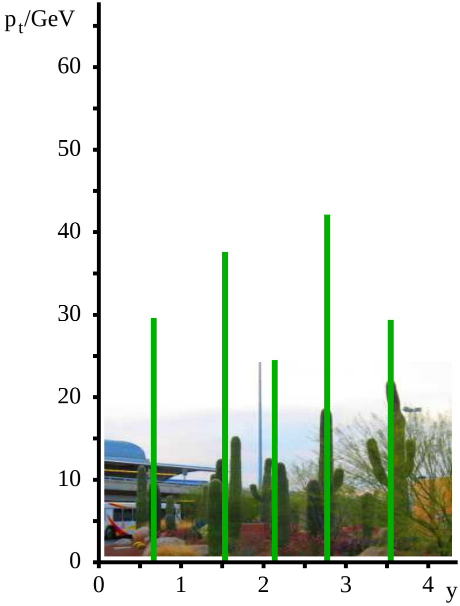
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

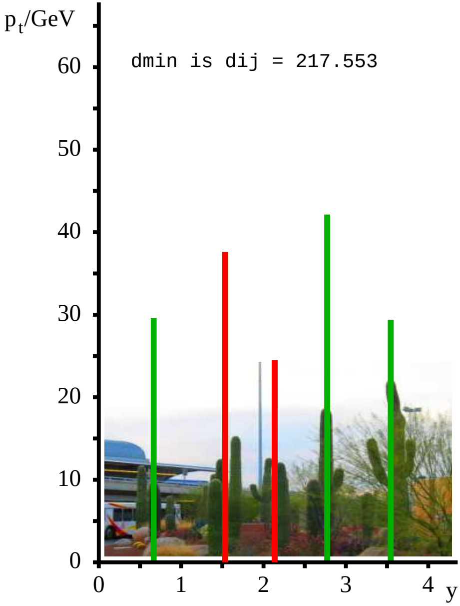
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

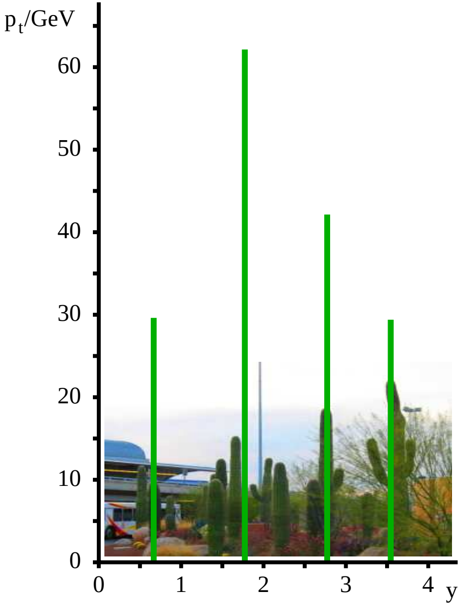
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

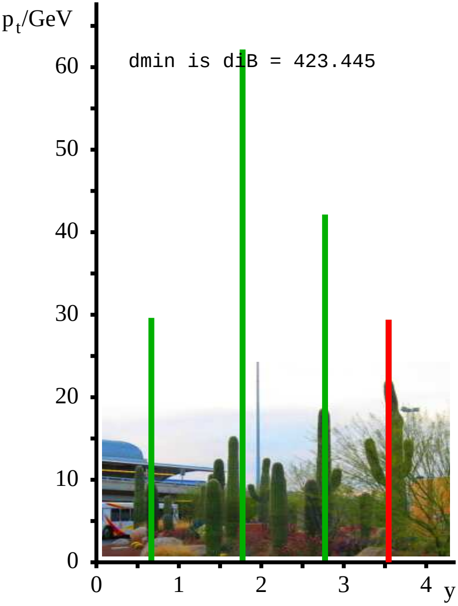
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



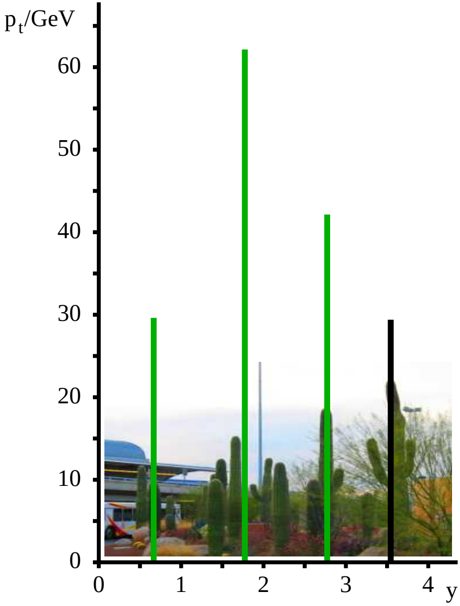
k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet
Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



$\mathbf{k_t}$ alg.: Find smallest of

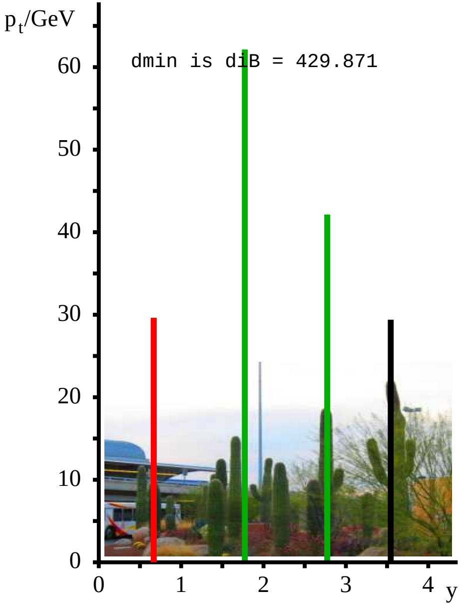
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

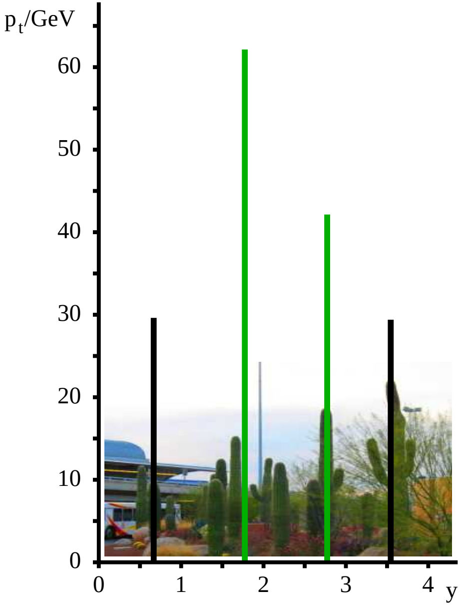
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



$\mathbf{k_t}$ alg.: Find smallest of

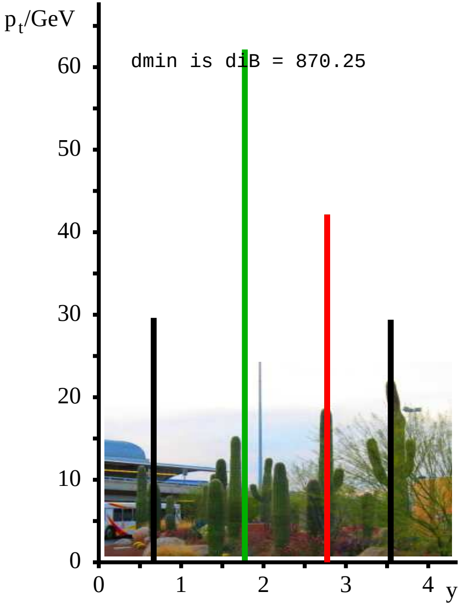
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

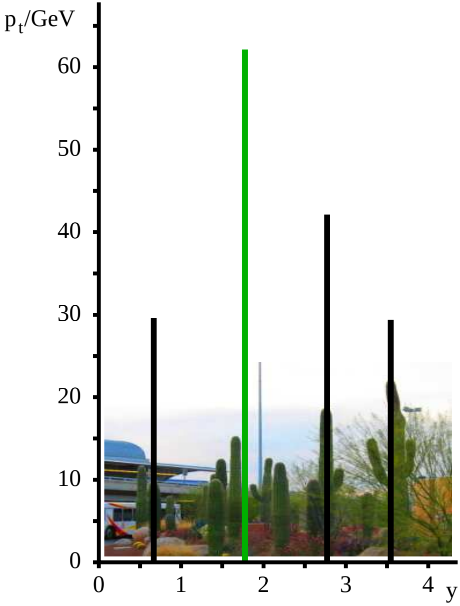
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

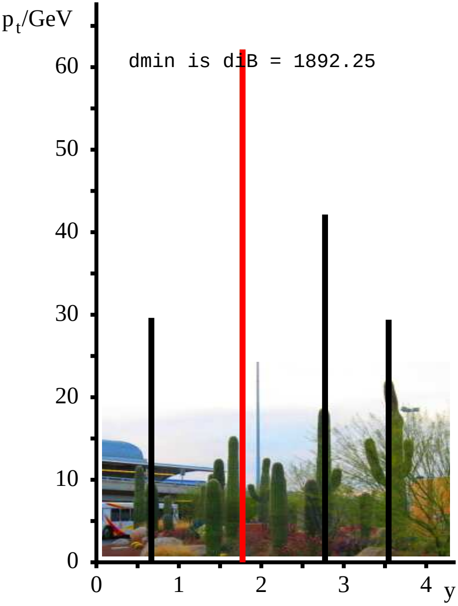
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

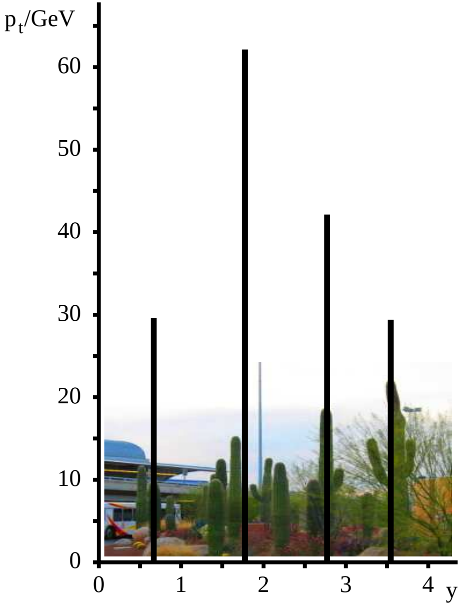
$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

Sequential recombination of cacti



k_t alg.: Find smallest of

$$d_{ij} = \min(k_{ti}^2, k_{tj}^2)\Delta R_{ij}^2/R^2, \quad d_{iB} = k_{ti}^2$$

If d_{ij} recombine; if d_{iB}, i is a jet

Example clustering with k_t algorithm, $R = 0.7$

ϕ assumed 0 for all towers

A full set of IRC-safe jet algorithms

Generalise inclusive-type sequential recombination with

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p})\Delta R_{ij}^2/R^2 \qquad d_{iB} = k_{ti}^{2p}$$

	Alg. name	Comment	time
$p = 1$	k_t CDOSTW '91-93; ES '93	Hierarchical in rel. k_t	$N \ln N$ exp.
$p = 0$	Cambridge/Aachen Dok, Leder, Moretti, Webber '97 Wengler, Wobisch '98	Hierarchical in angle Scan multiple R at once $\leftrightarrow$ QCD angular orderin	$N \ln N$
$p = -1$	anti-k_t Cacciari, GPS, Soyez '08 $\sim$ reverse-k_t Delsart, Loch et al.	Hierarchy meaningless. Behaves like IC-PR	$N^{3/2}$
SC-SM	SISCone GPS Soyez '07 + Tevatron run II '00	Replacement for IC-SM notably "MidPoint" cones	$N^2 \ln N$ exp.

One could invent/try others (e.g. OJF, etc.). Our [Paris+BNL] philosophy: 4 algs is enough of a basis to develop first physics understanding.

We already have far more than can be shown here

non-COMMERCIAL BREAK

Use FastJet — it's free!

One place to stop for your jet-finding needs:

FastJet

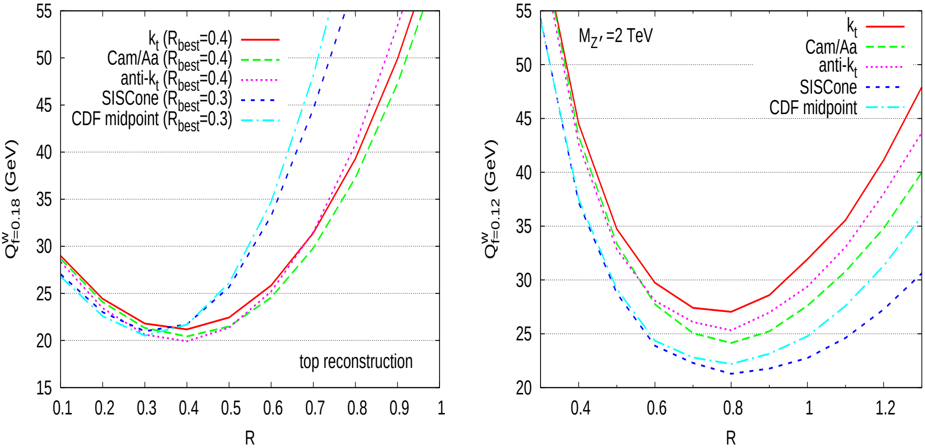
http://www.lpthe.jussieu.fr/~salam/fastjet

Cacciari, GPS & Soyez '05–08

- Fast, native, computational-geometry methods for k_t, Cam/Aachen, anti-k_t — Cacciari & GPS '05-06
- Plugins for SISCone (plus some other, deprecated, legacy cones)
- Documented user interface for adding extra algorithms of your own
- Tools for jet areas, pileup characterisation & subtraction

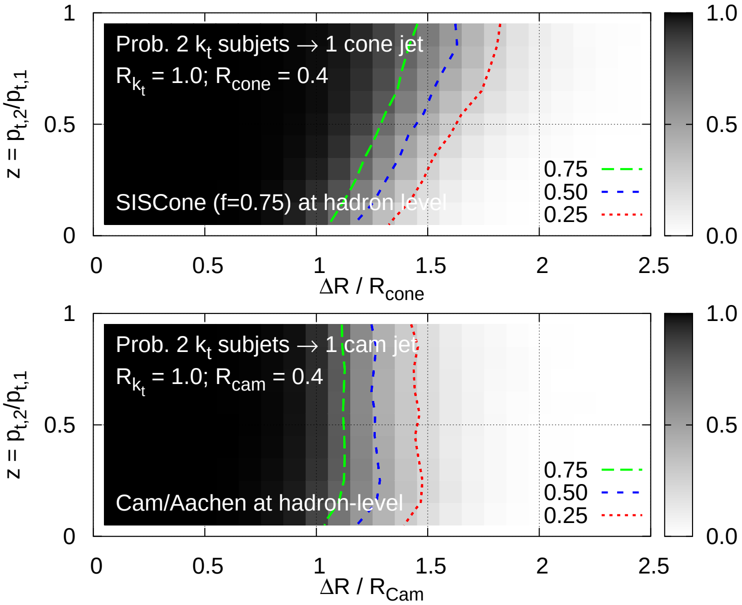
Physics quality measures

E.g. width of narrowest window (around mass-peak) containing $\sim 25\%$ of events that pass basic selection cuts, in hadronic $t\bar{t}$ and Z' events.



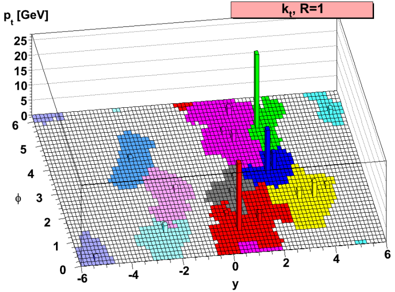
Les Houches: Cacciari, Rojo, GPS & Soyez '08

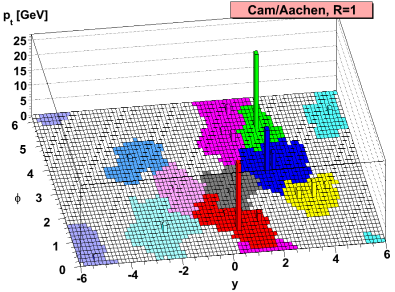
Reach of jet algorithms

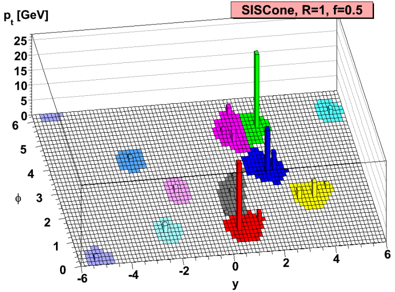


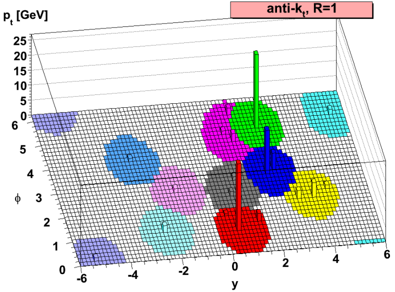
Herwig 6.510 + FastJet 2.1

Jet contours – visualised









LEP

LEP

- $M_{BSM} \sim 1$ TeV
- $M_{EW} \sim 100$ GeV $\sim \alpha_s \times M_{BSM}$
- $p_{t,\text{pileup}} \sim 25 - 100$ GeV/unit rap. $\sim M_{EW}$
- $p_{t,\text{UE}} \sim 2.5 - 5$ GeV/unit rap. $\sim \alpha_s \times M_{EW}$
- $p_{t,\text{hadr.}} \sim 0.5$ GeV/unit rap.

Multitude of scales forces us to go beyond "1 parton = 1 jet" equivalence

Tevatron is not LEP

Tevatron

- $M_{BSM} \sim 1$ TeV
- $M_{EW} \sim 100$ GeV $\sim \alpha_s \times M_{BSM}$
- $p_{t,\text{pileup}} \sim 25 - 100$ GeV/unit rap. $\sim M_{EW}$
- $p_{t,\text{UE}} \sim 2.5 - 5$ GeV/unit rap. $\sim \alpha_s \times M_{EW}$
- $p_{t,\text{hadr.}} \sim 0.5 - 1$ GeV/unit rap.

Multitude of scales forces us to go beyond "1 parton = 1 jet" equivalence

LHC is not Tevatron is not LEP

LHC

- $M_{BSM} \sim 1$ TeV
- $M_{EW} \sim 100$ GeV — $\sim \alpha_s \times M_{BSM}$
- $p_{t,\text{pileup}} \sim 25 - 100$ GeV/unit rap. — $\sim M_{EW}$
- $p_{t,\text{UE}} \sim 10 - 15$ GeV/unit rap. — $\sim \alpha_s \times M_{EW}$
- $p_{t,\text{hadr.}} \sim 0.5 - 1$ GeV/unit rap.

Multitude of scales forces us to go beyond "1 parton = 1 jet" equivalence

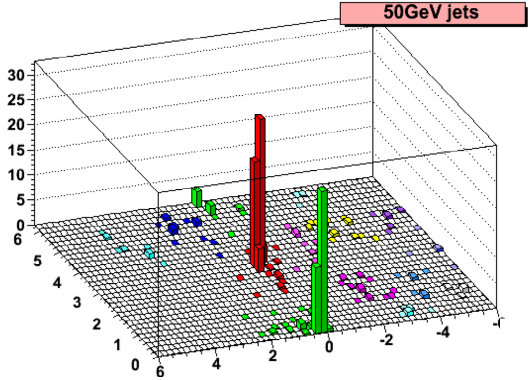
LHC is not Tevatron is not LEP

LHC

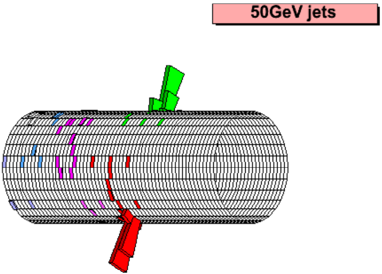
- $M_{BSM} \sim 1$ TeV
- $M_{EW} \sim 100$ GeV $\qquad \sim \alpha_s \times M_{BSM}$
- $p_{t,\text{pileup}} \sim 25 - 100$ GeV/unit rap. $\qquad \sim M_{EW}$
- $p_{t,\text{UE}} \sim 10 - 15$ GeV/unit rap. $\qquad \sim \alpha_s \times M_{EW}$
- $p_{t,\text{hadr.}} \sim 0.5 - 1$ GeV/unit rap.

Multitude of scales forces us to go beyond "1 parton = 1 jet" equivalence

Jets, pileup and areas



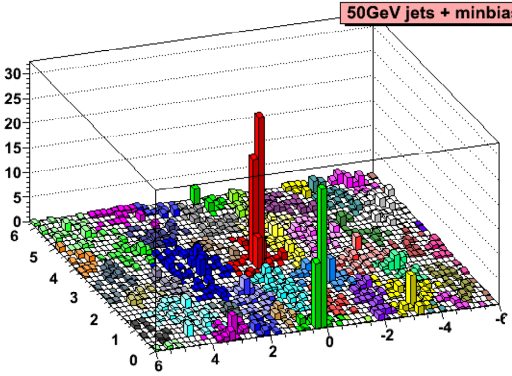
'Standard hard' event
Two well isolated jets



$\sim$ 200 particles

Clustering takes $\lesssim 1ms$

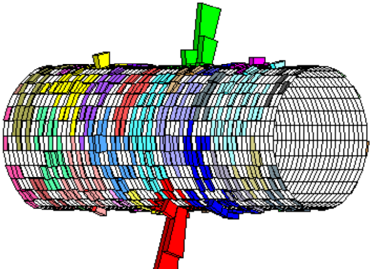
Jets, pileup and areas



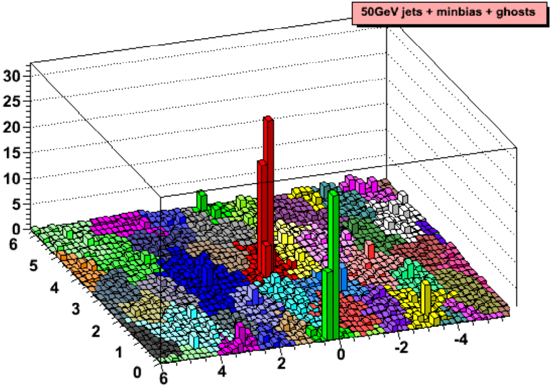
Add 10 min-bias events (moderately high lumi)

$\sim$ 2000 particles

Clustering takes $\sim 10ms$

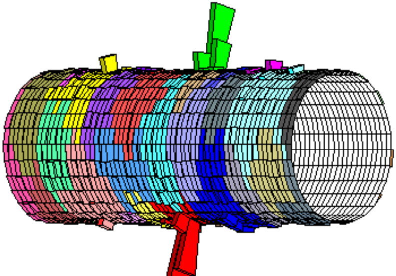


Jets, pileup and areas



Add dense coverage of infinitely soft *"ghosts"*

See how many end up in jet to measure jet area

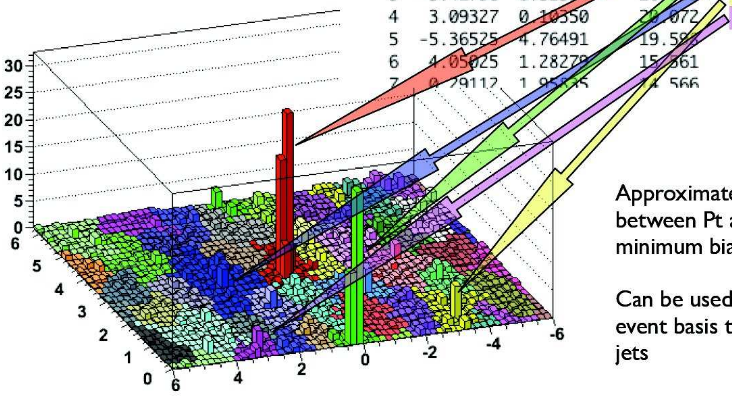


∼ 10000 particles

Clustering takes ∼ 0.2*s*

Jet areas

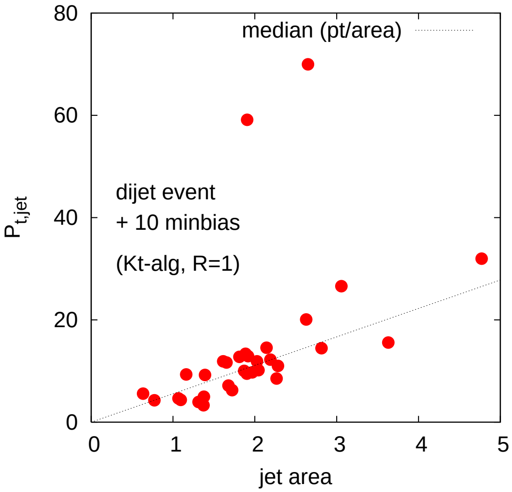
```
iev 0 (irepeat 24): number of particles = 1428
strategy used =  NlnN
number of particles = 9051
Total area: 76.0265
Expected area: 76.0265
 ijet    eta       phi        Pt        area  +-    err
    0   0.15050   3.24498    69.970     2.625 +-   0.020
    1   0.18579   0.13150    59.133     1.896 +-   0.020
    2   2.33840   3.23960    31.976     4.749 +-   0.028
    3  -3.41796   0.52394    26.595     3.084 +-   0.021
    4   3.09327   0.10350    20.072     2.688 +-   0.023
    5  -5.36525   4.76491    19.593     2.780 +-   0.012
    6   4.05025   1.28279    15.561     3.592 +-   0.028
    7   0.29112   1.95835    14.566     2.114 +-   0.018
```



Approximate linear relation between Pt and area for minimum bias jets.

Can be used on an event-by-event basis to correct the hard jets

Jet areas



Jet areas in k_t algorithm are quite varied

Because k_t-alg adapts to the jet structure

- Hard jets' contamination from min-bias $\sim$ area

  Area varies even for SISCone

- Soft jets' p_t/area tells you about min-bias normalisation and fluctuations

Median p_t/area across the set of jets in an event is a good estimator of pileup+UE in *that event*

Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

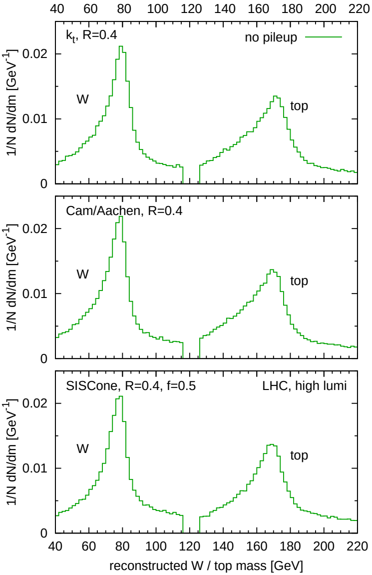
$$p_t \rightarrow p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

Same simple procedure works for a range of algorithms



Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

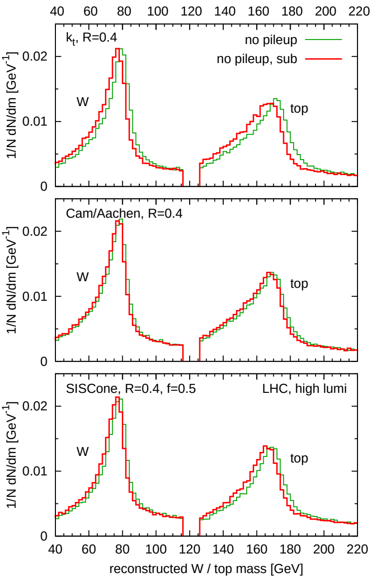
$$p_t \to p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

Same simple procedure works for a range of algorithms



Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

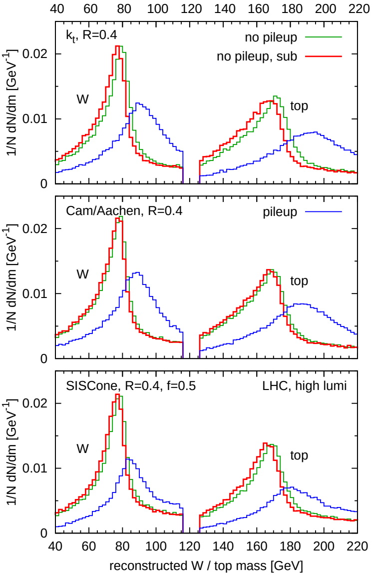
$$p_t \to p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

Same simple procedure works for a range of algorithms



Area-based subtraction

Basic Procedure:

- Use p_t/A from majority of jets (pileup jets) to get level, ρ, of pileup and UE in event
- Subtract pileup from hard jets:

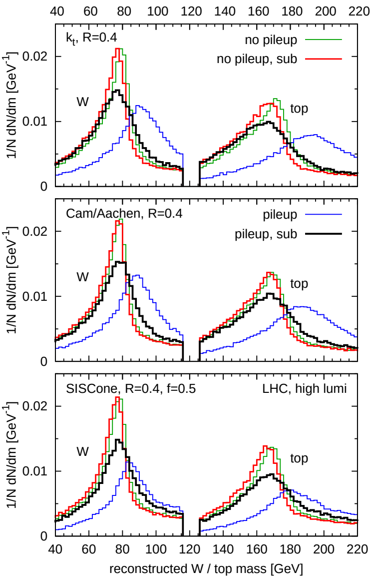
$$p_t \to p_{t,sub} = p_t - A\rho$$

Cacciari & GPS '07

Illustration:

- semi-leptonic $t\bar{t}$ production at LHC
- high-lumi pileup ($\sim$ 20 ev/bunch-X)

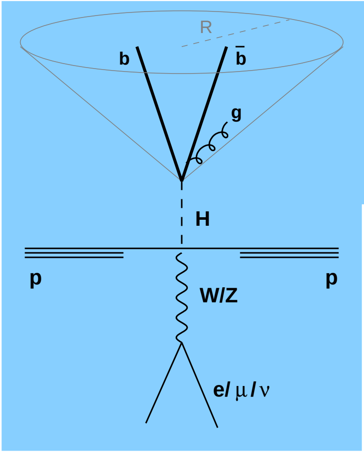
Same simple procedure works for a range of algorithms



Searching for high-p_t HW/WZ?

High-p_t light Higgs decays to $b\bar{b}$ inside a single jet. Can this be seen?

Butterworth, Davison, Rubin & GPS '08



Cluster with Cambridge/Aachen

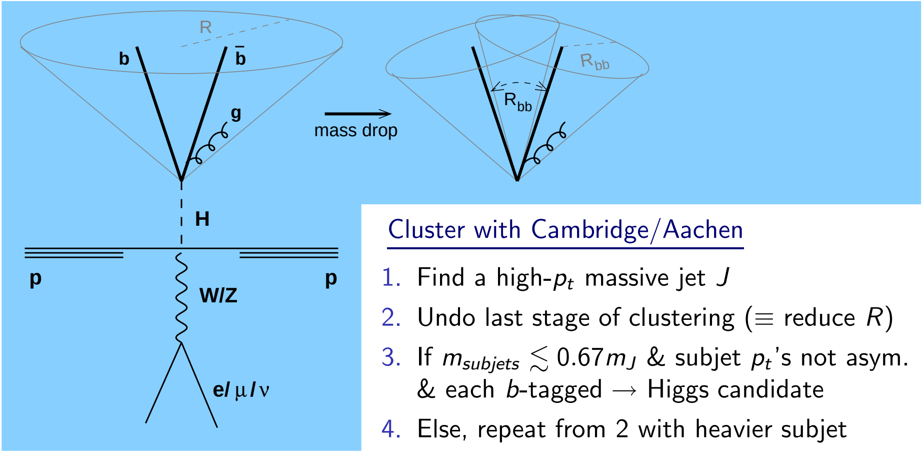
1. Find a high-p_t massive jet J
2. Undo last stage of clustering ($\equiv$ reduce R)
3. If $m_{subjets} \lesssim 0.67 m_J$ & subjet p_t's not asym. & each b-tagged $\rightarrow$ Higgs candidate
4. Else, repeat from 2 with heavier subjet

Then on the Higgs-candidate: *filter* away UE/pileup by reducing $R \rightarrow R_{filt}$, take *three hardest subjets* (keep LO gluon rad^n) + require b-tags on two hardest.

Searching for high-p_t HW/WZ?

High-p_t light Higgs decays to $b\bar{b}$ inside a single jet. Can this be seen?

Butterworth, Davison, Rubin & GPS '08



Cluster with Cambridge/Aachen

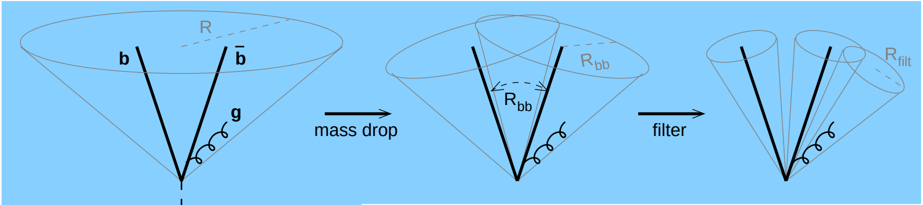
1. Find a high-p_t massive jet J
2. Undo last stage of clustering ($\equiv$ reduce R)
3. If $m_{subjets} \lesssim 0.67 m_J$ & subjet p_t's not asym. & each b-tagged $\rightarrow$ Higgs candidate
4. Else, repeat from 2 with heavier subjet

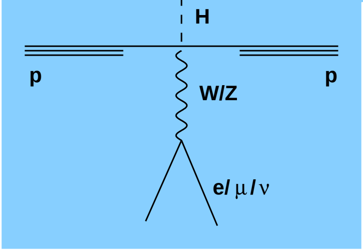
Then on the Higgs-candidate: *filter* away UE/pileup by reducing $R \rightarrow R_{filt}$, take *three hardest subjets* (keep LO gluon rad^n) + require b-tags on two hardest.

Searching for high-p_t HW/WZ?

High-p_t light Higgs decays to $b\bar{b}$ inside a single jet. Can this be seen?

Butterworth, Davison, Rubin & GPS '08





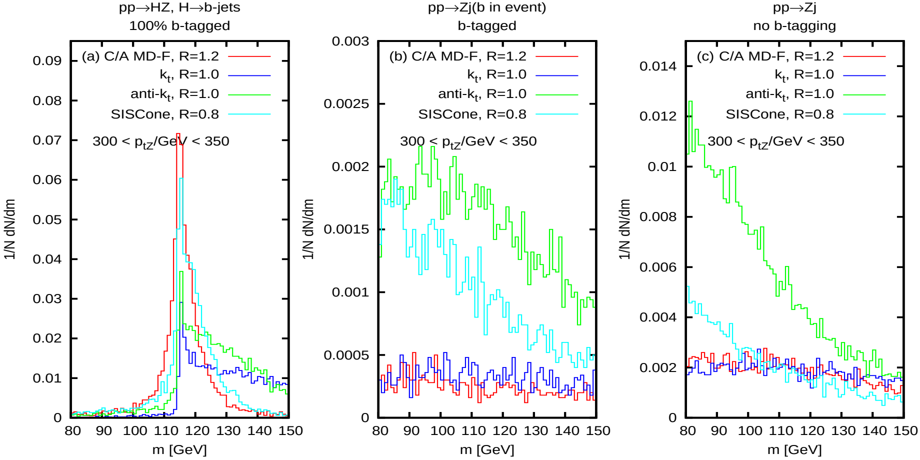
Cluster with Cambridge/Aachen

1. Find a high-p_t massive jet J
2. Undo last stage of clustering ($\equiv$ reduce R)
3. If $m_{subjets} \lesssim 0.67 m_J$ & subjet p_t's not asym. & each b-tagged $\rightarrow$ Higgs candidate
4. Else, repeat from 2 with heavier subjet

Then on the Higgs-candidate: *filter* away UE/pileup by reducing $R \rightarrow R_{filt}$, take *three hardest subjets* (keep LO gluon rad^n) + require b-tags on two hardest.

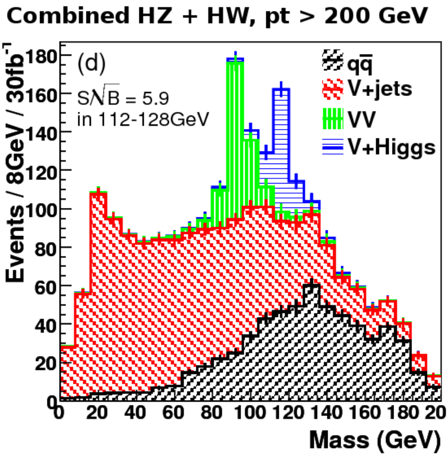
Compare with "standard" algorithms

Check mass spectra in HZ channel, $H \to b\bar{b}$, $Z \to \ell^+\ell^-$



Cambridge/Aachen (C/A) with mass-drop and filtering (MD/F) works best

combine HZ and HW, $p_t > 200$ GeV



- Take $Z \to \ell^+\ell^-$, $Z \to \nu\bar{\nu}$, $W \to \ell\nu$ $\quad \ell = e, \mu$
- $p_{tV}, p_{tH} > 200$ GeV
- $|\eta_V|, |\eta_H| < 2.5$
- Assume real/fake b-tag rates of 0.7/0.01.
- Some extra cuts in HW channels to reject $t\bar{t}$.
- Assume $m_H = 115$ GeV.

At 5.9σ for 30 fb^{-1} this looks like a competitive channel for light Higgs discovery. **Deserves serious exp. study!**

Conclusions

- Know the algorithms and make sure they're fully described somewhere
- Be sure they're infrared & collinear safe (there's plenty of choice)

- LHC's a very different environment from what we've had before
- One should explore a range of jet definitions for any given analysis
- Both to optimize the analysis
- And to verify robustness of conclusions wrt jet-def

- Jets are *not just partons*
- Tell you lots about UE, pileup $\longrightarrow$ model it better, subtract it better
- Have substructure, which can help reveal underlying physics

- As data arrives at LHC, new ideas in jet-definintion will arise, geared to actual LHC conditions. Keep your eyes open for best tools.

Conclusions

- Know the algorithms and make sure they're fully described somewhere
- Be sure they're infrared & collinear safe (there's plenty of choice)

- LHC's a very different environment from what we've had before
- One should explore a range of jet definitions for any given analysis
- Both to optimize the analysis
- And to verify robustness of conclusions wrt jet-def

- Jets are *not just partons*
- Tell you lots about UE, pileup $\longrightarrow$ model it better, subtract it better
- Have substructure, which can help reveal underlying physics

- As data arrives at LHC, new ideas in jet-definintion will arise, geared to actual LHC conditions. Keep your eyes open for best tools.

Conclusions

- Know the algorithms and make sure they're fully described somewhere
- Be sure they're infrared & collinear safe (there's plenty of choice)

- LHC's a very different environment from what we've had before
- One should explore a range of jet definitions for any given analysis
- Both to optimize the analysis
- And to verify robustness of conclusions wrt jet-def

- Jets are *not just partons*
- Tell you lots about UE, pileup $\rightarrow$ model it better, subtract it better
- Have substructure, which can help reveal underlying physics

- As data arrives at LHC, new ideas in jet-definintion will arise, geared to actual LHC conditions. Keep your eyes open for best tools.

Conclusions

- Know the algorithms and make sure they're fully described somewhere
- Be sure they're infrared & collinear safe (there's plenty of choice)

- LHC's a very different environment from what we've had before
- One should explore a range of jet definitions for any given analysis
- Both to optimize the analysis
- And to verify robustness of conclusions wrt jet-def

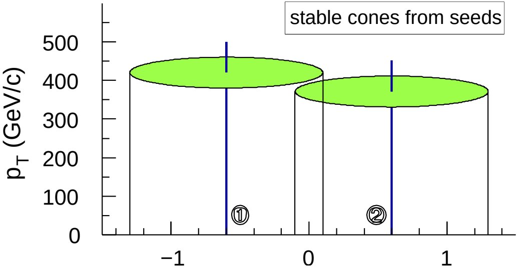
- Jets are *not just partons*
- Tell you lots about UE, pileup $\rightarrow$ model it better, subtract it better
- Have substructure, which can help reveal underlying physics

- As data arrives at LHC, new ideas in jet-definintion will arise, geared to actual LHC conditions. Keep your eyes open for best tools.

EXTRA MATERIAL

IC-SM theory issues

Use of seeds is *dangerous*



Extra soft particle adds new seed $\longrightarrow$ changes final jet configuration.

This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

Kilgore & Giele '97

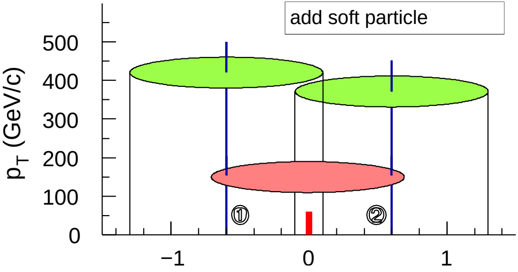
Solution: add extra seeds at midpoints of all pairs, triplets, ...of stable cones.

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses...

IC-SM theory issues

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

Kilgore & Giele '97

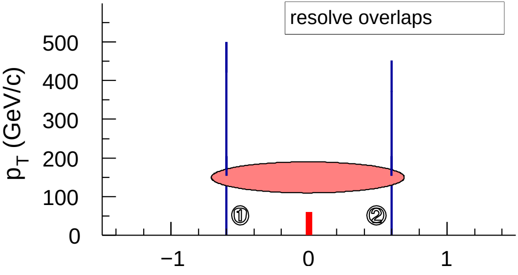
Solution: add extra seeds at midpoints of all pairs, triplets, …of stable cones.

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses…

IC-SM theory issues

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

Kilgore & Giele '97

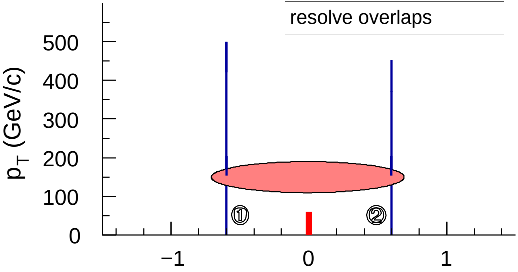
Solution: add extra seeds at midpoints of all pairs, triplets, ...of stable cones.

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses...

IC-SM theory issues

Use of seeds is *dangerous*



Extra soft particle adds new seed → changes final jet configuration.

This is **IR unsafe**.
Divergences of real and virtual contributions do not cancel at $\mathcal{O}\left(\alpha_s^4\right)$

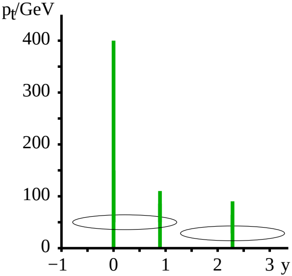
Kilgore & Giele '97

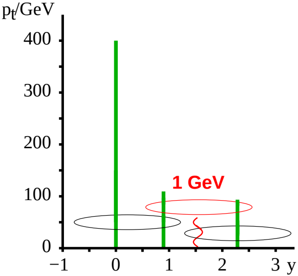
Solution: add extra seeds at midpoints of all pairs, triplets, . . . of stable cones.

Seymour '97 (?)

NB: only in past 3-4 years has this fix appeared in CDF and D0 analyses. . .

Midpoint IR problem





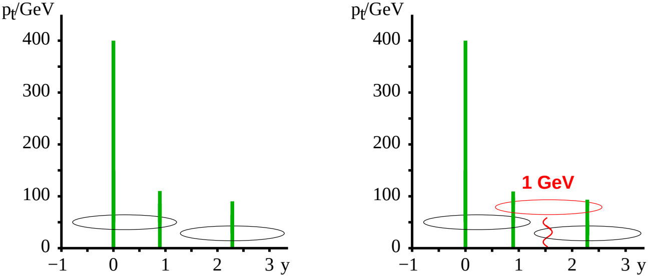
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem



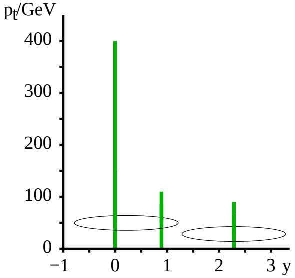
Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

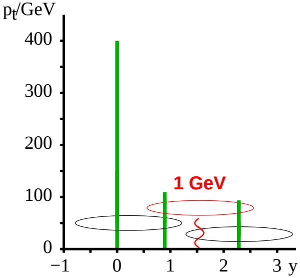
Midpoint cone alg. misses some stable cones; extra soft particle $\longrightarrow$ extra starting point $\longrightarrow$ extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

Midpoint IR problem





Stable cones with midpoint:	{1,2} & {3}	{1,2} & {2,3} & {3}
Jets with midpoint ($f = 0.5$)	{1,2} & {3}	{1,2,3}

Midpoint cone alg. misses some stable cones; extra soft particle → extra starting point → extra stable cone found

MIDPOINT IS INFRARED UNSAFE

Or collinear unsafe with seed threshold

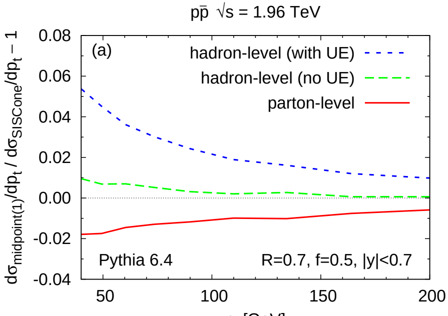
How much does IR safety *really* matter?

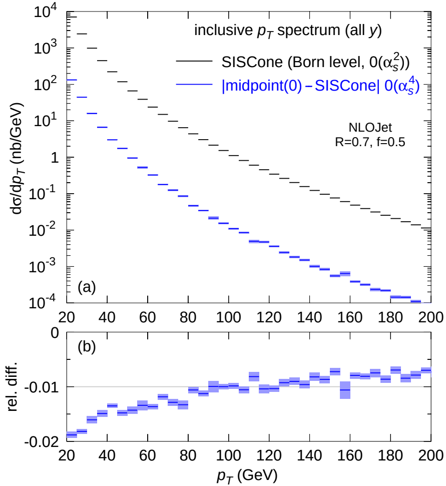
Compare midpoint and SISCone

Result depends on observable:

- inclusive jet spectrum is the least sensitive (affected at NNLO)
- larger differences (5 – 10%) at hadron level

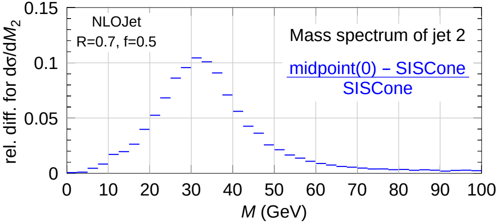
seedless reduces UE effect





IR safety & multi-jet observables

Look at jet masses in multijet events. NB: Jet masses reconstruct boosted *W*/*Z*/*H*/*top* in BSM searches



Select 3-jet events

$$p_{t1,2,3} > \{120, 60, 20\} \text{ GeV},$$

Calculate LO jet-mass spectrum for jet 2, compare midpoint with SISCone.

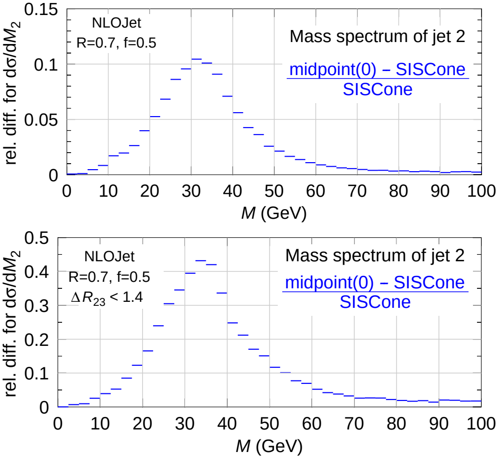
- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

e.g. for jets from common decay chain

In complex events, IR safety matters

IR safety & multi-jet observables

Look at jet masses in multijet events. NB: Jet masses reconstruct boosted $W/Z/H/top$ in BSM searches



Select 3-jet events

$p_{t1,2,3} > \{120, 60, 20\}$ GeV,

Calculate LO jet-mass spectrum for jet 2, compare midpoint with SISCone.

- 10% differences by default
- **40% differences** with extra cut $\Delta R_{2,3} < 1.4$

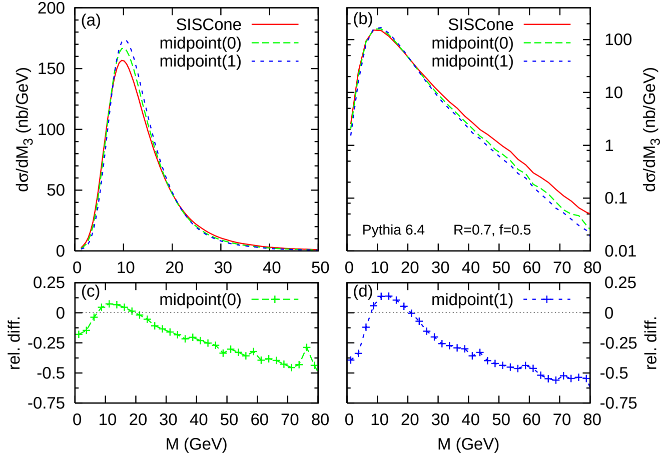
  e.g. for jets from common decay chain

In complex events, IR safety matters

Multi-jet observables: after showering

Showering puts in many extra seeds: missing stable cones (in midpoint) should be less important?

Look at 3rd jet mass distribution (no ΔR_{23} cut):



Missing stable cones → 50% effects even after showering

k_t: can we do better than N^2?

There are $N(N-1)/2$ distances d_{ij} — surely we have to calculate them all in order to find smallest?

k_t distance measure is partly *geometrical:*

- Consider smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2) R_{ij}^2$
- Suppose $k_{ti} < k_{tj}$
- Then: $R_{ij} \le R_{i\ell}$ for any $\ell \neq j$. [If $\exists\ \ell$ s.t. $R_{i\ell} < R_{ij}$ then $d_{i\ell} < d_{ij}$]

In words: if i, j form smallest d_{ij} then j is geometrical nearest neighbour (GNN) of i.

k_t distance need only be calculated between GNNs

Each point has 1 GNN → need only calculate N d_{ij}'s

k_t: can we do better than N^2?

There are $N(N-1)/2$ distances d_{ij} — surely we have to calculate them all in order to find smallest?

k_t distance measure is partly *geometrical:*

- Consider smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2) R_{ij}^2$
- Suppose $k_{ti} < k_{tj}$
- Then: $R_{ij} \leq R_{i\ell}$ for any $\ell \neq j$. [If $\exists\ \ell$ s.t. $R_{i\ell} < R_{ij}$ then $d_{i\ell} < d_{ij}$]

In words: if i, j form smallest d_{ij} then j is geometrical nearest neighbour (GNN) of i.

k_t distance need only be calculated between GNNs

Each point has 1 GNN $\rightarrow$ need only calculate N d_{ij}'s

k_t: can we do better than N^2?

There are $N(N-1)/2$ distances d_{ij} — surely we have to calculate them all in order to find smallest?

k_t distance measure is partly *geometrical:*

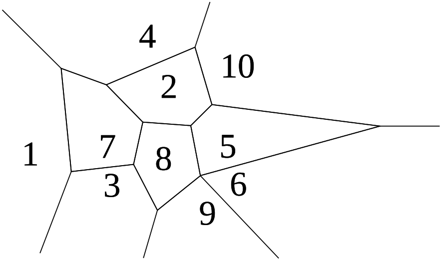
- Consider smallest $d_{ij} = \min(k_{ti}^2, k_{tj}^2) R_{ij}^2$
- Suppose $k_{ti} < k_{tj}$
- Then: $R_{ij} \le R_{i\ell}$ for any $\ell \neq j$. [If $\exists\ \ell$ s.t. $R_{i\ell} < R_{ij}$ then $d_{i\ell} < d_{ij}$]

In words: if i, j form smallest d_{ij} then j is geometrical nearest neighbour (GNN) of i.

k_t distance need only be calculated between GNNs

Each point has 1 GNN $\rightarrow$ need only calculate N d_{ij}'s

Finding Geom Nearest Neighbours



Given a set of vertices on plane (1...10) a *Voronoi diagram* partitions plane into cells containing all points closest to each vertex

Dirichlet '1850, Voronoi '1908

A vertex's nearest other vertex is always in an adjacent cell.

E.g. GNN of point 7 will be found among 1,4,2,8,3 (it turns out to be 3)

Construction of Voronoi diagram for N points: $N \ln N$ time — Fortune '88

Update of 1 point in Voronoi diagram: $\ln N$ time

Devillers '99 [+ related work by other authors]

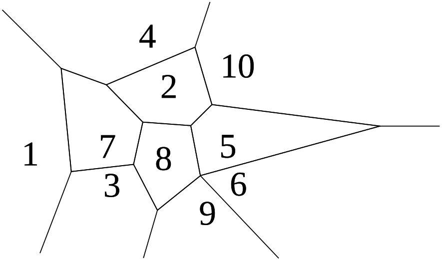
Convenient C++ package available: **CGAL** — `http://www.cgal.org`

Assemble with other comp. science methods: **FastJet**

Cacciari & GPS, hep-ph/0512210

`http://www.lpthe.jussieu.fr/~salam/fastjet/`

Finding Geom Nearest Neighbours



Given a set of vertices on plane (1...10) a *Voronoi diagram* partitions plane into cells containing all points closest to each vertex

Dirichlet '1850, Voronoi '1908

A vertex's nearest other vertex is always in an adjacent cell.

E.g. GNN of point 7 will be found among 1,4,2,8,3 (it turns out to be 3)

Construction of Voronoi diagram for N points: $N \ln N$ time   Fortune '88

Update of 1 point in Voronoi diagram: $\ln N$ time

Devillers '99 [+ related work by other authors]

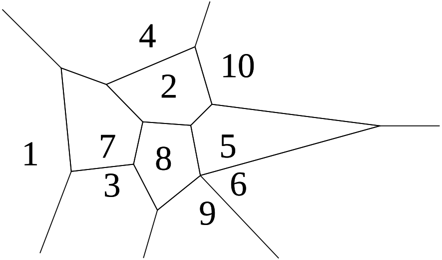
Convenient C++ package available: **CGAL**   http://www.cgal.org

Assemble with other comp. science methods: **FastJet**

Cacciari & GPS, hep-ph/0512210

http://www.lpthe.jussieu.fr/~salam/fastjet/

Finding Geom Nearest Neighbours



Given a set of vertices on plane (1...10) a *Voronoi diagram* partitions plane into cells containing all points closest to each vertex

Dirichlet '1850, Voronoi '1908

A vertex's nearest other vertex is always in an adjacent cell.

E.g. GNN of point 7 will be found among 1,4,2,8,3 (it turns out to be 3)

Construction of Voronoi diagram for N points: $N \ln N$ time Fortune '88

Update of 1 point in Voronoi diagram: $\ln N$ time

Devillers '99 [+ related work by other authors]

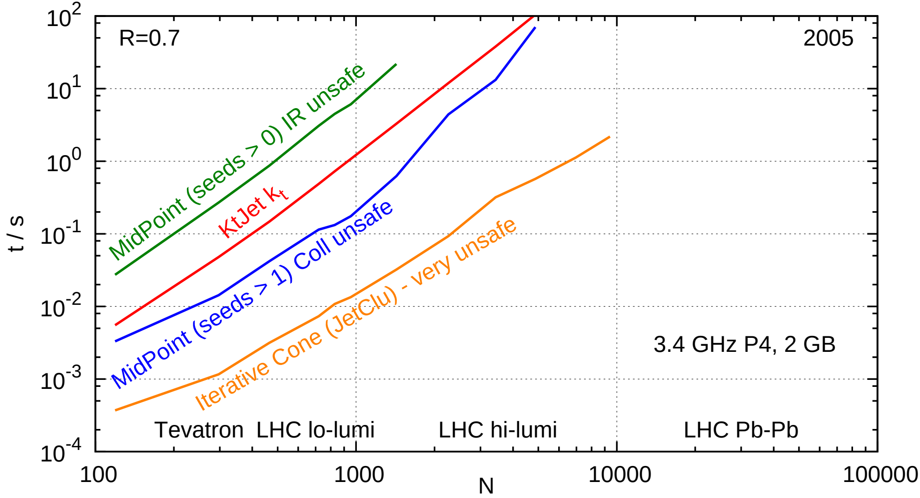
Convenient C++ package available: **CGAL** http://www.cgal.org

Assemble with other comp. science methods: **FastJet**

Cacciari & GPS, hep-ph/0512210

http://www.lpthe.jussieu.fr/~salam/fastjet/

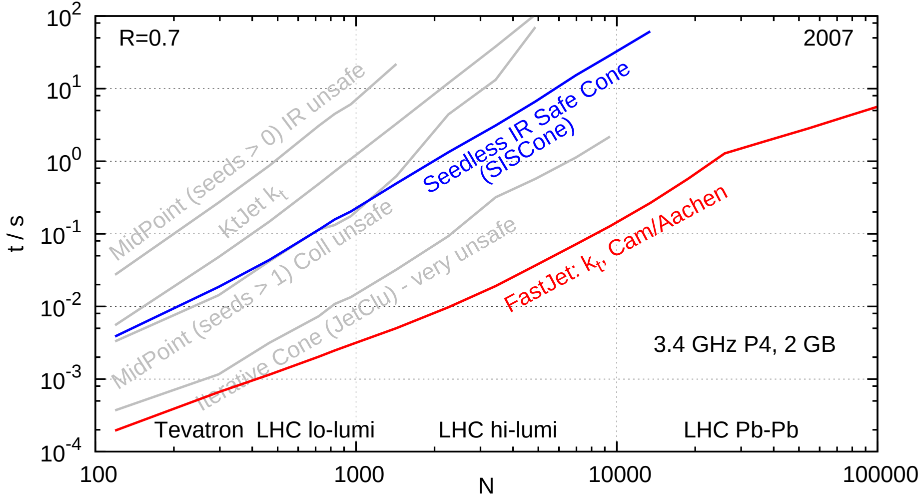
Status in 2005



Single package, **FastJet**, to access all developments, natively (k_t, Cam/Aachen) or as plugins (SISCone): Cacciari, GPS & Soyez '05–08

http://www.lpthe.jussieu.fr/~salam/fastjet/

Status in 2007



Single package, **FastJet**, to access all developments, natively (k_t, Cam/Aachen) or as plugins (SISCone): Cacciari, GPS & Soyez '05–08

`http://www.lpthe.jussieu.fr/~salam/fastjet/`

Optimal R?

	Jet $\langle \delta p_t \rangle$ given by product of dependence on			
	scale	colour factor	R	$\sqrt{s}$
pert. radiation	$\sim \frac{\alpha_s(p_t)}{\pi} p_t$	C_i	$\ln R + \mathcal{O}(1)$	–
hadronisation	Λ_h	C_i	$-1/R + \mathcal{O}(R)$	–
UE	Λ_{UE}	–	$R^2/2 + \mathcal{O}(R^4)$	s^ω

To get best experimental resolutions, minimise contributions from all 3 components.

Here: sum of squared means

Better still: calculate flucts

NB: this is rough picture

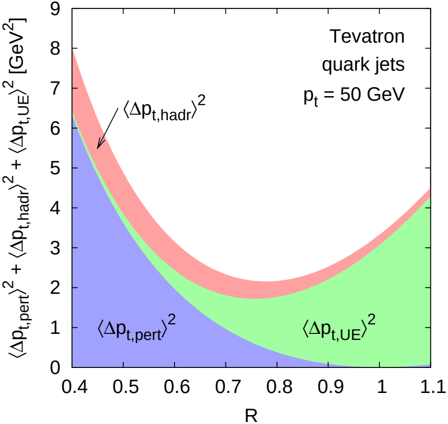
details of p_t scaling wrong

But can still be used to understand general principles.

Dasgupta, Magnea & GPS '07

Optimal R?

	Jet $\langle \delta p_t \rangle$ given by product of dependence on			
	scale	colour factor	R	$\sqrt{s}$
pert. radiation	$\sim \frac{\alpha_s(p_t)}{\pi} p_t$	C_i	$\ln R + \mathcal{O}(1)$	–
hadronisation	Λ_h	C_i	$-1/R + \mathcal{O}(R)$	–
UE	Λ_{UE}	–	$R^2/2 + \mathcal{O}(R^4)$	s^ω



To get best experimental resolutions, minimise contributions from all 3 components.

Here: sum of squared means

Better still: calculate flucts

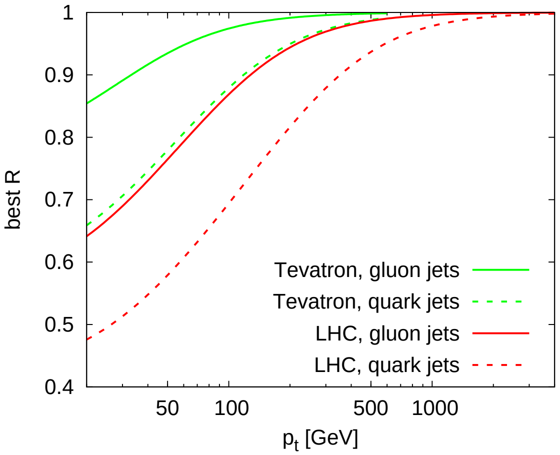
NB: this is rough picture

details of p_t scaling wrong

But can still be used to understand general principles.

Dasgupta, Magnea & GPS '07

Optimal R v p_t, proc., collider

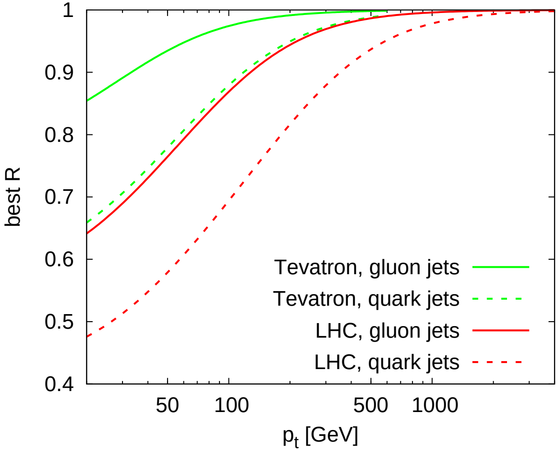


Basic messages

- higher $p_t \longrightarrow$ larger R
  Most say opposite
- larger R for gluons than quarks Gluon jets wider
- smaller R at LHC than Tevatron UE larger

This kind of information is the start of what might go into **"auto-focus for jetography"**

Optimal R v p_t, proc., collider

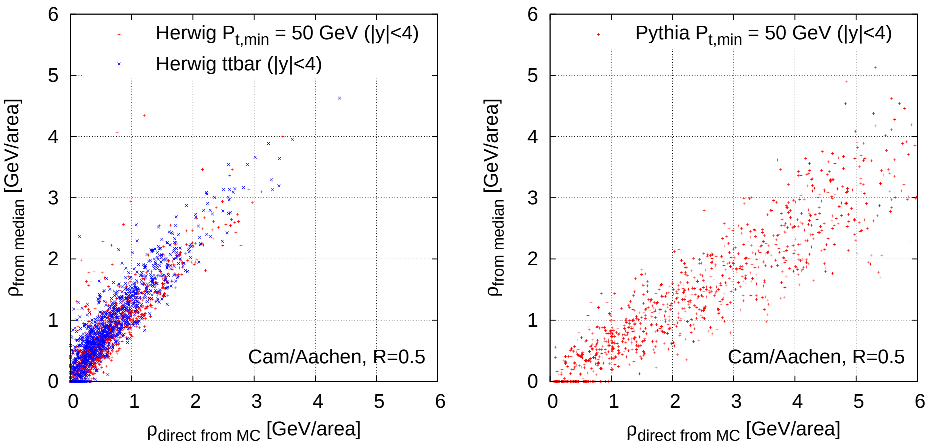


Basic messages

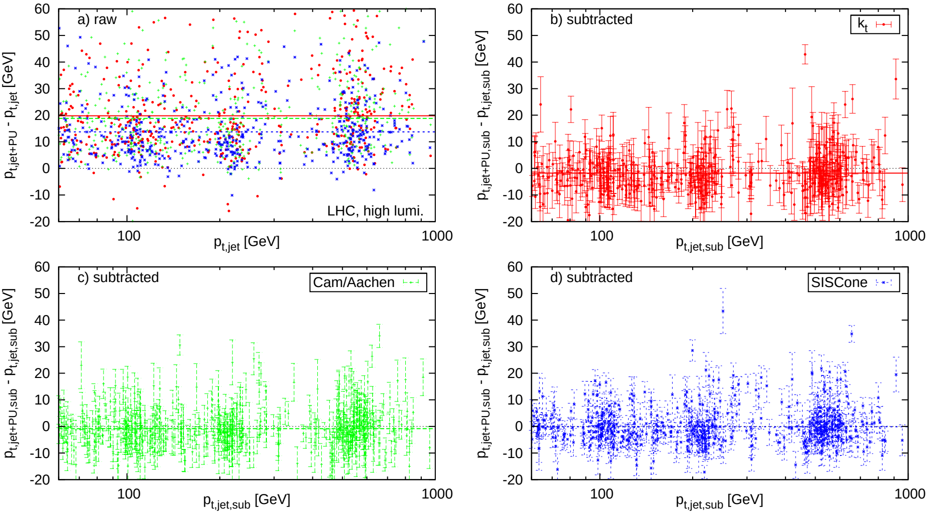
- higher $p_t \longrightarrow$ larger R
  Most say opposite
- larger R for gluons than quarks Gluon jets wider
- smaller R at LHC than Tevatron UE larger

This kind of information is the start of what might go into **"auto-focus for jetography"**

Correlate MC & measured UE (LHC)



Check subtraction



High-p_t HV: are you serious?

How can high-p_t HV do better than low p_t search?

- Acceptance improves at high-p_t
  - H and V both more likely to be central
  - H and V, since boosted, have central decay products
  - Decay products have high p_t, so you always see them

- Backgrounds fall faster than signal
  - for V$b\bar{b}$, not gigantic effect (factor 2–3), but there
  - for $t\bar{t}$, top mass $\to$ b's with $p_t = 60 - 70$ GeV, just like light Higgs at rest; much harder to fake high-p_t H

- At high-p_t capture most of gluon radiation from $H \to b\bar{b} + X$, without too much UE; much harder at low p_t. $\to$ Better mass resolution.

b-tagging, mass-dep for high-$p-t$ HV