



GPU implementation of HGCal 2D Clustering

Z. Chen, T. Di Pilato, B. Kilian, 15-17 April, 2019



Day 3

GPU `calculateLocalDensity` (Ziheng & Ben):

1. Now fully implemented on GPU, verified results with those of CPU function.
2. Changed threading from 1 thread per bin to 1 thread per RecHit to avoid running kernels for empty bins.

CPU `calculateDistanceToHigher` (Tony):

1. Fully implemented the histogram-based function on CPU and synchronized with the CPU kdTree-based one.
- 



Status

Original code	New CPU code	New GPU code
<code>calculateLocalDensity()</code>	<code>calculateLocalDensity()</code>	<code>calculateLocalDensity()</code>
<code>calculateDistanceToHigher()</code>	<code>calculateDistanceToHigher()</code>	<code>calculateDistanceToHigher()</code>
<code>findAndAssignClusters()</code>	<code>findAndAssignClusters()</code>	<code>findAndAssignClusters()</code>

Color code:

- NOT STARTED
 - IN PROGRESS
 - IN VALIDATION
 - FULLY VALIDATED
- 



Future Plans


- Implement a new GPU-friendly algorithm for `calculateDistanceToHigher`
 - Finish implementing clustering algorithm (`calculateDistanceToHigher`, `findAndAssignClusters`) on GPU
 - Optimization, optimization, optimization!
- 



Starting Point

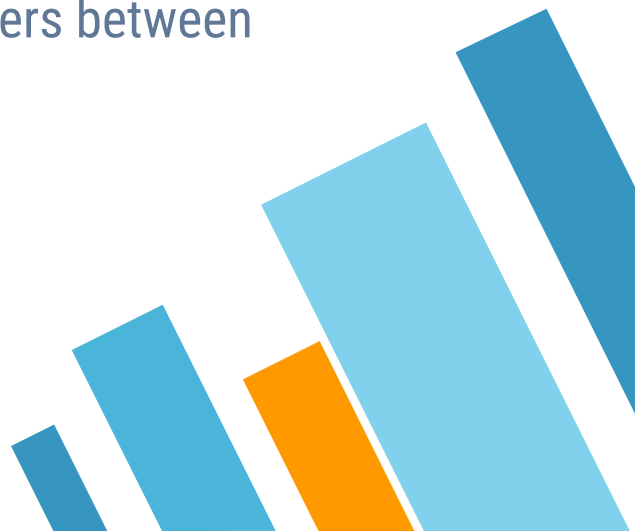
1. Latest **CPU kdTree-based** version of HGCal 2D clustering algorithm in CMSSW_10_6_0_pre2
2. Input data structure -- 2D histogram
3. Code of this effort from previous Hackathon

Goal

1. CPU histogram-based clustering
 2. GPU histogram-based clustering
 3. Get the same result as **CPU kdTree-based** clustering
- 




Day 1

1. Established starting point: setup and run.
 2. Checked the result from existing code from previous Hackathons. Discrepancy in density and clusters between CPU kdTree-based
 3. Started to develop our functions
 - a. `calculateLocalDensity`
- 



Plan for Day 2

1. Complete `calculateLocalDensity` for CPU and GPU histogram-based functions. Fully synchronized with CPU kdTree-based one.
 2. Work on `calculateDistanceToHigher` CPU and GPU histogram-based functions. And compare the results with CPU kdTree-based one
- 




Day 2

CPU `calculateLocalDensity` (Tony & Ziheng):

1. Bin-based CPU algorithm fully synchronized with current kdTree version

GPU `calculateLocalDensity` (Ben):

2. Resolved data structure issues from previous hackathon (never pass vectors to a kernel)
 3. Can now run our GPU kernel and access meaningful data
- 



Plan for Day 3

1. complete the implementation of `calculateLocalDensity` on GPU and verify GPU results with CPU code
 2. eventually start to implement the new `calculateDistanceToHigher` function on CPU
- 