Proposal for a time-based clustering algorithm for muons on silicon sensors

Introduction

The goal of this document is starting a discussion about the way to implement a time-based clustering algorithm for hits on silicon sensors. The software developed draws inspiration from the algorithm described in the CMS wiki page. The source code is available in github.
The designed processors covers the first part of the digitization, i.e. all the steps related to the sensor response but the last one (cluster charge re-weighting):

- the track segmentation computes the ionization points
- the drifted points on the sensor surface are calculated from the ionization points
- a diffusion process is applied to drifted points in order to obtain charges on sensor pixels
- charges on pixels are aggregated.

Question: is it necessary to take into consideration the cluster charge re-weighting for the muon collider detector?

For a time-based clustering algorithm the basic requirement is to have all the hits sorted according to the time; the collection of hits read by the processor must be considered without any time order relation. Since the collection of hits is stored in memory, a suitable in-memory sorting algorithm can be used. A major improvement is partitioning the sorted collection according to the detector element where the hit takes place. If it is acceptable to consider the entire digitization process for a given hit completely contained within the sensitive or sensor then it is possible to perform the computation in parallel over all the detector elements.

Questions: is it acceptable to perform the digitization for a sensitive element in isolation? What kind of inter-sensitive physical processes must be taken into consideration?

The track segmentation and the calculation of drifted points are steps related only with the hit, without any correlation with the time. They can be computed once and for all and stored in memory, tagging each drifted point with time of the corresponding hit. The core of the time-base clustering algorithm takes place starting from the diffusion of the charge over the matrix of pixels and further aggregation.

The vertex barrel consists of many layers, each layer contains a sensitive ladder segmented in different sensors, without any gap between sensors. The values of the charge of all the pixels for a given sensor are acquired at the same time by the readout.

Questions: is it acceptable to have the same clock time (sampling frequency) for all the sensitive elements? Is it necessary to take into consideration possible delays between sensors for a given ladder? If so, is the delay a constant or some kind of randomized value?

The basic requirements for a time aggregation is processing a bunch of sorted hits within a time window. The length of the window depends on the implementation of the time-based clustering algorithm, it can be related to the clock time of the readout or be related to other physical aspects. The time window can slide over the time with or without overlapping. The length of the window itself can change over the time. The clustering algorithm can deal with a single time window for all the detector elements or can have different windows evolving independently, one window per sensitive element or sensor.

A simple time-based clustering algorithm makes use of a non-overlapping time window with constant length. All the hits that occur within the time window are processed once and for all, with no side effects in the previous or the next window. The charge of a pixel for a drifted point is considered as an instantaneous event, with no charge evolution over the time. For the situation previously described the diffusion process and aggregation follow the same steps already implemented by the processor. If it is required to calculate a single time for a given charged pixel it can be used the time of the first drifted point in the window and then a randomized offset can be applied.

Questions: what could be the value of the constant length? Is it related to any physical feature? Is it acceptable the requirement of non-overlapping window? Can the single time for charged pixels be evaluated in other ways?

A more complex algorithm could resort to the simulation of the evolution of the charge for any drifted point of a given pixel. The charging process could produce effects in consecutive windows, if the length of the window is comparable with the time of the process itself. As a coarse approximation the evolution could be modelled as a time-dependent function that multiplies the smearing transformation.

In any case the outcome of the clustering algorithm is a set of matrices of pixel-digits, one matrix per sensor and time. Each matrix is tagged with the timestamp calculated by the clustering algorithm and with detector element ID. Any other information contained in the original hit is lost.

Questions: is it necessary to keep track, somehow, of the momentum associated with the hit?