Simulation of the RunIIb L1 Cal Trigger and EM Algorithm Optimization

Sabine Lammers\textsuperscript{1}, Gregory Pawloski\textsuperscript{2}

\textsuperscript{1}Columbia University, New York, NY
\textsuperscript{2}Rice University, Houston, TX

Abstract
A study of the search for electromagnetic objects using the sliding windows algorithm in the RunIIb L1 calorimeter trigger is presented. An optimized algorithm was found which consists of trigger objects formed by 2 trigger towers, no separation requirement between trigger objects, an isolation fraction requirement that uses a 2x3 region of EM trigger towers, and a EM/HAD fraction requirement that uses a 2x1 region of hadronic trigger towers.

Send comments to: pawloski@fnal.gov and lammers@fnal.gov
## Contents

1 **Introduction**  

2 **RunIIb L1 Calorimeter Sliding Windows Algorithm**  
   2.1 Algorithm Flow  
   2.2 Finding EM, Jet and Tau objects  
   2.3 Hardware Implementation  

3 **Software Simulation Tools**  
   3.1 Trigger Rate Tool  
   3.2 tsim_11cal2b  

4 **EM Algorithm Study**  
   4.1 Potential EM Algorithms  
   4.2 Real Data and Monte Carlo Samples  
   4.3 Efficiencies and Rates  
      4.3.1 1x1 Trigger Objects or Atlas Trigger Objects  
      4.3.2 Minimum Separation between Trigger Objects  
      4.3.3 Cuts Using Thresholds or Fractions  
      4.3.4 Choice of Geometry for Cuts  
   4.4 Final RunIIb EM Algorithm  

2  

5  

6  

7  

8  

8  

14  

14  

17  

17  

1
1 Introduction

The introduction of clustering in the Level 1 calorimeter trigger via the sliding windows algorithm is one of the major upgrades of the RunIIb detector. This note documents a study that was performed to choose the optimal parameters for the sliding windows algorithm for EM object identification. In Section 2, a short description of the algorithm is given, including the differentiation between EM, hadronic and tau-like clusters, along with some features of the EM algorithm hardware implementation. Section 3 documents some of the software tools that were adapted and used for the studies presented here. These tools are and will continue to be the basis for most RunIIb trigger studies. Section 4 gives a detailed discussion of the parameters of the EM sliding windows algorithm, the efficiency and rate studies that were performed to find the optimal configuration of those parameters, and the final algorithm choice.

2 RunIIb L1 Calorimeter Sliding Windows Algorithm

The L1 calorimeter trigger is being upgraded for RunIIb in order to perform clustering for object identification at level 1. The sliding windows algorithm aims to find the optimal region of the calorimeter for the inclusion of energy from jets or EM objects by moving a window’s grid across the calorimeter eta-phi space so as to maximize the transverse energy seen within the window. Variants of the sliding windows algorithm have been studied extensively at different HEP experiments, and have been found to be highly efficient at triggering on electromagnetic and jet objects, while not having the latency drawbacks of iterative clustering algorithms. For a full discussion of the merits of the sliding windows algorithm, see the RunIIb TDR [1].

2.1 Algorithm Flow

The sliding windows algorithm is implemented in the DØ RunIIb simulation in essentially three steps. A diagrammatic example is given in Figure 1. The trigger tower energies taken from the electromagnetic and hadronic calorimeter are fed as input to the algorithm. Regions of interest (ROI) are constructed by summing all the trigger tower energies in a sliding window across the entire calorimeter eta-phi plane. Each window is indexed by its lower left (LL) corner, and a new ROI-space grid with the same dimension as the TT-space grid is constructed containing the summed window energies. The size of the sliding window is a fixed parameter of the algorithm and is chosen based upon the type of object being searched for. The ROI-grid is constructed with the idea that each entry should represent the greater portion of the transverse energy of a candidate object. In the next step, a local maximum (LM) is searched for in the ROI-grid using a declustering scheme, which avoids multiple counting of jet or EM object candidates. The parameters of the declustering scheme are again chosen dependent on the object being searched for. An example of one possible declustering scheme is shown in Figure 2.

A comparison between each entry of the ROI-grid and every other entry in the declustering region
Figure 1: The four stages of algorithm flow for the sliding windows algorithm. The grid represents a subset of the eta-phi plane. In the first stage, the TT transverse energies are mapped into EM and HAD grid arrays indexed by the L1Cal eta-phi numbering scheme. In the second stage, regions of interest (ROI) are formed by summing the TT transverse energies in each sliding window indexed by the lower left corner (in this example, the sliding window is 2x2). In the third stage, local maxima are found by a declustering algorithm which identifies the likely position of a candidate object. In the fourth stage, TT’s are summed according to the algorithm prescription (in this example, the summed region is 4x4) and shaping cuts are made to identify objects.

Figure 2: Examples of 5x5 (for jets) and 3x3 (for EM) declustering schemes. The candidate ROI is marked with an “x” in the center of the grid. If the candidate ROI is more energetic than the neighboring ROI’s marked with a > and at least as energetic as those marked with an >=, it becomes the local maximum (LM).

of the ROI-grid is performed, to find the maximal energy. There are three basic parameters of sliding windows algorithm: the size in TT’s of the sliding window, the separation in TT’s required between local maxima, and the size of the total region surrounding the LM summed to estimate the total transverse energy of the object. The declustering region used is dependent upon the first two parameters.
For example, in Figure 2, a 5x5 decluster region would be necessary for a 2x2 sliding window region and a 1 TT separation between local maxima. In the final step, the transverse energy from the local maxima and a surrounding region are summed, cuts are made on the isolation, EM/HAD fraction and energy density in order to choose EM, jet and tau objects. These cuts are different for each algorithm.

### 2.2 Finding EM, Jet and Tau objects

EM, jet and tau objects are differentiated by the signatures of their energy deposits in the calorimeter. Jet objects tend to be broad and deposit energy in both the electromagnetic and hadronic portions of the calorimeter. The sliding windows algorithm optimized for jets in the DØ detector uses a 2x2 sliding window, has 1 TT LM-separation and has a 1 TT ring of energy used for the transverse energy sum (the total region is then 0.8x0.8 in the eta-phi plane) [2]. This algorithm is denoted (2,1,1).

Because EM objects are narrower and do not tend to deposit energy in the hadronic calorimeter, shaping cuts can be applied on the EM/HAD fraction and isolation of the candidate EM object. Section 4 of this note documents the parameters of the sliding windows algorithm chosen for optimal performance.

Tau objects look similar to jets, but have a very energetic core. Therefore, an additional cut is made on the fraction of energy of the entire tau candidate object (sliding window and surrounding region) contained in the sliding window. The parameters for this algorithm have not yet been optimised.

### 2.3 Hardware Implementation

The sliding windows algorithm is implemented in the RunIIb L1 Cal Trigger hardware via a set of programmable Altera Stratix FPGA’s located on the Trigger Algorithm Board (TAB). The TAB receives trigger tower energies from the Analog Digital Filter (ADF) boards, which digitize the calorimeter signals from the BLS system. It outputs the number of EM, Jet and tau objects that pass threshold requirements to the Global Algorithm Board (GAB) and passes object properties to the Cal-Track match system and to the Trigger Framework. All arithmetic in the hardware is performed bit serially on 12-bit data streams.

The studies of the sliding windows EM algorithm, which this note documents, found that the optimal algorithm was somewhat different from the nominal one chosen. The hardware implementation of this final algorithm is documented online [3]. The major impacts of the final EM algorithm choice are:

- there will be one choice of the cut value for EM isolation and EM/HAD fraction applied to each object. In principle, one could have multiple thresholds for EM isolation and EM/HAD fraction, but due to limited hardware resources, having only one cut allows for the full capacity of 7 transverse energy thresholds to remain. Six out of the seven energy thresholds will be passed to the GAB for AND/OR term construction. The seventh threshold can be used for the L1 cal-track matching terms.

- an ’isolation bit’ is defined for each EM object as: (object passes EM isolation cut) & (object
passes EM/HAD cut). This bit can then be used in conjunction with an $E_T$ threshold to trigger an event.

- the EM isolation and EM/HAD fraction decisions for all EM objects are ORed over three separate eta regions (north, central, south) for each of the four phi regions that are included in a TAB. Therefore, only decisions based on the 12 (3x4) regions of eta-phi are passed from the TAB to the GAB, and EM isolation and EM/HAD fraction of individual EM objects is lost.

3 Software Simulation Tools

3.1 Trigger Rate Tool

Trigger Rate Tool is a framework package designed to do a fast simulation of L1, L2 and L3 trigger decisions for the purpose of algorithm design and threshold tuning within trigger rate allowances. It calculates the overlap between different triggers which fire for the same event, and can make rate extrapolations for target luminosities when run over an unbiased data sample.

Simulation of the triggers is performed by reading in the L1L2Chunk or L3Chunk, reconstructing the trigger logic, and saving the trigger decision to an ascii file which is then used to calculate rates. There are two main programs: GenerateTriggerTable controls the simulation of the L1, L2 and L3 trigger terms and writes the trigger decision for each event to the ascii file trigger.dump; Rates reads the trigger.dump file and calculates rates for a given luminosity. The program can be run over data or MC thumbnails.

To study the sliding windows algorithm in the trigger, an additional package was created, l1cal2b_sliding_windows, that simulates the sliding windows algorithm, and constructs sliding windows trigger terms in the level 1 calorimeter. Trigger Rate Tool was modified to be used in conjunction with l1cal2b_sliding_windows, which allows the same rates estimates of sliding windows triggers, integrated into the entire trigger menu, as with Run2a triggers.

For more details of Trigger Rate Tool, see D0 Note 4640 [4]. A D0 note on l1cal2b_sliding_windows is in preparation.

3.2 tsim_l1cal2b

The following applies to version v00-02-14 of tsim_l1cal2b. tsim_l1cal2b is a DØ framework package that simulates the RunIIb Level 1 calorimeter trigger. The package contains two main processes, TsimL1Cal and TriggerCalo.

TsimL1Cal processes real or Monte Carlo (MC) data from a L1CalTTChunk, CalDataChunk, L0CALTowerData, or L1L2Chunk and simulates its treatment by the ADF. The resulting trigger tower information is stored in an L1CalDataChunk.

TriggerCalo simulates the Trigger Algorithm Board (TAB) by performing global, jet, electron, or tau triggers. TriggerCalo will find trigger objects and store their location, $E_T$, and quality cut values (i.e. the $E_T$ of the isolation region of an EM object) in the L1CalTot2bChunk, L1CalJet2bChunk,
L1CalEle2bChunk or L1CalTau2bChunk for global, jet, electron or tau triggers, respectively. The trigger object data is also outputted to a root tree.

For the studies considered in Section 4, the optimal EM algorithm parameters were determined by producing efficiency vs. rate curves for the various algorithm parameters. To produce the values of efficiency and rate, local modifications were made to tsim_11cal2b to store the $E_T$ values of possible electron quality cut geometries. To calculate efficiencies a separate program was used to read in the outputted root tree, apply cuts on the trigger objects, and determine if the surviving objects matched in $n_{\text{Detector}}$ and $\phi_{\text{Detector}}$ with reconstructed electrons in the sample being considered.

Rates were produced by processing the Trigger Rate Tool Enhanced Bias Run 189075 sample through tsim_11cal2b. The root tree produced by tsim_11cal2b was processed through a second program that applied cuts on the trigger objects. A rate was calculated by multiplying the number of passing events by a normalization factor. The normalization factor was derived from the Trigger Rate Tool, which uses a factor of $0.368732 \text{Hz/Event}$ to calculate rates at an instantaneous luminosity of $60 \text{E30 cm}^{-2}\text{s}^{-1}$. Assuming a linear extrapolation, a factor of $1.229107 \text{Hz/Event}$ was used to calculate a rate at an instantaneous luminosity of $2\text{E32 cm}^{-2}\text{s}^{-1}$.

## 4 EM Algorithm Study

### 4.1 Potential EM Algorithms

The L1 Cal trigger upgrade will be able to construct EM trigger objects from a cluster of two neighboring trigger towers by using an ATLAS-like version of the sliding windows algorithm [5]. The ATLAS-like (which will just be referred to as Atlas) algorithm has nearly the same conceptual structure as the 2x2 sliding window algorithm described in Section 2, but instead of reporting object $E_T$ for the whole 2x2 area, it only reports the $E_T$ of the hottest 2x1 cluster located at the lower left corner of the 2x2 search window. The Atlas algorithm has the capability of applying electron quality cuts to the 2x1 objects that are formed in the 2x2 search window.

The purpose of the study presented in this section is to determine if it is beneficial to use such a clustering scheme and, if so, to determine which cuts should be applied to an Atlas object. Possible cuts include variants on an EM/HAD fraction and isolation fraction cut.

For the EM/HAD fraction cut, three geometries are considered: a 3x3 region of hadronic towers, a 2x3 area of hadronic towers, and a 2x1 area of hadronic towers. A diagram of these geometries is shown in Figure 3. Both the 2x1 and 2x3 regions are centered behind the 2x1 EM RoI. These regions can have "horizontal" or "vertical" orientations, making them oblong in either eta or phi. Due to hardware restrictions the 3x3 region is not always centered on the hottest tower in the 2x1 object. Instead the 3x3 area is centered on the lower left corner of the Atlas search window. This is the only tower that is shared by the horizontal and vertical 2x1 objects in the search window.

For the isolation cut, three geometries are considered: a 3x3 region of EM towers that excludes the 2x1 object region, a 2x3 region of EM towers that excludes the 2x1 object region, and a 2x3 region of EM and hadronic towers that excludes the 2x1 EM and hadronic regions. A diagram of
Figure 3: Possible EM/HAD fraction cut geometries. The green blocks are hadronic towers, and the blue blocks form the EM trigger object.

these geometries is shown in Figure 4. The 2x3 regions are centered around the 2x1 object, while the

Figure 4: Possible isolation fraction cut geometries. The green blocks are hadronic towers, the orange blocks are EM towers, and the blue blocks form the EM trigger object.

3x3 region is centered on the lower left corner of the Atlas search window.

For the above geometries, it is possible to cut either by using a threshold for the $E_T$ sum in the region or by using a fraction formed by comparing the object $EMET$ to the $E_T$ of the surrounding area. Since the L1 Cal hardware uses bit serial arithmetic, latency requirements prevent the use of actual division operations. This forces cuts involving fractions to rely on bit shift operations, which restrict the cuts to the form: $2^{\text{integer}} \times E_{\text{Area}} < E_{\text{Object}}$.

In addition to quality cuts, the new EM algorithm can require a minimum separation between trigger objects. Whether this is an advantage will be considered below.

4.2 Real Data and Monte Carlo Samples

For the studies presented in this note, efficiencies are calculated with Monte Carlo and real data samples. The MC samples include 4,500 $W \rightarrow e\nu$ events, 4,000 $t\bar{t} \rightarrow e^{+}+\text{jets}$ events, and 14,500 Drell-Yan events. The samples were produced by PYTHIA with the following card files and were processed through d0gstar, d0sim, and d0reco:

- /D0/dist/packages/cardfiles/v00-06-01/npir/pythia_w_enu.cards
- /D0/dist/packages/cardfiles/p14-br-04/top/pythia_ttbar_wenu+wjj.cards
- /D0/dist/packages/cardfiles/p14-br-04/npir/pythia_gam-z_ee.cards

¹Modified in a local directory so that $10 \text{ GeV} < m(e^+e^-) < 60 \text{ GeV}$
The real data sample contains 21,616 events that the CALGO group has chosen as good $Z \rightarrow ee$ events. The dataset definition for this sample is TMBfix2-recoT_calgo_zee_mrg_1-4.raw_p14.06.01_csg_p14.fixtm2.02. More information about the data sample can be found at [6]. Electrons were chosen from the EM particles which passed one of the following two sets of cuts:

1. Two EM particles with:
   - $\text{EM}_{\text{hm7}} < 12$
   - $\text{fabs(EM}_{\text{id})} < 12$
   - $\text{EM}_{\text{pt}} > 20.0$
   - $\text{EM}_{\text{emfrac}} > 0.9$
   - $\text{EM}_{\text{iso}} < 0.15$

2. Two EM particles with:
   - $\text{fabs(EM}_{\text{id})} < 12$
   - $\text{EM}_{\text{pt}} > 20.0$
   - $\text{EM}_{\text{emfrac}} > 0.9$
   - $\text{EM}_{\text{iso}} < 0.15$
   - At least one EM in CC with $\text{EM}_{\text{prbmatch2}} > 0.000001$

4.3 Efficiencies and Rates

4.3.1 1x1 Trigger Objects or Atlas Trigger Objects

To determine if it is beneficial to produce EM trigger objects from clusters of trigger towers instead of from individual towers, a comparison was made between a characteristic EM trigger that uses a 1x1 sliding windows algorithm and a characteristic EM trigger that uses an Atlas algorithm. In order to choose a model 1x1 algorithm, the efficiency and rate relationships for potential cuts were studied for the $W \rightarrow e\nu$ and $t\bar{t} \rightarrow e+"jets"$ MC samples. The results of the comparisons are found in Figure 5.

For any given rate, a 1x1 RunIIb algorithm which uses an EM/HAD fraction cut and no isolation requirement produces the highest efficiencies. Hence, the algorithm with only the EM/HAD fraction requirement was chosen as a characteristic 1x1 algorithm. A similar comparison was performed for potential Atlas cuts, and the results can be found in Figure 6.

For the lower values of rate, the Atlas algorithm which applies a cut on the sum of a 3x3 EM region and a 3x3 hadronic region has the highest values of efficiency, and hence it was chosen as a characteristic Atlas algorithm. A comparison of the efficiency and rate relations for the two model algorithms is presented in Figure 7. For any given rate the Atlas algorithm has an overall efficiency that is higher than or similar to the 1x1 algorithm; this is due to a steeper turn-on in the efficiency of a cluster algorithm. However, using a cluster of two trigger towers increases the acceptance of
background for a given $E_T$ threshold. Hence, for a given rate, a 1x1 algorithm can be set to a lower $E_T$ threshold than the Atlas algorithm. It could potentially be beneficial to consider an “or” of a low $E_T$ 1x1 algorithm trigger and a high $E_T$ Atlas algorithm trigger.

The RunIIb L1 Cal hardware has the capability of running two EM algorithms in parallel, such as a 1x1 and Atlas algorithm. However, this comes at an increased hardware expense, with consequences such as a reduction in the number of outputted trigger object thresholds [5]. Figure 8 contains the turn-on curves for the Drell-Yan MC sample (treated as a single EM sample) for various combinations of algorithms running in parallel. The thresholds for the algorithms shown in Figure 8 have been set so that each algorithm produces a similar rate. Consequently the $E_T$ threshold of a 1x1 sliding windows algorithm is set lower than the $E_T$ threshold of the Atlas algorithm. This causes the combination of the 1x1 and Atlas algorithms to have the highest efficiency at the lowest values of $E_T$. This effect is most noticeable at a reconstructed $E_T$ of 13 GeV for the CC electrons, where the combination of the 1x1 and Atlas algorithms has 3 times the efficiency of the Atlas only algorithm. However, this advantage was outweighed by two of the main characteristics of the Atlas only algorithm. First, the Atlas only algorithm has the sharpest turn-on curve and overtakes the 1x1 or Atlas combination efficiency when the reconstructed $E_T$ reaches 16 GeV for the CC electrons in Figure 8. Second, the Atlas only algorithm suffers from less hardware constraints and would be capable of outputting the complete seven thresholds, while the 1x1 or Atlas combination would be limited to at most 4 thresholds in total.
Figure 5: Plots of the efficiency and rate relations for various 1x1 algorithms. The points are generated by scanning over values of $E_T$ threshold. The label CEM_ CUBE(n,x,i) refers to a trigger term that requires $n$ trigger objects with $E_T \geq x$ that pass an isolation cut of the form, $E_{T_{\text{iso}}EM} > i \times (E_{T_{\text{iso}}EM} - E_{T_{\text{iso}}EM})$. The label CEM_ EMF(n,x,i) refers to a trigger term that requires $n$ trigger objects with $E_T \geq x$ that pass an EM/HAD fraction cut of the form, $E_{T_{\text{EM}}EM} > i \times E_{T_{\text{HAD}}}$. The label CEM_ CUBE_ EMF(n,x,i,k) refers to a trigger term that requires $n$ trigger objects with $E_T \geq x$ that pass both an isolation cut of the form, $E_{T_{\text{iso}}EM} > i \times (E_{T_{\text{iso}}EM} - E_{T_{\text{iso}}EM})$, and an EM/HAD fraction cut of the form, $E_{T_{\text{EM}}EM} > k \times E_{T_{\text{HAD}}}$. Black represents the trigger term CEM_ CUBE(1,x,1). Pink represents the trigger term CEM_ CUBE(1,x,2). Red represents the trigger term CEM_ EMF(1,x,8). Blue represents the trigger term CEM_ CUBE_ EMF(1,x,1,8).
Figure 6: Plots of the efficiency and rate relations for various Atlas algorithms. The points are generated by scanning over values of $E_T$ threshold. The label CEM_ CUBE(n,x,i) refers to a trigger term that requires $n$ trigger objects with $E_T \geq x$ that pass an isolation cut of the form, $E_{T_{2\text{EM}}} > i \ast (E_{T_{3\text{TOT}}} - E_{T_{2\text{EM}}})$. The label CEM_ EMF(n,x,i) refers to a trigger term that requires $n$ trigger objects with $E_T \geq x$ that pass an EM/HAD fraction cut of the form, $E_{T_{2\text{EM}}} > i \ast E_{T_{2\text{HAD}}}$. The label CEM_ CUBE_ EMF(n,x,i,k) refers to a trigger term that requires $n$ trigger objects with $E_T \geq x$ that pass both an isolation cut of the form, $E_{T_{2\text{EM}}} > i \ast (E_{T_{3\text{TOT}}} - E_{T_{2\text{EM}}})$, and an EM/HAD fraction cut of the form, $E_{T_{2\text{EM}}} > k \ast E_{T_{2\text{HAD}}}$. Black represents the trigger term CEM_ CUBE(1,x,4). Red represents the trigger term CEM_ EMF(1,x,8). Blue represents the trigger term CEM_ CUBE_ EMF(1,x,4,8).
Figure 7: Plots of the efficiency and rate relations for a model Atlas algorithm and a model 1x1 algorithm. To obtain a fully efficient EM trigger at high $E_T$, the triggers contain the “or” of a trigger term that requires an electron quality cut with a term that only cuts on $E_T$. The threshold of the 1x1 $E_T$-only term was set to 11.75 GeV to make its rate equal to the Atlas $E_T$-only term that had a threshold set to 15 GeV. The points are generated by scanning over values of $E_T$ threshold for the terms that use electron quality cuts. The value of the $E_T$ threshold starts at 6 GeV and is incremented by 0.5 GeV. Black represents an algorithm that applies a cut of the form, $E_{T_{1x1EM}} > 4 \times (E_{T_{3x3TOT}} - E_{T_{1x1EM}})$, on the sum of a 3x3 hadronic TT region and a 3x3 EM TT region. Red represents an algorithm that applies an EM/HAD fraction cut of the form, $E_{T_{1x1EM}} > 8 \times E_{T_{1x1HAD}}$. 

12
Figure 8: Plots of the efficiency turn-on curves for combinations of various RunIIb algorithms. The efficiencies are for a single EM trigger object to match an individual electron from the Drell-Yan MC sample. The plot on the top is for contributions from the CC, and the plot on the bottom is for contributions from the EC. The above algorithms consist of the “or” of an $E_T$-only trigger term, CEM(1,15), and a trigger term, CEM_ CUBE(1,x,4), that requires an isolation cut of the form, $E_{T21EM} > 4 \times (E_{T21TOR} - E_{T21EM})$ and $E_{T21OBJECT} \geq x$. The value of the threshold, x, is set for each algorithm to give them the same rate. Black points represent a combination of an Atlas and a 1x1 algorithm with a CEM(1,15) Atlas term “or-ed” with a CEM_ CUBE(1,11,4) 1x1 term. Blue points represent an Atlas only algorithm with a CEM(1,15) Atlas term “or-ed” with a CEM_ CUBE(1,12,75,4) Atlas term. Red points represent an algorithm that only forms 2x1 objects that are “2-in-$\eta$” x “1-in-$\phi$” and does not form objects that are “1-in-$\eta$” x “2-in-$\phi$”. When studying the CC contributions, the red points represent an “or” of a CEM(1,15) “2-in-$\eta$” x “1-in-$\phi$” term with a CEM_ CUBE(1,11.5,4) “2-in-$\eta$” x “1-in-$\phi$” term. When studying the EC contributions, the red points represent an “or” of a CEM(1,15) “2-in-$\eta$” x “1-in-$\phi$” term with a CEM_ CUBE(1,11,4) “2-in-$\eta$” x “1-in-$\phi$” term. Green points represent a combination of an Atlas and a “2-in-$\eta$” x “1-in-$\phi$” algorithm with a CEM(1,15) Atlas term “or-ed” with a CEM_ CUBE(1,12.25,4) “2-in-$\eta$” x “1-in-$\phi$” term.
4.3.2 Minimum Separation between Trigger Objects

The effect of a minimum separation between trigger objects should have the most pronounced effect on the efficiency at which a trigger object correctly matches a non-isolated electron. To determine this, the $t\bar{t} \rightarrow e+\text{"jets"}$ MC sample and the Enhanced Bias data sample were processed through tsim_11cal2b to determine the efficiency and rate relationships for an Atlas algorithm requiring 1 TT of separation between trigger objects and for an Atlas algorithm requiring 0 TT of separation between trigger objects. From Figure 9 it is seen that for any given rate, an Atlas algorithm requiring 0 TT of separation between trigger objects has a greater matching efficiency than an Atlas algorithm requiring 1 TT of separation.

![Figure 9: Plot of efficiency for matching an electron from $t\bar{t}$ MC to an Atlas object vs Rate. Each data points represents a different value of $E_T$ threshold cut. No electron quality cuts are applied.](image)

4.3.3 Cuts Using Thresholds or Fractions

The plots in Figure 10 and Figure 11 contain comparisons between using direct thresholds or fractions for the electron quality cuts. In this study two possible geometries are considered, one that uses the sum of the 3x3 hadronic TT region and the 3x3 EM TT region and one that uses the sum of the 2x3 hadronic TT region and the 2x3 EM TT region.

Figure 10 contains the results for a single EM trigger that requires the “or” of two trigger terms, $\text{CEM}(1,1,15) \parallel \text{CEM}(1,1,12.75,n)$\(^2\). Efficiencies are calculated with tsim_11cal2b using the $W \rightarrow e\nu$

\(^2\text{CEM}(x,y)\) means there are $x$ EM objects with $E_T \geq y$. $\text{CEM}(x,y,n)\) means there are $x$ EM objects with $E_T \geq y$ that also pass a quality cut. For cuts based on a fraction, this means that $E_{T_{\text{EM/Obj}}}/E_{T_{\text{area}}} > 2^n$ for some integer $n$. For cuts based on a threshold this means that $E_{T_{\text{area}}} < n$ where $n$ is any increment of 0.25 GeV.
Figure 10: Single EM Triggers. Black is with a 3x3 isolation fraction cut. Red is with a 2x3 isolation fraction cut. Green is with a 3x3 isolation threshold cut. Blue is with a 2x3 isolation threshold cut.

and $t\bar{t} \rightarrow e+“jets”$ MC samples and the rates are calculated with tsim_11cal2b using the Enhanced Bias run data sample. For the four EM algorithms, points are generated by scanning over potential values of $n$. The horizontal black line through the plot represents a reference rate. The values of $n$ that produce rates closest to this reference are located in Table 1.

Figure 11 contains the results for a di-EM trigger that requires the following trigger terms, $\text{CEM}(1,15) \parallel \text{CEM}(1,12.75,m) \parallel \text{CEM}(2,9) \parallel \text{CEM}(2,3)\text{CEM}(1,15) \parallel \text{CEM}(2,6.25,n)$, where $m$ is the value of $n$ that places the single EM triggers closest to the reference rate in Figure 10. Efficiencies are calculated with tsim_11cal2b using the Drell-Yan MC sample and the rates are calculated with tsim_11cal2b us-
Table 1:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Value of n</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3 w/ fraction</td>
<td>2</td>
</tr>
<tr>
<td>2x3 w/ fraction</td>
<td>2&lt;n&lt;3</td>
</tr>
<tr>
<td>3x3 w/ threshold</td>
<td>3.75 GeV</td>
</tr>
<tr>
<td>2x3 w/ threshold</td>
<td>2.75 GeV</td>
</tr>
</tbody>
</table>

Figure 11: Di-EM Triggers. Black is with a 3x3 isolation fraction cut. Red is with a 2x3 isolation fraction cut. Green is with a 3x3 isolation threshold cut. Blue is with a 2x3 isolation threshold cut.

ing the Enhanced Bias run data sample. The values of n that produce rates closest to the reference rate are located in Table 2.

Table 2:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Value of n</th>
</tr>
</thead>
<tbody>
<tr>
<td>3x3 w/ fraction</td>
<td>2</td>
</tr>
<tr>
<td>2x3 w/ fraction</td>
<td>3</td>
</tr>
<tr>
<td>3x3 w/ threshold</td>
<td>1.5 GeV</td>
</tr>
<tr>
<td>2x3 w/ threshold</td>
<td>1.5 GeV</td>
</tr>
</tbody>
</table>

The efficiency and rate curves for a 2x3 or 3x3 geometry using either a threshold cut or a fraction cut are the same within the errors of the data points. As a hardware consideration, the threshold cut is more complicated, since it requires two thresholds for a single EM trigger and a lower $E_T$ di-EM trigger. Among the 2x3 or 3x3 geometries, it is more preferential to use the 2x3 geometry since by
4.3.4 Choice of Geometry for Cuts

In the plots shown in Figures 13-20, efficiencies for potential isolation and EM/HAD fraction cuts are calculated for 6 GeV and 10 GeV trigger objects that match reconstructed electrons. The properties of the 6 GeV and 10 GeV trigger objects are approximated by using the high $E_T$ $Z \rightarrow ee$ data sample. $E_T$ distributions for the different isolation and EM/HAD fraction TT geometries are found for the objects that match electrons. The assumption is made that the nonzero value of the $E_T$ distributions are caused by noise and are independent of the electron $E_T$. Hence, the $E_T$ distributions found with the $Z \rightarrow ee$ data sample’s high $E_T$ objects should be similar to the distributions for 6 GeV and 10 GeV objects. The efficiencies for a 10 GeV (6 GeV) trigger object are determined by dividing the $E_T$ of the distributions by 10 GeV (6 GeV), applying the isolation and/or EM/HAD fraction cut(s) on that ratio and counting the number of objects that survive. Rates are determined by processing the Enhanced Bias data sample with tsim_11cal2b and by using a second program to apply the particular quality cut being studied along with either a 10 GeV or 6 GeV $E_T$ threshold cut. In Figures 13-20, it is seen that there is little difference in the efficiency and rate relations for the various combinations of geometries for the isolation and EM/HAD fraction cuts. It is preferential to choose a geometry which separates hadronic and EM trigger towers, since it is possible for there to be unforeseen hardware issues between the two independent readouts. A smaller geometry is also preferential since it should have a higher efficiency for non-isolated electrons.

4.4 Final RunIIb EM Algorithm

Based on the above results the EM algorithm for the RunIIb Calorimeter upgrade will have the following parameters:

- Trigger objects will be created from 2 TTs via an Atlas algorithm
- There will be 0 TTs of separation between trigger objects
- The isolation fraction cut will use a 2x3 region of EM TTs by applying a cut of the form, $2^{\text{integer}} \times [(EM \ E_T_{2x3\text{Area}}) - (EM \ E_T_{\text{AtlasObject}})] < (EM \ E_{T_{\text{AtlasObject}}})$. See Figure 12.
- The EM/HAD fraction cut will use a 2x1 region of hadronic TTs by applying a cut of the form, $2^{\text{integer}} \times (HAD \ E_{T_{\text{AtlasObject}}}) < (EM \ E_{T_{\text{AtlasObject}}})$. See Figure 12.
Figure 12: The chosen isolation fraction geometry is on the left. The chosen EM/HAD fraction geometry is on the right. The green blocks are hadronic towers, the orange blocks are EM towers, and the blue blocks form the EM trigger object.

References

   Online: http://www.nevis.columbia.edu/evans/l1cal/docs/tdr/D0_Run2b_TDR.pdf

   Online: http://www.nevis.columbia.edu/evans/l1cal/meetings/020710/mitrevski.pdf

   Online: http://www.nevis.columbia.edu/evans/l1cal/algos/em_algo/em_final.html


   Online: http://www.nevis.columbia.edu/evans/l1cal/algos/algos.html

   Online: http://www-d0.fnal.gov/Run2Physics/cs/RAW/raw.html
Figure 13: Scanning EM/HAD fraction cut of the form $E_{T_{\text{HadRegion}}} \ast 2^n < E_{T_{\text{EMObject}}}$ for $2^n=0,1,2,4,8,16,32$
Figure 14: Scanning EM/HAD fraction cut of the form \( E_{T_{\text{HadRegion}}} \times 2^n < E_{T_{\text{EMObject}}} \) for \( 2^n = 0, 1, 2, 4, 8, 16, 32 \)
Figure 15: Scanning isolation fraction cut of the form $E_{T_{\text{HadRegion}}} \times 2^n < E_{T_{\text{EMObject}}} \times 2^n$ for $n=0,1,2,4,8,16,32$
Figure 16: Scanning isolation fraction cut of the form $E_{T_{\text{HadRegion}}} \cdot 2^n < E_{T_{\text{EMObject}}}$ for $2^n = 0, 1, 2, 4, 8, 16, 32$
Figure 17: Scanning isolation fraction cut and EM/HAD fraction cut for $2^{n_{iso}} = 0, 1, 2, 4, 8, 16, 32$ and $2^{n_{emf}} = 0, 1, 2, 4, 8, 16, 32$. 
Figure 18: Scanning isolation fraction cut and EM/HAD fraction cut for $2^{n_{iso}}=0,1,2,4,8,16,32$ and $2^{n_{em\ f}}=0,1,2,4,8,16,32$
Figure 19: Scanning isolation fraction cut and EM/HAD fraction cut for $2^{n_{iso}} = 0,1,2,4,8,16,32$ and $2^{n_{emf}} = 0,1,2,4,8,16,32$
Figure 20: Scanning isolation fraction cut and EM/HAD fraction cut for $2^{n_{isw}}=0,1,2,4,8,16,32$ and $2^{n_{emf}}=0,1,2,4,8,16,32$