Version 2.0
September 23th, 1996

A Clustering Engine for the SVT Hit Finder Board

1 Introduction

The Hit Finder board of the SVT receives the SVX-II data after a Level 1 Accept. The board suppresses bad channels, does a pedestal subtraction, and forms groups of adjacent hit strips into clusters, calculates the charge-weighted centroid of those clusters, and passes the result to the pattern recognition stage of the SVT. In this note we investigate the clustering algorithm for the Hit Finder board.

The purpose of the SVT is to find large-impact parameter tracks with good efficiency, high signal-to-noise, and in time for the Level 2 decision. The most significant way the clustering algorithm can influence these goals is through what it does with the 10% of clusters that contain more than 3 strips.

2 Evaluation of the Algorithms

The first data sample we use is 15,000 dijet triggers. The second is also dijet data but every event has a SECVTX tag. The events have the raw data banks so we can look at individual strips, including the channel-by-channel pedestal means and widths.

We use the standard SVX code to do pedestal subtraction and make the SVXE element bank. We perform custom clustering algorithms and make a new QSVC cluster bank. The SVX tracking is then rerun and the results are evaluated.

We will use three figures of merit to evaluate the algorithms. The first is a measure of the efficiency for b-like objects. We find all pairs of tracks with both legs having $P_t > 2$GeV and $d > 90\mu$m. We require the two legs separated by less than $90°$ and call the bisector of the angle formed by the tracks the "B" direction. We fit the pair with CTVMFT, require a good χ^2, calculate L_{xy} as the intervertex separation projected on the B direction, and require $|L_{xy}| > 1000\mu$m. This pseudo-tag is similar to a $\pi\pi$ trigger. It is tighter than SECVTX in P_t, d, and N_{hit} cuts but a little looser in other ways such as cluster sharing cuts.

We run on the sample of SECVTX tags and take the number of positive L_{xy} vertices minus the number of negative vertices to be the signal for evaluating efficiency. Typical signal-to-noise for these pseudo-tags in the dijet data is 1:1 and the positive excess is 2% of events. In the tagged sample the signal-to-noise is 1.2:1 and the positive excess in pseudo-tags is 67% of the positive excess of SECVTX tags.

The other two figures of merit are measures of how much background the trigger has to deal with. One of the potential problems with a clustering algorithm is making too many clusters so each outer track can be matched with many combinations of clusters. This can slow down the SVT pattern recognition and track fitting. To make a measure of how many combinations the pattern recognition will find, we count how many clusters on each layer are within $\pm$1mm of the track's trajectory. The product of these four numbers is taken to be the number of pattern recognition combinations that will be output by the AM system and fit by the track fitting system. This number is evaluated on the untagged dijet data.

Finally we take the number of positive L_{xy} vertices to be a measure of the trigger rate which we would like to minimize. This number is also evaluated on the untagged dijet data.

3 Clustering Algorithms

The algorithm should be simple because it will have to go into a PLD. On the other hand, it should have flexibility to deal with the unexpected once the data starts coming in.

The Hit Finder will have a channel-by-channel pedestal subtraction followed by a universal ADC threshold applied to the data before it enters the Hit Finder engine. The engine itself will probably only be able to look at 4 strips simultaneously to determine adjacencies. However it could look at more provided the algorithm allows the engine to evaluate a long cluster on the fly. The engine will be able to place the ADC's from 3 strips on to the address of a memory which outputs a cluster centroid and a bit indicating if this is a legitimate cluster pattern. Three ADC's of seven bits each would be 2Mbytes of memory for each engine which would be very good to avoid if possible.

One possibility is to cut on the total cluster charge. Figure 1 shows the charge distribution for all clusters and for clusters used on high-P_t, 4-hit tracks. We could remove 4% of clusters with a cut at 90 counts with no loss in efficiency.

Another possibility is to require that a 3 or 4-strip cluster show a reasonable pattern of charge deposition. For example, we could remove 3-strip clusters with a pattern of high-low-high. Figure 1 shows the distribution of an index determined by the charge distribution pattern for 3 and 4-strip clusters for all clusters and those used on good tracks. There doesn't seem to be much to gain without significant inefficiency. This algorithm may be improved by asking how much the charges go up and down, rather than just if they went up and down. However, many longer clusters are caused by delta rays which tend to leave more charge where they range out, creating extra peaks.

Table 1 and figure 2 display the results of several algorithms:

0. "Offline." To establish a baseline, we run these tests using the current offline clustering algorithm with 1, 2, and 3-strip cluster resolutions set to 15μm but long-cluster resolutions set to the number of strips$\times$pitch/$\sqrt{12}$, as it is offline. A three-or-more-strips cluster is found if the adjacent strips are 2σ over pedestal; a 2-strip cluster is found if the adjacent strips are 2.5σ over pedestal; and a 1-strip cluster if the strip is 4σ over pedestal.

1. "Offline Q." Same as "Offline" but with a strip required to be 3σ over pedestal to be included in a cluster. This is intended to reduce the number of strips the engine has

to process. It also may be the level of the zero suppression in the readout stream in run II.

2. "Offline 15μm" We make a 3σ cut on all strips. All cluster resolutions set to 15μm.

3. "All 3." We make a 3σ cut on all strips. We keep all clusters with three or fewer strips. Longer clusters are split by making the first three strips into a cluster, then the next three, *etc.* All cluster resolutions set to 15μm.

4. "Biggest." We make a 3σ cut on all strips. We keep all clusters with three or fewer strips. Longer clusters are split by taking the highest charge strip and the two adjacent strips to make a cluster, the rest of the strips are ignored. If the highest charge strip is the first or last strip, the cluster is made of two strips. All cluster resolutions set to 15μm.

5. "Peaks." We make a 3σ cut on all strips. We look at each remaining strip. If either of the neighboring strips are present and have a larger charge than this strip, no cluster is made. If this strip is larger than its neighbors, or its neighbors are not present, this becomes a candidate cluster. Next, compute the total charge of the central strip and its neighbors. If the total charge of the three strips is greater than 1/3MIP (50 ADC counts), a cluster is made. The cluster centroid is the charge-weighted mean of the strips, including the neighbors if they are present. All cluster resolutions set to 15μm.

6. "Mixed." We make a 3σ cut on all strips. We make clusters of adjacent strips up to 6 strips long. If the cluster is 3 or fewer strips, the centroid is the charge-weighted centroid and the resolution is set to 15μm. If the cluster is 4 or more strips, the centroid is placed at the middle of the cluster and the resolution is set the offline value for a 5-strip cluster.

7. "Mixed 5-bit." This algorithm is the same as "Mixed" but now we only use a limited amount of information from the ADC's. We scale the ADC by 24/140 to move the one-MIP peak to 24 counts, truncate it at 127 counts, and round it to the nearest integer. This simulates the SVX-III chip ADC. When we calculate the cluster centroid we use only the 5 most significant bits - this corresponds to putting only 15 bits on the centroid look-up instead of 21 bits for a significant hardware savings. If all ADC's have no significant bits on, we place the centroid in the middle of the cluster.

4 Conclusions

There doesn't seem to be much to gain by cutting on the total cluster charge or on the cluster pattern. This probably is saying that unused clusters are mostly real, caused by low-P_t tracks. However, a cut on the total charge of a cluster will probably be implemented since it is simple enough.

The first conclusion from the table is that allowing long clusters but forcing their resolution to be the nominal 15μm value causes a 14% loss in signal. Splitting the long cluster

Number	Algorithm	"B"s	roads	triggers
0	Offline	597	120	653
1	Offline Q	578	126	573
2	Offline 15μm	515	112	487
3	All 3	553	180	504
4	Biggest	512	108	424
5	Peaks	555	143	450
6	Mixed	584	126	597
7	Mixed 5-bit	588	126	598

Table 1: The figures of merit for several clustering algorithms described in the text. Roads are in thousands.

into smaller ones and giving the smaller ones a 15μm resolution reduces the loss to 8%. The losses are reflected in the total number of tracks as well as the number of tags. This supports the assumption that the loss is from tracks failing the χ^2 cut because a cluster with poor resolution was included in the fit with a small σ.

The mixed algorithm shows that the loss can be recovered by allowing long clusters with appropriate resolution, even if the centroid and the resolution is approximated. The very large resolution used for long clusters makes the cluster unimportant in the track fit. However, long clusters cannot simply be ignored since there are too many of them and the pattern recognition must require hits on all 4 layers to maintain rejection.

"All 3" brings in 30% more hit combinations which will delay the SVT output. Using only 5 bits of ADC information to calculate the cluster centroid seems to be no different from using the full ADC information.

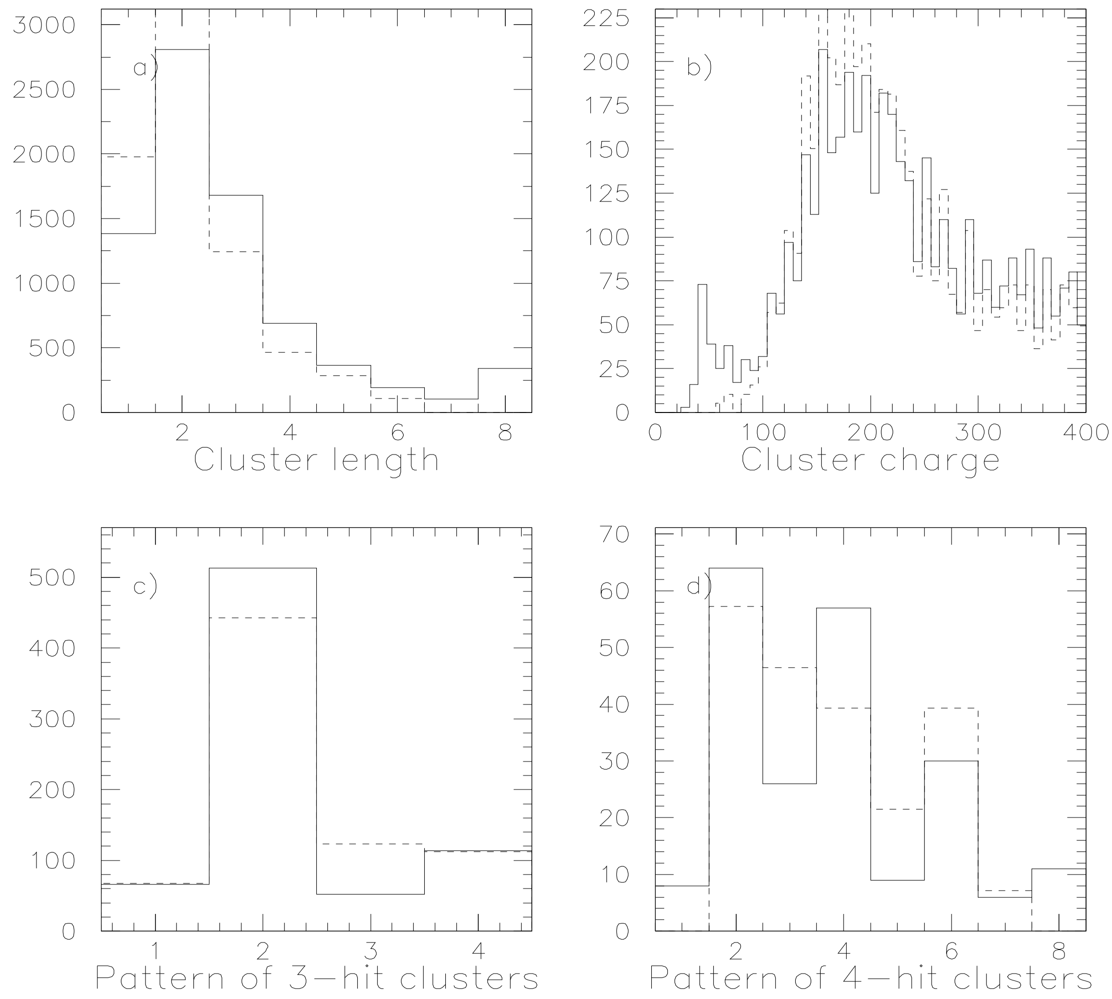

Figure 1: a) The length of clusters used on good tracks. b) The charge of all clusters. c) The pattern of charge deposition for 3-strip clusters. d) The pattern of charge deposition for 4-strip clusters. Solid histograms are for all clusters and dashed is for those used on tracks. The patterns are determined by the charges on the cluster. For example the 3-cluster pattern low-high-low is pattern number 2.

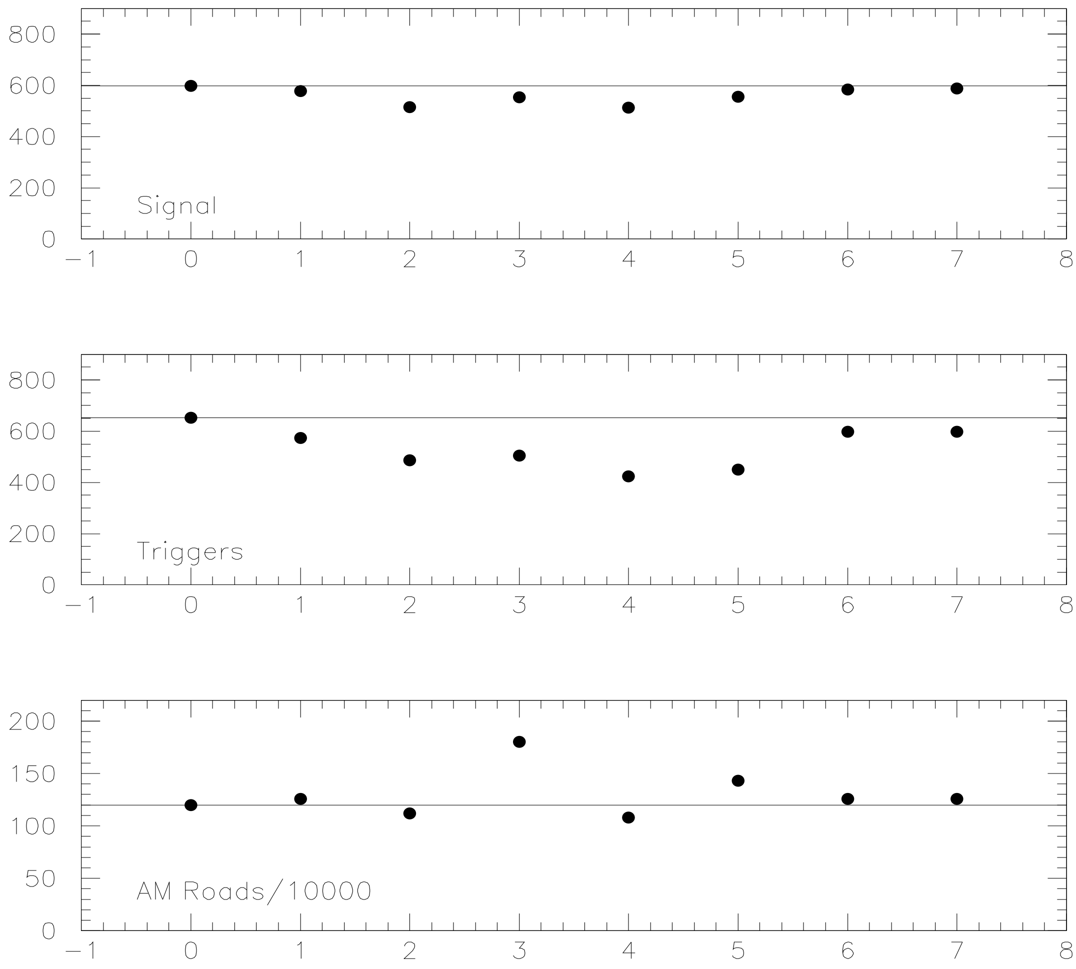

Figure 2: The results of the different clustering algorithms. Signal is the number of pseudo-tags in a SECVTX tagged sample, Triggers is the number of pseudo-tags in the jet sample, and AM Roads is the number of hit combinations the SVT will fit in the jet sample.