CDF/DOC/TRIGGER/CDFR/xxxx
Version 0
August 23, 2007

L2 XFT Stereo Upgrade Overview

D. Cox, R. Erbacher, R. Forrest, A. Ivanov, A. Lister, A. Soha
University of California, Davis

S. Holm, T. Liu, P. Wilson
Fermilab

Abstract

This document describes the XFT stereo upgrade for the L2 trigger system, starting from the Stereo Finder boards, up to the XFT stereo track algorithm implementation in the L2 PC. This note also describes the simulation tools available for developing triggers using stereo tracking information and discusses the implementation of the versions of the high p_T lepton triggers that use stereo tracking at L2.

Contents

1 System Overview

The XFT Stereo upgrade provides many benefits over the purely axial triggering system used previously. One of the aims of this project is to reduce the rate of fake tracks in many triggers. The fake rates increase very rapidly with luminosity, much faster than the real track rates. By removing as many of the fake tracks as possible at trigger level, it is possible to keep these triggers, without pre-scaling them, up to much higher instantaneous luminosities. The L1 path is used to confirm that the existing track (reconstructed with the axial COT layers only) goes through the stereo layers at the expected locations. At L2 the segmentation is much finer than at L1 allowing a better fake rejection rate and also providing information about the position of the track. In particular it's possible to measure the angle θ of the track with respect to the beam axis as well as the distance z from the center of the detector along the beam axis and use that information to point the track to other detectors like the muon chambers, to see if the track is matched with a muon stub. This 3D tracking opens up additional capabilities for trigger level multi-track mass calculations or isolation requirements and z-vertex reconstruction at L2.

Because of the slight angular inclination with respect to the beam-pipe of the stereo layers, the $\cot(\theta)$ of the track can be calculated from the distances between where the track is expected to cross the stereo layers and where it actually does cross. This is illustrated in figure 1. SL3 and SL7 have the same stereo angle whereas SL5 has the opposite angle. The $\cot(\theta)$ of the track is measured using the difference in the displacements between the expected track position and the measured one as follows

$$\cot(\theta) = \frac{\Delta(SL7) - \Delta(SL5)}{r_7 - r_5} = \frac{\Delta(SL5) - \Delta(SL3)}{r_5 - r_3} = \frac{\Delta(SL7) - \Delta(0)}{r_7} \qquad (1)$$

where r_7, r_5 and r_3 are the radii of the 3 stereo superlayers, $\Delta(SL7)$, $\Delta(SL5)$ and $\Delta(SL3)$ are the deviations between the expected and measured track positions for the 3 stereo suplerlayers and $\Delta(0)$ is the location with respect to the centre of the detector where the track would cross the beam-axis.

To illustrate the fake rate rejection by using the stereo XFT system to confirm the tracks, the measured and expected trigger rates for the CMX trigger are shown in figure 2 as a function of instantaneous luminosity. At luminosities around $150 \cdot 10^{30}\,\mathrm{cm}^2\mathrm{s}^{-1}$, an additional factor of 2 in the trigger rate can be gained by using L2 stereo confirmation and pointing as opposed to the L1 stereo confirmation. At the predicted highest luminosity running, around $300 \cdot 10^{30}\,\mathrm{cm}^2\mathrm{s}^{-1}$, the additional gain is expected to be a factor of 4.

The system used to obtain the XFT stereo information at L2 is described here and shown schematically in figure 3. The Stereo Finder board is used both for the L1 and L2 paths. There are 36 finder boards, 12 for each superlayer, split into two different crates (b006 and b0xft07). Each board contains 2

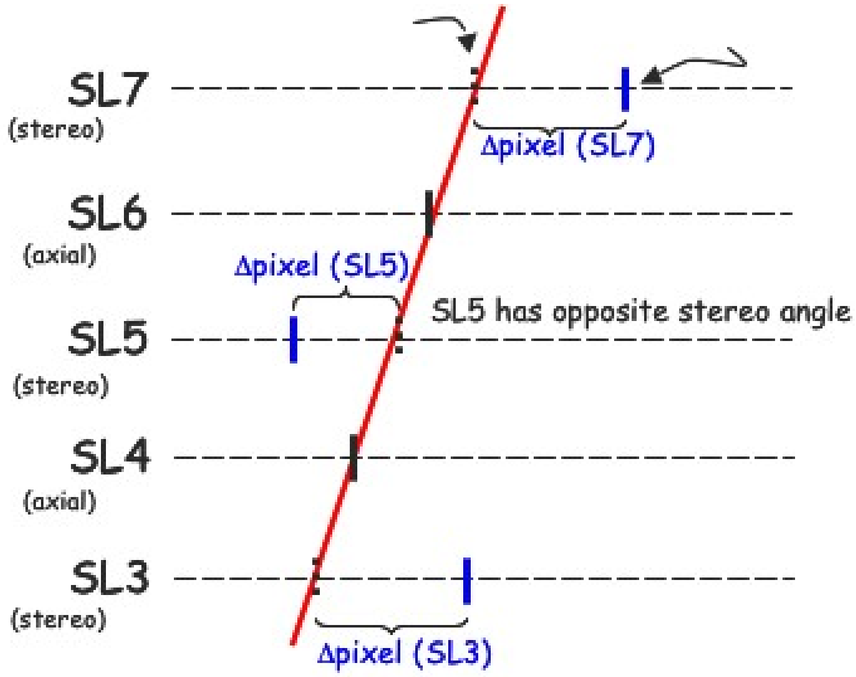

Figure 1: Schematic illustration of the axial and stereo COT layers.

chips (chip A and chip B) which each look at only some of the input fibres. The input fibres come out of the COT XTC cards. The COT hits are combined into track segments and the ϕ position and slope are measured. The slope bits are set according to the L2 90 bit format described in section 2. These 90 bits per cell are concatenated into the 12 L1 bits which are then sent to the SLAM cards. The higher resolution data is sparsified as described in section 3 then sent both to the T2FD bank, see section 4, and to one of the three PULSAR boards via the transition boards on the backplanes of the finder crates. The data format for the finder to PULSAR connection is described in section 5. The PULSAR boards merge and concatenate the stereo pixel information. The order in which the pixel information from each finder is sent out of the PULSAR is defined by a mapping described in section 6. The PULSAR sends the data over optical fibres into the L2 PC, see section 8. The L2 PC runs the stereo track reconstruction algorithm and L2 trigger algorithms, as described in section 10. Because of timing constraints, the stereo reconstruction algorithm is only run on those tracks that already pass all other L2 trigger requirements and are computed at most once for each track in an event. Section 9 describes how this task is facilitated by the use of the TL2D bank; the stereo track information for the processed tracks is appended to the track information in that bank. The stereo track reconstruction algorithm will only be run on a selected number of triggers and can be used in a number of different ways; e.g. using the track $\cot(\theta)$ to verify that the track points to a muon stub. A tool, presented in section 11 is used to simulate the L2 stereo tracking algorithm. It enables the implementation of new triggers or the improvement of existing ones.

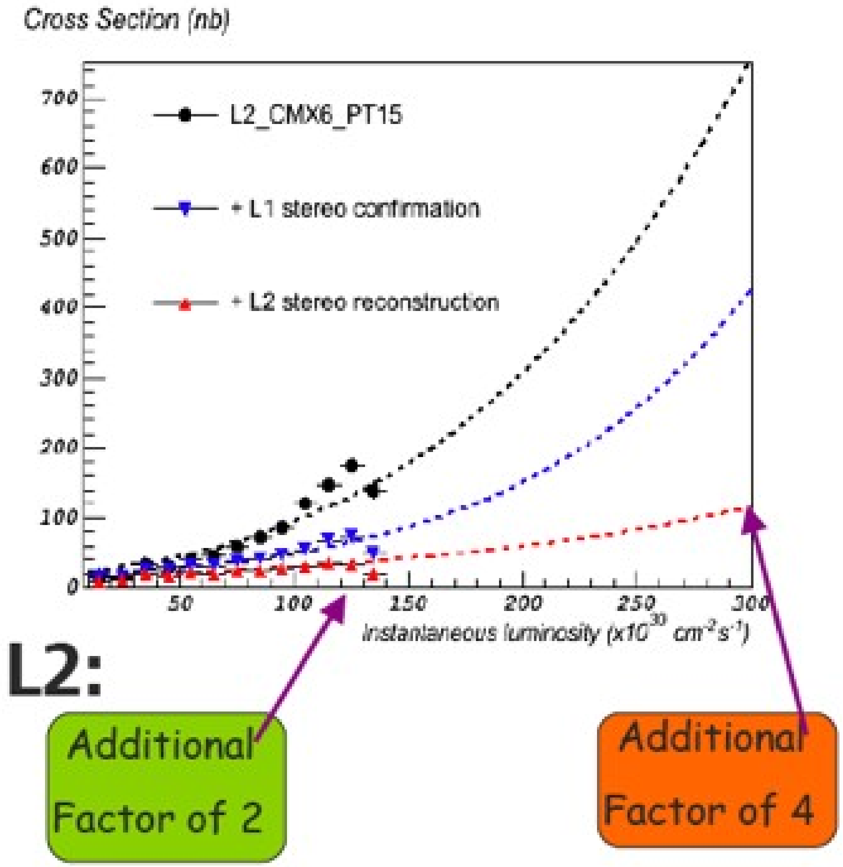

Figure 2: Measured and predicted CMX trigger rates using no stereo confirmation (black), L1 stereo track confirmation (blue) and L2 track confirmation (red).

2 L2 Pixel Design

Note: CDF Note 7789 contains a more detailed definition of the L1 and L2 pixel mappings [1]
To provide flexibility and simplicity, the design of the L2 pixels is matched to that of the L1 pixels [1]. A cell contains 6 pixels at L1, each with 2 slope bits: a positive and a negative slope bit. When both bits are set, the track is either a high p_T track (i.e. essentially straight, $p_T > xxx\,\mathrm{Gev/c}$) or the hits are consistent with either a positive or a negative curvature. The basic unit of a cell at L1 is thus represented by 6x2=12 bits. At L2, a cell contains 6 sub-cells, each of which contain 3 pixels, each with 5 slope bits: positive curvature low p_T ($p_T < xxx\,\mathrm{GeV/c}$), positive curvature higher p_T ($xxx < p_T < yyy\,\mathrm{GeV/c}$), high p_T ($p_T > yyy\,\mathrm{GeV/c}$), negative curvature higher p_T and negative curvature low p_T. The basic unit of a cell at L2 is thus represented by 6x3x5 = 90 bits. The basic bit structure for L1 cells and L2 sub-cells are depicted in tables 1 and 2, respectively. It should be noted that any given pixel can have multiple tracks passing through it with different slope values, resulting in multiple slope bits being set.

[1]Note: In our naming convention a L1 pixel corresponds to the same ϕ size as a L2 sub-cell

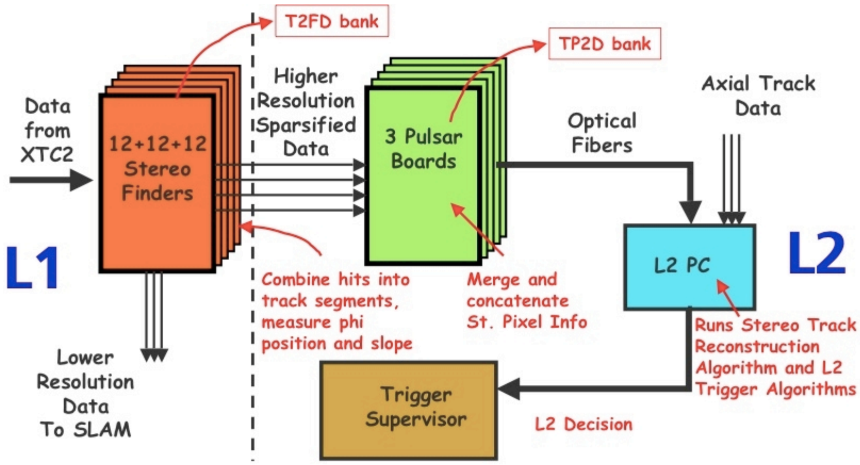

Figure 3: Schematic diagram of the data flow for the L2 XFT Upgrade project.

Bit #	11	10	9	8	7	6	5	4	3	2	1	0
Pixel #	5		4		3		2		1		0	
Slope	+	-	+	-	+	-	+	-	+	-	+	-

Table 1: L1 pixel format for a single cell. The 2 slope bits correspond to positively and negatively charged tracks or positive and negative track curvatures. Both bits are set for high p_T tracks or if the hits can be fitted with either a positive or a negative curvature track.

3 Data Sparsification

The well-known idea behind sparsification is to send data only for portions of the detector that contain at least one hit. Using data sparsification results in a smaller data packet size and faster data transfers. The sparsification scheme adopted for the L2 path provides the data in two distinct sections. The first 10 (18,18) words for SL3 (SL5, SL7) are used as the mask section. Each sub-cell is represented by a single bit. If the bit is set (=1) then at least 1 of the 15 slope bits (5 bits for each pixel) is set for this sub-cell. Then follows the data section which contains the 15 bits only for each sub-cell for which the mask bit was set. The information for sub-cells with no hits (for which all 15 bits are 0) are not sent. Numerous studies were carried out to investigate the the data transfer time from the Finder Boards to the L2 Pulsar boards [2]. This time was found to not exceed 4 μs when the data is sparsified.

4 T2FD Bank Format

Note: This section, included here for completeness, is a copy of the section included in CDF note 4152 [3].
The T2FD bank is designed for diagnostic readout, and to facilitate development of algorithms at level 2. The pixel information from the stereo finders for superlayers 3, 5, and 7 is made available in this bank through the format shown in table 3.

Bit #	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pixel #			2					1					0		
Track p_T	+	++	hpt	- -	-	+	++	hpt	- -	-	+	++	hpt	- -	-
Curvature	++	+	0	-	- -	++	+	0	-	- -	++	+	0	-	- -

Table 2: L2 pixel format for a single sub-cell. The slope bit assigments are shown as a function of track p_T and of track curvature. The 3rd slope bit is set for high p_T tracks (hpt).

The format reflects the fact that the outermost three stereo superlayers, here called SL3, SL5, and SL7, each have 12 finder boards, but those boards are split among two crates (18 boards each). The current hardware plan is to have six boards of each superlayer type in each of the two crates. The T2FD bank definition has a flexible format for the finder pixel data, to include the possibility of sparsification. Note that the number of cells per finder board depends on the superlayer (SL3, SL5, and SL7 have 20, 28, and 36 cells/finder, respectively), so the block length can be different for each of the two blocks, depending on how the finders are distributed among the two crates, regardless of whether or not sparsification is used. For the stereo information available to level 2, each cell consists of 18 ϕ bins, and hits in any given ϕ position can have any combination of up to five different slopes. For the purpose of sparsification, each cell is split into 6 *sub-cells* that each contain 3 ϕ positions (again with 1 to 5 available slopes).

Table 4 shows the details of the pixel data format. The information from each of the 18 successive finders within a given crate appears sequentially in 18 repeated sections, each with the format shown in table 4. The first word contains header information, including superlayer # (2 bits), buffer # for finder chip A (2 bits), buffer # for finder chip B (2 bits), the number of 32 bit words in the buffer [excluding the header word] (7 bits), bunch counter # for finder chip A (7 bits), and bunch counter # for finder chip B (7 bits). The header word is identified by the fact that bit 15 is set to 1 and bit 14 is set to 0. Also, bits 13, 30, and 31 are set to 0.

The next 5, 9, or 9 words, for SL3, SL5, or SL7, are used for the sparsification mask. For these 32-bit words, bits 0-5 serve as the mask for the 6 sub-cells of cell#0, bits 6-11 serve as the mask for the 6 sub-cells of cell#10 or cell#18 (depending on the SL), bits 12-14 are unused, and bit 15 has the value of 0, bits 16-21 serve as the mask for the 6 sub-cells of cell#1, bits 22-27 serve as the mask for the 6 sub cells of cell#11 or cell#13 (depending on the SL), bits 28-30 are unused, and bit 31 has the value 0. The actual pattern of cells in the mask section is somewhat complex due to the hardware implemenation, but here is an overview. The stereo finders for each of the superlayers have two chips, called chip A and chip B. Two different versions of firmware have been written for the chips, such that they can either handle 10 cells or 18 cells (here, we ignore the added complexity of the fact that the chips also can readout neighboring cell information). For SL3 there are 20 cells (generically numbered 0 through 19 for our purposes here). Chip A reads out cells 0-9 and chip B reads out cells 10-19. The firmware that provides the cell information for L2 (for the T2FD bank and for the L2 PULSAR system), puts the SL3 cell masks into the following order: 0, 10, 1, 11, 2, 12, 3, 13, 4, 14, 5, 15, 6, 16, 7, 17, 8, 18, 9, 19. For SL5 there are 28 cells (generically numbered 0 through 27). For simplicity of firmware, chip A handles cells 0-17 and chip B handles cells 18-27. Therefore the cell mask ordering for SL5 is the following: 0, 18, 1, 19, 2, 20, 3, 21, 4, 22, 5, 23, 6, 24, 7, 25, 8, 26, 9, 27, 10, X, 11, X, 12, X, 13, X, 14, X, 15, X, 16, X, 17, X, where each X means that there are six unused bits in the mask due to the fact that chip B is “empty” while chip A is still being read

“T2FD”
Bank number
Bank version
Bank length
Bank type (I*4)
Number of blocks (=2)
Pointer to block 0
Pointer to block 1
Pointer to end of data
Block 0: Number of cards (=18)
Pointer to card 0
...
Pointer to card 17
Pointer to end of block
Finder pixel data
...
Finder pixel data
Block 1: Number of cards (=18)
Pointer to card 0
...
Pointer to card 17
Pointer to end of block
Finder pixel data
...
Finder pixel data

Table 3: Structure of the T2FD bank.

out. For SL7 there are 36 cells (generically numbered 0 through 35). Here, chip A handles cells 0-17 and chip B handles cells 18-35. The cell mask ordering for SL7 is the following: 0, 18, 1, 19, 2, 20, 3, 21, 4, 22, 5, 23, 6, 24, 7, 25, 8, 26, 9, 27, 10, 28, 11, 29, 12, 30, 13, 31, 14, 32, 15, 33, 16, 34, 17, 35. Each mask word specifies the masks for 4 sub-cells, except for the last 4 mask words for SL5, which each have the masks for 2 sub-cells alternating with two "empty" slots.

The correspondence between the above generic cell numbers within a particular finder and the global cell location in the COT is also somewhat complex. For SL3, chip A on the finder that reads out the $\phi = -90$ region of the drift chamber reads global cells 230-239 and chip B reads global cells 0-9. For SL5, chip A on the $\phi = -90$ finder reads global cells 322-335 and global cells 0-3, while chip B reads global cells 4-13. For SL7, chip A on the $\phi = -90$ finder reads global cells 414-431, while chip B reads global cells 0-17. For the finders that read out each of the other 30 degree segments of the COT, it can more simply be stated that chip A reads out the global cell positions with smaller cell numbers while chip B reads out the global cell positions with larger cell numbers.

Within the mask, a bit set to 1 corresponds to a sub-cell that contains at least one hit, while a bit set to 0 corresponds to a sub-cell with no hits. If sparsification is not desired, all of the cell masks can be set to 1's. The next set of words provide the actual hit data for any sub-cells that were indicated by a bit set to 1 in the mask. The sub-cells appear in a *different* order than as given in the mask. They appear in numerical cell order (0-19, 0-27, or 0-35, depending on SL). Each word contains the hit information for the 3 ϕ pixels of 2 sub-cells (the last word will sometimes contain just 1 sub-cell, and the remaining bits will belong to the trailer). Within the data section, bit 15 has the value 0, and bit 31 has the value 0, except sometimes for the last word where bit 31 might be part of the trailer section. For each pixel, there are 5 bits corresponding to the 5 slope bins. If there is an odd number of sub-cells being read out, then the last word is a combination of data and trailer, in which case bits 16-29 are used for the trailer information, bit 30 is set to 1, and bit 31 is set to 1. If there is an even number of sub-cells being read out, then one additional word appears with bits 0-13 used for the trailer information, bit 14 set to 1, bit 15 set to 1, and bits 16-31 unused. The trailer contains three bits for the turn #, four bits for the finder #, four bits for the fiber #, and the bit-pattern 101 appearing in bits 11-13 or 27-29.

If sparsification were not used (and the cell masks would then all be set to 1's), then there would be 67, 95, and 119 words required for each finder of SL3, SL5, and SL7, respectively.

5 Data Format for the Transfer Between Finder and Pulsar Boards

A 16-bit communication protocol is used for the data transfer from the L2 output chips of the Finder boards to the Pulsar mezannine cards. The data from each stereo Finder board is packed into a packet, one per Finder per event. The size of the data packet will in general depend on how busy a particular event is, which is reflected by the COT hit occupancy, and the number of the stereo segments found. The data packet starts with the header, which has a size of two 16-bit words, and ends with the trailer word, one word-size (see Table 5). The bit 15 in each of the 16-bit words sent from the Finder to the Pulsar is reserved to define the beginning and the end of the data envelope, which is recognized by the PULSAR firmware. It is set to 1 in the first and the last word of the data envelope, otherwise it is set to 0. The bit 14

Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Header	0	0	Bunch Counter														1	0	0	# 32 bit words in buffer (excluding header)							Buffer #				SL #	
			Finder B								Finder A																Fin B		Fin A			
Mask	0	Unused			Cell #11 (or 19)						Cell #1						0	Unused			Cell #10 (or 18)						Cell #0					
Mask	0	Unused			Cell #13 (or 21)						Cell #3						0	Unused			Cell #12 (or 20)						Cell #2					
Mask	0	...															0	...														
Mask	0	Total of 5 words for SL3															0															
Mask	0	Total of 9 words for SL5															0															
Mask	0	Total of 9 words for SL7															0															
Data	0	Sub-cell 1, Pix 3					Sub-cell 1, Pix 2					Sub-cell 1, Pix 1					0	Sub-cell 0, Pix 3					Sub-cell 0, Pix 2					Sub-cell 0, Pix 1				
Data	0	...															0	...														
Data	0	Up to 60 data words for SL3															0															
Data	0	Up to 84 data words for SL5															0															
Data	0	Up to 108 data words for SL7															0															
Data	1	1	1	0	1	Fiber #				Finder #				Turn #			0	Sub-cell N, Pix 3					Sub-cell N, Pix 2					Sub-cell N, Pix 1				

Table 4: Format of header, mask, pixel data, and trailer for T2FD bank, corresponding to the data from one finder. The mask section can be used to sparsify based on sub-cells that are 1/6$^{\text{th}}$ of a cell (details are given in the text). For the data section, the notation "Sub-cell 1, Pix 1" means that the five corresponding bits indicate which combination of the 5 slope bins for ϕ pixel $\#1$ of sub-cell $\#1$ have hits. Note that when bit 15 or bit 31 has the value 1, then bit 14 or bit 30 is also a control bit, and indicates the beginning or end of the envelope (these control bits sometimes appear in the middle of the word since the structure of this data format was driven by the 16-bit word format used for a stereo finder to PULSAR transfer). If there is an odd number of sub-cells with data, then the last word is a combination of data and trailer word as shown above. In this case bits 16-29 are used for the trailer, bit 30 is set to 1, and bit 31 is set to 1. If there is an even number of sub-cells with data, one additional trailer word appears with bits 0-13 used for the trailer, bit 14 set to 1, bit 15 set to 1, and bits 16-31 unused.

<table>
<tr><th>Bit #</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2">Header</td><td rowspan="2">1</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="7" rowspan="2">unused</td><td colspan="4">Buffer #</td><td colspan="2" rowspan="2">SL #</td></tr>
<tr><td colspan="2">Fin B</td><td colspan="2">Fin A</td></tr>
<tr><td rowspan="2">Header</td><td rowspan="2">0</td><td rowspan="2">0</td><td colspan="14">Bunch Counter</td></tr>
<tr><td colspan="7">Finder B</td><td colspan="7">Finder A</td></tr>
<tr><td>Data</td><td>0</td><td colspan="15">Data</td></tr>
<tr><td>...</td><td>0</td><td colspan="15">...</td></tr>
<tr><td>Data</td><td>0</td><td colspan="15">Data</td></tr>
<tr><td>Trailer</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td colspan="4">Fiber #</td><td colspan="4">Finder #</td><td colspan="3">Turn #</td></tr>
</table>

Table 5: Data envelope structure between Finder and Pulsar boards.

in the first and the last word is used to distinguish the beginning (set to 0) from the end (set to 1). The first header word starts with 2 bits for the superlayer number (coded as: bit 1=0, bit 0=0 for SL 3; bit 1=0, bit 0=1 for SL 5; and bit 1=1, bit 0=1 for SL 7). Bits 2-3 (4-5) give the buffer number that was read out for chip A (B) on the Finder. The second header word contains the bunch counter information, ranging from 0 to 127, for Finder chip A in bits 0-6 and Finder chip B in bits 7-13. In the trailer word, bits 0-2 give the turn number, which specifies b0 markers, bits 3-6 give the finder ID number, bits 7-10 give the fiber ID number, and bits 11-15 are set to: bit 15=1; bit 14=1; bit 13=1; bit 12=0; and bit 11=1.

At L2 18×5 stereo pixel binning is employed (18 ϕ-pixel bins by 5 slope bins), so that each cell requires 90 bits of information. To speed up the data transfer the information from each cell is sparsified into 6 parts (sub-cells), and only those sub-cells with at least one segment on are passed to L2 (see Section 3). The data field has the following structure (see Tables 6, 7, 8).

The first several words contain the list of sub-cells masks, where one bit is reserved per sub-cell, i.e. 6 bits specify one cell. If a particular sub-cell has at least one hit, then the bit is set to 1, otherwise it remains 0. Each word specifies information for two cells.

To speed up the sparsification and mask creation in the Stereo Finder firmware, the order of cells in the mask list is the following. The first word contains masks for cell $\#0$ followed by cell $\#$ 10 (18,18) for superlayers $SL = 3(5, 7)$. The second word contains masks for cells $\#1$ and$\#$ 11 (19,19) and so on. The bits 12-14 in each word remain unassigned. For the $SL5$ Stereo Finder bits 6-14 are unassigned starting from 11-th word in the mask section. The size of the mask list is fixed and is 10 words for super-layer 3 and 18 words for super-layers 5 and 7.

The second part of the data field contains the sub-cell stereo pixel information for those sub-cells, which are not masked (corresponding to bit=1 in the mask section). The sub-cells are appended in the numerical order corresponding to cells: 0,1,2, ... (different from the one in the mask section!) Masked sub-cells (bit=0) are discarded. One word is assigned to each sub-cell.

Bit #	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
10 words for SL 3	n.a. bits			Cell # 10						Cell # 0					
	n.a. bits			Cell # 11						Cell # 1					
	... Sub-cell bit masks bit=1 if has a hit bit=0 if no hit ...														
	n.a. bits			Cell # 19						Cell # 9					
Sub-cell #0	Pixel # 2					Pixel # 1					Pixel # 0				
Sub-cell #1	Pixel # 2					Pixel # 1					Pixel # 0				
	...														
Last Sub-cell	Pixel # 2					Pixel # 1					Pixel # 0				

Table 6: Data field for the superlayer $SL = 3$ for Finder to Pulsar data transfer.

Bit #	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
18 words for SL 5	n.a. bits			Cell # 18						Cell # 0					
	n.a. bits			Cell # 19						Cell # 1					
	... Sub-cell bit masks bit=1 if has a hit bit=0 if no hit ...														
	n.a. bits			Cell # 27						Cell # 9					
	n.a. bits									Cell # 10					
	...														
	n.a. bits									Cell # 17					
Sub-cell #0	Pixel # 2					Pixel # 1					Pixel # 0				
Sub-cell #1	Pixel # 2					Pixel # 1					Pixel # 0				
	...														
Last Sub-cell	Pixel # 2					Pixel # 1					Pixel # 0				

Table 7: Data field for the superlayer $SL = 5$ for Finder to Pulsar data transfer.

<table>
<tr><th>Bit #</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="4">18
words
for SL 7</td><td colspan="3">n.a. bits</td><td colspan="6">Cell # 18</td><td colspan="6">Cell # 0</td></tr>
<tr><td colspan="3">n.a. bits</td><td colspan="6">Cell # 19</td><td colspan="6">Cell # 1</td></tr>
<tr><td colspan="15">...
Sub-cell bit masks
bit=1 if has a hit
bit=0 if no hit
...</td></tr>
<tr><td colspan="3">n.a. bits</td><td colspan="6">Cell # 35</td><td colspan="6">Cell # 17</td></tr>
<tr><td>Sub-cell #0</td><td colspan="5">Pixel # 2</td><td colspan="5">Pixel # 1</td><td colspan="5">Pixel # 0</td></tr>
<tr><td>Sub-cell #1</td><td colspan="5">Pixel # 2</td><td colspan="5">Pixel # 1</td><td colspan="5">Pixel # 0</td></tr>
<tr><td></td><td colspan="15">...</td></tr>
<tr><td>Last Sub-cell</td><td colspan="5">Pixel # 2</td><td colspan="5">Pixel # 1</td><td colspan="5">Pixel # 0</td></tr>
</table>

Table 8: Data field for the superlayer $SL = 7$ for Finder to Pulsar data transfer.

6 Data Re-ordering in Pulsar

The data leaving each of the 36 Stereo Finders is sent to one of the 3 Pulsar boards. The Finders are arranged in 2 XFT crates (b0xft06 and b0xft07), with 18 boards in each crate, in such a way that the first 6 slots of both crates correspond to SL3, the next 6 to SL5 and the last 6 to SL7. In order to get a roughly equal occupancy in each of the 3 PULSAR boards, a mapping was used. Each of the 3 PULSAR boards gets the information from 4 SL3 Finders, 4 SL5 Finders and 4 SL7 Finders, such that each Pulsar recieves all the data for a complete slice of the layers, see figure 4.
The mapping is shown in table 9.

Figure 4: Schematic view of the 3 stereo SL showing the way the hardware reads out each slice.

7 TP2D Bank Format

Note: This section, included here for completeness, is similar to the section included in CDF note 4152 [3].

8 Data Transfer from the Pulsar to the L2 PC

Each of the three XFT Pulsar boards has a single output, each containing the information from 12 finder boards. After leaving the PULSAR, the data is transferred to the L2 PC via optical links. There are thus three optical inputs to the L2 PC. The PC has 2 PCI buses. One of the PCI buses has 2 FILAR cards connected to it. The other has 1 FILAR and 1 SOLAR card (used for the output to the L2TS board, that

Pulsar board 0			Pulsar board 1			Pulsar board 2	
Finder Crate	Finder Board #		Finder Crate	Finder Board #		Finder Crate	Finder Board #
0	12		0	16		1	14
0	6		0	10		1	8
0	0		0	4		1	2
0	13		0	17		1	15
0	7		0	11		1	9
0	1		0	5		1	3
0	14		1	12		1	16
0	8		1	6		1	10
0	2		1	0		1	4
0	15		1	13		1	17
0	9		1	7		1	11
0	3		1	1		1	5

Table 9: Mapping between Finder crate and card number and concatenation ordering in Pulsar board.

contains the information that is written into the TL2D bank). Because of the large sizes of the XFT inputs we split them over the 2 PCI buses. Two inputs are on 1 FILAR and the third is on the FILAR that shares the PCI bus with the SOLAR card.
The specifications for the PC, FILAR and SOLAR boards can be found elsewhere.
....

9 TL2D Bank Format

The TL2D bank contains a wide range of information used in the level 2 trigger decision. For the purpose of the level 2 XFT stereo upgrade, it is used to store the stereo track parameters generated by the stereo tracking algorithm, running either offline or in the level 2 PC.

The stereo track parameters, specified below, have been added to the XTRP card of the TL2D bank. We have recycled the previously defined, but unused, so-called "Match" word. The structure of the XTRP card of the TL2D bank is given in table 10. More details about the 32-bit words for XTRP track data and the End of Event word can be found in CDF note 4152 (section 31.16.4 at the time of this writing).

Table 11 shows the assignment of the 32-bits used for each stereo parameter word. Within each word, bit 0 is set to 1 if the stereo algorithm was attempted and is set to 0 otherwise (in which case the rest of the word should be ignored). Bit 1 is set to 1 if a stereo track was found at level 2 and is set to 0 otherwise. The remainder of the bits should be ignored if no stereo track was found at level 2. Bits 2-10 store the integer-encoded (explained below) magnitude of the z position at SL5. Bit 11 carries the sign of z at SL5, where 1 corresponds to positive and 0 corresponds to negative. Bits 12-20 store the integer-encoded magnitude of $\cot(\theta)$. Bit 21 carries the sign of $\cot(\theta)$, where 1 corresponds to positive and 0 corresponds to negative. The remainder of the bits, 22-31, are unused.

Here is a brief explanation of the stereo track parameters and how they are encoded from float-

TL2D Bank, XTRP Card
Number of XTRP tracks (N_{trk})
Track 1 XTRP track data
Track 1 stereo parameters
Track 2 XTRP track data
Track 2 stereo parameters
...
Track N_{trk} XTRP track data
Track N_{trk} stereo parameters
Track End of Event word

Table 10: Structure of XTRP card of the TL2D bank.

Bit	Definition
31:22	Unused
21	Sign of $\cot(\theta)$
20:12	Integer-encoded absolute value of $\cot(\theta)$
11	Sign of z at SL5
10:2	Integer-encoded absolute value of z at SL5
1	Stereo track found at level 2
0	Stereo algorithm attempted

Table 11: Bit assignment for each stereo parameter word of the XTRP card of the TL2D bank.

ing point numbers to integers and back. The direct results of the stereo reconstruction algorithm are two floating point numbers: the z position at $r = 0$ (along the beamline) and $\cot(\theta)$. For storage in the TL2D bank, we first calculate the z coordinate at SL5 using the expression $z_{\mathrm{SL5}} = z_0 + 94.0\cot(\theta)$, where 94.0 cm is the radius of SL5. The z resolution at SL5 is estimated to be 1 cm, so the minimum granularity for encoding as an integer is taken to be 0.5 cm. The maximum value of z at SL5 would be 122 cm, so nine bits are used to store the magnitude as an integer. The magnitude of the z position is encoded into an integer using, schematically, int(abs(z_{SL5})$/0.5 + 0.5$). The $\cot(\theta)$ resolution is estimated to be 0.1. The maximum value of $\cot(\theta)$ would be about 1.85, given the COT geometry. An encoding granularity of 0.005 is chosen (0.05 would be sufficient, but we have extra bits and prefer to match the number of bits used for z position encoding). The magnitude of $\cot(\theta)$ is encoded into an integer using logic of the form int(abs($\cot(\theta)$)$/0.005 + 0.5$). To extract these parameters from the TL2D bank and, in the case of the z position at SL5 and $\cot(\theta)$, decode the integers back to floating point values, accessors have been written for the TL2D_StorableBank class of the TriggerObjects package. An accessor is also available to recalculate the z position at $r = 0$ from the z position at SL5.

10 Stereo Tracking Algorithm Implementation

The purpose of the stereo tracking algorithm is to quickly desparsify the L2 pixel data in the T2FD bank format and, based on these pixels, find the best stereo quantities per axial track. The algorithm is designed to quickly return the stereo quantities $\cot(\theta)$ and Z_5 for any given axial track, so these quantities can then be used in the trigger decision. Because the algorithm runs on the L2 PC and could potentially be run several times per event, it is crucial it be fast enough to execute without introducing additional dead-time into the system.

The algorithm can be broken down into three major parts, each of which will be described in greater detail below. For each axial track, the first task is to desparsify the pixel data contained in the TP2D banks of the PULSAR. We only unpack the pixel data for every relivent stereo superlayer (3,5,7) in a given window around the extrapolated axial track. The second part is then to mask and cluster these pixels. This masking eliminates fake pixels by removing those pixels whose slope does not match expected similar slopes from the curvature of the axial track. We then cluster consecutive pixels, decreasing possible track multiplicity in the next step. The third and final part of the algorithm is to loop over combinations of clusters in the three layers, apply fiducial cuts, and choose the best quality stereo tracks.

10.1 Initialization

The initialization of the algorithm corresponds to a per-event overhead, so the code has been minimized in this section. Besides clearing memory, the only function that takes a significant amount of time is finding each of the finder cards' length in the T2FD bank for later use. This corresponds to a $\sim 1\mu S$ per event overhead. Potentially, this could be decreased in the future by finding card length in hardware and then appending the bank format to include the length of the cards.

10.2 Unpacking

We unpack from the bank only those stereo pixels in a window around the given axial extrapolated phi-slice. The difference between axial extrapolated phi and the stereo hit phi is directly proportional to Z position of the track at that layer, so fiduciality requirements dictate the size of the window and vary at each of the three layers. Specifically, the track must lie within $|Z_0| \leq 90$ cm and must be contained in the tracker $|Z_8| \leq 155$ cm. To save computation time, lookup tables were developed to map axial XFT track quantities (ϕ_6 and p_t bins) into appropriate stereo layer pixel locations.
Because of the format of the T2FD bank, the whole sub-cell mask section must be read before pixel data is extracted. This mask section parsing costs some time, so the result is stored for use in future tracks. Lookup tables were also developed to quickly determine the absolute stereo pixel value of a corresponding mask section bit. Using this machinery, we can determine the exact bank location of our needed pixels. It should also be noted that a window may possibly span two finder cards and therefore two sections in T2FD; this will increase the time needed, as the next card's mask section will need to be parsed. Also, as a time saver, the clustering and masking is done as the pixels are being unpacked.

10.3 Clustering and Masking

Clustering and masking both serve as quality cuts as well as techniques to reduce time spent looping over possible track configurations (part 3 of the algorithm).
Masking makes use of the five slope bins per pixel provided by the finder. We cut fake pixels by assuming real pixels have similar slopes to their associated axial track. Based on studies, we divide the extrapolated stereo pixel window up into three sections in ϕ whose selected hits would correspond to position in z, (again $z_7 \propto \Delta SL7$ where $\Delta SL7$ is the axial track extrapolation position at SL7 minus stereo hit position at SL7). We allow stereo pixel slopes to be similar or slightly different than the axial slope based on the following table:

	High ϕ Third	Mid ϕThird	Low ϕ Third
SL7	(0,1)	(-1, 0, 1)	(-1, 0)
SL5	(-1, 0)	(-1, 0, 1)	(0, 1)
SL3	(0, 1, 2)	(-1, 0, 1)	(-2, -1, 0)

$\Delta\phi$ vs. Δslope for different superlayers.

So for example, stereo pixels corresponding to the central part of all superlayer windows will be masked with the axial slope $\pm$ one slope bin. Similarly, at SL7, pixels on the high ϕ part of the window will be masked with the axial slope bin and one slope bin less than the axial. This masking serves to reject fakes and leave clean pixels that are most likely to be real.

We cluster stereo pixels in order to reduce the number of potential combinations of tracks from which to choose. This is motivated by studies that show real tracks often leave several concurrent pixels hit in a row at this Level 2 resolution. See the table below for a single track Monte Carlo study of real pixel clusters.

	1	2	3
SL3	27.6%	69.2%	3.2%
SL5	18.7%	72.6%	8.7%
SL7	20.2%	71.5%	8.3%

Number of adjacent pixels in ϕ of simulated single track stereo hits.

To simplify and speed up clustering, we cluster by T2FD sub-cell. One group of three pixels in a sub-cell will form a cluster whose location will be determined by the average phi location of the valid, hit pixels in that sub-cell. Therefore, for every mask bit set in the T2FD mask header, there will be one cluster formed assuming the pixels contain hits in the requisite slope bins. The value of this cluster corresponds to the difference between the axial extrapolated ϕ and the actual stereo pixel hits in a particular superlayer. We call this value, in superlayer seven for example, $\Delta SL7$.

10.4 Looping Over Combinations

At this point, we have candidate pixel clusters for each of the superlayers 3, 5, 7. Any combination of these clusters, one from each layer, corresponds to a stereo track with a distinct Z_5 and $\cot(\theta)$ value. We do a χ^2 likelihood minimization to derive a linear equation for these quantities:

$$Z_0 = -4.606\,\Delta SL7 + 1.032\,\Delta SL5 + 6.444\,\Delta SL3 \quad (2)$$

$$\cot(\theta) = 0.0589\,\Delta SL7 + 0.0008\,\Delta SL5 - 0.0581\,\Delta SL3 \quad (3)$$

This allows us to quickly calculate these quantities. We also define the quantity $\Delta_{3,5,7} = \Delta SL3 + \Delta SL7 - 2 * \Delta SL5$ which should ideally be zero for a track that does not curve in the $r - z$ plane. This is a straight line quality requirement.

With several clusters at each superlayer, the combinatorics of this loop over clusters can be very substantial and therefore take substantial time. As the loop progresses, combinations from layer seven and three that lie far outside a reasonable Z value (95 cm), are skipped. From the complete loop over clusters we keep all tracks with $\Delta_{3,5,7} \leq 25$ and -90 cm $< z_0 < 90$ cm. The $\cot(\theta)$ fiducial cuts are enforced by appropriate choice of stereo pixel window size at each layer, limiting Z and therefore angle in θ.These are the candidate tracks to be used in the trigger logic.

11 Algorithm Performance and Stereo Track Triggers

The resolution and efficiency of the reconstruction algorithm will dictate the ultimate performance of the upgraded triggers. With this in mind, we have performance results for the baseline algorithm and maintain the flexibility to use either the best quality track combination, or all of the highest quality tracks found for each axial track. We also will leave open the possibility of improving the reconstruction algorithm at a later date.

Three main inclusive lepton triggers will initially make use of the L2 XFT Stereo Upgrade. The inclusive CMX trigger stands to benefit most from stereo information due to the ability to point a track in either the East or West rings of the CMX detector. The CMUP trigger will also benefit from pointing to either the East or West side of the CMU. Finally for electrons, the CEM trigger now has the ability to point to specific trigger towers in rapidity to reduce the L2 rate.

11.1 Reconstruction Algorithm Performance

The performance of the trigger will be determined by the resolution of the stereo quantities, the efficiency of finding tracks, and the time it takes to do so. Resolution studies were performed by matching axial XFT tracks to offline tracks for tight CMX muons. SLAM confirmed tracks were required to match within six Level 2 pixels at each of the three used stereo superlayers. We then compare the best Z_0 and $\cot(\theta)$ to the matched offline values. The results are shown in figures 5 and 6. We can then use these numbers to estimate the pointing resolution at the outer lepton detectors.

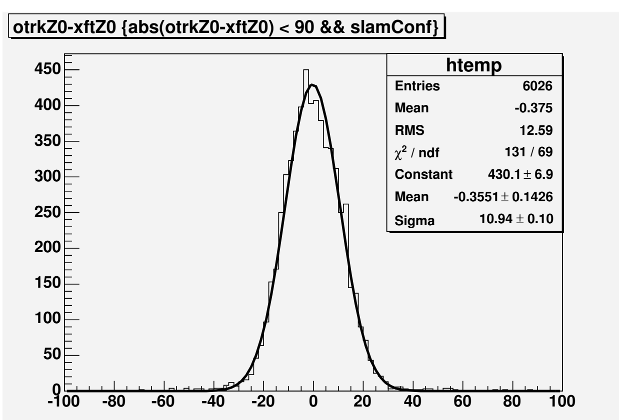

Figure 5: Z_0 resolution for the best track, $\sigma = 11\ cm$.

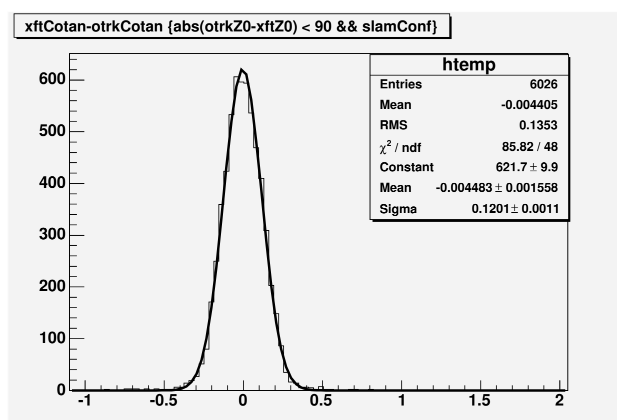

Figure 6: $\cot(\theta)$ resolution for the best track, $\sigma = .12$.

The efficiency of reconstructing tracks can suffer in the pixel masking, quality and fiducial cuts. We define reconstruction efficiency for this same sample of good muons as the number of SLAM confirmed tracks that have been successfully reconstructed over the number of SLAM confirmed tracks that were attempted to be reconstructed. This number is representative of the reconstruction itself and is independent of any specific trigger cut inefficiencies. The results versus luminosity are in figure 7.

The final figure of interest is the timing of the reconstruction. We put a lot of effort into making sure reconstruction is as fast as possible so as not to introduce latency in the L2 trigger system. We time the reconstruction with on-line tests in the backup L2 PC. Based on these tests we can tell how much time it will take, per track, to execute the reconstruction in the reconstructTrack() method. This time is a per track reference for trigger construction and does not include several other overheads, such as getting the data into the L2 PC. We see that the reconstruction code alone should take from 1.3 - 2.0 μS per track depending on luminosity, see figure 8. Triggers constructed with this reconstruction code should note that this does not include the encapsulating trigger code and take this into account accordingly.

11.2 'Best Track' versus 'Valid Track'

Because of drops in efficiency versus Luminosity, the original method of finding a single 'Best Track' choice of stereo quantities for every axial track was found to be insufficient. We now return from the function every possible combination of stereo pixels that fulfill the fiducial requirement and has $\Delta_{3,5,7} \leq 25$. This is shown to flatten out the efficiency curve at the expense of rate rejection. However, we will maintain the ability to use both methods in the future.

11.3 CMX

The existing CMX trigger at L2 uses all axial $r - \phi$ projected tracks to match to muon stubs in the detector. This means that non-CMX fiducial tracks may fake legitimate muons based on a coincidental ϕ correspondence with a muon stub. The ability to point a track to a specific region in rapidity allows a third dimension of fiducial fake rejection. The upgraded CMX trigger $\mathbf{CMX6\_PT15\_3DMATCH}$ used the 'Best Track' approach to extrapolate candidate stereo tracks to the radius of both the inner and outer edge of the CMX detector. The trigger then simply checks that the projected Z position of the candidate tracks at the inner CMX radius is closer to the origin than an inner radius cut. It then checks that it is farther from the origin than an outer radius cut, see figure 9. These cuts are chosen based on offline muon samples to be efficient at 99.6% for sucessfully reconstructed CMX axial tracks. The resolution of the tracks is not good enough to distinguish the slightly varied geometry of the keystone and miniskirt, so no special adjustments are made for these areas.

The rate reduction projections are calculated with the HTTF rate tools method and agree nicely with results from data, see figure 11.

Timing plots to be put in later

The efficiency of L2 CMX initially was found to decrease with luminosity with the 'best track' approach. The new trigger is found to be flat relative to Luminosity at the expense of some rate reduction. Efficiency can be measured relative to good offline muons as well as relative to the existing L3 muon trigger. When compared to offline, we look at tight muons and the associated offline track. These tracks are then matched to a certain XFT axial track. The offline efficiency is then the number of those reconstructed tracks that pass the trigger, over the total number of well matched, SLAM confirmed tracks. Note this does not take into account the reconstruction inefficiency, see figure 7. The second efficiency measurement approach is to simply measure the number of events that pass our trigger relative to the number that end up passing the existing **L3_MUON_CMX18** trigger path. The results of the former approach are shown in figure 12 , while the results of the latter approach are shown in figure 13 We are currently in the process of understanding the differences between these approaches.

11.4 CMUP

The **CMUP6_PT15_3DMATCH** trigger has essentially the same functionality as the CMX trigger. Each axial track is still matched in ϕ as in the original trigger, but with the addition of stereo pointing, a specific side of the CMU detector can be used eliminating fakes from coincidental muon stub hits on the opposite side of the detector. Stereo tracking can return if the track hit the East CMU, West CMU, non-fiducial or 'both' if the track is indistinguishable between both sides. As in the CMX trigger method, we place cuts designed to be 99.6% efficient offline on the extrapolated Z_{CMU}. The results are shown below in figure 16.

11.5 CEM

For the CEM trigger we again extrapolate the candidate track to r_{cem} and then look at ± 1 trigger tower. Investigation of performance is still underway.

References

[1] J. Dittman et al., *Run 2b XFT Stereo Mask Finding Specifications'*, CDF Note 7789

[2] R. Erbacher et al., *Specification of the Data Format from Level 1 to Level 2 of Stereo XFT'*, CDF Note 7772

[3] W. Badgett et al., *The CDF Run II Event Structure*, CDF Note 4152

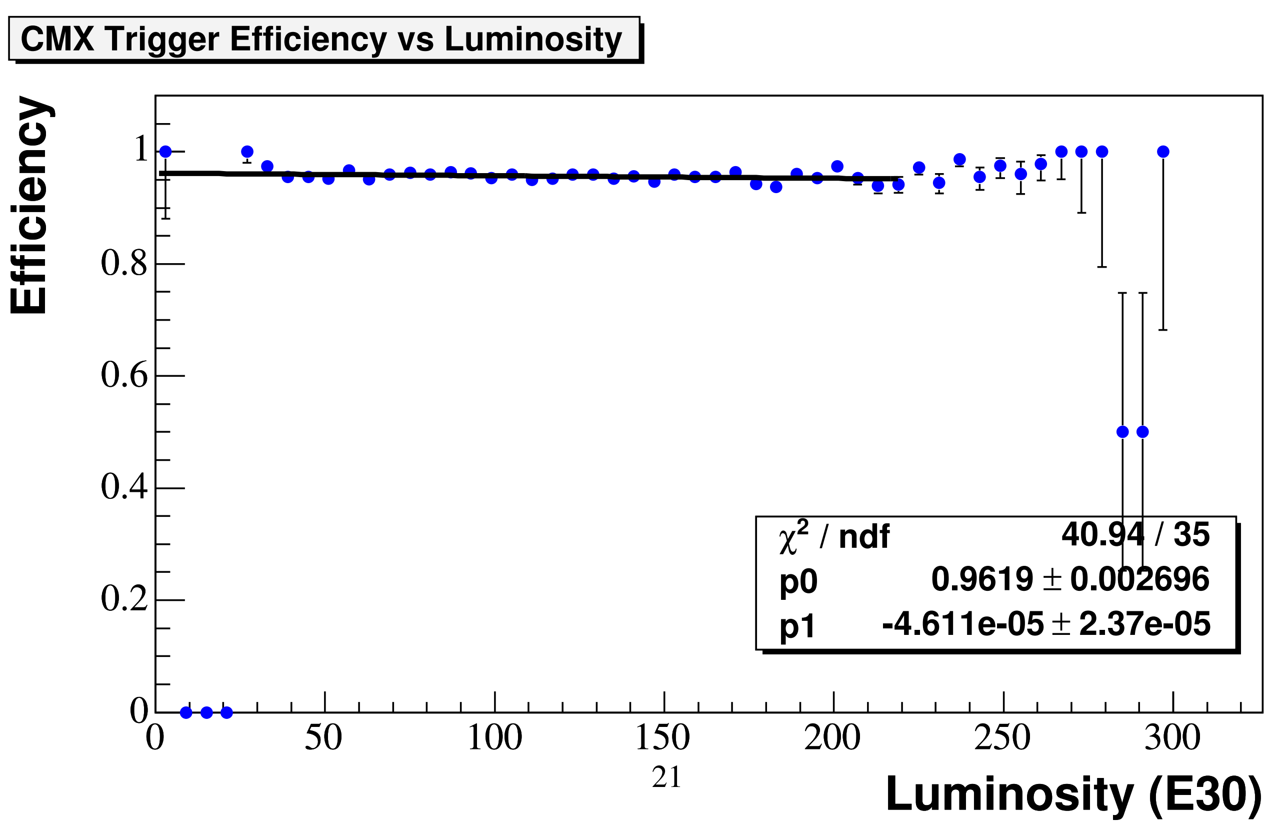

Figure 7: Reconstruction efficiency for SLAM confirmed Axial tracks.

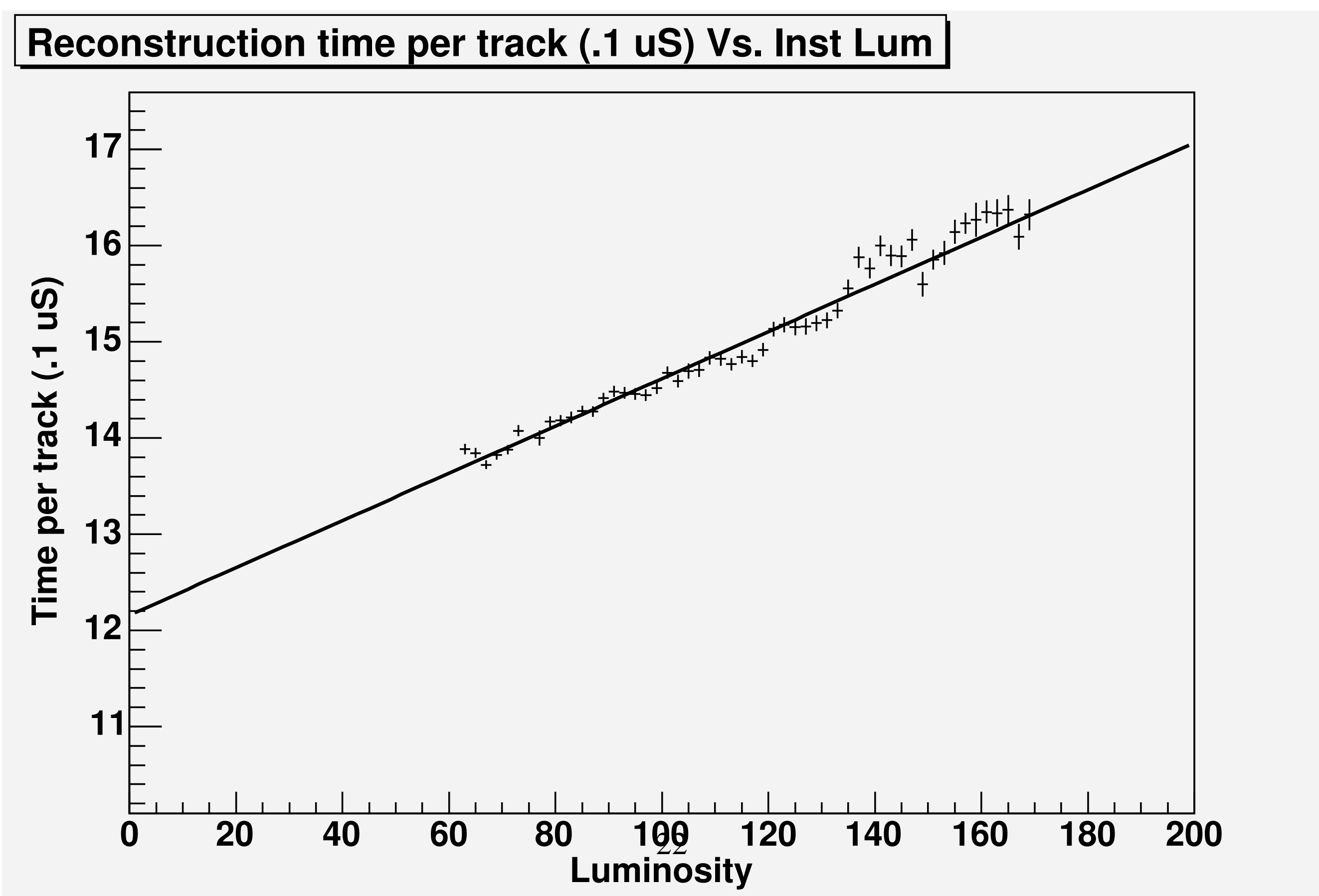

Figure 8: Time per XFT Axial Track Stereo Reconstruction, in units of .1 μS.

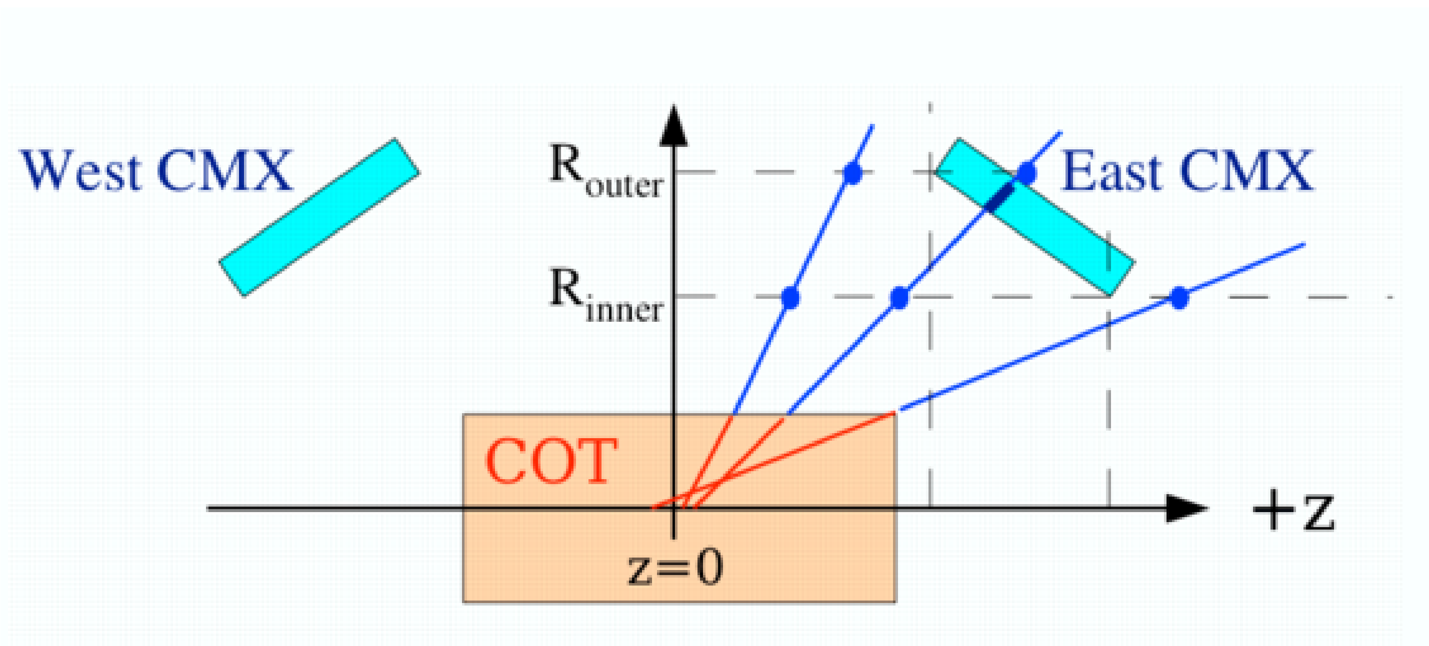

Figure 9: CMX stereo track reconstruction.

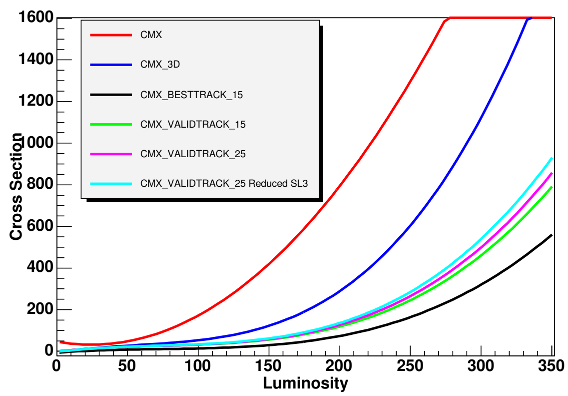

Figure 10: L2_CMX_3DMATCH rates for different possible versions of the trigger. Existing CMX and CMX_3D are shown for consistency. The 'Best track' method is shown in black, as well as the all or 'Valid Track' methods for different cuts on $\Delta_{3,5,7} \leq 15, 25$.

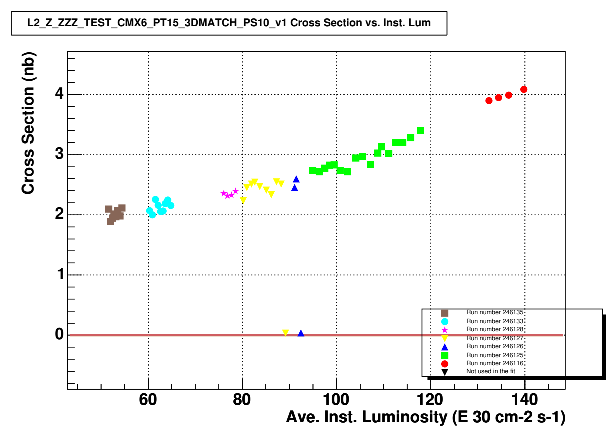

Figure 11: The rates in data agree with simulation. Shown here is the final CMX_3DMATCH trigger corresponding to CMX_VALIDTRACK_25 in 10, prescaled by 10.

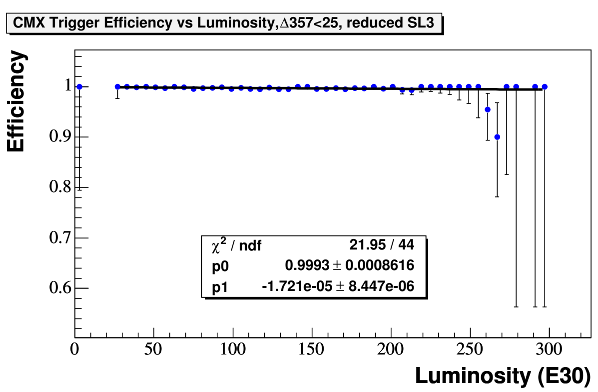

Figure 12: L2_CMX_3DMATCH efficiency when compared to good, well matched offline muon tracks.

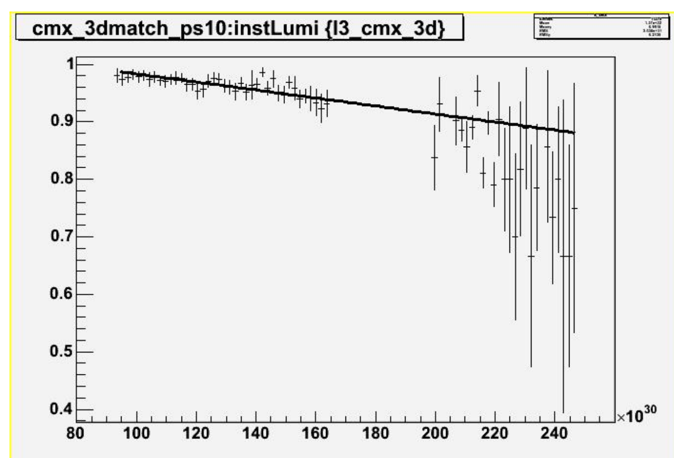

Figure 13: Efficiency from data.

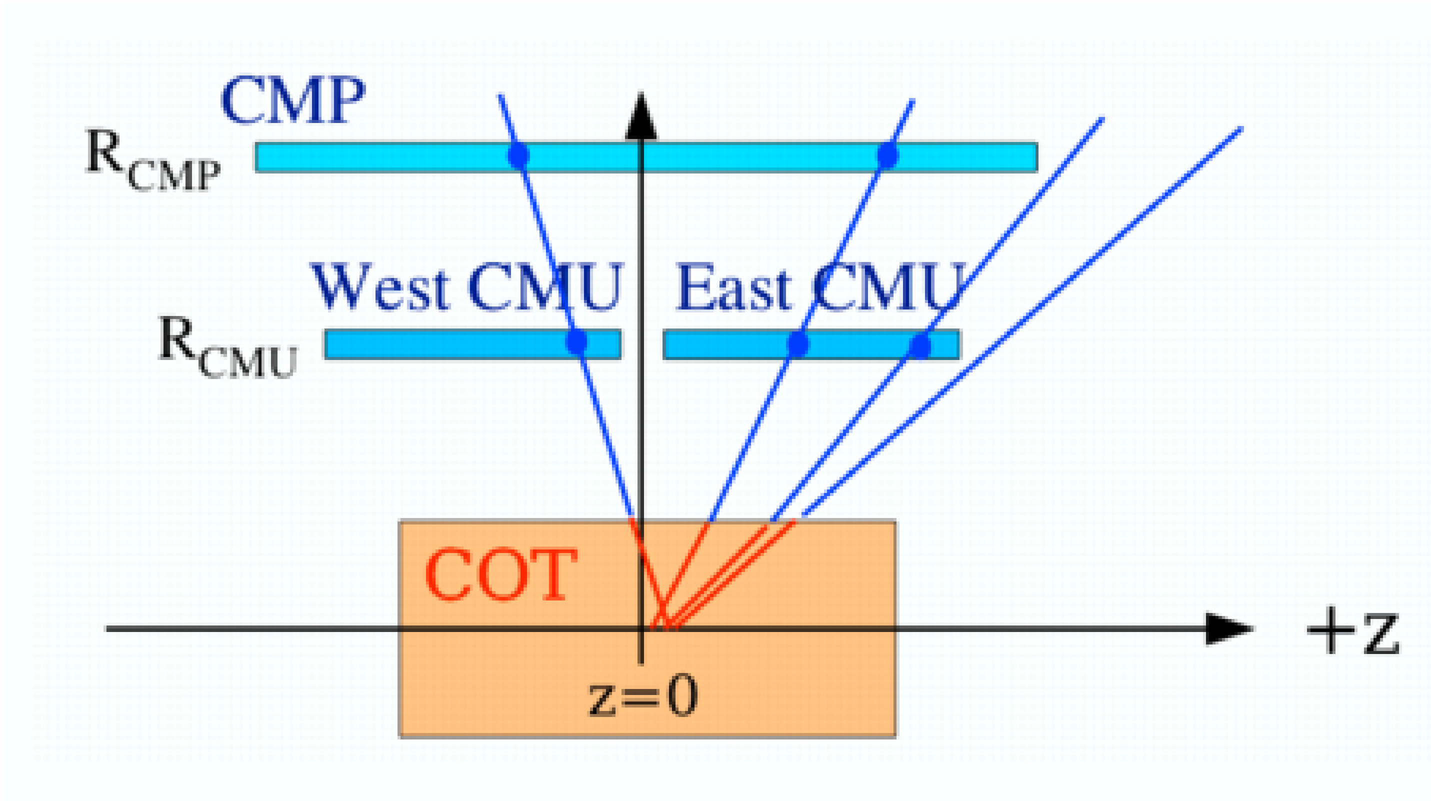

Figure 14: CMUP stereo track reconstruction.

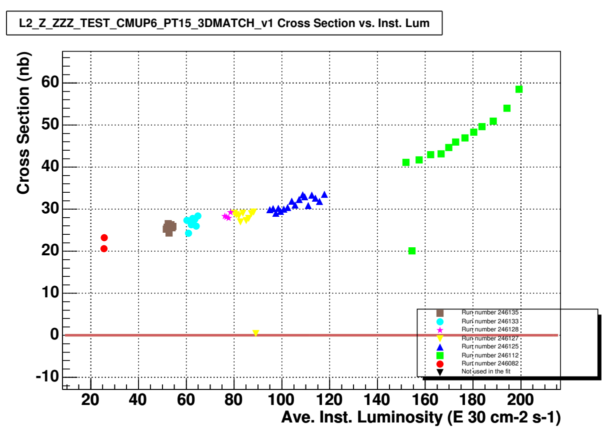

Figure 15: L2_CMUP_3DMATCH 'valid track' rates from data. They match projections from the HTTF rate tools.

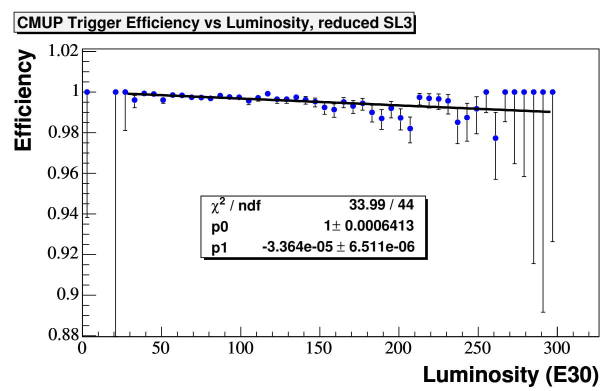

Figure 16: Offline method of finding efficiency for good matched CMUP offline muons.