

Updates over spring break



Anton Kapliy
University of Chicago
March 30, 2009

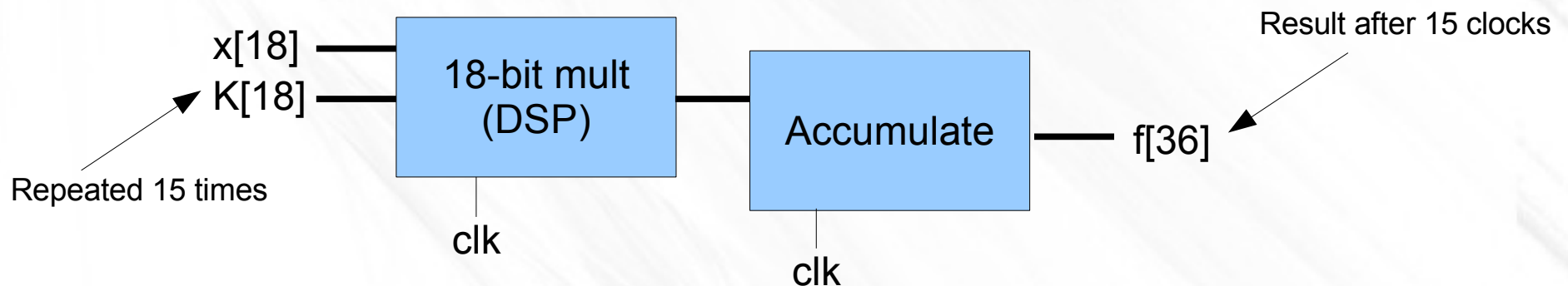
Technology preview

- FPGA at 1.5 GHz
- DSP's work at 1.5 GHz
- 90-200 DSP/chip (now)
- Just released in 2009
- Startup Achronix Inc.
 - \$100M venture \$\$\$
- A lot of excitement in the blogosphere



Timing estimate (if new FPGA's work)

- 1.5 GHz clk = 0.7 ns. Suppose 800 DSP/chip
- One track requires 14 parallel fits:
 - $f\{14\} = K\{14,14\} * x\{14\} + C\{14\}$
 - 5 f's are (d0,z0,curv,phi,eta)
 - 9 f's are constraints (used for chi2)
- Each $f[i]$ needs (14+1) multiply-accum cycles:



$$2 * [(15 \text{ cycles}) * 0.7 \text{ ns}] * 14 \text{ parameters} / (800 \text{ DSP}) = \mathbf{0.4 \text{ nanoseconds per fit}}$$

Factor of "2" to give time for missing point guess and other operations that can't be pipelined

In 1 millisecond, we can do **2.5M fits**

Progress on 1/2-SS banks

- Implemented existing fit-saving methods:
 - Max=1000 fits per road
 - Cross-sector roadwarrior
- Pretty much state-of-the art settings, minus
 - Cross-subregion roadwarrior
 - Bank cleaning (using pattgen-only banks)
 - Expect 25% reduction in #fits for 5% loss in eff
- Looking at hilumi Whbb with 50x64x144
 - This is the ~67% coverage bank!

Whbb @ 10³³

Bank1
only

∃ road in
bank2
that
matches
ss-by-ss

Only do
fits with
hits
found in
bank2
road

Wrt
IPAT

In
1000's

	0	2	3	$\Delta(3-0)$	
eff	80.1	77.1	74.4	-5.7	%
fakes	14.7	14.6	14.4	-0.3	%
#patt	158	107	107	1.48	(factor)
#sec	47	27	27	1.74	(factor)
#fits(11/11)	12	11	1.1	10.91	(factor)
#fits(10/11)	11	9	1.1	10	(factor)
#fits(maj)	116	111	11	10.55	(factor)
#fits(tot)	139	131	13.2	10.53	(factor)

Whbb @ 10³⁴

Bank1
only

∃ road in
bank2 in
the same
sector

∃ road in
bank2
that
matches
ss-by-ss

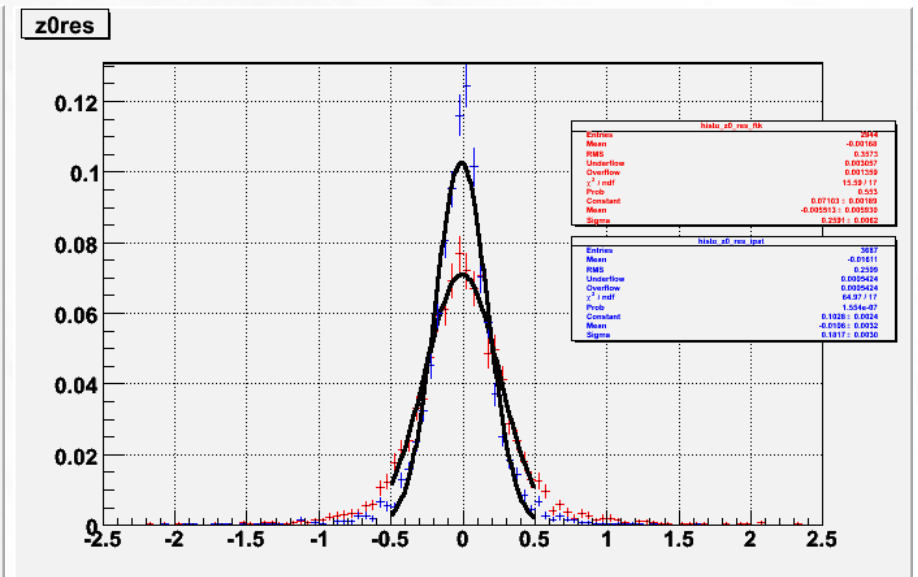
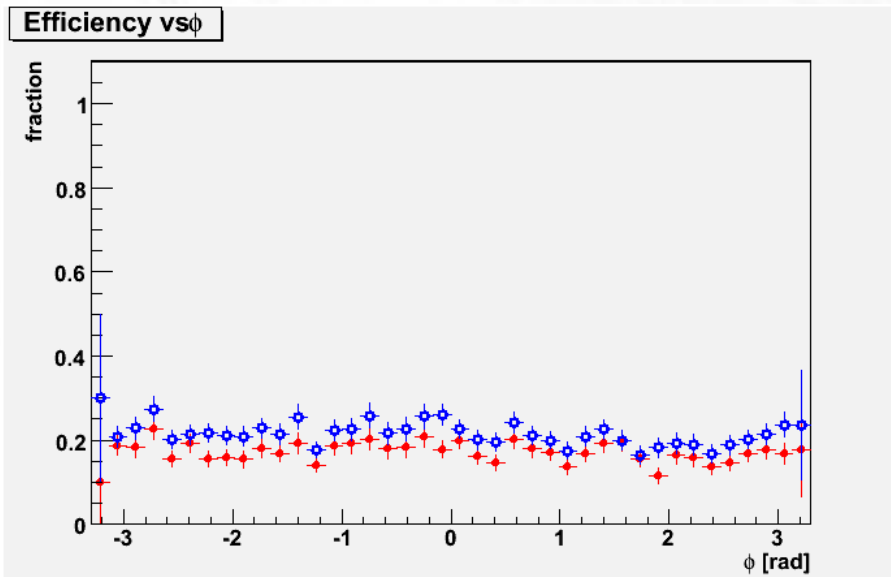
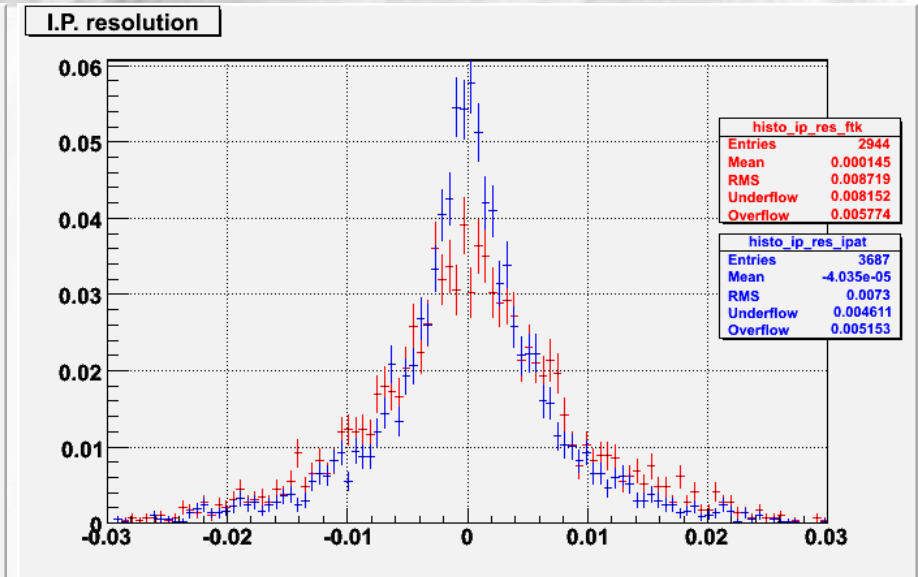
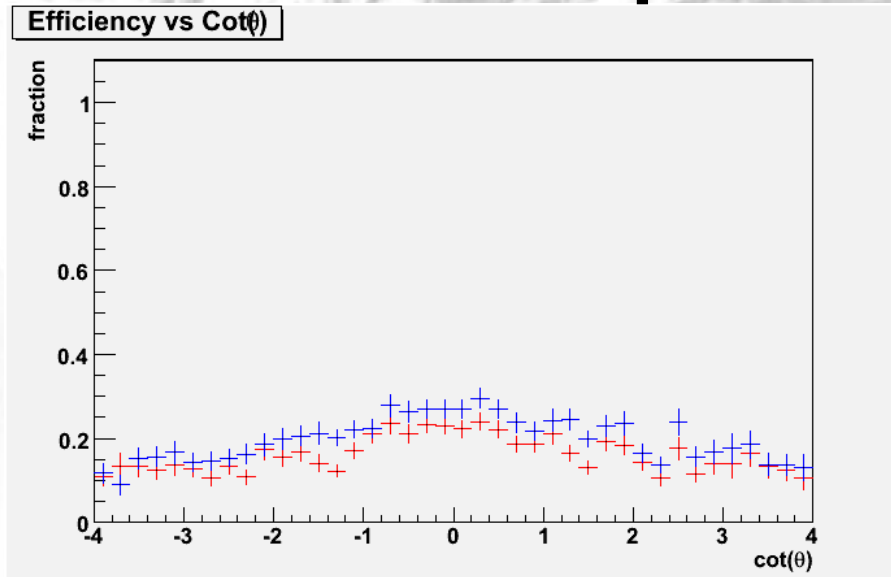
Only do
fits with
hits
found in
bank2
road

	0	1	2	3	$\Delta(3-0)$	
eff	76.9	75.8	75.3	71.3	-5.6	%
fakes	19.8	19.6	19.5	18.8	-1	%
#patt	6786	5194	3706	3706	1.83	(factor)
#sec	1790	1037	861	861	2.08	(factor)
#fits(11/11)	199	189	168	21	9.48	(factor)
#fits(10/11)	391	335	257	34	11.5	(factor)
#fits(maj)	1993	1888	1676	211	9.45	(factor)
#fits(tot)	2583	2412	2101	266	9.71	(factor)

Wrt
IPAT

In
1000's

Whbb @ 10^{34} General plots for case 3 (1)



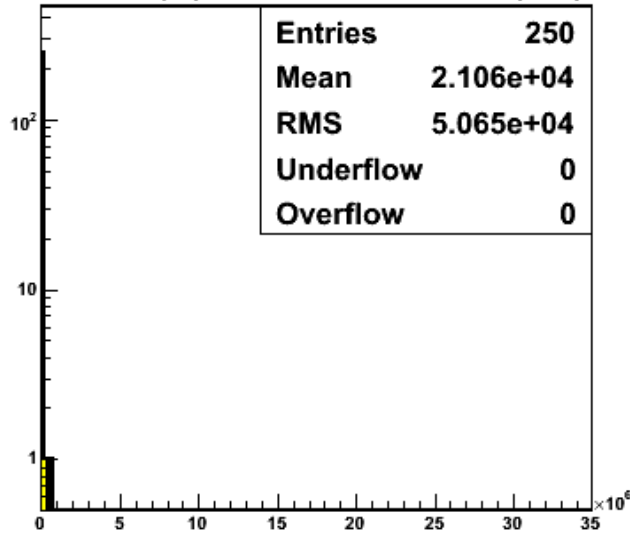
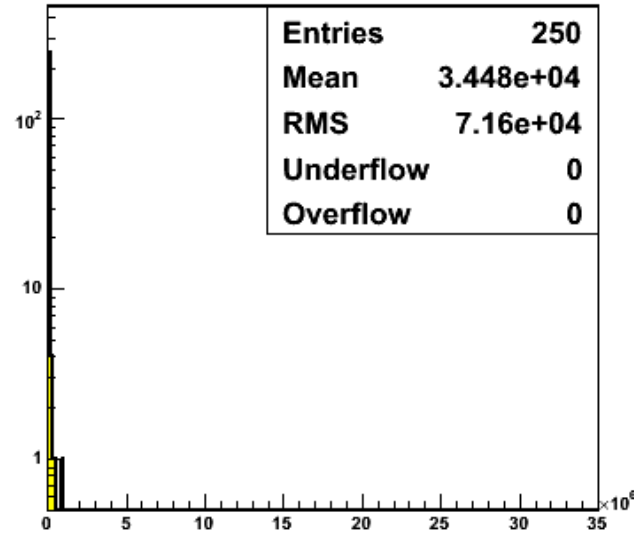
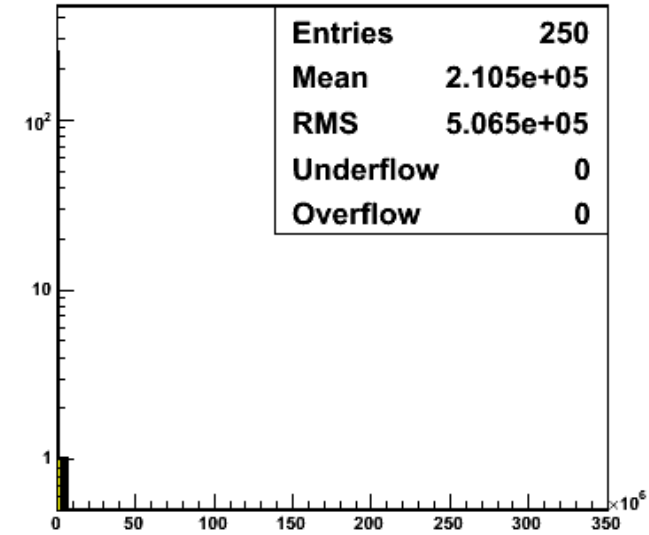
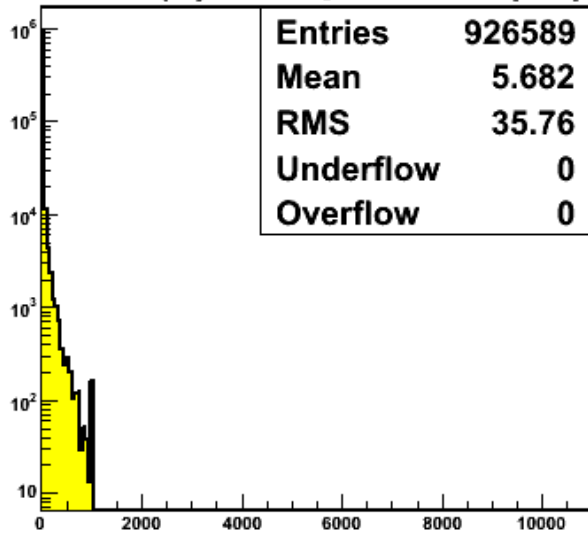
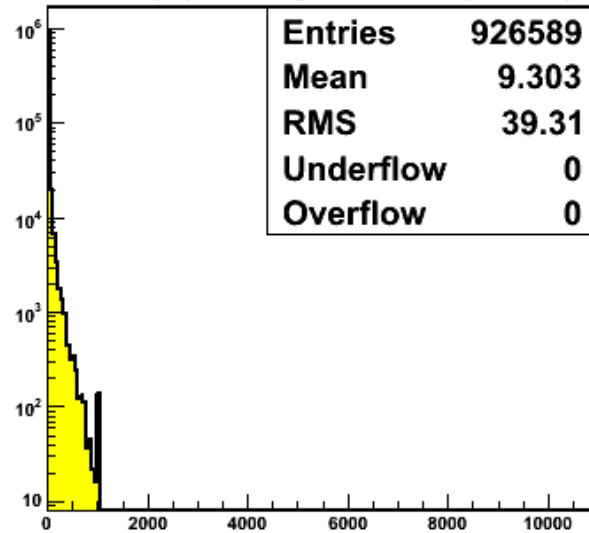
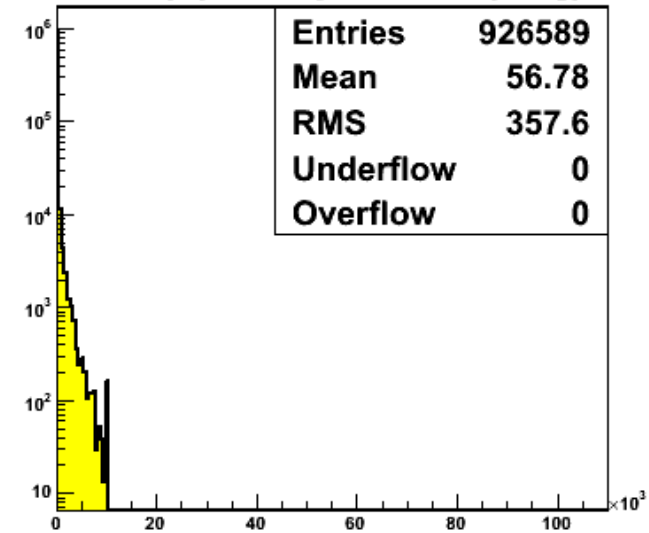
Everything looks fine: no biases observed anywhere.

General plots for case 3 (2)

11/11

10/11

10/11 (maj)

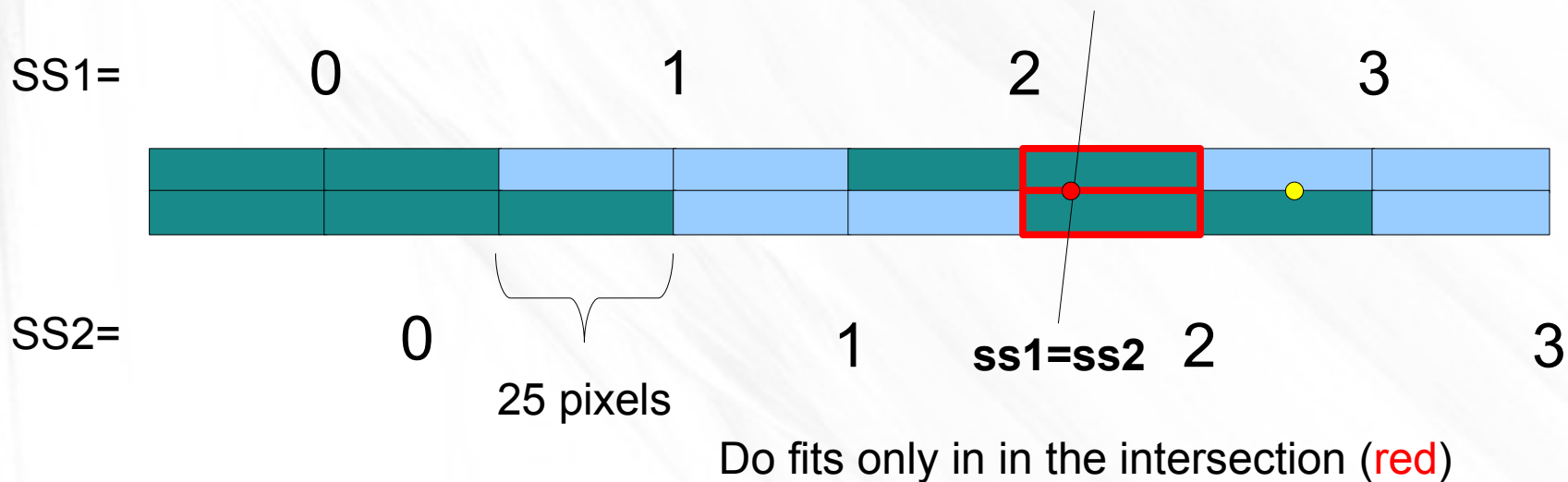
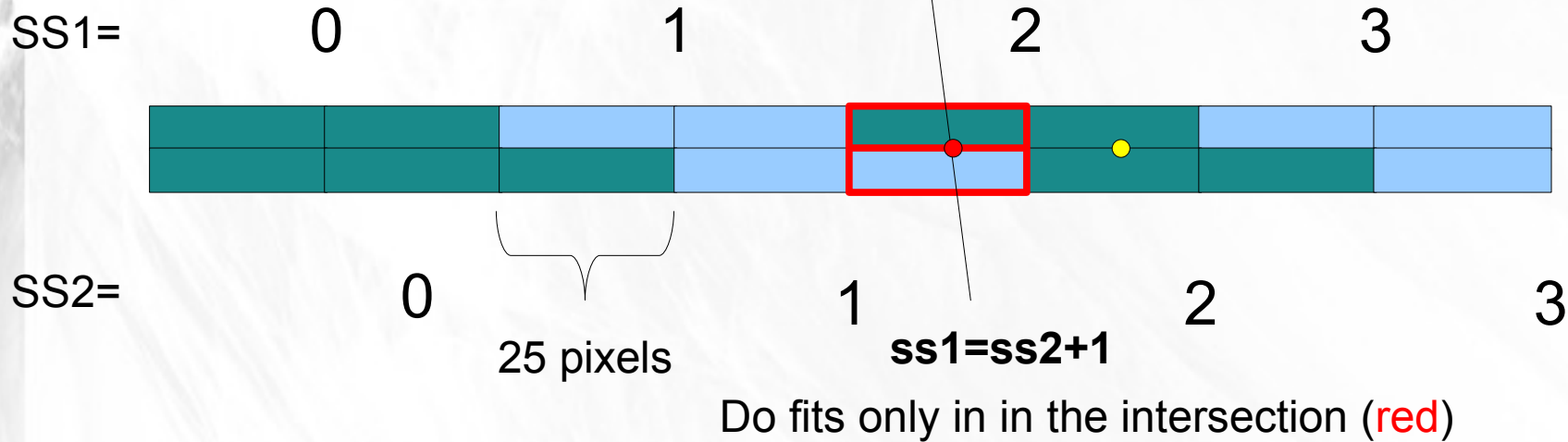
ALL(1): fits per event(all)

ALL(1): fits per event(miss)

ALL(1): fits per event(maj)

ALL(1): fits per road(all)

ALL(1): fits per road(miss)

ALL(1): fits per road(maj)


Case 3 summary:

- **5.5%** loss in efficiency in Whbb and single mu
- Factor of **2** increase in pattern bank size
- A factor of **10** reduction in #fits to **260k** (!)
- These results are **on top of other fixes**
 - Don't depend on roadwarrior details
- Hardware viability in data organizer - ?
 - Extra complication: matching multiple roads in 1/2-ss shifted bank
 - See next pages for details

Case 3: road matching and fits

- Real hits
- Noise hits

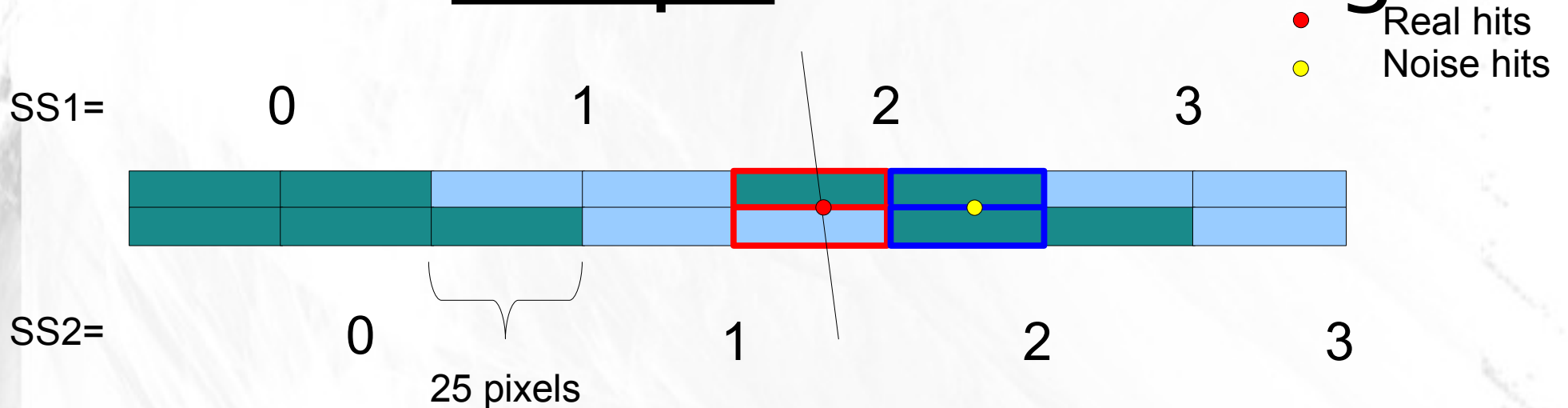


Case 3: ftksim implementation

- Load both banks
- Find roads in bank1 and bank2
- Run roadwarrior in bank1 and bank2
 - Final collections of roads are $\{r1\}$ and $\{r2\}$
- Loop over $\{r1\}$ and choose subset $\{r1'\}$:
 - RW** – Each $\{r1'\}$ had a list of $\{r2'\}$ that “match”
 - Match if $ss1=ss2$ or $ss1=ss2+1$ in each layer
- For each road in $\{r1'\}$, loop over hit combs:
 - Hit filtering** – Drop hit combs that aren't found in any $\{r2'\}$
 - I.e., only fit hits in the intersection of $r1$ & $r2$

(single-layer case shown for simplicity)

Case 3: multiple road matching



$$ss1=ss2+1 \text{ OR } ss1=ss2$$

We can match two roads in bank2 to road SS1=2 in bank1:
SS2=1 OR SS2=2

We want to consider hits in the intersection of SS1 AND SS2.

But there are two roads in SS2 bank that “match”
⇒ We must consider two intersections (red and blue).

roads {r2'} matched to each r1

- Let r1 be roads in bank1; r2 in bank2 (shifted)
- Multiple roads r2 can match to the same r1:
 - r1=r2 if $ss1=ss2$ OR $ss1=ss2+1$
 - Presence of 10/11 roads in both banks
- Call maximum # roads to match mroads:

No additional benefit

	1	5	10	20
eff	48.4	69.6	71.3	71.6
#fits(11/11)	15.6	17.9	21.1	22.3
#fits(10/11)	13.5	30.9	34.5	35.4
#fits(maj)	68.4	179	211	223
#fits(tot)	97.5	227.8	266.6	280.7

Whbb @ 10³⁴

Treatment of 10/11 roads

- A missing layer in either r1 or r2 is ignored in road “matching”(i.e. automatically satisfied)
 - Thus, we may require 11, or 10, or 9 SS values to be “the same or offset by 1”

<i>Layer</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	
SS1	2	2	2	2	match 4/4
SS2	2	1	2	1	

<i>Layer</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	
SS1	2	2		2	match 3/4
SS2	2	1	2	1	

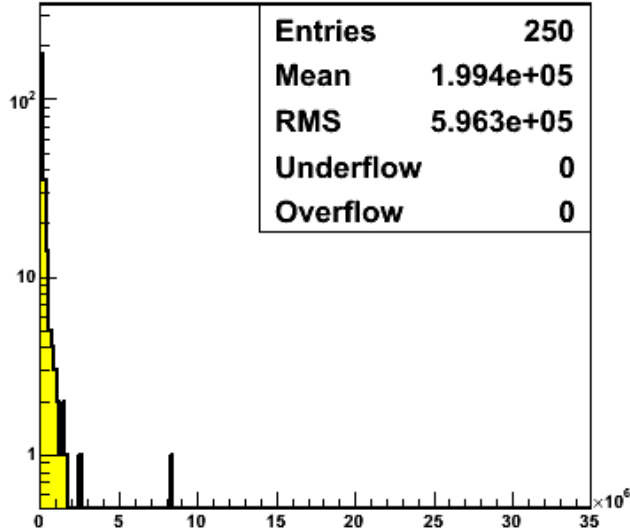
<i>Layer</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	
SS1	2		2	2	match 2/4
SS2	2	1		1	

- This is done to keep efficiency high
- In choosing which combs to do, we only do fits for hits found in r1 and some r2
 - If \exists missing layer in r2 its r1 hit is accepted

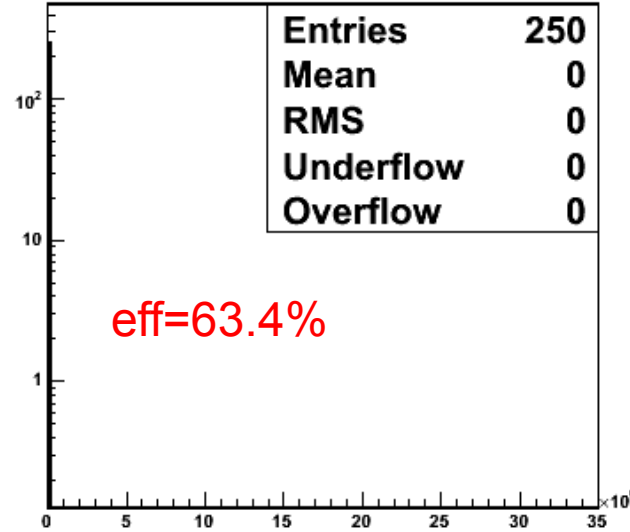
Case of 11/11 roads only

Case 0: default bank, only match 11/11 roads

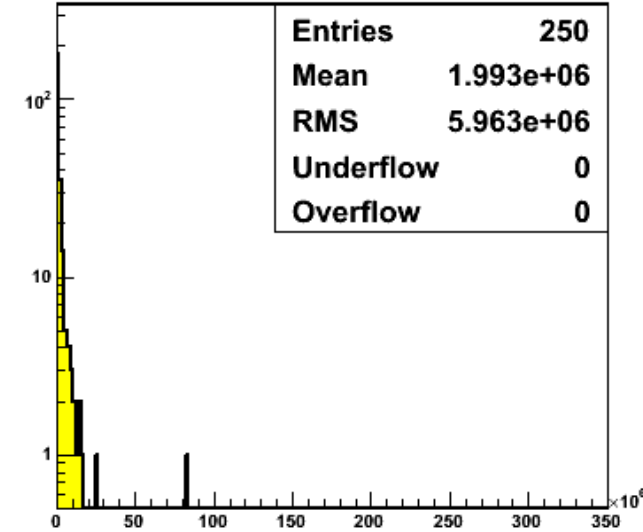
ALL(1): fits per event(all)



ALL(1): fits per event(miss)

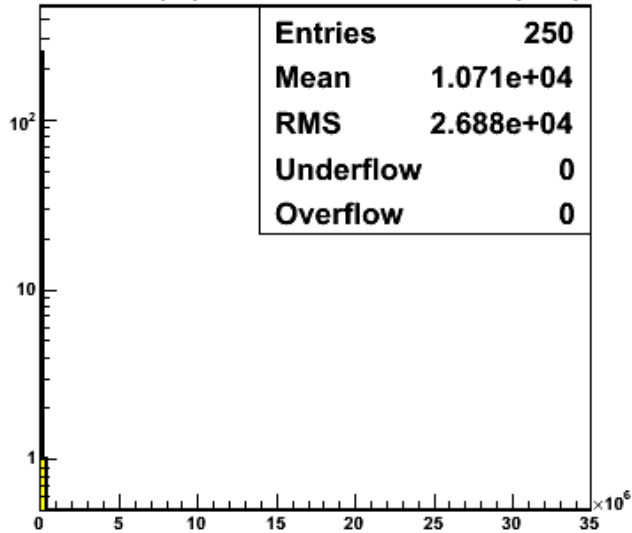


ALL(1): fits per event(maj)

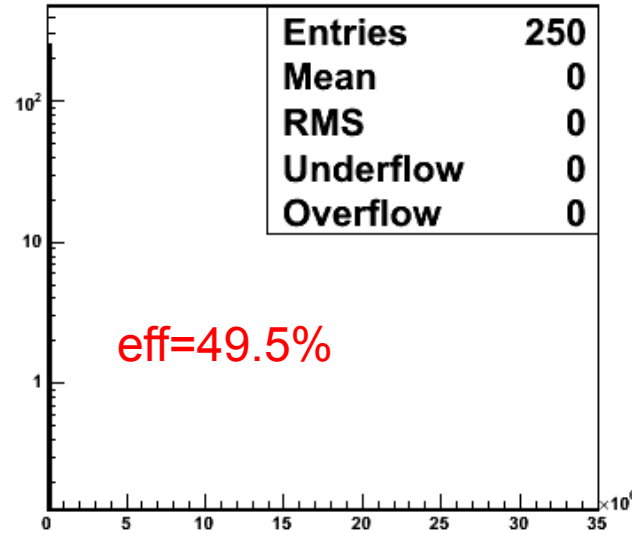


Case 3: two banks + hit filtering, only match 11/11 roads

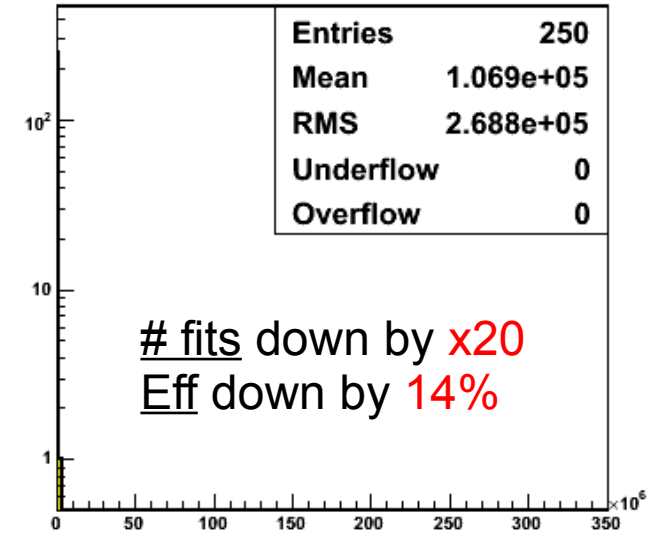
ALL(1): fits per event(all)



ALL(1): fits per event(miss)

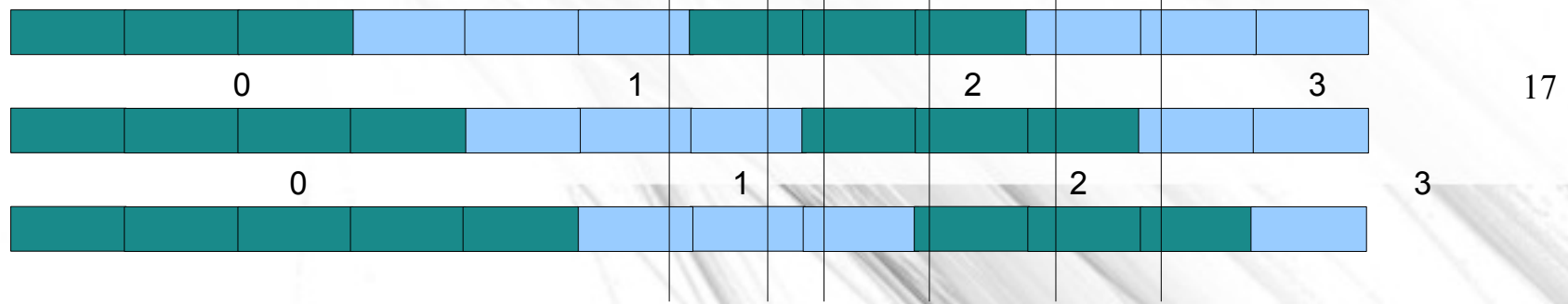


ALL(1): fits per event(maj)



Further discussion

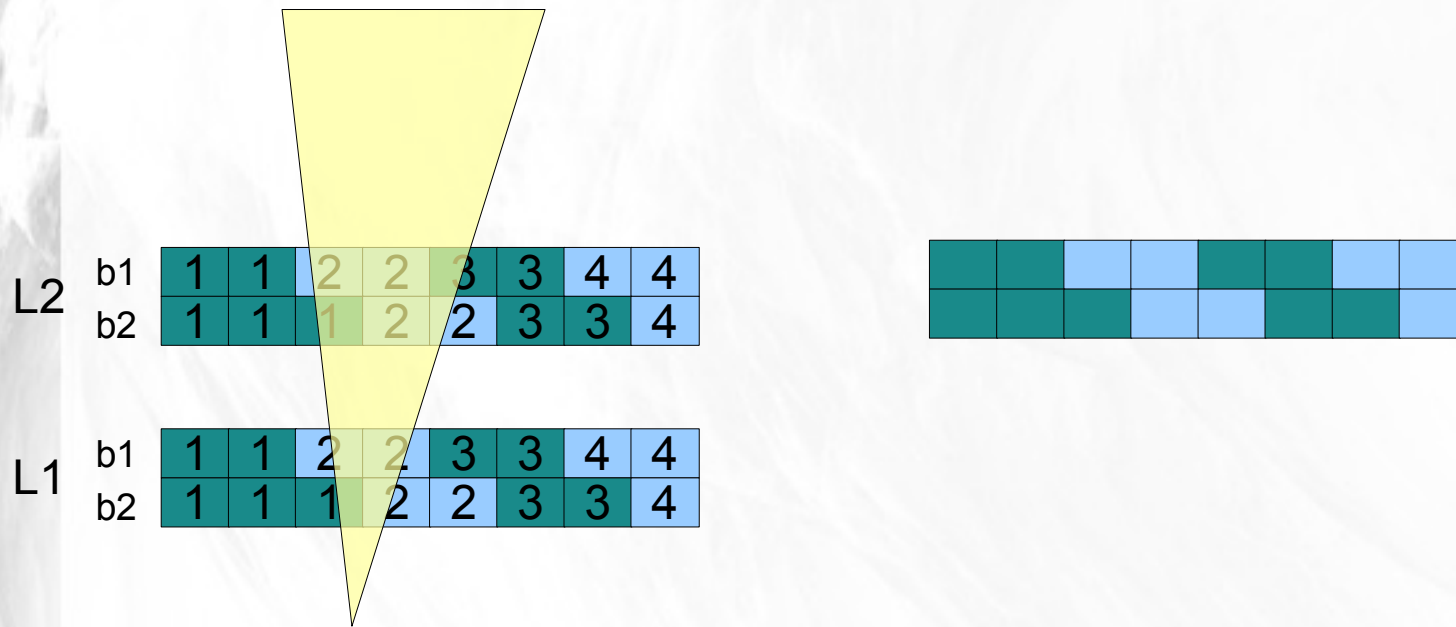
- Method shows excellent prospects
- Additional free parameters to be tuned:
 - What if we drop 10/11 roads in bank2?
 - What if we reduce size of bank2?
 - What if we don't do roadwarrior in bank 1 or 2?
 - No RW in bank2 = better efficiency!
- Possible extension to > 2 parallel banks?



ADDITIONAL SLIDES

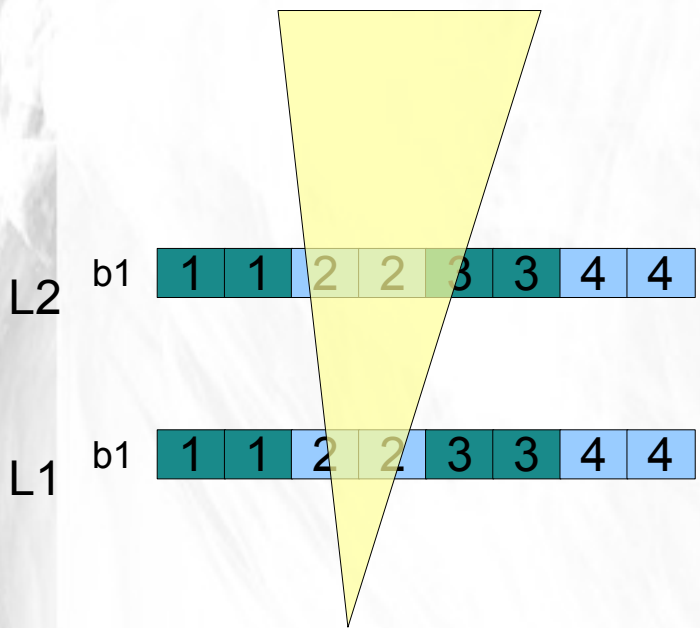
- Here is my attempt to explain what's happening in pictures
- It might be too technical to follow without my comments, so it's mostly here for my own benefit
- Considered here is a two-layer case

Training tracks spread



- Consider a 2-layer geometry with two alternative SS divisions (b1 and b2)
- Suppose yellow cone shows spread of training tracks
- What are possible bank configurations?

Training tracks

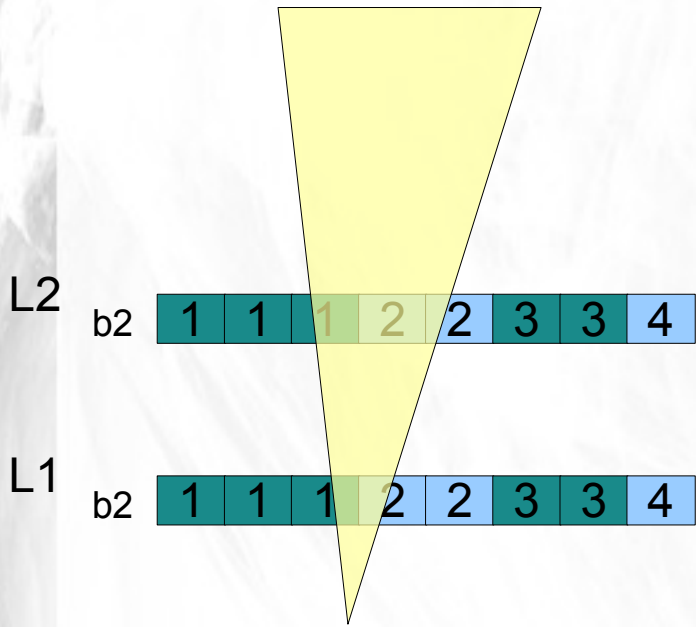


Bank1:

L1 L2

2 2
2 3

Training tracks

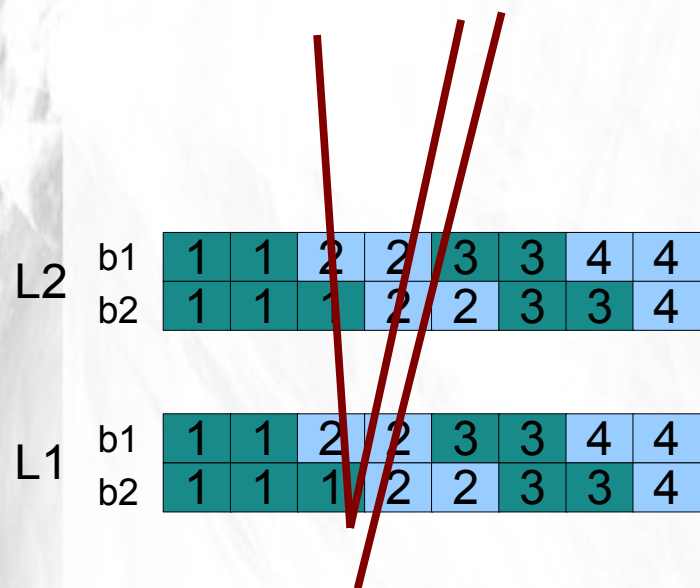


Bank2:

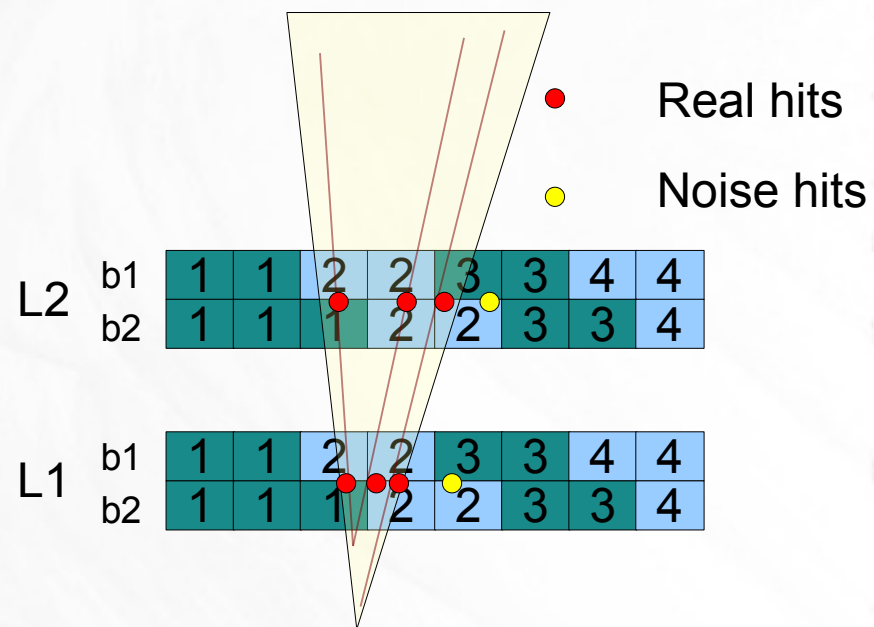
L1 L2

1	1
1	2
2	2

WH event (tracks)



WH event (hits)



Roads in individual banks and road matching

Bank1:

L1 L2

2 2
2 3

3x2 = 6 fits
3x2 = 6 fits
Tot = 12 fits

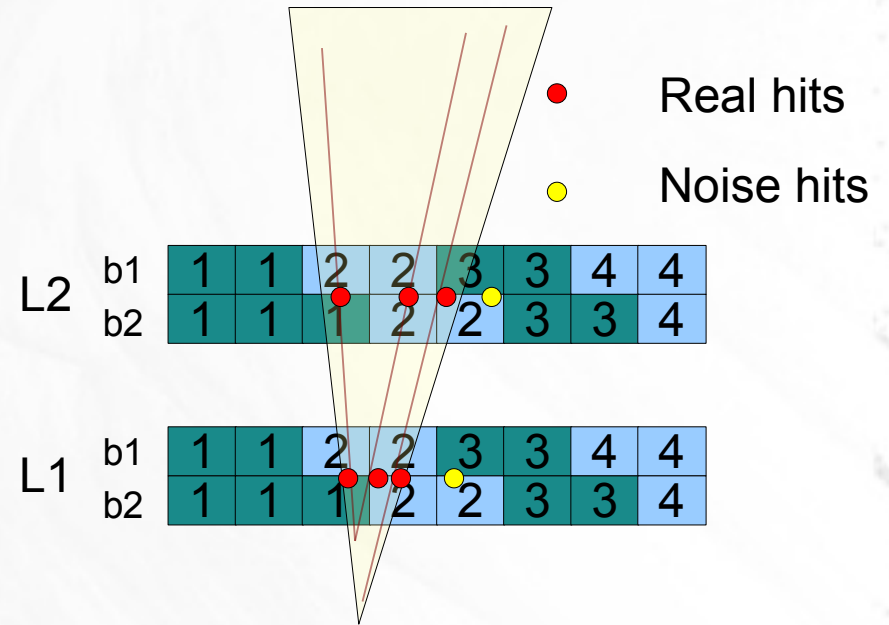
Bank2:

L1 L2

1 1
1 2
2 2

1x1 = 1 fit
1x3 = 3 fits
3x3 = 9 fits
Tot = 13 fits

WH event (hits)



Both roads in bank1 are accepted

Road matching:

ss1=ss2 || ss1=ss2+1

ss1	2,2	2,3
matching ss2	1,1	1,2
	1,2	2,2
	2,2	

Hit filtering for fits within matched roads:

Current algo: loop over hits in r1; loop over r2 & count matched hits

Road matching:

(2,2): (2,3):

```

1 1  1 2
1 2  2 2
2 2
    
```

12 fit combinations

(2,2): 3x2 possibilities

11: YES (1,1) REAL
 12: YES (1,2)

21: NO in (2,1)
 22: YES in (2,2) REAL

Removed two fits

31: NO in (2,1)
 32: YES in (2,2)

(2,3): 3x2 possibilities

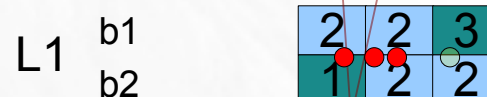
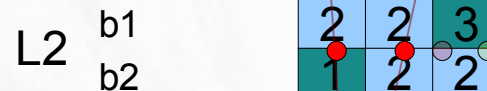
11: YES in (1,2)
 12: YES in (1,2)

21: YES in (1,2)
 22: YES in (1,2)

31: YES in (1,2) REAL
 32: YES in (1,2)

WH event (hits)

● Real hits
 ● Noise hits



WH event (hits)

● Real hits
 ● Noise hits



Hit filtering for fits within matched roads:
 Only do fits over hits found in both roads

Road matching:

(2,2): (2,3):

1 1 1 2
 1 2 2 2
 2 2

10 fit combinations (2,2): 3x2 possibilities

(1,1):
 11 REAL
 (1,2):
 12
 (2,2):
 22 REAL
 32

Removed two fits

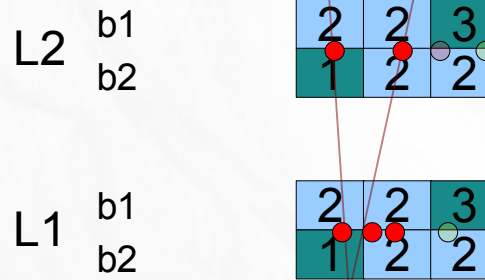
(2,3): 3x2 possibilities

(1,2):
 11
 12
 (2,2):
 21
 22
 31 REAL
 32

**SAME AS
 PREVIOUS
 PAGE**

WH event (hits)

● Real hits
 ● Noise hits



WH event (hits)

● Real hits
 ● Noise hits

