

I've run some further simulations for my associative memory track finder using a data file provided by Mike Procaro. The file contains 996 events with a mix of minimum-bias events and B events. The station spacing is 40 mm. The plane spacing is 4 mm. The file was "pixelated" to reflect a pixel size of 50 x 400 with a resolution of 10 x 400.

My simulations are mostly concerned with measuring data flow. Data flow is probably the most difficult problem to solve for this part of the trigger. Without extreme effort, data rates between individual modules are limited to 200 to 400 MBytes/Sec unidirectional. 400 MB/S corresponds to a 32-bit bus cycling at 100MHz.

Global data rates are more of a problem. When data must be shared among many modules, or must be collected from all modules, the requirements can get very high. In addition, since the modules are working independently, data must be sorted by timestamp in order to insure that each module operates on the proper event. This requires a smart switching network of the type used in the baseline triplet trigger. Each of its processing farms is capable of accepting data from 32 sources, sorting it by timestamp, and delivering it to one of 192 buffers in 64 processors. It can sustain a data rate of 800 MB/S. With a little additional effort, this type of network can be pushed to 6400 MB/S.

My simulation also includes some measures of physics performance. I measure how well found doublets and tracks correspond to real HEP tracks. We'll have to wait until the algorithm can be incorporated into the triggering simulations to determine how effective it is for physics.

Here are my results:

Hits:

The average number of hits per station-quadrant is 5.53. This equates to ~366 Megabytes/Sec into each doublet finder. $366 \text{ MB/S} = (5.53 \text{ Hits/Event} + 1 \text{ timestamp/Event}) * 7 \text{ MHz} * 2 \text{ planes/station} * 4 \text{ bytes/Hit}$

Doublets:

Doublets are found by choosing a hit in one plane and finding all hits in the second plane that fall within 4000 microns in both X and Y. The 4000 micron window is adjustable, and corresponds to tracks with a slope less than 1.

For the best case, the average number of doublets generated per station-quadrant is 5.28. This corresponds to ~352 MB/Sec. In addition to doublet rate, I measured two other paramters:

Bad doublets are those that aren't really on a track. They're generated when two tracks are close enough that plane hits from one track fall within the search window of the other track.

Good doublets are doublets that are on a real track, but were not found. These tracks have slopes that are steeper than the search window.

I also try two variations on the algorithm:

RejectUsed: This rejects a doublet if it has a steep slope and one of its plane hits has already been used to form a doublet. This helps in the case where two parallel tracks are near each other, but will tend to favor smaller slopes when, perhaps, it shouldn't. The effect is also dependent on data ordering.

In the tables, a RejectUsed value of 4000 correponds to no rejection, while a value of 1000 will reject doublets that have a slope greater than 1/4. The RejectUsed feature seems to eliminate a significant number of bad doublets without significantly increasing the loss of good doublets.

ConvertUnused: This converts a plane hit into a doublet if it isn't part of another doublet. This is intended to deal with pixel inefficiencies. Unfortunately, it also increases the effect of noise hits. The ConvertUnused feature adds about 1 extra doublet per station.

Table 1. Doublet rates per station-quadrant for four algorithm variations.

Reject	Conv			
<u>Used</u>	<u>Unused</u>	<u>Average</u>	<u>Bad</u>	<u>Good</u>
4000	0	7.05	2.07	0.14
4000	1	7.97	2.07	0.14
1000	0	5.28	0.33	0.17
1000	1	6.28	0.33	0.17

Tracks:

Tracks are found from left to right by matching a station doublet to a partial track. Doublets that match are added to the track and projected to the next station. Doublets that do not match a track are sent to the next station as the start of a new track. Match window sizes are determined by pixel geometry, scattering, and curvature. For the first match of a track, a $4100\mu \times 4100\mu$ window is used. This roughly corresponds to the coarse pixel resolution (400μ) times the ratio of station spacing to plane spacing ($40/4$). For each additional match, a window of $500\mu \times 4000\mu$ is used. The X window size (500μ) is dominated by scattering. The size of the Y window is dominated by curvature. (Note that by ‘window’ I mean the distance from center to edge.) See figures 3, 4, and 5 for measured window sizes.

I tried a variation of the algorithm that uses the curvature of the track when predicting the next station hit. This had almost no effect on performance (see Table 2). I don’t understand why, but it probably has something to do with the fact that very curvy tracks also have a lot of scattering. Table 3 shows that using curvature, 59.6 percent of the doublets fall within 250 microns of their predicted position. Not using curvature, 57 percent are within 250 microns of the prediction. With a window size of 3100 microns, the two cases are almost identical.

For the best case, there were an average of 5.62 tracks per station-quadrant. This corresponds to a data rate of about 433 MB/S.

Track Ends

Track ends are emitted when the first three or last three hits of a track are found. The best case generated 1.7 track ends per station. Since all of the track ends must be gathered into a single processor, this represents a global data rate of ~4430 megabytes per second per quadrant. Such a high data rate will probably require a 2-level processing farm to handle all 4 quadrants. The first level can handle quadrants independently, trim down the data, and pass only vertex-specific data to a vertex farm, while saving the rest of the track data for triggered readout.

Figure 2 shows a single event including track ends circled in blue. Track ends include 3 hits unless the track evaporates before leaving the pixel array **B**. The yellow lines represent predicted station hits. Purple lines represent a prediction after a station miss **A**. Green lines connect plane hits that form good doublets. Red lines connect plane hits that form bad doublets **C**.

Track Windows

Figures 3, 4, and 5 are histograms of window sizes.

Figure 3 shows that all but 2% of hits in a doublet are within 4000 microns of each other.

Figure 4 shows that 90% of all first track hits are within 4100 microns of their predictions.

Figure 5 shows that 90% of other track hits are within 500 microns of their predicted X position and 3000 microns of their predicted Y position.

Table 2. Track rates for all variations

Use Curve	Reject Used	Conv Unused	Average	Bad	Good	Ends
0	4000	0	7.73	2.21	2.14	2.90
0	4000	1	9.25	2.35	2.74	3.05
0	1000	0	5.62	0.67	0.75	1.70
0	1000	1	7.24	0.83	1.39	1.88
1	4000	0	7.72	2.20	2.18	3.02
1	4000	1	9.24	2.34	2.78	3.17
1	1000	0	5.62	0.66	0.75	1.73
1	1000	1	7.24	0.82	1.39	1.90

Figure 1 Curve following. ddY is used to predict next station hit.

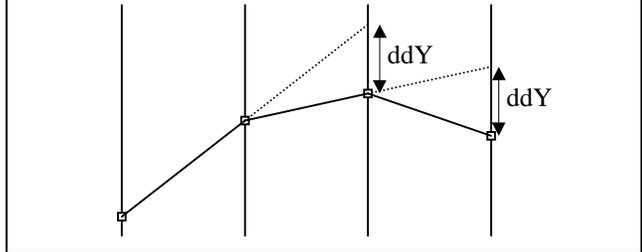
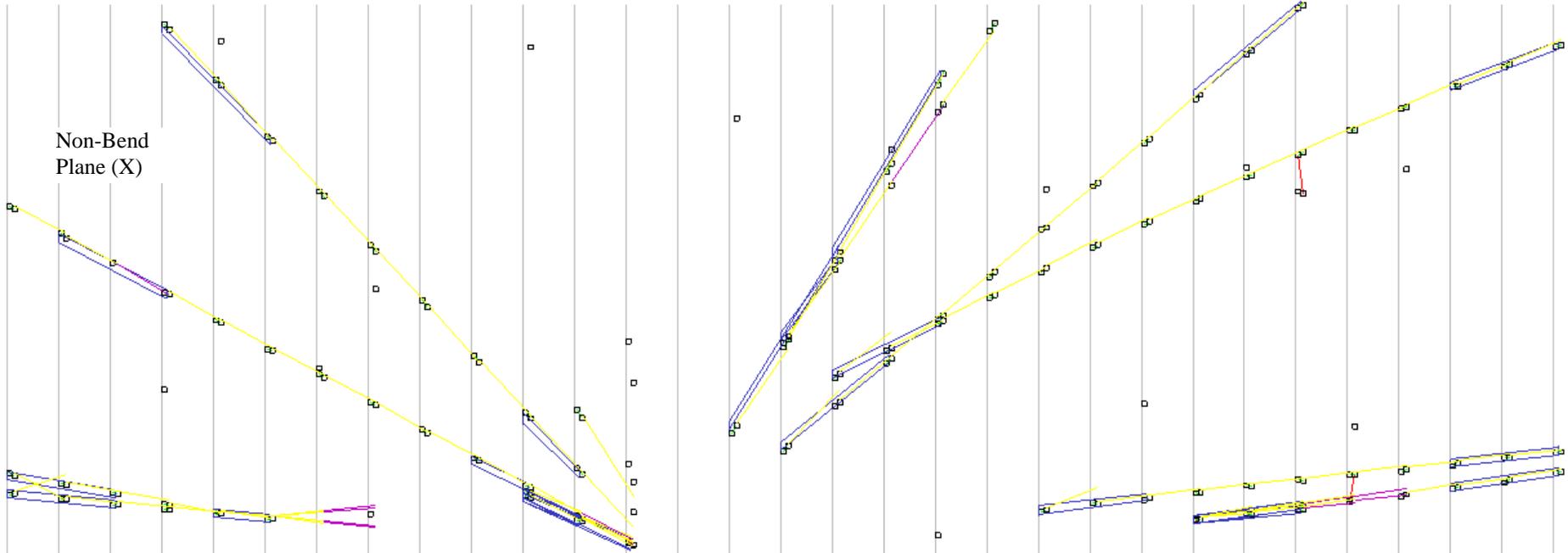
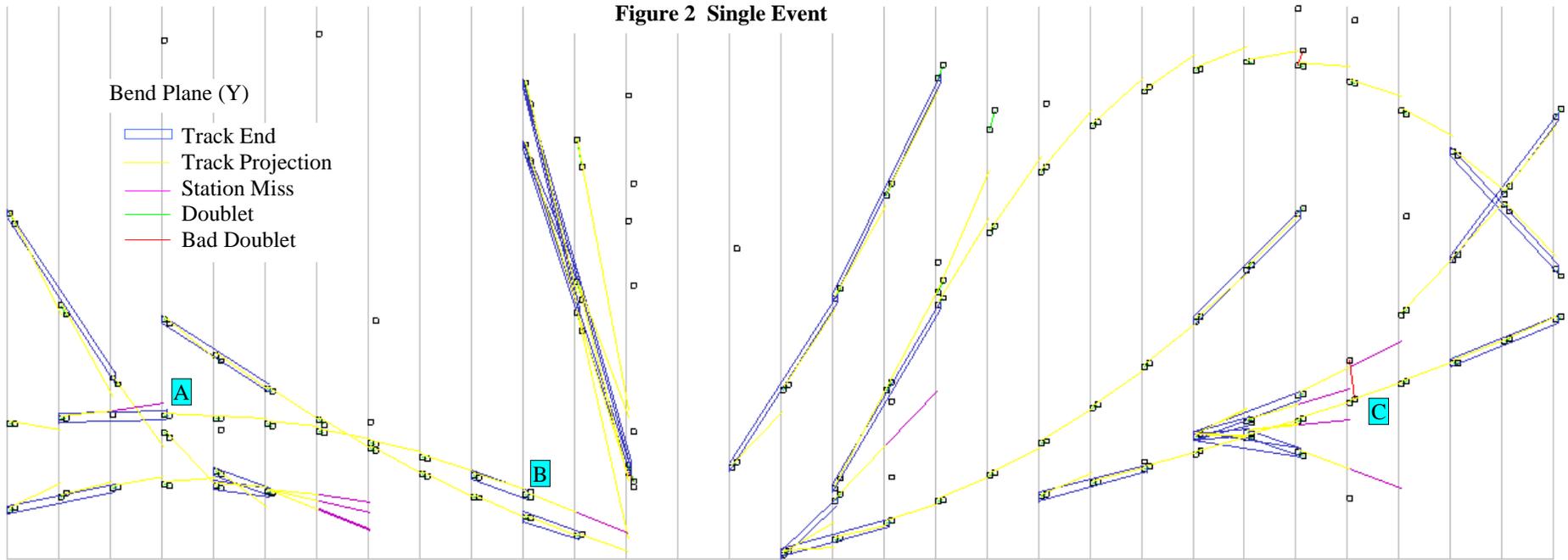


Table 3. Track window size with and without curve following. Percent of tracks falling inside match window.

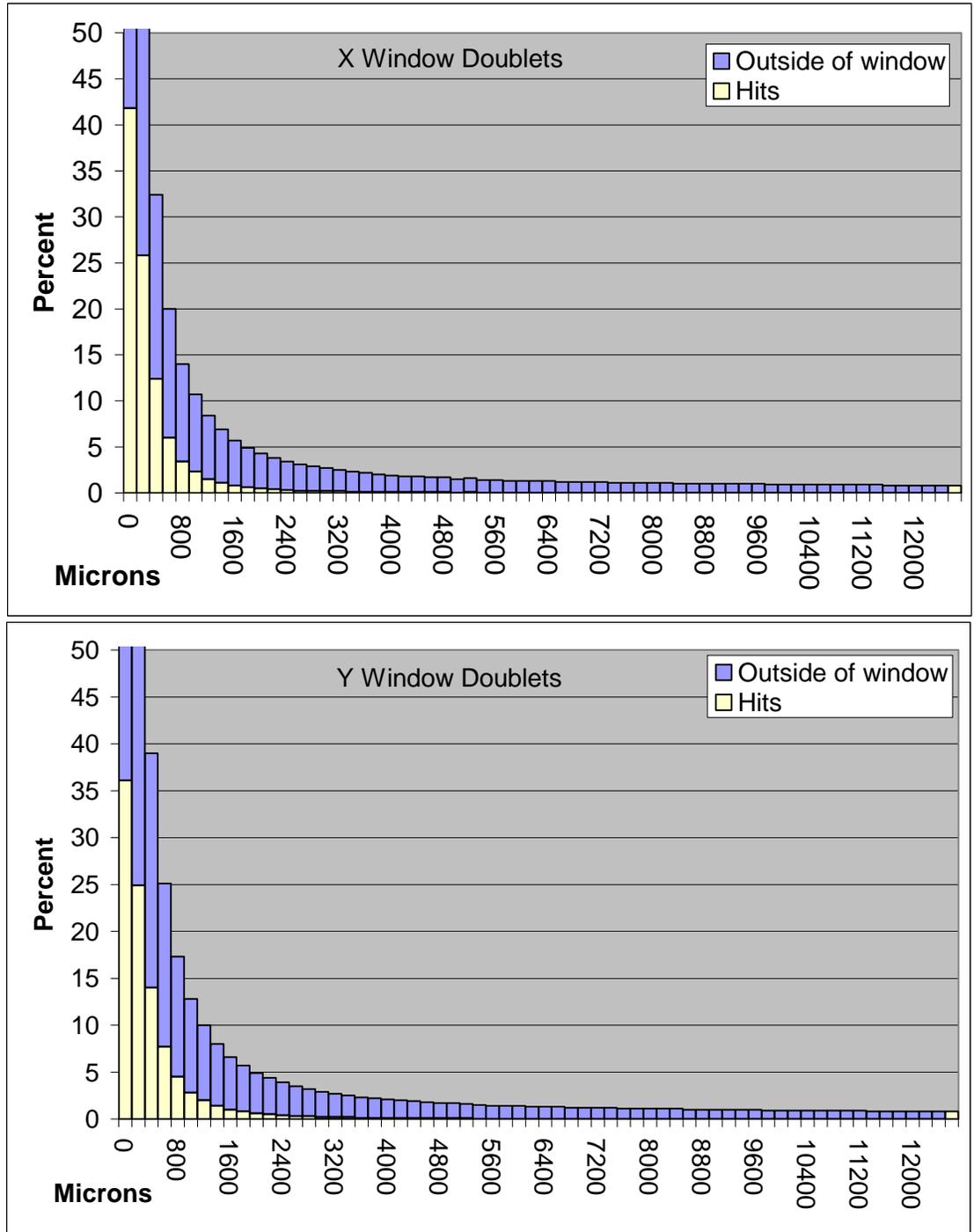
Window Size (μ)	No Curve		Use Curve	
	Tracks	(Accum)	Tracks	(Accum)
0	12.6	12.6	16.7	16.7
50	13.4	26.1	14.3	31.0
100	10.9	37.0	10.5	41.5
150	8.5	45.5	7.9	49.4
200	6.5	52.0	5.8	55.2
250	5.0	57.0	4.4	59.6
3100	0.1	91.6	0.1	91.1

Figure 2 Single Event



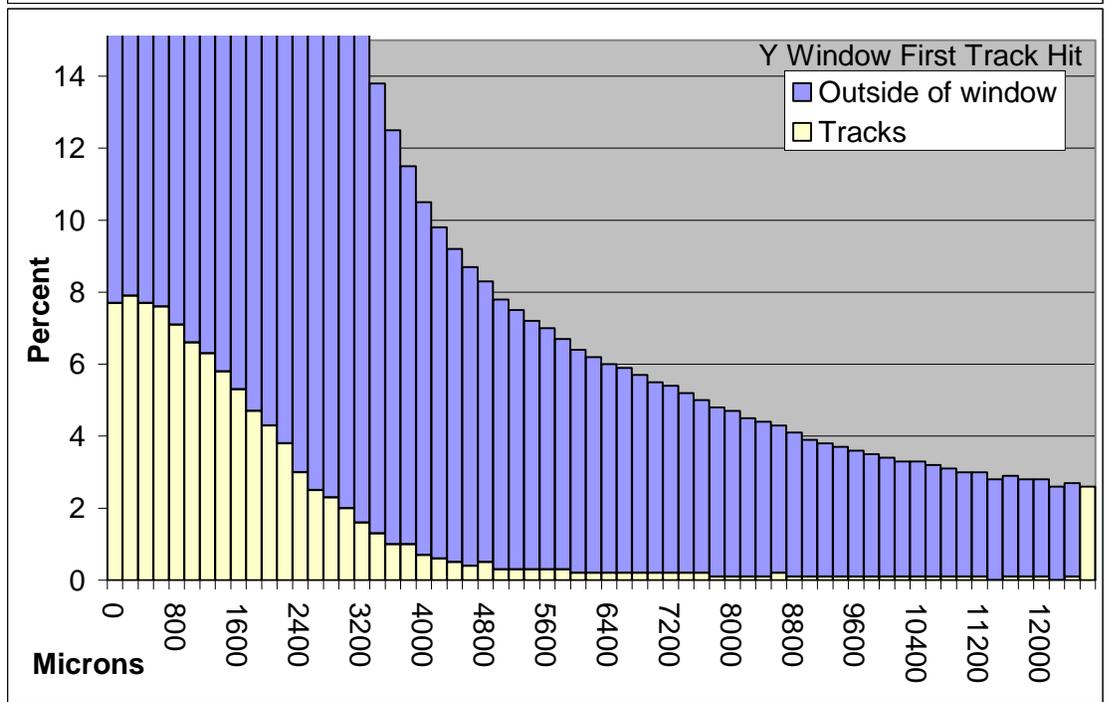
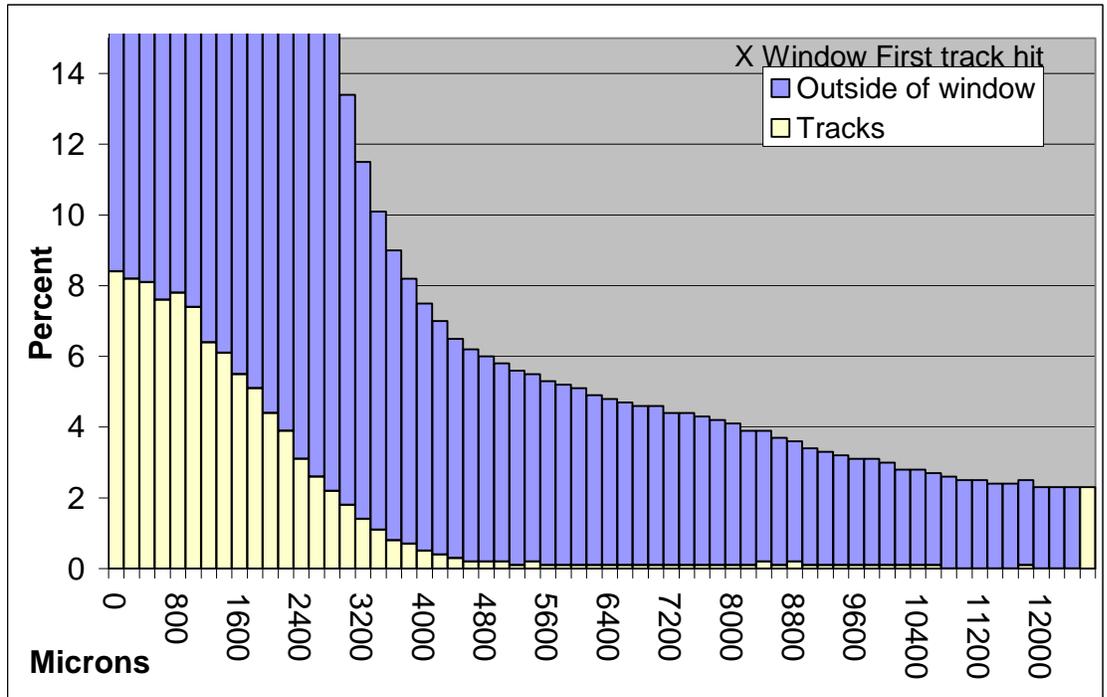
0	41.8	36.1
200	25.8	24.9
400	12.4	14.0
600	6.0	7.7
800	3.4	4.5
1000	2.3	2.8
1200	1.5	2.0
1400	1.1	1.4
1600	0.8	1.0
1800	0.6	0.8
2000	0.5	0.6
2200	0.4	0.5
2400	0.3	0.4
2600	0.2	0.3
2800	0.2	0.3
3000	0.2	0.2
3200	0.2	0.2
3400	0.1	0.2
3600	0.1	0.1
3800	0.1	0.1
4000	0.1	0.1
4200	0.1	0.1
4400	0.1	0.1
4600	0.1	0.1
4800	0.1	0.1
5000	0.0	0.1
5200	0.1	0.1
5400	0.0	0.0
5600	0.0	0.0
5800	0.0	0.0
6000	0.0	0.0
6200	0.0	0.0
6400	0.0	0.0
6600	0.0	0.0
6800	0.0	0.0
7000	0.0	0.0
7200	0.0	0.0
7400	0.0	0.0
7600	0.0	0.0
7800	0.0	0.0
8000	0.0	0.0
8200	0.0	0.0
8400	0.0	0.0
8600	0.0	0.0
8800	0.0	0.0
9000	0.0	0.0
9200	0.0	0.0
9400	0.0	0.0
9600	0.0	0.0
9800	0.0	0.0
10000	0.0	0.0
10200	0.0	0.0
10400	0.0	0.0
10600	0.0	0.0
10800	0.0	0.0
11000	0.0	0.0
11200	0.0	0.0
11400	0.0	0.0
11600	0.0	0.0
11800	0.0	0.0
12000	0.0	0.0
12200	0.0	0.0
12400	0.0	0.0
12600	0.8	0.8

Figure 3 Histogram of doublet windows



0	8.4	7.7
200	8.2	7.9
400	8.1	7.7
600	7.6	7.6
800	7.8	7.1
1000	7.4	6.6
1200	6.4	6.3
1400	6.1	5.8
1600	5.5	5.3
1800	5.1	4.7
2000	4.4	4.3
2200	3.9	3.8
2400	3.1	3
2600	2.6	2.5
2800	2.2	2.3
3000	1.8	2
3200	1.4	1.6
3400	1.1	1.3
3600	0.8	1
3800	0.7	1
4000	0.5	0.7
4200	0.4	0.6
4400	0.3	0.5
4600	0.2	0.4
4800	0.2	0.5
5000	0.2	0.3
5200	0.1	0.3
5400	0.2	0.3
5600	0.1	0.3
5800	0.1	0.3
6000	0.1	0.2
6200	0.1	0.2
6400	0.1	0.2
6600	0.1	0.2
6800	0.1	0.2
7000	0.1	0.2
7200	0.1	0.2
7400	0.1	0.2
7600	0.1	0.2
7800	0.1	0.1
8000	0.1	0.1
8200	0.1	0.1
8400	0.2	0.1
8600	0.1	0.2
8800	0.2	0.1
9000	0.1	0.1
9200	0.1	0.1
9400	0.1	0.1
9600	0.1	0.1
9800	0.1	0.1
10000	0.1	0.1
10200	0.1	0.1
10400	0.1	0.1
10600	0.1	0.1
10800	0	0.1
11000	0	0.1
11200	0	0.1
11400	0	0
11600	0	0.1
11800	0.1	0.1
12000	0	0.1
12200	0	0
12400	0	0.1
12600	2.3	2.6

Figure 4 Histogram of track windows for first track hit



0	62	12.7
50	19.2	13.5
100	4.1	11
150	1.7	8.5
200	1	6.5
250	0.7	5
300	0.5	3.9
350	0.4	3.3
400	0.3	2.6
450	0.3	2.2
500	0.2	1.9
550	0.3	1.7
600	0.3	1.4
650	0.2	1.2
700	0.2	1.1
750	0.2	1
800	0.2	1
850	0.1	0.9
900	0.2	0.7
950	0.1	0.8
1000	0.1	0.6
1050	0.1	0.6
1100	0.1	0.6
1150	0.1	0.5
1200	0.1	0.5
1250	0.1	0.5
1300	0.1	0.4
1350	0.1	0.4
1400	0.1	0.4
1450	0.1	0.4
1500	0.1	0.3
1550	0.1	0.3
1600	0.1	0.3
1650	0.1	0.3
1700	0.1	0.3
1750	0.1	0.3
1800	0.1	0.3
1850	0.1	0.2
1900	0.1	0.2
1950	0.1	0.2
2000	0.1	0.2
2050	0.1	0.2
2100	0.1	0.2
2150	0.1	0.2
2200	0.1	0.2
2250	0.1	0.2
2300	0.1	0.2
2350	0.1	0.1
2400	0.1	0.1
2450	0.1	0.2
2500	0.1	0.1
2550	0.1	0.1
2600	0.1	0.1
2650	0	0.1
2700	0	0.1
2750	0	0.1
2800	0	0.1
2850	0	0.1
2900	0	0.1
2950	0	0.1
3000	0	0.1
3050	0	0.1
3100	0	0.1
3150	4.6	8.4

Figure 5 Histogram of track windows

