DIALS Dispersion Spotfinding calculation

Starting with an image array, for a given pixel,

0	1	2	0	1	0	1	0	..
0	0	1	0	2	2	1	0	..
1	0	2	1	5	3	4	0	..
0	0	1	5	40	10	2	1	..
2	1	1	99	20	3	0	2	..
3	0	1	3	2	1	0	1	..
3	0	1	2	1	2	3	1	..
0	1	0	3	1	2	1	2	..
..	..	..	..	..	..	..	..	..

We want to calculate the mean, variance of all pixels surrounding it, for a particular kernel width (3x3 here)

0	1	2	0	1	0	1	0	..
0	0	1	0	2	2	1	0	..
1	0	2	1	5	3	4	0	..
0	0	1	5	40	10	2	1	..
2	1	1	99	20	3	0	2	..
3	0	1	3	2	1	0	1	..
3	0	1	2	1	2	3	1	..
0	1	0	3	1	2	1	2	..
..	..	..	..	..	..	..	..	..

x

1	5	3
5	40	10
	20	3

x^2

1	25	9
25	1600	100
	400	9

n

1	1	1
1	1	1
0	1	1

$$X = \sum x \quad = 1+5+3+5+40+10+20+3 \quad = 87$$
$$Y = \sum x^2 = 1+25+9+\ldots+9 \quad = 2169$$
$$N = \sum n \quad = 1+1+1+1+1+1+0+1+1 = 8$$

$$\text{Mean} = \mu = \frac{X}{N}$$
$$\text{Variance} = \sigma^2 = \frac{NY - X^2}{N(N-1)}$$

- Calculating variance this way prevents the need to precalculate the mean, meaning only one pass over the data (at this step)

Further calculations are then done, comparing the mean and variance and the individual pixels value in relation to them.

However, doing this calculation for every pixel involves a lot of redundancy - obviously the next pixel horizontally has some number of overlapping pixels that it re-sums every time, and larger kernels would be slower to calculate.

So, we use summed area table (SAT, aka integral images, aka prefix sum).

We build lookup tables for each of the three variables we need to calculate the mean and variance, each of which describes the inclusive sum of all values to the above and to the left

From the pixel values

	i = 0	1	2	3
j = 0	0	1	2	..
1	0	0	1	..
2	1	0	2	..
3	..	..	..	..

$$X[j,i] = \sum_{n<=j} \sum_{m<=i} x[n,m]$$

e.g. each entry in a row is the sum of the previous items.....

$$\sum x[j=0, i=5] = 0+1+2+0+1+0 = 4$$

	i = 0	1	2	3	4	5	6	7	..
j = 0	0	1	3	3	4	4	5	5	..
	..	..	..	..	..	..	..	..	..

And each row is the sum of the previous rows.....

Row only j = 0	0	1	3	3	4	4	5	5	..
+									
Row only j = 1	0	0	1	1	3	5	6	6	..
+									
Row only j = 2	1	1	3	4	9	12	16	16	..
=									
Final, 2D row j = 2	1	2	7	8	16	21	27	27	..

We iterate over the data, calculating three tables for each of the values that we are reading from:

Pixel Values

0	1	2	0	1	0	1	0	..
0	0	1	0	2	2	1	0	..
1	0	2	1	5	3	4	0	..
0	0	1	5	40	10	2	1	..
2	1	1	99	20	3	0	2	..
3	0	1	3	2	1	0	1	..
3	0	1	2	1	2	3	1	..
0	1	0	3	1	2	1	2	..
..	..	..	..	..	..	..	..	..

$\Rightarrow$

$\sum x$

0	1	3	3	4	4	5	5	..
0	1	4	4	7	9	11	11	..
1	2	7	8	16	21	27	27	..
1	2	8	14	62	77	85	86	..
3	5	12	18	86	104	112	115	..
6	8	16	25	95	114	122	126	..
8	10	19	30	101	122	133	128	..
8	11	20	34	106	129	141	148	..
..	..	..	..	..	..	..	..	..

$\sum x^2$

0	1	5	5	6	6	7	7	..
0	1	6	6	11	15	17	17	..
1	2	11	12	42	55	73	73	..
1	2	12	38	1668	1781	1803	1804	..
5	7	18	44	2074	2196	2218	2223	..
14	16	28	63	2097	2220	2242	2248	..
18	20	33	72	2107	2234	2265	2272	..
18	21	34	82	2118	2249	2281	2292	..
..	..	..	..	..	..	..	..	..

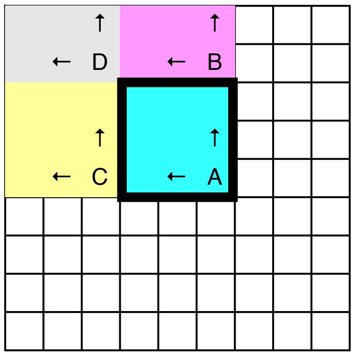
$\sum n$

1	2	3	4	5	6	7	8	..
2	4	6	8	10	12	14	16	..
3	6	9	12	15	18	21	24	..
4	8	12	16	20	24	28	32	..
5	10	15	19	24	29	34	39	..
6	12	18	23	29	35	41	47	..
7	14	21	27	34	41	48	55	..
8	16	24	31	39	47	55	63	..
..	..	..	..	..	..	..	..	..

Now, for a kernel *of any size*, we can calculate the summed area of any of the values we have a table for by fetching the four values at each corner of the kernel:

$$\sum_{\text{kernel}} = A - B - C + D$$

(D's area is subtracted twice by the overlapping areas of B and C, so needs to be added back on)



Thus,

$$\sum x = 104 - 9 - 12 + 4 = 73$$

$$\sum x^2 = 2196 - 15 - 18 + 6 = 2169$$

$$\sum n = 29 - 12 - 15 + 6 = 8$$