

Summed-area table

A **summed-area table** is a data structure and algorithm for quickly and efficiently generating the sum of values in a rectangular subset of a grid. In the image processing domain, it is also known as an **integral image**. It was introduced to computer graphics in 1984 by Frank Crow for use with mipmaps. In computer vision it was popularized by Lewis^[1] and then given the name "integral image" and prominently used within the Viola–Jones object detection framework in 2001. Historically, this principle is very well known in the study of multi-dimensional probability distribution functions, namely in computing 2D (or ND) probabilities (area under the probability distribution) from the respective cumulative distribution functions.^[2]

1.

31	2	4	33	5	36
12	26	9	10	29	25
13	17	21	22	20	18
24	23	15	16	14	19
30	8	28	27	11	7
1	35	34	3	32	6

2.

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

$15 + 16 + 14 + 28 + 27 + 11 =$
 $101 + 450 - 254 - 186 = 111$

Using a summed-area table (2.) of an order-6 magic square (1.) to sum up a subrectangle of its values; each coloured spot highlights the sum inside the rectangle of that colour.

Contents

- The algorithm
- Extensions
- References
- External links

The algorithm

As the name suggests, the value at any point (x, y) in the summed-area table is the sum of all the pixels above and to the left of (x, y) , inclusive.^{[3][4]}

$$I(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$

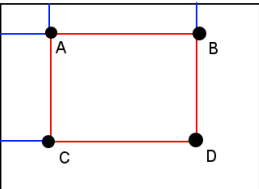
where $i(x, y)$ is the value of the pixel at (x,y) .

The summed-area table can be computed efficiently in a single pass over the image, as the value in the summed-area table at (x, y) is just:^[5]

$$I(x, y) = i(x, y) + I(x, y - 1) + I(x - 1, y) - I(x - 1, y - 1)$$

Once the summed-area table has been computed, evaluating the sum of intensities over any rectangular area requires exactly four array references regardless of the area size. That is, the notation in the figure at right, having $A=(x_0, y_0)$, $B=(x_1, y_0)$, $C=(x_0, y_1)$ and $D=(x_1, y_1)$, the sum of $i(x,y)$ over the rectangle spanned by A, B,C and D is:

$$\sum_{\substack{x_0 < x \leq x_1 \\ y_0 < y \leq y_1}} i(x, y) = I(D) + I(A) - I(B) - I(C)$$



Sum = D - B - C + A

A description of computing a sum in the summed-area table data structure/algorithm

Extensions

- This method is naturally extended to continuous domains.^[2]
- The method can be also extended to high-dimensional images.^[6] If the corners of the rectangle are \mathbf{x}^p with p in $\{0, 1\}^d$, then the sum of image values contained in the rectangle are computed with the formula

$$\sum_{p \in \{0,1\}^d} (-1)^{d - \|\mathbf{p}\|_1} I(\mathbf{x}^p)$$

where $I(\mathbf{x})$ is the integral image at \mathbf{x} and d the image dimension. The notation \mathbf{x}^p correspond in the example to $d = 2$, $A = \mathbf{x}^{(0,0)}$, $B = \mathbf{x}^{(1,0)}$, $C = \mathbf{x}^{(1,1)}$ and $D = \mathbf{x}^{(0,1)}$. In neuroimaging, for example, the images have dimension $d = 3$ or $d = 4$, when using voxels or voxels with a time-stamp.

- This method has been extended to high-order integral image as in the work of Phan et al.^[7] who provided two, three, or four integral images for quickly and efficiently calculating the standard deviation (variance), skewness, and kurtosis of local block in the image. This is detailed below:

To compute variance or standard deviation of a block, we need two integral images:

$$I(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$
$$I^2(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i^2(x', y')$$

The variance is given by:

$$\text{Var}(X) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2.$$

Let S_1 and S_2 denote the summations of block $ABCD$ of I and I^2 , respectively. S_1 and S_2 are computed quickly by integral image. Now, we manipulate the variance equation as:

$$\begin{aligned} \text{Var}(X) &= \frac{1}{n} \sum_{i=1}^n (x_i^2 - 2 \cdot \mu \cdot x_i + \mu^2) = \frac{1}{n} \left(\sum_{i=1}^n (x_i)^2 - 2 \cdot \sum_{i=1}^n (\mu \cdot x_i) + \sum_{i=1}^n (\mu^2) \right) = \frac{1}{n} \left(\sum_{i=1}^n (x_i)^2 - 2 \cdot \sum_{i=1}^n (\mu \cdot x_i) + n \cdot (\mu^2) \right) \\ &= \frac{1}{n} \left(\sum_{i=1}^n (x_i)^2 - 2 \cdot \mu \cdot \sum_{i=1}^n (x_i) + n \cdot (\mu^2) \right) = \frac{1}{n} (S_2 - 2 \cdot S_1/n \cdot S_1 + n \cdot ((S_1/n)^2)) = \frac{1}{n} (S_2 - (S_1)^2/n) \end{aligned}$$

Where $\mu = S_1/n$ and $S_2 = \sum_{i=1}^n (x_i^2)$.

Similar to the estimation of the mean (μ) and variance (Var), which requires the integral images of the first and second power of the image respectively (i.e. I, I^2); manipulations similar to the ones mentioned above can be made to the third and fourth powers of the images (i.e. $I^3(x, y), I^4(x, y)$) for obtaining the skewness and kurtosis.^[7] But one important implementation detail that must be kept in mind for the above methods, as mentioned by F Shafait et al.^[8] is that of integer overflow occurring for the higher order integral images in case 32-bit integers are used.

References

- Lewis, J.P. (1995). *Fast template matching*. *Proc. Vision Interface*. pp. 120–123.
- Finkelstein, Amir; neeratsharma (2010). "Double Integrals By Summing Values Of Cumulative Distribution Function" (<http://demonstrations.wolfram.com/DoubleIntegralsBySummingValuesOfACumulativeDistributionFunction/>). *Wolfram Demonstration Project*.
- Crow, Franklin (1984). "Summed-area tables for texture mapping" (<http://www.soe.ucsc.edu/classes/cms160/Fall05/papers/p207-crow.pdf>) (PDF). *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. pp. 207–212.
- Viola, Paul; Jones, Michael (2002). "Robust Real-time Object Detection" (<http://www.hpl.hp.com/techreports/Compaq-DEC/CRL-2001-1.pdf>) (PDF). *International Journal of Computer Vision*.
- BADGERATI. "Computer Vision – The Integral Image" (<https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/>). *computersciencesource.wordpress.com*. Retrieved 2017-02-13.
- Tapia, Ernesto (January 2011). "A note on the computation of high-dimensional integral images". *Pattern Recognition Letters*. **32** (2). doi:10.1016/j.patrec.2010.10.007 (<https://doi.org/10.1016%2Fj.patrec.2010.10.007>).
- Phan, Thien; Sohoni, Sohumi; Larson, Eric C.; Chandler, Damon M. (22 April 2012). "Performance-analysis-based acceleration of image quality assessment" (http://vision.okstate.edu/pubs/ssiai_tp_1.pdf) (PDF). *2012 IEEE Southwest Symposium on Image Analysis and Interpretation*: 83. doi:10.1109/SSIAI.2012.6202458 (<https://doi.org/10.1109%2FSSIAI.2012.6202458>).
- Shafait, Faisal; Keysers, Daniel; M. Breuel, Thomas (January 2008). "Efficient implementation of local adaptive thresholding techniques using integral images" (<http://www.csse.uwa.edu.au/~shafait/papers/Shafait-efficient-binarization-SPIE08.pdf>) (PDF). *Electronic Imaging*. doi:10.1117/12.767755 (<https://doi.org/10.1117%2F12.767755>).

External links

- Summed table implementation in object detection (<http://www.codeproject.com/Articles/441226/Haar-feature-Object-Detection-in-Csharp>)

Lecture videos

- An introduction to the theory behind the integral image algorithm (<https://www.youtube.com/watch?v=mM5JY-Q6hiM>)
- A demonstration to a continuous version of the integral image algorithm, from the Wolfram Demonstrations Project (<https://www.youtube.com/watch?v=SI117NdjJ8>)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Summed-area_table&oldid=823716734"

This page was last edited on 2 February 2018, at 23:34 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.