

积分图像

一、定义

积分图最早提出于论文 *Rapid Object Detection using a Boosted Cascade of Simple Features*(cvpr2001)中，其作为一种新的图像特征表示方式，用于把检测的 Haar-like 特征快速计算出来，以便用于人脸实时检测的任务。积分图像计算一句话概括：**某个像素点的积分值等于该点像素值和之前所有像素值的和。**

假设我们需要计算图像中某一点的积分值，则需要求解该点区域（包括该点）的像素值之和，需要将所有的元素相加得到结果，如果工程需要多次重复进行求和操作，耗费资源。如果可以先构造一张“积分图”（integral image），也称其为 Summed Area Table。在需要多次重复计算像素值之和的场景，使用该积分图即类似查找表的方式，大大减少资源消耗。

a	b	c
d	e	f
g	h	i

图 1

以 a, b, ..., h, i 表示上图 1 中 9 个图像区域的位置，图 2 为积分前后的图像像素值。

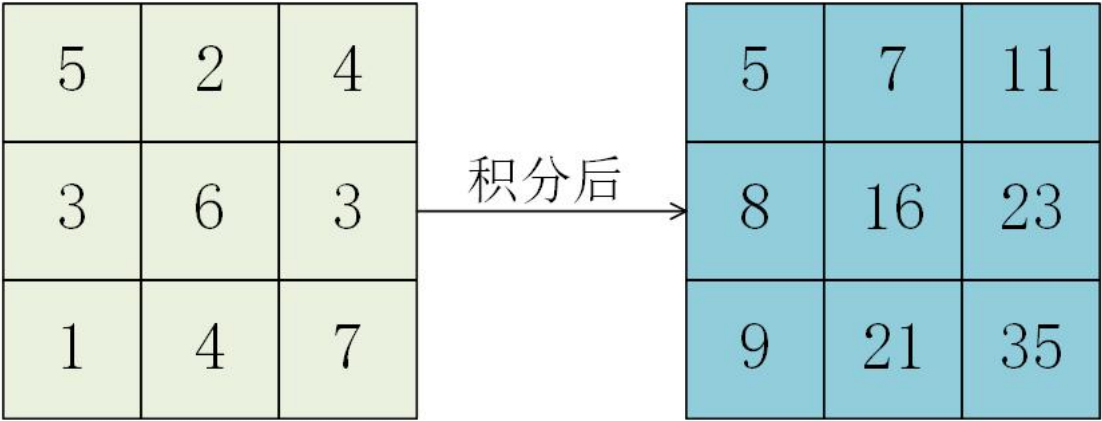


图 2

我们以 F 表示积分后的结果，f 表示积分前的像素值。f(a)，F(a)分别表示 a 位置处的积分前后像素值。

3 个准则：

- (1) 最左列：只在该列进行元素加和；
- (2) 最上行：只在该行进行元素加和；
- (3) 其余位置：该处元素值+以该点所在行和列的左上区域元素值；

比如：F(g) = f(a)+f(d)+f(g)，F(c) = f(a)+f(b)+f(c)，F(e) = f(a)+f(b)+f(d)+f(e)

（相当于该点向上，向左画两条射线，两条射线左上角的区域为需要求解的所有元素值之和）

## 二、通用公式

(1) 假设以一组四邻域作为子区域分析积分图像的通用公式，如图 3 所示：

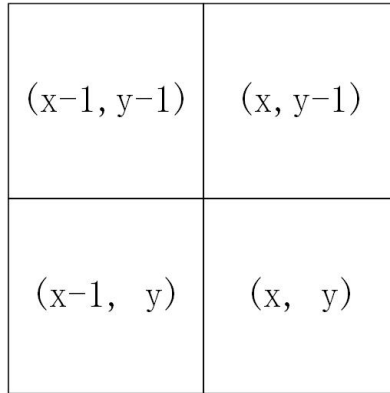


图 3

则通用公式可表示为：

$$F(x, y) = f(x, y) + F(x-1, y) + F(x, y-1) - F(x-1, y-1) \quad \text{公式 1}$$

(2) 或者换一种表达如下图 4：

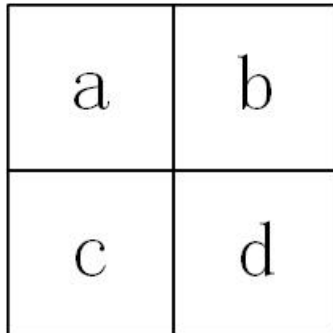


图 4

则 d 处的积分值为：

$$F(d) = \text{sum}(a+b+c+d) - \text{sum}(a+b) - \text{sum}(a+c) + \text{sum}(a) \quad \text{公式 2}$$

当需要求解  $(x, y)$  处的积分值时，仅需要知道和其相邻的三个点的积分结果即可。因此，在图像中，若需要求解某点积分值，需要找到该点相邻的三个点的积分值即可，同理，这三个点的每一个积分值找到其相邻的三个邻域积分值即可，依次迭代。


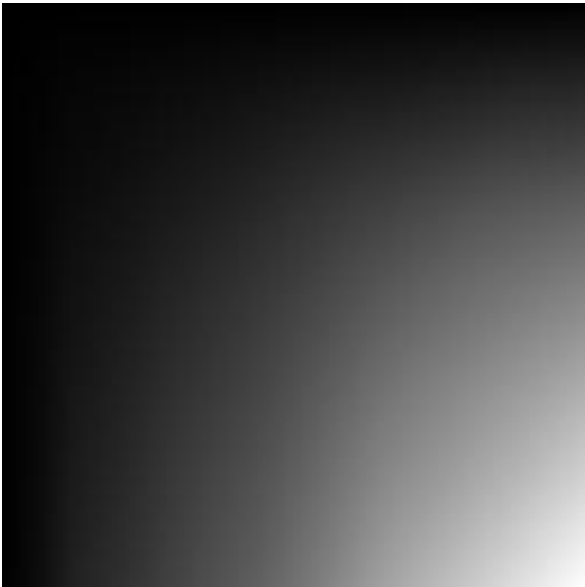


## 三、代码实现

按照上述的图像积分图原理分析，在实际的编程实现过程中对图像进行扩展，左边扩展一列，顶端扩展一行。

1、Opencv 自带计算积分图函数

```
void cv::integral( InputArray src, OutputArray sum, OutputArray sqsum, OutputArray  
tiled, int sdepth = -1, int sqdepth = -1 )
```

src: 输入图像  $W \times H$  8 位浮点数或者 32f, 64f;  
sum: 积分图像为 $(W + 1) \times (H + 1)$ , 类型为 32 位或 64 位浮点数;  
sqsum: 平方像素值的积分图像, 它是 $(W + 1) \times (H + 1)$ 双精度 64f 数组;  
titled: 旋转 45 度的图像积分, 它与 sum 相同数据类型;  
sdepth: 期望的积分深度和倾斜的积分图像, CV\_32S, CV\_32F 或 CV\_64F;  
sqdepth: 要求的深度的积分图像的平方像素值, CV\_32F 或 CV\_64F;

	
	
原始图	积分后归一化图