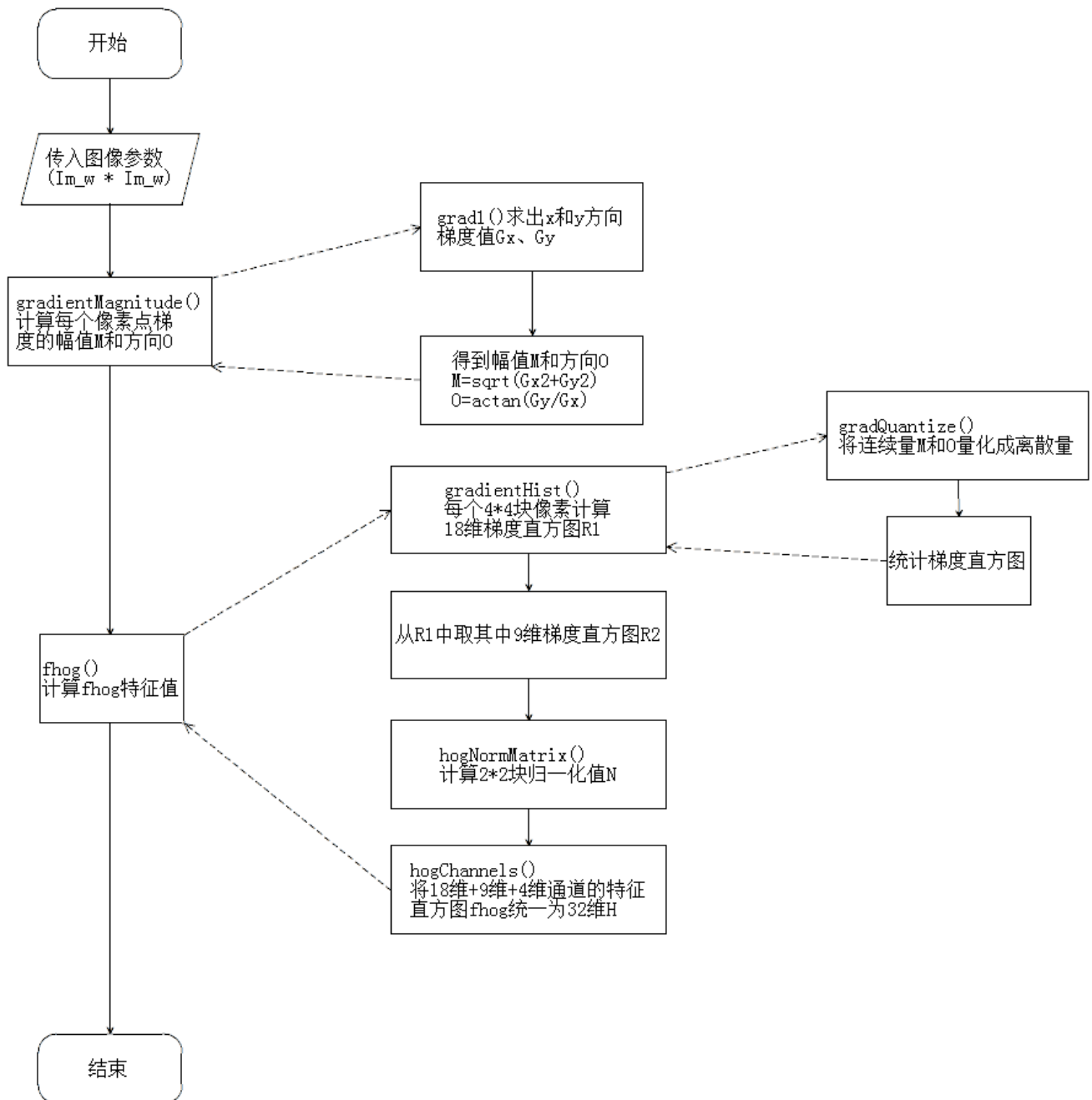


FHOG多通道融合特征提取学习笔记---qw2017.5.9

程序流程（.c文件中的子函数）



1. 梯度计算

边缘是由图像局部特征包括灰度、颜色和纹理的突变导致的。一维离散微分模板在将图像的梯度信息简单、快速且有效地计算出来，其公式如下：（函数`grad1()`作用）

$$\begin{cases} G_x(x, y) = H(x+1, y) - H(x-1, y) \\ G_y(x, y) = H(x, y+1) - H(x, y-1) \end{cases}$$

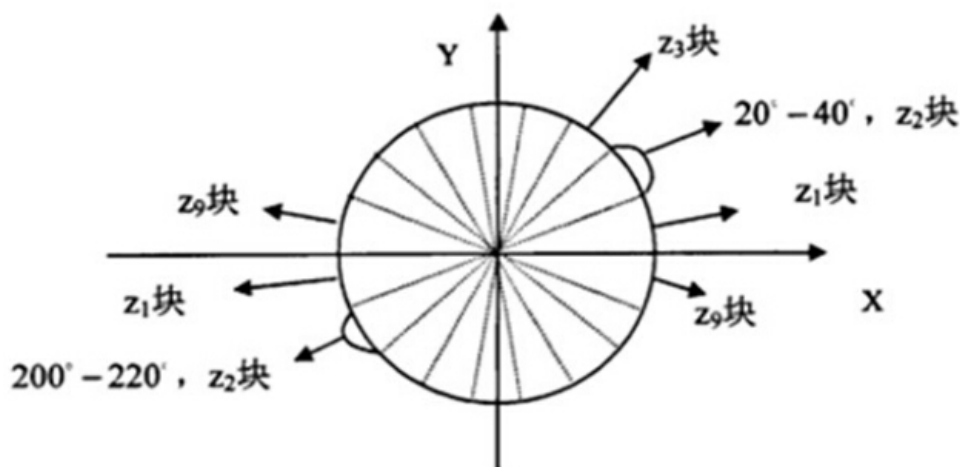
式中， G_x ， G_y ， $H(x, y)$ 分别表示的是像素点 (x, y) 在水平方向上及垂直方向上的梯度以及像素的灰度值，其梯度的幅值及方向计算公式如下：（函

数gradientMagnitude()作用)

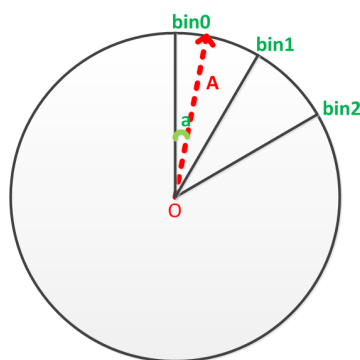
$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y(x, y)}{G_x(x, y)} \right)$$

2.梯度方向量化

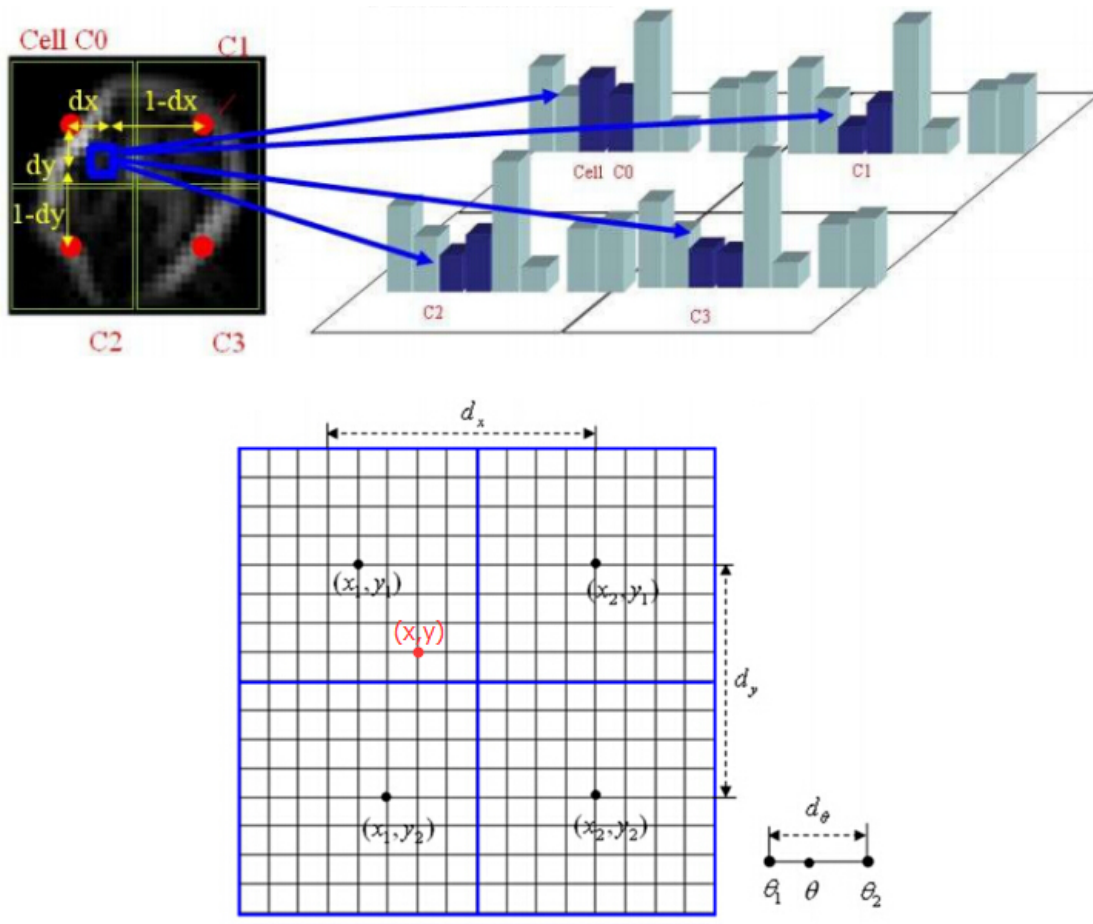


因为这里的梯度和角度都是用到了二线插值的。每一个点的梯度角度可能是0~360度之间的任意值，而程序中将其离散化为18个bin，即每个bin占20度。所以滑动窗口中每个像素点的梯度角度如果要离散化到这18个bin中，则一般它都会有2个相邻的bin(如果恰好位于某个bin的中心，则可认为对该bin的权重为1即可)。程序中量化有两种选择，1使用临近两个bin双线性插值如下图，2是取最近的bin，程序中参数选择决定了使用的是第二种方式。源码中可以看到梯度的幅值是用来计算梯度直方图时权重投票的，每个像素点的梯度幅值M作了均衡化，M_new等于M*1/16。（函数gradQuantize()作用）



3.梯度直方图统计

（3Tri-linear interpolation三线性插着）在计算每个cell的梯度直方图时，可以用三线性插值来提高计算速率。对于每个cell里的点，我们认为都是一个三维向量(x,y,θ)，如下图所示某一待处理像素点它位于block中的C0单元中，利用该点与四个cell中的中心像素点（图中4个圆点，(x₁,y₁),...,(x₄,y₄)）的距离计算权值，将待处理像素点的梯度幅值分别加权累加到C0、C1、C2、C3中相应的直方图上，与θ相邻的两个bin上(θ₁, θ₂)。（函数gradientHist()作用）



参见上图，以 (x,y,θ) 对cell0的梯度直方图即 $h(x_1,y_1,\theta_1)$ 的加权投影为例，三线性插值公式如下：

$$h(x_1,y_1,\theta_1) \leftarrow h(x_1,y_1,\theta_1) + |\nabla f(x,y)| \left(1 - \frac{x-x_1}{d_x}\right) \left(1 - \frac{y-y_1}{d_y}\right) \left(1 - \frac{\theta(x,y)-\theta_1}{d_\theta}\right), \quad (1)$$

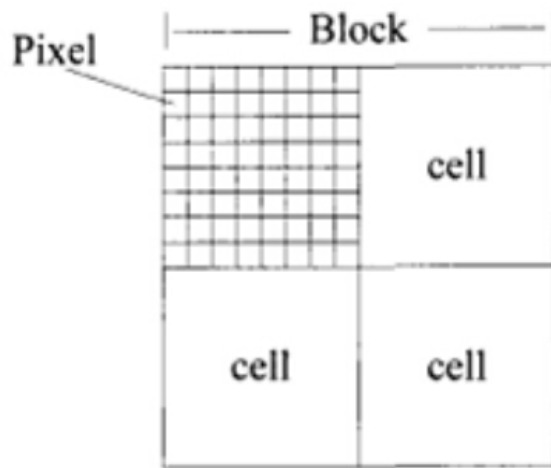
$$h(x_1,y_1,\theta_2) \leftarrow h(x_1,y_1,\theta_2) + |\nabla f(x,y)| \left(1 - \frac{x-x_1}{d_x}\right) \left(1 - \frac{y-y_1}{d_y}\right) \left(\frac{\theta(x,y)-\theta_1}{d_\theta}\right). \quad (2)$$

where $d_x=x_2-x_1$, $d_y=y_2-y_1$, and $d_\theta=\theta_2-\theta_1$. $h(x_1,y_2,\theta_1)$, $h(x_1,y_2,\theta_2)$, $h(x_2,y_1,\theta_1)$, $h(x_2,y_1,\theta_2)$, $h(x_2,y_2,\theta_1)$ and $h(x_2,y_2,\theta_2)$ can be computed in the similar way.

这样不断累加更新得到最终的 $h(x_1,y_1,\theta_1), \dots, h(x_4,y_4,\theta_4)$ ，就是我们所要4个cell的梯度直方图。

4.块归一化处理 Normalization and Descriptor Blocks

由于局部光照的变化以及前景-背景对比度的变化，使得梯度强度的变化范围非常大。这就需要对梯度强度做归一化。归一化能够进一步地对光照、阴影和边缘进行压缩，使特征向量空间对光照，阴影和边缘变化具有鲁棒性。本文的做法是：把各个细胞单元组合成大的、空间上连通的区域（blocks）。这样，一个block内所有cell的特征向量串联起来便得到该block的HOG特征。这些区间是互有重叠的，这就意味着：每一个单元格的特征会以不同的结果多次出现在最后的特征向量中。归一化之后的块描述符（向量）就称之为HOG描述符。归一化单位都是Block（即块）。



如上图，一般一个块（Block）都由若干单元（Cell）组成，一个单元都有若干个像素点组成。

假设行人检测的参数设置是：2×2 cell/block、8×8像素/cell、9个直方图通道（9 bins），一个块的特征向量长度为：2×2×9。

对block块内的HOG特征向量进行归一化。一般采用的归一化函数有以下四种：（函数作用）

- (a) $L2-norm, v \leftarrow v / \sqrt{\|v\|_2^2 + \varepsilon^2}$ (ε 是一个极小的常数，避免分母为 0)
- (b) $L2-Hys$ (方法同上，只是限制 v 的最大值到 0.2，然后再次重新标准化)
- (c) $L1-norm, v \leftarrow v / \sqrt{\|v\|_1 + \varepsilon}$
- (d) $L1-sqrt, v \leftarrow v / \sqrt{\|v\|_1 + \varepsilon}$

Dalal的文章也验证了对于人体检测系统使用L2-norm的时候效果最好，**程序中进行HOG计算时使用L2-norm。**

4.DPM（DeformablePart Model可变性模型）及维度计算

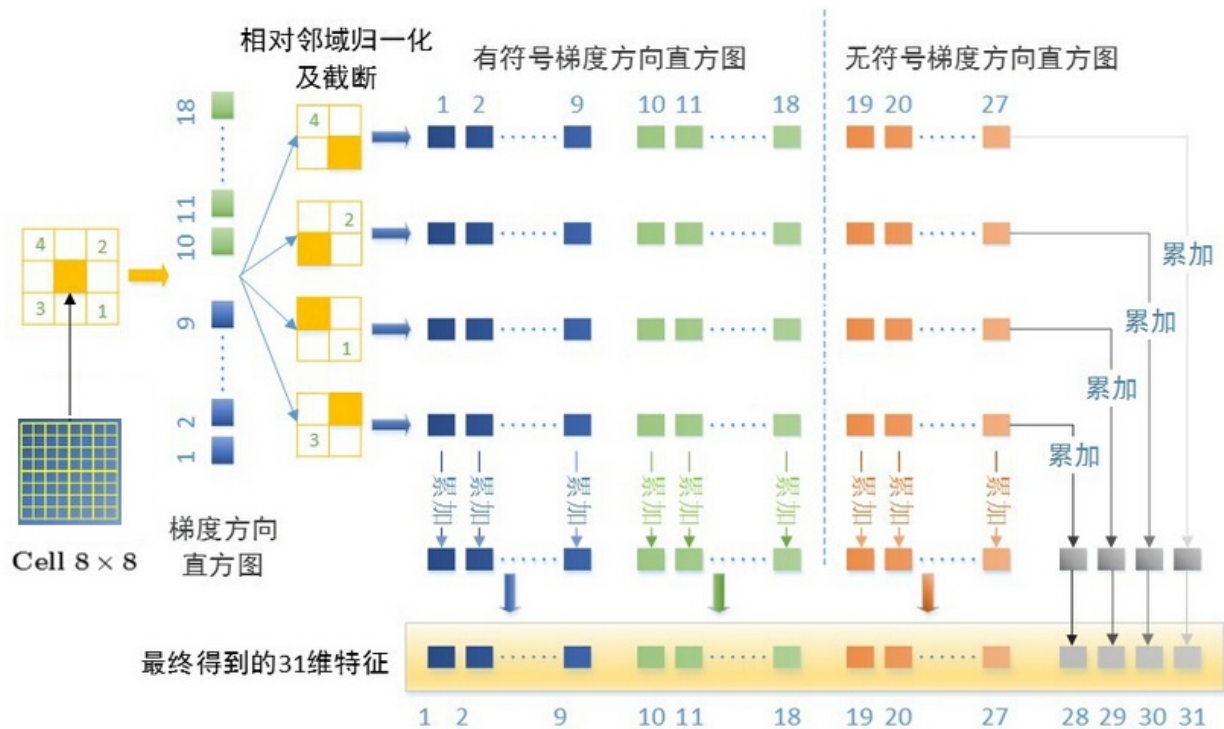
DPM检测流程：

对于任意一张输入图像，提取其DPM特征图，然后将原始图像进行高斯金字塔上采样，然后提取其DPM特征图。对于原始图像的DPM特征图和训练好的Root filter做卷积操作，从而得到Root filter的响应图。对于2倍图像的DPM特征图，和训练好的Part filter做卷积操作，从而得到Part filter的响应图。然后对其精细高斯金字塔的下采样操作。这样Root filter的响应图和Part filter的响应图就具有相同的分辨率了。然后将其进行加权平均，得到最终的响应图。亮度越大表示响应值越大。

维度计算：

DPM首先采用的是HOG进行特征的提取，但是又有别于HOG，DPM中，只保留了HOG中的Cell。如下图所示，假设，一个8*8的Cell，将该细胞单元与其对角线临域的

4个细胞单元做归一化操作。提取有符号的HOG梯度，0-360度将产生18个梯度向量，提取无符号的HOG梯度，0-180度将产生9个梯度向量。首先，只提取无符号的HOG梯度，下图中右侧橙色部分，将其看成一个4*9的矩阵，分别将行和列分别相加，最终将生成4+9=13个特征向量，为了进一步提高精度，将提取的18维有符号的梯度特征也加进来，如下图中左侧蓝色绿色部分，这样一共产生13+18=31维梯度特征，实现了很好的目标检测。



4.程序数据流

1: void gradientMagnitude(float *I, float *M, float *O, int h, int w, int d, bool full);

%输入: I图像, M梯度幅值矩阵首地址, O梯度方向矩阵首地址, h, w图像高宽, d彩色图像或灰度图, full参数代表方向bin数如果为1则取0-2PI, 0则取0-PI。

%输出: M算出的幅值 $\sqrt{G_x^2 + G_y^2}$, O算出的方向 $(\arctan(G_y/G_x))$ 。

2: 函数1调用此表指针float* acosTable(void);

%通过查表法求出反正切值, 此处为一个数组指针。

3: 函数的1结果传入此函数void fhog(float *M, float *O, float *H, int h, int w, int binSize, int nOrients, int softBin, float clip);

%输入: M梯度幅值, O梯度方向, H梯度方向直方图首地址, h, w图像高宽, binSize细胞大小, nOrients通道维度, softBin表示选择梯度幅值加权形式, 本程序采用三线性插值, clip最终剪辑拼接直方图的选择参数。

%输出: 特征直方图H (32维)

4: 函数3调用此函数void gradientHist(float *M, float *O, float *H, int h, int w, int bin, int nOrients, int softBin, bool full);

%输入：M梯度幅值，O梯度方向，H梯度方向直方图首地址，h、w图像高宽，binSize细胞大小，nOrients通道维度，softBin表示选择梯度幅值加权形式，本程序采用三线性插值，full选择方向bin数。

%输出：H计算出的方向统计直方图，程序中为18维矩阵R1。

5：函数4调用此函数void gradQuantize(float *O, float *M, int *O0, int *O1, float *M0, float *M1, int nb, int n, float norm, int nOrients, bool full)

%输入：O梯度方向，M梯度幅值，O0、O1量化后的方向首地址，M0、M1量化后的幅值首地址，nb宽高乘积，n一行高度，norm幅值均衡化因子，nOrients通道维度，full选择方向bin数。

%输出：O0、O1量化后的方向，M0、M1量化后的幅值。

5：函数3调用此函数float* hogNormMatrix(float *N, float *H, int nOrients, int hb, int wb, int bin)

%输入：N块归一化首地址，H梯度方向直方图(程序中为9维矩阵R2),nOrients通道数，hb、wb块的高宽，bin几个细胞组成一个块。

%输出：N归一化后的梯度方向直方图。

6：函数3调用此函数void hogChannels(float *H, const float *R, const float *N, int hb, int wb, int nOrients, float clip, int type)

%输入：H 32维特征矩阵首地址，R低维度梯度方向直方图，N归一化梯度方向直方图，hb、wb块的高宽，nOrients通道维度，直方图bin剪辑组合选择参数。

%输出：H组合后的梯度方向直方图。