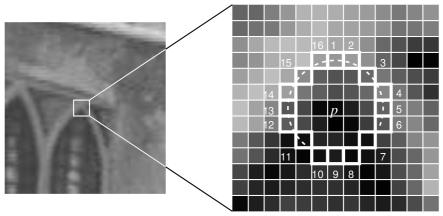
**ORB-SLAM2 庖丁解牛之A-1-(2)**

上一回说到ORB Extraction，而首先就是要对FAST角点进行提取。

* 先理论基础：



1、选择图中的一个像素点，亮度为Ip；

2、选择近似的阈值t；

3、以待检测像素为圆心，3为半径，做一个圆，与圆相交的共有16个像素，检测这16个像素中与中心点的像素差大于某个阈值T时n +1，，若这16个点检测完成时n>N（N一般取11或者9）则认为该待检测像素为角点，然后循环这个过程检测没一个像素。

4、通常情况下，还会进行预测试来加快检测步伐，即检测1,5,9,13这四个点，如果有其中3个与中心像素值的差大于阈值T，而进一步进行检查，否则认为该点不是角点，同时还有极大值检测的方式，防止角点扎堆现象。

一般来说，非最大抑制方法步骤是：

A、对所有检测到的特征点 计算得分函数V。V 是p 和16个周围的像素值 之间的绝对差之和。

B、考虑两个相邻的关键点并计算它们 V 。

C、丢弃较低 V 值的特征点。

*参考：https://docs.opencv.org/master/df/d0c/tutorial\_py\_fast.html*

* 后工程实践：

在上文（记得放链接）当中，函数ORBextractor::operator()调用的内容并没有之间体现FAST角点这一步，但封装在哪里了呢？答案是：ComputeKeyPointsOctTree。FAST检测被封装在该函数体中。

FAST声明是在opencv2/features2d.hpp中，是在ORBextractor.cc中这么被调用的：

FAST(mvImagePyramid[level].rowRange(iniY,maxY).colRange(iniX,maxX),

//待检测的图像，这里就是当前遍历到的图像块

vKeysCell,

//存储角点位置的容器

iniThFAST,

//检测阈值

true

//使能非极大值抑制

);

OpenCV官网的定义是这样的：

void FAST(InputArray image, vector<KeyPoint>& keypoints, int threshold, bool nonmaxSuppression=true )

Parameters:

image – grayscale image where keypoints (corners) are detected.

keypoints – keypoints detected on the image.

threshold – threshold on difference between intensity of the central pixel and pixels of a circle around this pixel.

nonmaxSuppression – if true, non-maximum suppression is applied to detected corners (keypoints).

type – one of the three neighborhoods as defined in the paper: FastFeatureDetector::TYPE\_9\_16, FastFeatureDetector::TYPE\_7\_12, FastFeatureDetector::TYPE\_5\_8

*参考：https://docs.opencv.org/2.4/modules/features2d/doc/feature\_detection\_and\_description.html*