



计算金字塔图像，金字塔的上层是下层按照`scaleFactor`（默认=1.2）进行降低采样得到的，不过在实现时不是基于下一层降采样得到，而是直接从原始图像进行降采样得到（这里可以进行优化）

而是代码中使用了一个技巧，金字塔每一层的图像是原始图像进行降低采样并进行边界扩充后，取其子图像后得到的。这样子就不需要担心坐标越界的问题了。

将金字塔每层的图像均匀分割成大小相同的图像块（图像块默认是30x30的大小，但是由于不整除的原因，实际大小在30x30~59x59之间波动），然后对每个图像块调用opencv的FAST函数提取FAST特征。

由于要限制每层特征点的数量，所以如果提取出来的特征点超过了每层所需要的特征点的数量，那么就要进行特征点的删除，那么该如何删除呢？当然是使得删除之后，剩下的特征点尽可能均匀地覆盖整张图片。所以ORB-SLAM使用四叉树来决定删除哪些点

对当前金字塔层的图像进行四叉树划分，四叉树的每个结点管理在该范围内的特征点。不管对图像进行划分，直到四叉树的叶子结点数量满足所需特征点的数量，然后在每个叶子结点所管理的特征点中保留响应值最大那个特征点

对于保留下来的特征点进行角度计算，具体是使用灰度质心法来计算角度。

根据特征点的方向计算其ORB描述子，从而实现了旋转不变性

在降采样的图像上计算特征点的描述子，从而实现了尺度不变性。



尺度不同的图像，计算特征点描述子时，需要使用不同大小的Patch来计算特征描述子，在降采样的图像上计算特征点的描述子，正好解决了这个问题。