

1.按此文的分类，光流法可分为哪几类？

可将算法分类为加性算法(additive)或组合法(compositional)，以及正向算法(forwards)或逆算法(inverse)

2.在 compositional 中，为什么有时候需要做原始图像的 wrap？该 wrap 有何物理意义？

需要在当前位姿估计之前引入增量式 warp (incremental warp) 以建立半群约束要求 (the semi-group requirement)。

3.forward 和 inverse 有何差别？

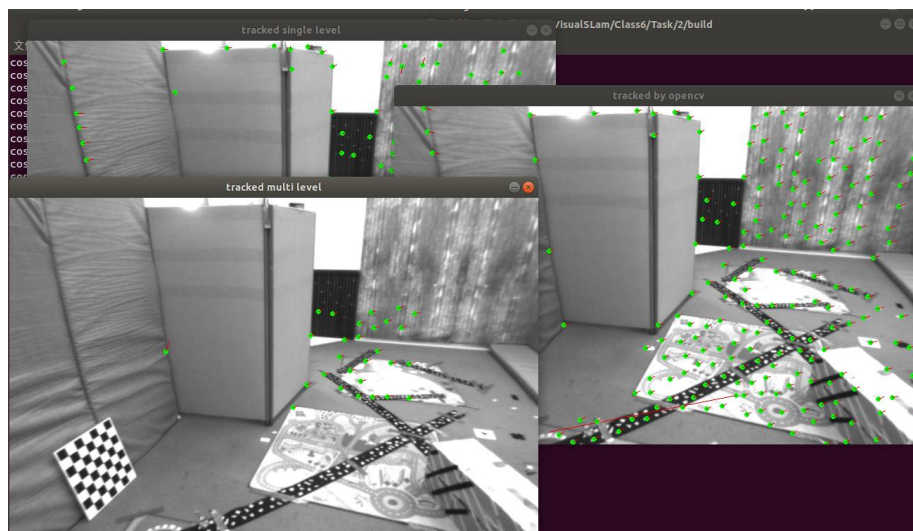
向前方法对于输入图像进行参数化(包括仿射变换及放射增量)。反向方法则同时输入图像参数和模板图像，其中输入图像参数化仿射变换，模板图像参数化仿射增量。因此反向方法的计算量显著降低，计算效率较高。由于图像灰度值和运动参数非线性，整个优化过程为非线性。

向前方法和反向方法在目标函数上不太一样，一个是把运动向量  $p$  都是跟着  $I$  (被匹配图像)，但是向前方法中的迭代的微小量  $\Delta p$  使用  $I$  计算，反向方法中的  $\Delta p$  使用  $T$  计算，这样计算量便小了

利用 OpenCV 中的 `cv::calcOpticalFlowPyrLK()` 函数求解光流结果

单层图像的 LK 光流

多层图像的 LK 光流



1.所谓 coarse-to-fine 是指怎样的过程？

由粗到精的过程，以原始图像作为图像金字塔的底层，每往上一层，对下层图像进行缩放，计算光流时，先从顶层图像开始计算，将上一层的结果作为下一层的初始值。

2.光流法中的金字塔用途和特征点法中的金字塔有何差别？

光流法中的图像金字塔通过逐层迭代寻找最佳的关键点位置和光流方向，目的解决光流在快速运动中难以检测的问题

特征点法中的图像金字塔通过检测每一层上的角点来增添尺度的描述，解决了特征点的尺度不变性。

- 我们优化两个图像块的灰度之差真的合理吗？哪些时候不够合理？你有解决办法吗？

光流法基于灰度不变假设，在缓慢移动时，可以粗略视为灰度不变，是合理的，但是当运动过快的时候，灰度变化大，光流跟踪不了，无法满足基本假设。

可以建立光流金字塔

- 图像块大小是否有明显差异？取 16x16 和 8x8 的图像块会让结果发生变化吗？

建立图像金字塔时，窗口固定，在每一层金字塔上都用同样大小的窗口进行计算所以图像块大小不会有明显差异

当不使用图像金字塔时，设置窗口时，窗口大光流鲁棒性好，窗口小光流精确性好。

- 金字塔层数对结果有怎样的影响？缩放倍率呢？

金字塔层数一般越多越好，但是在 4-5 层之后，图像会变得很小，特征点像素太过紧密容易引起误追踪。

放大倍率小，金字塔层数增加，迭代层数增加，效果变好

### 3. 直接法

1. 直接法是否可以类似光流，提出 inverse, compositional 的概念？它们有意义吗？

直接法通过求解相机运动来计算第二帧图像对应关键点的位置，如果提出 inverse, compositional 的概念，第二帧图像的信息将会失效，没有意义。

2. 请思考上面算法哪些地方可以缓存或加速？

加窗口部分，合适大小的窗口，会使计算加速

3. 在上述过程中，我们实际假设了哪两个 patch 不变？

灰度不变假设，同窗口内深度不变假设

4. 为何可以随机取点？而不用取角点或线上的点？那些不是角点的地方，投影算对了吗？

直接法对灰度值优化，不是角点也没有关系

