1 光流法的分类主要有：

Forwards Additive

Forwards Compositional

Inverse Additive

Inverse Compositional

2 因为**compositional只能用于warps that form groups.**

而FCIA: warp 集合包含 identitywarp, warp 集合包含在 Composition 操作上是闭

的(semi-group), 其中包括 Homograph, 3D rotation 等。

ICIA: semi-group, 另外要求增量 warp 可逆, 其中包括 Homograph, 3D rotation

等, 但不包括 piece wise affine。

而wrap的物理意义在于 对图像做 微小的平移或者放射变换。

3前向与后向对比

前向方法对于输入图像进行参数化(包括仿射变换及放射增量)。后向方法则

同时参数输入图像和模板图像，其中输入图像参数化仿射变换，模板图像参数参

数化仿射增量。因此后向方法的计算量显著降低。

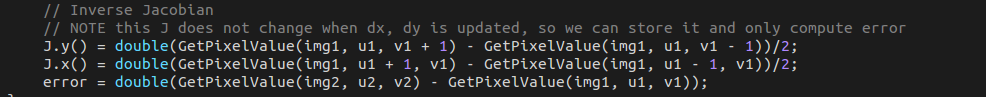
前向方法和后向方法在目标函数上不太一样，一个是把运动向量**p**都是跟着I（被匹配图像），但是前向方法中的迭代的微小量Δ**p**使用I计算的，后向方法中的Δ**p**使用T图像计算的。因此计算雅克比矩阵的时候，一个的微分在Δ**p**处，而另外一个在0处。

2.2 根据lucas-kanade的 第五页的内容，可以定义如下：

**I (W(x; p + delta（p）)) − T (x)**

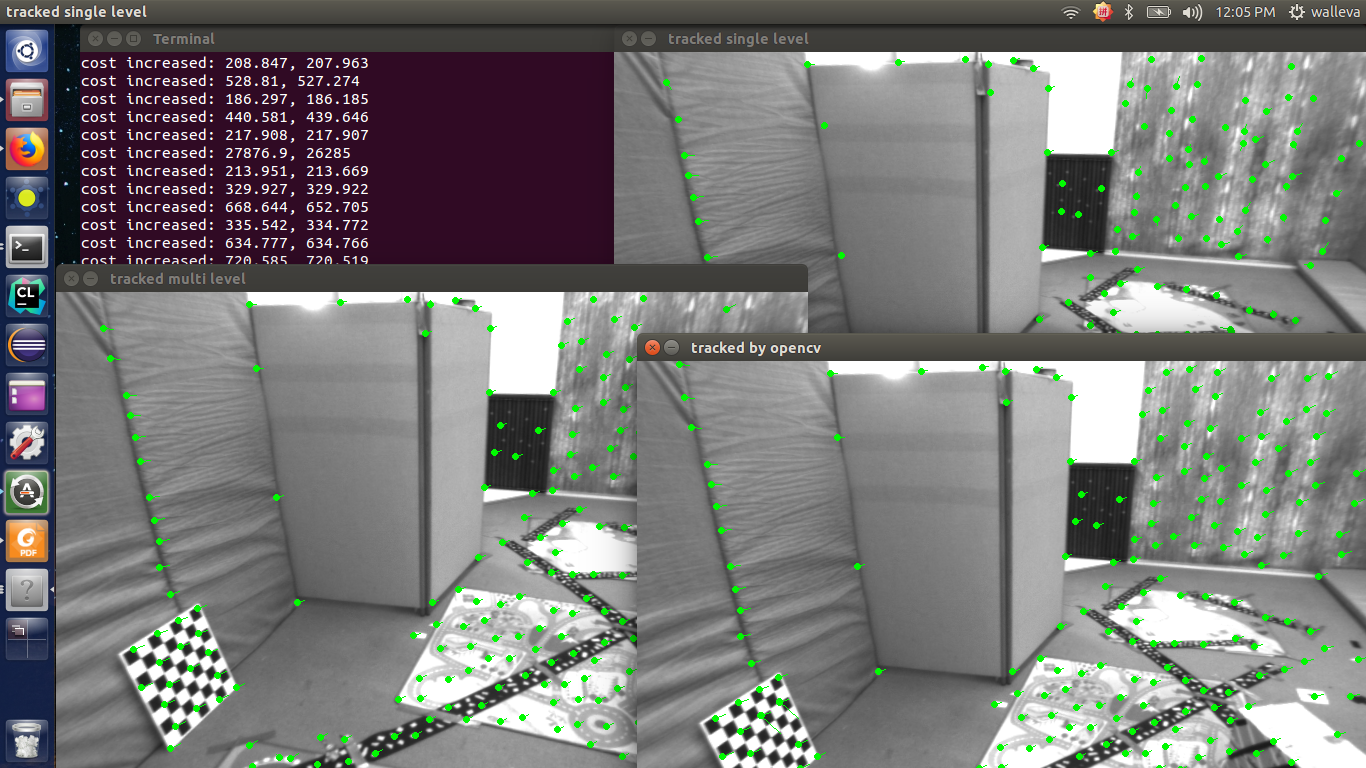
关于子变量的导数定义如下： ∇I = ( ∂I/∂x , ∂I/∂y )

**2.3 反向法：**

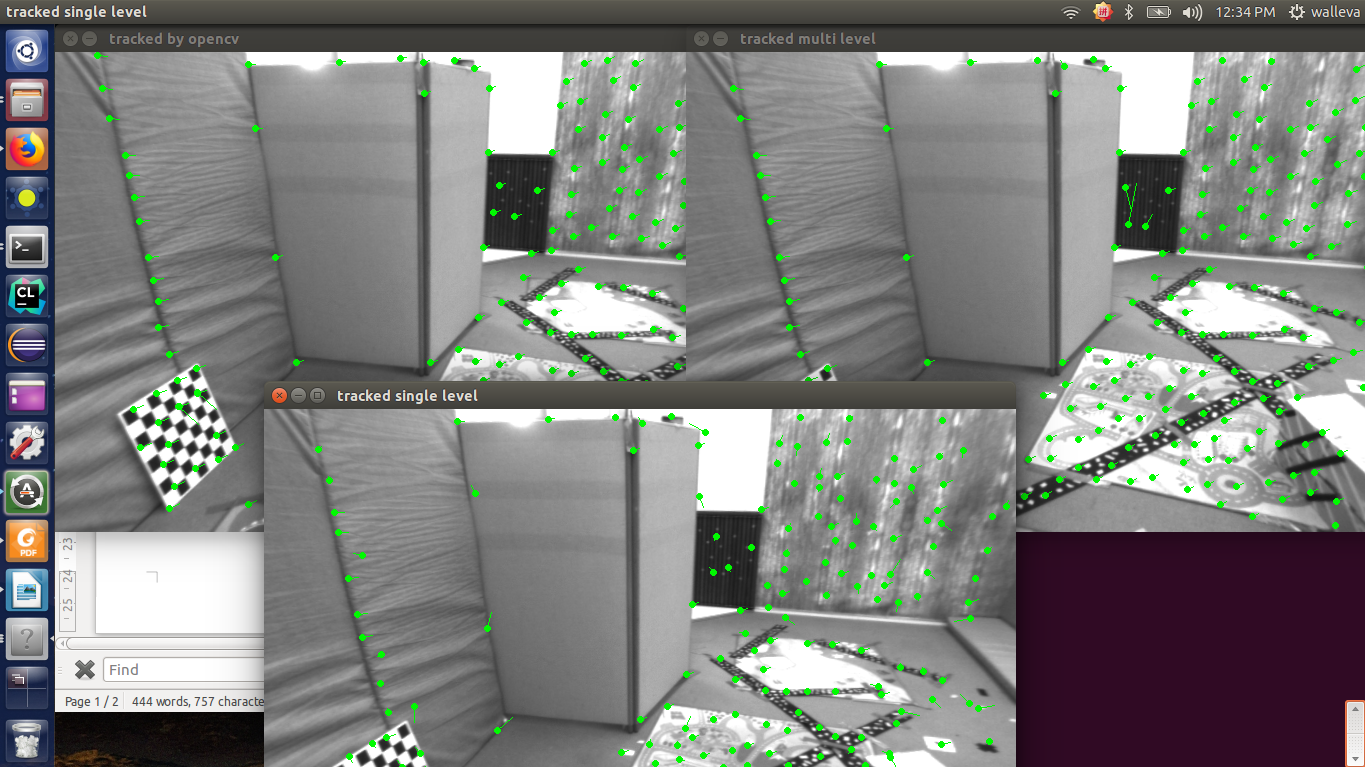
I1(xi; yi) 处的梯度不随迭代改变，所以只需计算一次，就可以在后续的迭代中一直使用，节省了大量计算时间。 （超棒的思路）

**2.4 实验结果：**

反向法：



正向法：



1 coarse to fine，就是 将图片缩放，然后计算出最优的点，然后传递给下一层，再进行估计，以此叠加到原图片的那一层。

2 光流中的金字塔的目的 主要是为了 抑制因为图像运动过大过快，导致无法跟踪，因此可以采用corse to fine的方法来达到更好的跟踪效果。而之前所说的特征点中运用金字塔是为了避免一个特征点中的角点， 由于射影变换，可能近看就不是了，为了达到尺度不变性，那么利用金字塔的方法来跟踪的话， 效果就比较好。

**2.5 讨论：**

1 不太合理， 主要是假设光度不变性， 光流必须要求相机的运动是微小的，所以健壮性偏差，很容易受到突变扰动和 曝光的影响！ 对于突然的亮光造成的跟踪问题，可以在硬件上 采用 偏光镜， 可以克服亮光带来的影响/。

2 图像块的话，从16\*16到8\*8的话，感觉实验的差异不是很明显。

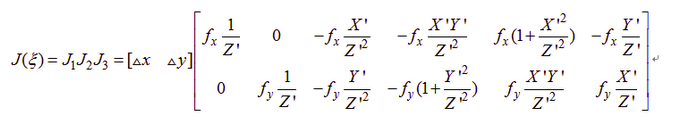
3 金字塔的层数如果降低，会有影响，增加的话，影响不明显， 缩放倍率的话， 降低之后，影响不是很明显。

**直接法：**

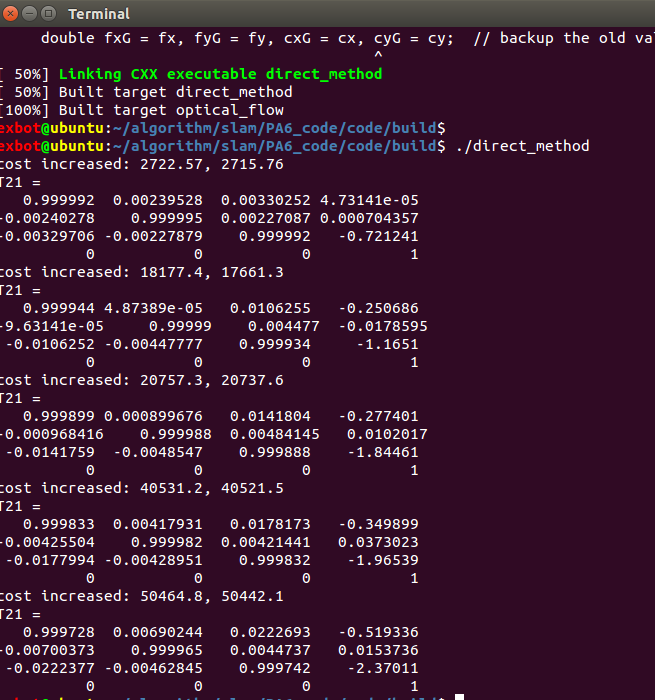
1 该问题的误差项为， **Iref** (π(pi)) − **Icur** (π (Tcur;ref pi))

2 最终的jacobean矩阵的维度 为 6\*1 ，

由图像的像素的梯度的梯度矩阵 乘以 第二副图相机坐标系下的三维坐标相对于李代数的求导矩阵

3 窗口可以取单个点。

单个点的单层效果如下：





4 图像的点只要具有梯度，直接法正常也可以估计运动状态。所以可以不用选取角点。

**多层直接法：**

1 可以像光流法那样 进行 inverse和compositional，有意义， 我觉得能很大程度减少代码的运行时间，而得到更优化的结果。

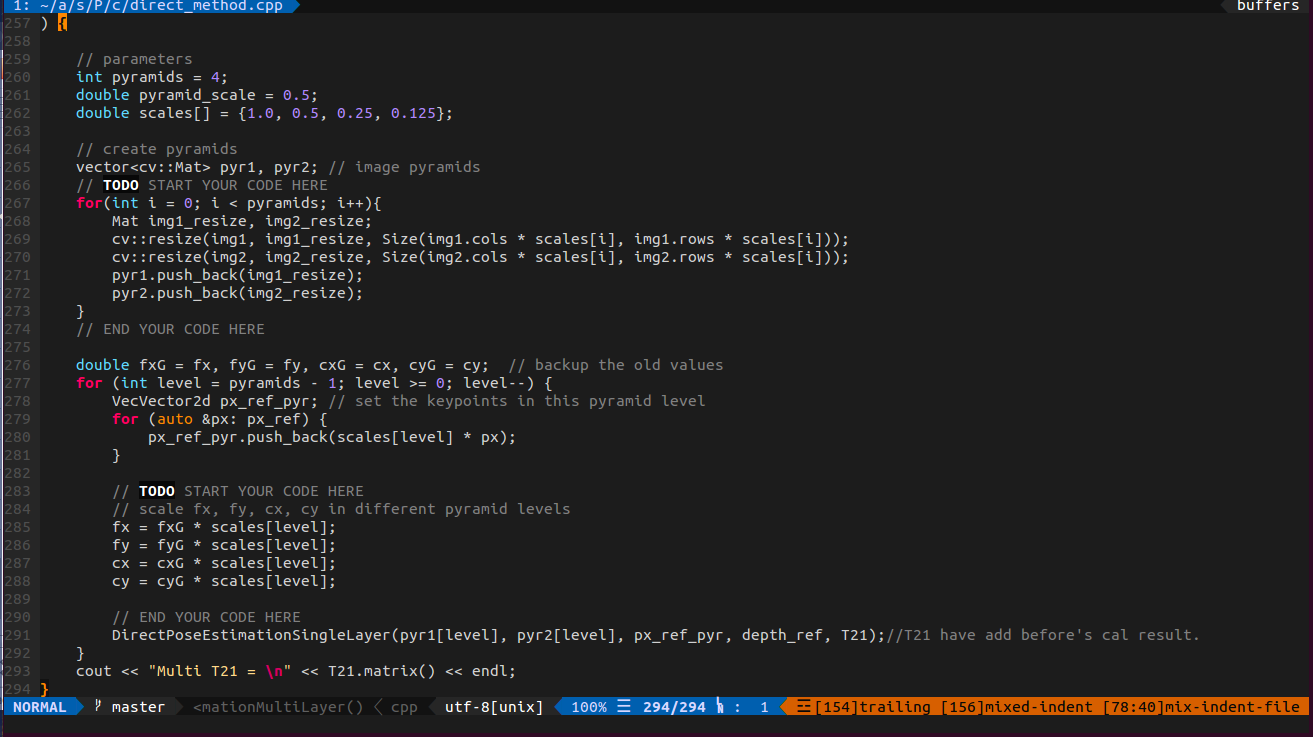
2 加速的话，因为 每次的缩小的比例都是 1/2, 那么 可以通过移位 来加速运算？

2 8\*8的的窗口尺寸不变， 灰度梯度3\*3 不变。

4 直接法中 只要大部分情况下，这个点有梯度，那么就可以求出来，而且 在明暗相近的 场景中，比特征点的优势更大。

5 特征点：需要找匹配点，不容易丢失， 但是浪费时间。

直接法，对于有梯度的点，既可以进行相对应的计算匹配， 但是因为 图像非凸。

代码如下：

得到的结果如下：

