2.LK光流

2.1 光流文献综述

1.光流可以分为哪几类？

forward additive algorithm
forward compositional algorithm
inverse additive algorithm
inverse compositional algorithm

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？

The incremental update to the wrap $W(x;\Delta p)$ must be composed with the current estimate fof the wrap $W(x;p)$ in the compositional algorithm.

3.forward和inverse有何差别？

As a number of authors have pointed out, there is a huge computational cost in re-evaluating the Hessian in every iteration of the LK algorithm. If the Hessian were constant it could be precomputed and then re-used.

2.2 forward-addtive Gauss-Newton光流的实现

1.从最小二乘角度来看，每个像素的误差怎么定义？

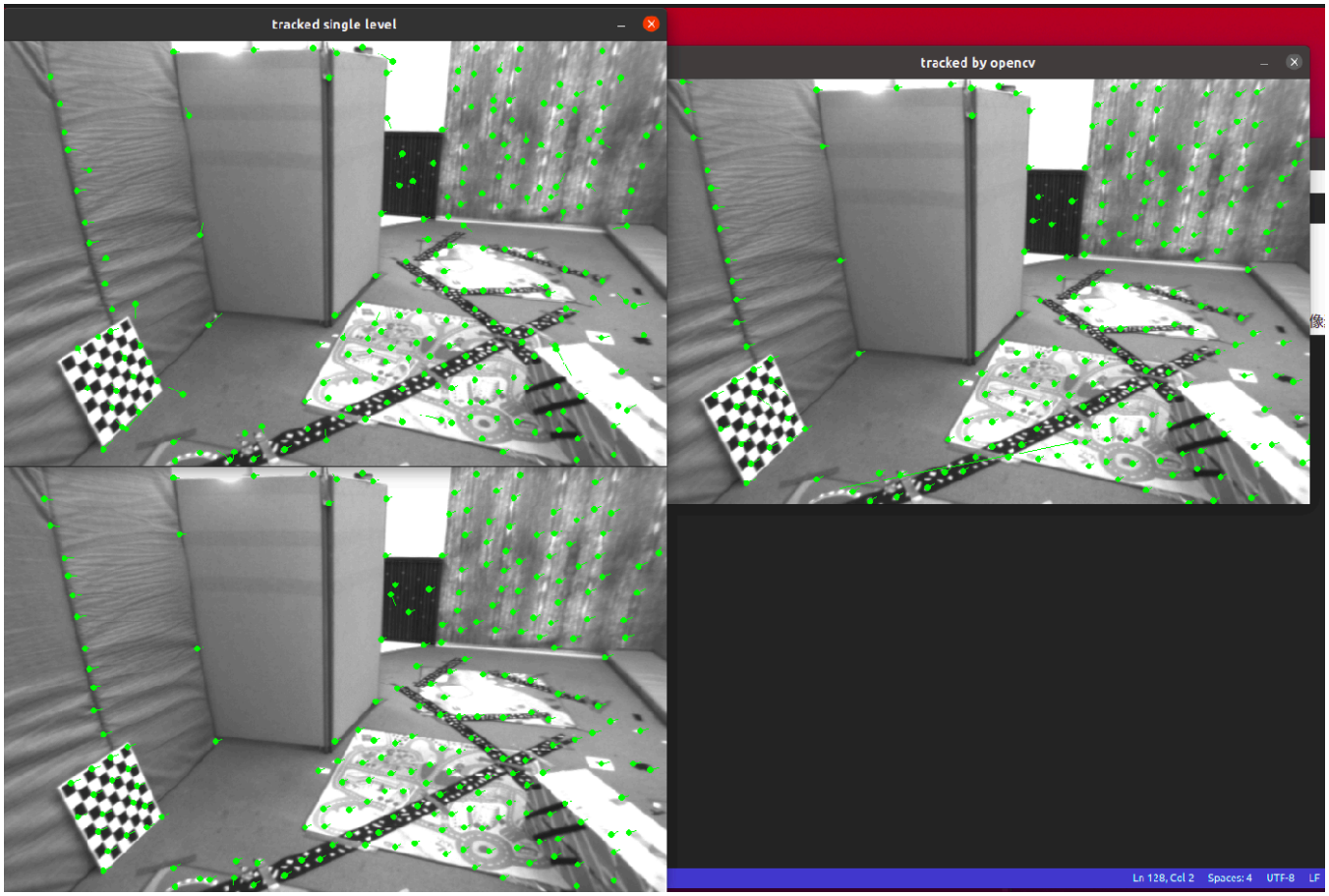
第一张图像在特征点(x,y)的像素值减去第二张图像在$(x+dx,y+dy)$处的像素值

$$\boldsymbol{I_1(x_i,y_i)-I_2(x_i+\Delta x_i,y_i+\Delta y_i)}$$

2.误差相对于自变量的导数如何定义？

自变量为$[\Delta x,\Delta y]^T$，用中值差分表示图像梯度，导数则为：

$$\begin{bmatrix} -\dfrac{\mathrm{dI_2}}{\mathrm{d\Delta x}} \\ -\dfrac{\mathrm{dI_2}}{\mathrm{d\Delta y}} \end{bmatrix} = \begin{bmatrix} -\dfrac{\mathrm{I_2(x+\Delta x+1,y+\Delta y)-I_2(x+\Delta x-1,y+\Delta y)}}{2} \\ -\dfrac{\mathrm{I_2(x+\Delta x,y+\Delta y+1)-I_2(x+\Delta x,y+\Delta y-1)}}{2} \end{bmatrix}$$



2.4 推广至金字塔

1.所谓coarse-to-fine是指怎样的过程？

光流法中通过对图像缩放不同倍率，计算光流时先从最顶层图像开始计算，然后把上一层计算结果作为下一层计算的初始值，是一个由粗到精的过程。使用该方法后，图像实际运动太大时，图像缩小后运动的像素比较小，也能较好的追踪到。

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？

特征点法金字塔用于在图像缩放不同尺度下对比特征点，保证特征点的尺度不变性，不论相机是向前运动还是向后运动，同一特征点都能在像素平面上匹配上。

2.6 讨论

1.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？

LK光流有三个假设：亮度恒定，相邻帧运动幅度小，空间（局部）一致。在满足上述条件的情况下是合理的，但是当相邻图像整体灰度值变化较大，相机运动过快，特征点局部领域内灰度值不一致时不合理。

尽量保证相机曝光恒定（把曝光时间作为待估计变量），对图像块减去均值计算相对亮度信息，图像金字塔的方法可以改善上述情况。

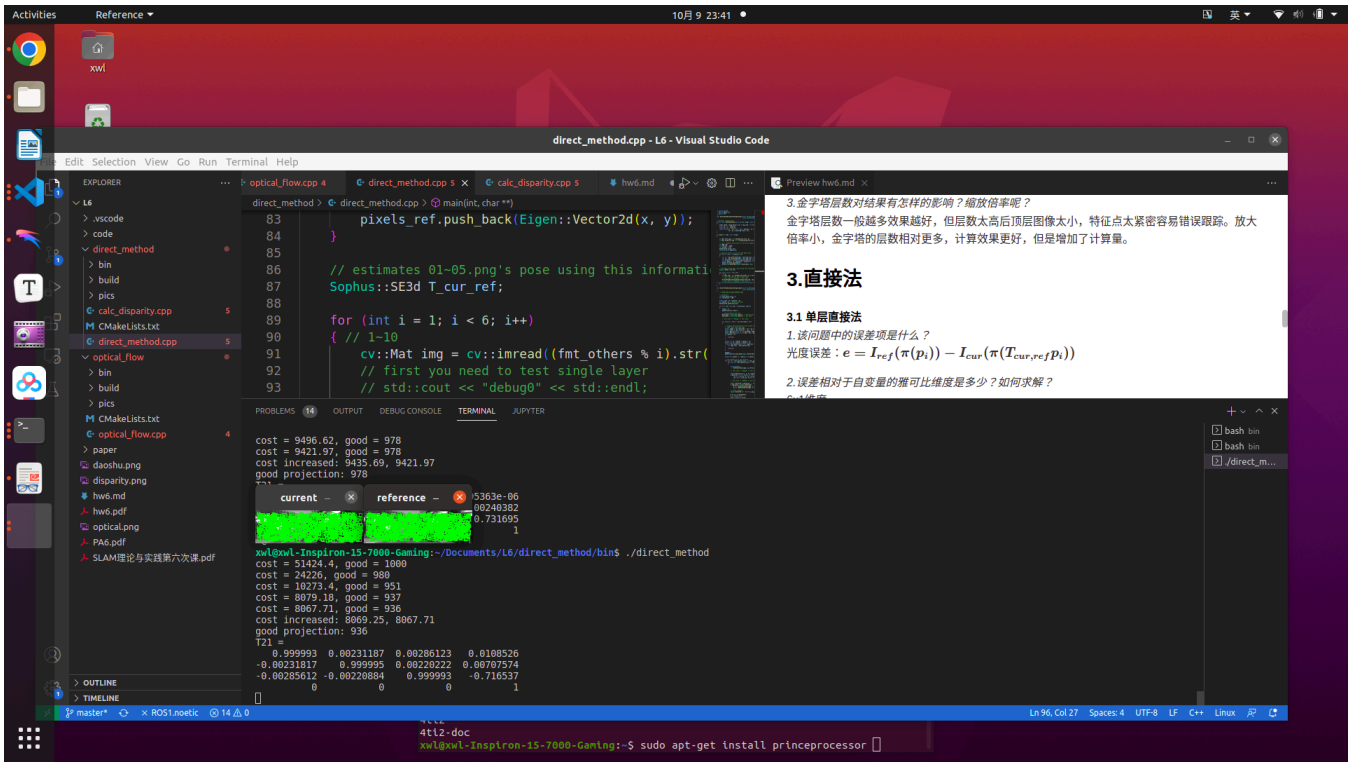
2.图像块大小是否有明显差异？取16x16和8x8的图像块会让结果发生变化吗？

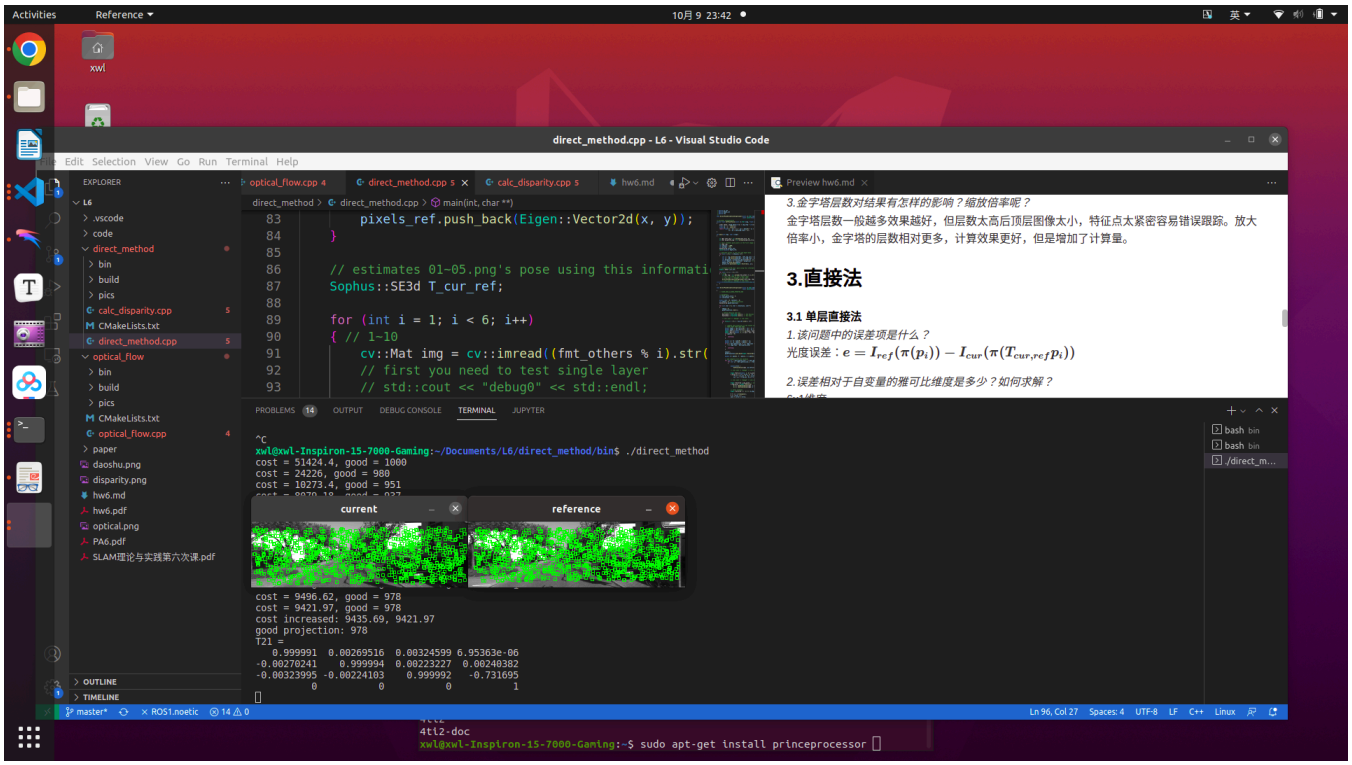
实际测试发现，图像块大效果更好，但窗口较大时，受野值影响小，结果更加鲁棒，但是窗口较小时，光流的计算更加精确。

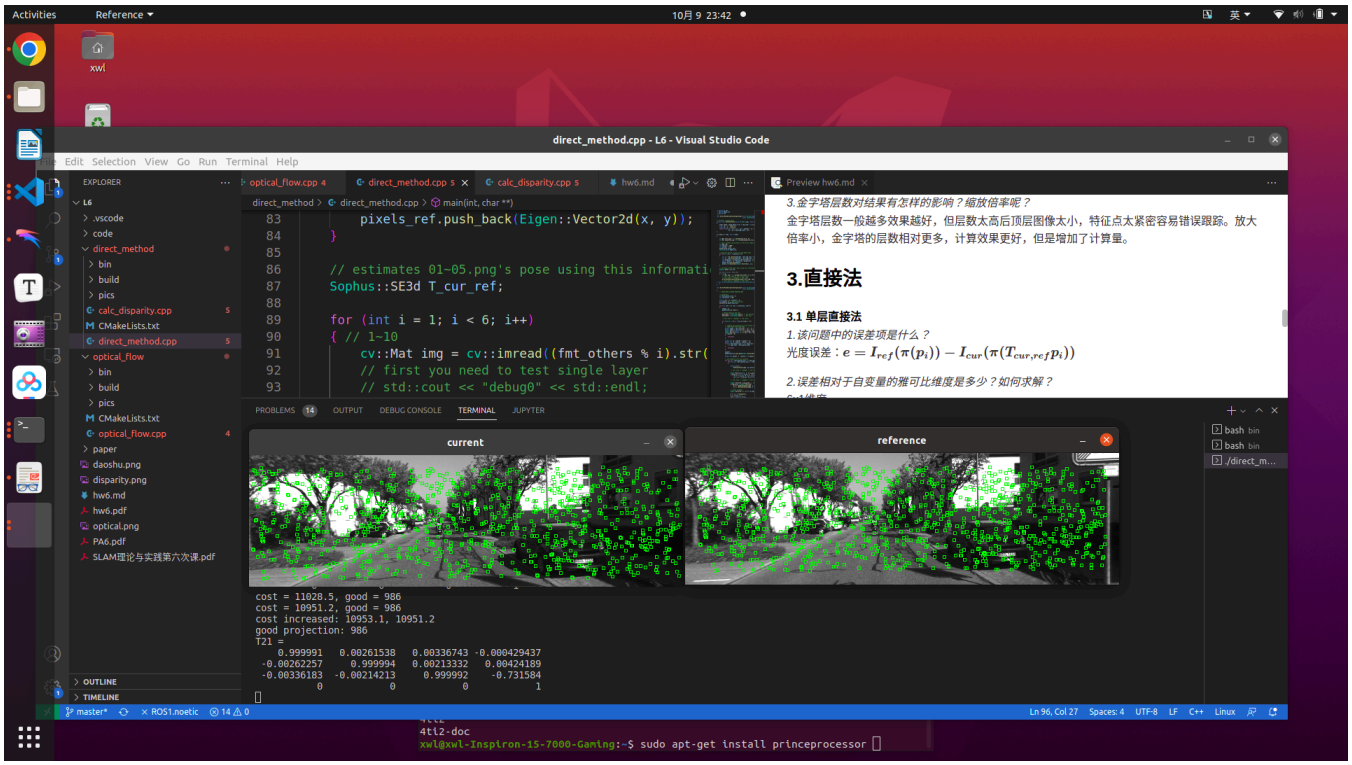
3.金字塔层数对结果有怎样的影响？缩放倍率呢？

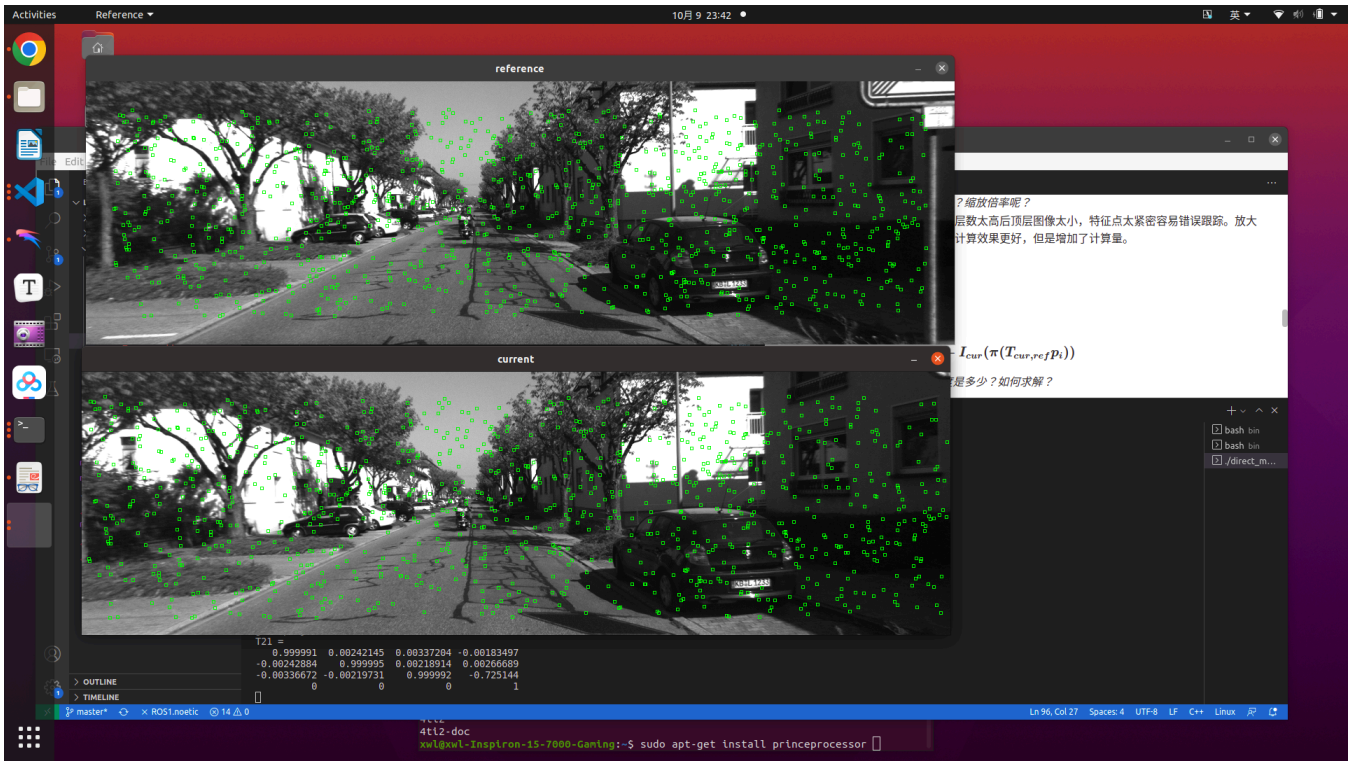
金字塔层数一般越多效果越好，但层数太高后顶层图像太小，特征点太紧密容易错误跟踪。放大倍率小，金字塔的层数相对更多，计算效果更好，但是增加了计算量。

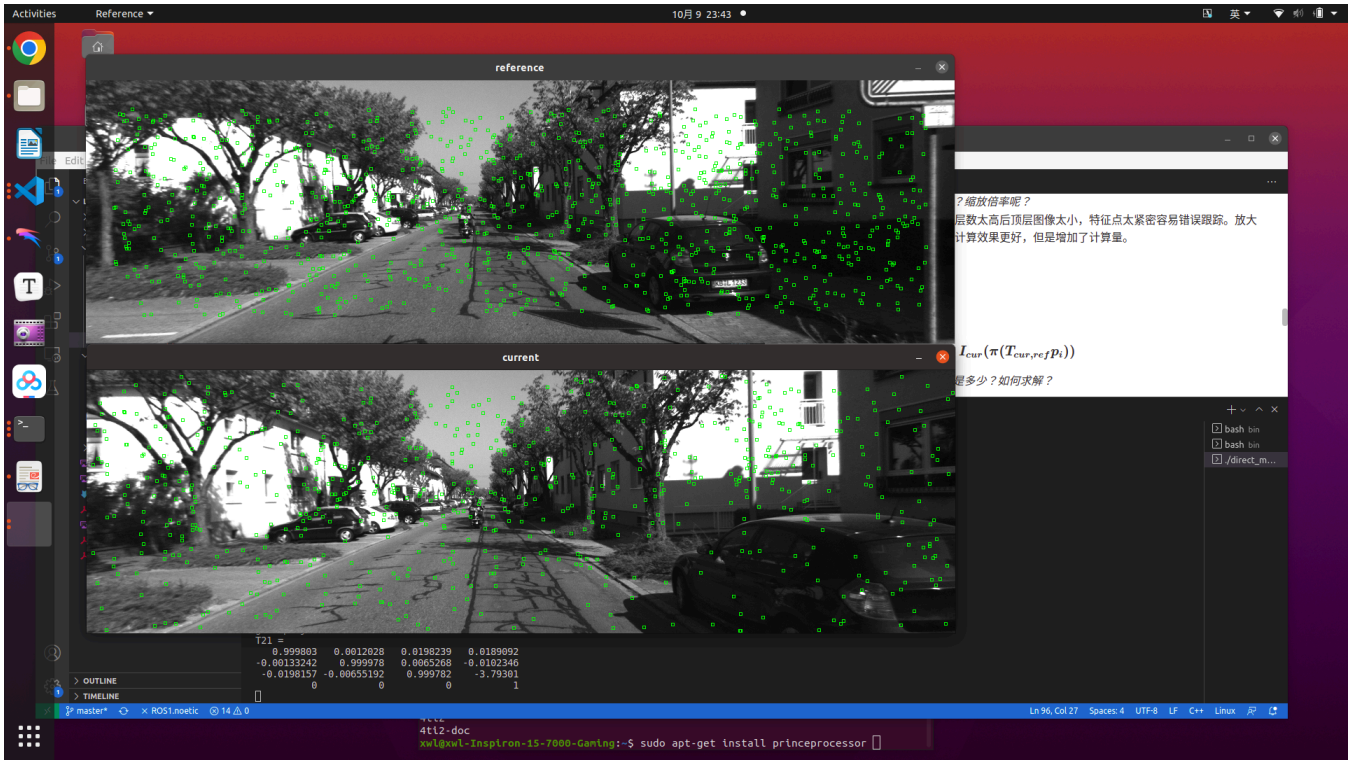
3.直接法




Activities
Reference
10月 9 23:42
direct_method.cpp - L6 - Visual Studio Code
Edit Selection View Go Run Terminal Help
EXPLORER
optical_flow.cpp 4
direct_method.cpp 5
calc_disparity.cpp 5
hw6.md
Preview hw6.md
direct_method > direct_method.cpp > main(int, char **)
pixels_ref.push_back(Eigen::Vector2d(x, y));
}
// estimates 01-05.png's pose using this informati
Sophus::SE3d T_cur_ref;
for (int i = 1; i < 6; i++)
{ // 1~10
cv::Mat img = cv::imread((fmt_others % i).str(
// first you need to test single layer
// std::cout << "debug0" << std::endl;
3.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效果越好，但层数太高后顶层图像太小，特征点太紧密容易错误跟踪。放大倍率小，金字塔的层数相对更多，计算效果更好，但是增加了计算量。
3.直接法
3.1 单层直接法
1.该问题中的误差项是什么？
光度误差：$e = I_{ref}(\pi(p_i)) - I_{cur}(\pi(T_{cur,ref}p_i))$
2.误差相对于自变量的雅可比维度是多少？如何求解？
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
xwl@xwl-Inspiron-15-7000-Gaming:~/Documents/L6/direct_method/bin$ ./direct_method
cost = 51424.4, good = 1000
cost = 24226, good = 980
cost = 10273.4, good = 951
current
reference
cost = 9496.62, good = 978
cost = 9421.97, good = 978
cost increased: 9435.69, 9421.97
good projection: 978
T21 =
0.999991 0.00269516 0.00324599 6.95363e-06
-0.00270241 0.999994 0.00223227 0.00240382
-0.00323995 -0.00224103 0.999992 -0.731695
0 0 0 1
bash
bash
./direct_m...
OUTLINE
TIMELINE
master*
ROS1.noetic
Ln 96, Col 27
Spaces: 4
UTF-8
LF
C++
Linux
4tl2-doc
xwl@xwl-Inspiron-15-7000-Gaming:~$ sudo apt-get install princeprocessor


Activities
Reference
10月 9 23:42
direct_method.cpp - L6 - Visual Studio Code
Edit Selection View Go Run Terminal Help
EXPLORER
optical_flow.cpp 4
direct_method.cpp 5
calc_disparity.cpp 5
hw6.md
Preview hw6.md
direct_method > direct_method.cpp > main(int, char **)
pixels_ref.push_back(Eigen::Vector2d(x, y));
}
// estimates 01-05.png's pose using this informati
Sophus::SE3d T_cur_ref;
for (int i = 1; i < 6; i++)
{ // 1~10
cv::Mat img = cv::imread((fmt_others % i).str(
// first you need to test single layer
// std::cout << "debug0" << std::endl;
3.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效果越好，但层数太高后顶层图像太小，特征点太紧密容易错误跟踪。放大倍率小，金字塔的层数相对更多，计算效果更好，但是增加了计算量。
3.直接法
3.1 单层直接法
1.该问题中的误差项是什么？
光度误差：$e = I_{ref}(\pi(p_i)) - I_{cur}(\pi(T_{cur,ref}p_i))$
2.误差相对于自变量的雅可比维度是多少？如何求解？
PROBLEMS 14
OUTPUT
DEBUG CONSOLE
TERMINAL
JUPYTER
current
reference
cost = 11028.5, good = 986
cost = 10951.2, good = 986
cost increased: 10953.1, 10951.2
good projection: 986
T21 =
0.999991 0.00261538 0.00336743 -0.000429437
-0.00262257 0.999994 0.00213332 0.00424189
-0.00336183 -0.00214213 0.999992 -0.731584
0 0 0 1
bash
bash
./direct_m...
OUTLINE
TIMELINE
master*
ROS1.noetic
Ln 96, Col 27
Spaces: 4
UTF-8
LF
C++
Linux
4tl2-doc
xwl@xwl-Inspiron-15-7000-Gaming:~$ sudo apt-get install princeprocessor





3.1 单层直接法

1.该问题中的误差项是什么？

光度误差：$\boldsymbol{e = I_{ref}(\pi(p_i)) - I_{cur}(\pi(T_{cur,ref}p_i))}$

2.误差相对于自变量的雅可比维度是多少？如何求解？

6x1维度

$$\frac{\partial(u)}{\partial\delta\xi} = \begin{bmatrix} \frac{f_x}{Z} & 0 & -\frac{f_x X}{Z^2} & -\frac{f_x XY}{Z^2} & f_x + \frac{f_x X^2}{Z^2} & -\frac{f_x Y}{Z} \\ 0 & \frac{f_y}{Z} & -\frac{f_y Y}{Z^2} & -f_y - \frac{f_y Y^2}{Z^2} & \frac{f_y XY}{Z^2} & \frac{f_y X}{Z} \end{bmatrix}$$

$$\boldsymbol{J^T} = -\frac{\partial \boldsymbol{I_2}}{\partial \boldsymbol{u}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\delta\xi}}$$

3.窗口可以取多大？是否可以取单个点？

可以取4x4或者6x6等大小，如果取单个点会降低鲁棒性。

3.4 延伸讨论

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？

可以。光流中优化的是dx, dy，inverse使用参考图片在(x, y)处的梯度可以减少计算量。

2.请思考上面算法哪些地方可以缓存或者加速？

可以减小图像块的大小，使用inverse的方法可以免去梯度重复计算。

3.在上述过程中，我们实际假设了哪两个patch不变？

对应点处patch灰度值、深度信息不变

4.为什么可以随机取点？而不用取角点或者线上的点？那些不是角点的地方，投影算对了吗？

直接法是通过最小化光度误差对位姿进行优化，不是角点也可以

5.请总结直接法相对于特征点法的异同与优缺点

4.*使用光流计算视差

```
the calculated disparity is -1.76575    and the ground_truth disparity is 27
the calculated disparity is 9.83838    and the ground_truth disparity is 26
the calculated disparity is -10.0763    and the ground_truth disparity is 23
the calculated disparity is 9.87018    and the ground_truth disparity is 10
the calculated disparity is 21.8598    and the ground_truth disparity is 20
the calculated disparity is 10.3151    and the ground_truth disparity is 10
the calculated disparity is 16.7521    and the ground_truth disparity is 17
the calculated disparity is 21.0757    and the ground_truth disparity is 19
the calculated disparity is 23.5785    and the ground_truth disparity is 23
the calculated disparity is 19.795    and the ground_truth disparity is 18
the calculated disparity is 10.0251    and the ground_truth disparity is 9
the calculated disparity is 2.02045    and the ground_truth disparity is 25
the calculated disparity is 16.7381    and the ground_truth disparity is 17
the calculated disparity is 4.60495    and the ground_truth disparity is 6
the calculated disparity is 38.44    and the ground_truth disparity is 21
the calculated disparity is 30.5611    and the ground_truth disparity is 29
the calculated disparity is 5.22272    and the ground_truth disparity is 5
the calculated disparity is 31.8062    and the ground_truth disparity is 32
the calculated disparity is 2.02142    and the ground_truth disparity is 25
the calculated disparity is 11.701    and the ground_truth disparity is 12
the calculated disparity is 16.2327    and the ground_truth disparity is 16
the calculated disparity is 15.2127    and the ground_truth disparity is 21
the calculated disparity is 16.857    and the ground_truth disparity is 17
the calculated disparity is 16.4348    and the ground_truth disparity is 17
the calculated disparity is 3.00192    and the ground_truth disparity is 2
the calculated disparity is 23.2293    and the ground_truth disparity is 22
the calculated disparity is 16.8336    and the ground_truth disparity is 17
the calculated disparity is 12.6601    and the ground_truth disparity is 12
the calculated disparity is -23.6251    and the ground_truth disparity is 21
the calculated disparity is -4.28964    and the ground_truth disparity is 17
the average error is 6
```