第十九章：手写 VIO 课程大作业
作业思路提示

主讲人 Horizon

作业内容

1、更优的优化策略

（a）选择更优的 LM 阻尼因子策略，使得 VINS-Mono 在 MH-05 数据集上收敛速度更快或者精度更高；

（b）实现 Dog-Leg 算法替代 LM 算法，并测试替换后的 VINS-Mono 在 MH-05 上算法精度；

详细的实验报告包括：对迭代时间和精度的评估。其中，精度评估可以采用 evo 工具，github，地址为：https://github.com/MichaelGrupp/evo。

2、更快的 MakeHessian 矩阵

采用任何一种或多种加速方式（多线程、SSE指令集等）对信息矩阵的拼接函数加速，给出详细的实验对比报告。

更优的优化策略

1、更优的优化策略

(a) 更优的 LM 阻尼因子更新策略；

(b) 实现 Dog-Leg 算法替代 LM 算法；

(c) 限制最大迭代步数；

(d) 设置残差改变量阈值；

(e) 设置优化参数增量阈值。

阻尼因子更新策略

深蓝学院 shenlanxueyuan.com

1. $\lambda_0 = \lambda_o$; λ_o is user-specified [8].
   use eq'n (13) for $\boldsymbol{h}_{\mathsf{lm}}$ and eq'n (16) for ρ
   if $\rho_i(\boldsymbol{h}) > \epsilon_4$: $\boldsymbol{p} \leftarrow \boldsymbol{p} + \boldsymbol{h}$; $\lambda_{i+1} = \max[\lambda_i / L_\downarrow, 10^{-7}]$;
   otherwise: $\lambda_{i+1} = \min[\lambda_i L_\uparrow, 10^7]$;

2. $\lambda_0 = \lambda_o \max\left[\mathsf{diag}[\boldsymbol{J}^\mathsf{T}\boldsymbol{W}\boldsymbol{J}]\right]$; λ_o is user-specified.
   use eq'n (12) for $\boldsymbol{h}_{\mathsf{lm}}$ and eq'n (15) for ρ
   $\alpha = \left(\left(\boldsymbol{J}^\mathsf{T}\boldsymbol{W}(\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p}))\right)^\mathsf{T} \boldsymbol{h}\right) / \left(\left(\chi^2(\boldsymbol{p}+\boldsymbol{h}) - \chi^2(\boldsymbol{p})\right)/2 + 2\left(\boldsymbol{J}^\mathsf{T}\boldsymbol{W}(\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p}))\right)^\mathsf{T} \boldsymbol{h}\right)$;
   if $\rho_i(\alpha\boldsymbol{h}) > \epsilon_4$: $\boldsymbol{p} \leftarrow p + \alpha\boldsymbol{h}$; $\lambda_{i+1} = \max\left[\lambda_i/(1+\alpha), 10^{-7}\right]$;
   otherwise: $\lambda_{i+1} = \lambda_i + |\chi^2(\boldsymbol{p}+\alpha\boldsymbol{h}) - \chi^2(\boldsymbol{p})|/(2\alpha)$;

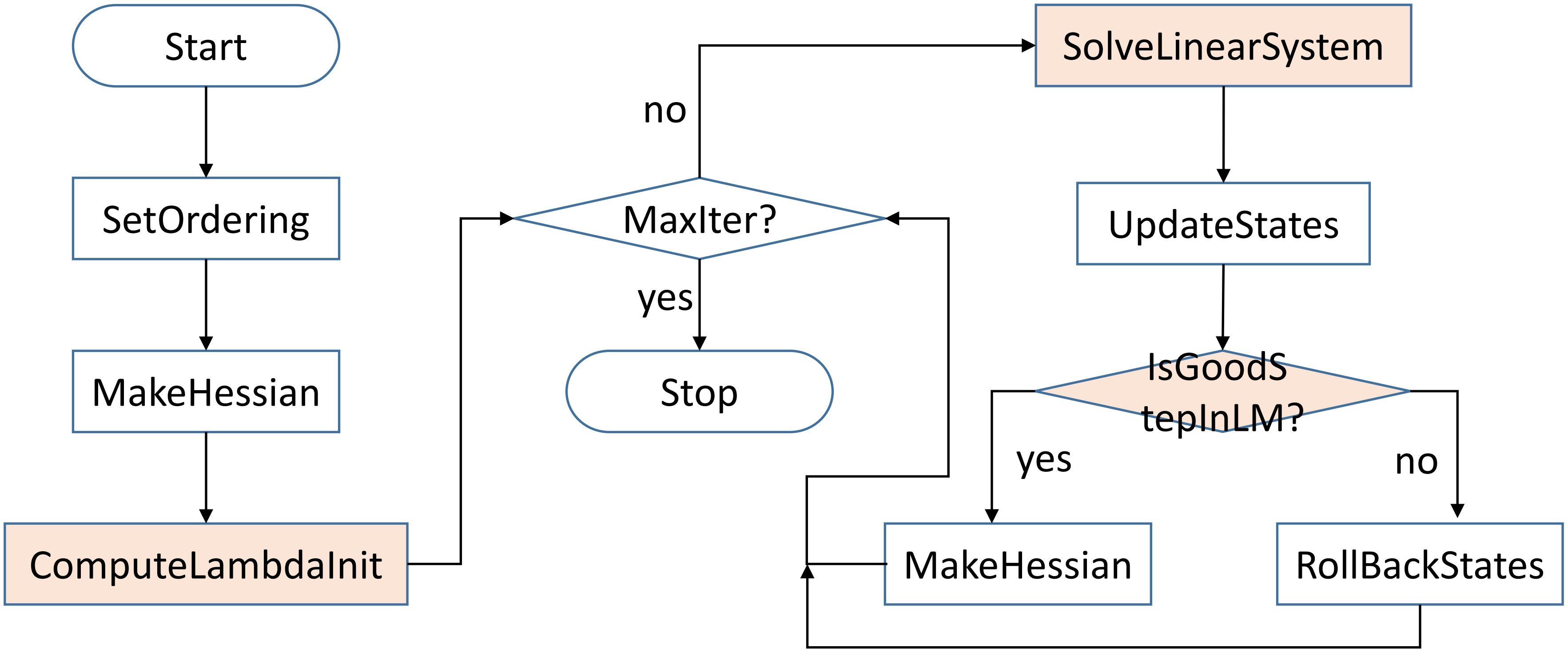
3. $\lambda_0 = \lambda_o \max\left[\mathsf{diag}[\boldsymbol{J}^\mathsf{T}\boldsymbol{W}\boldsymbol{J}]\right]$; λ_o is user-specified [9].
   use eq'n (12) for $\boldsymbol{h}_{\mathsf{lm}}$ and eq'n (15) for ρ
   if $\rho_i(\boldsymbol{h}) > \epsilon_4$: $\boldsymbol{p} \leftarrow \boldsymbol{p} + \boldsymbol{h}$; $\lambda_{i+1} = \lambda_i \max\left[1/3, 1 - (2\rho_i - 1)^3\right]$; $\nu_i = 2$;
   otherwise: $\lambda_{i+1} = \lambda_i \nu_i$; $\quad \nu_{i+1} = 2\nu_i$;

阻尼因子更新策略

三种 LM 阻尼因子更新策略对比：

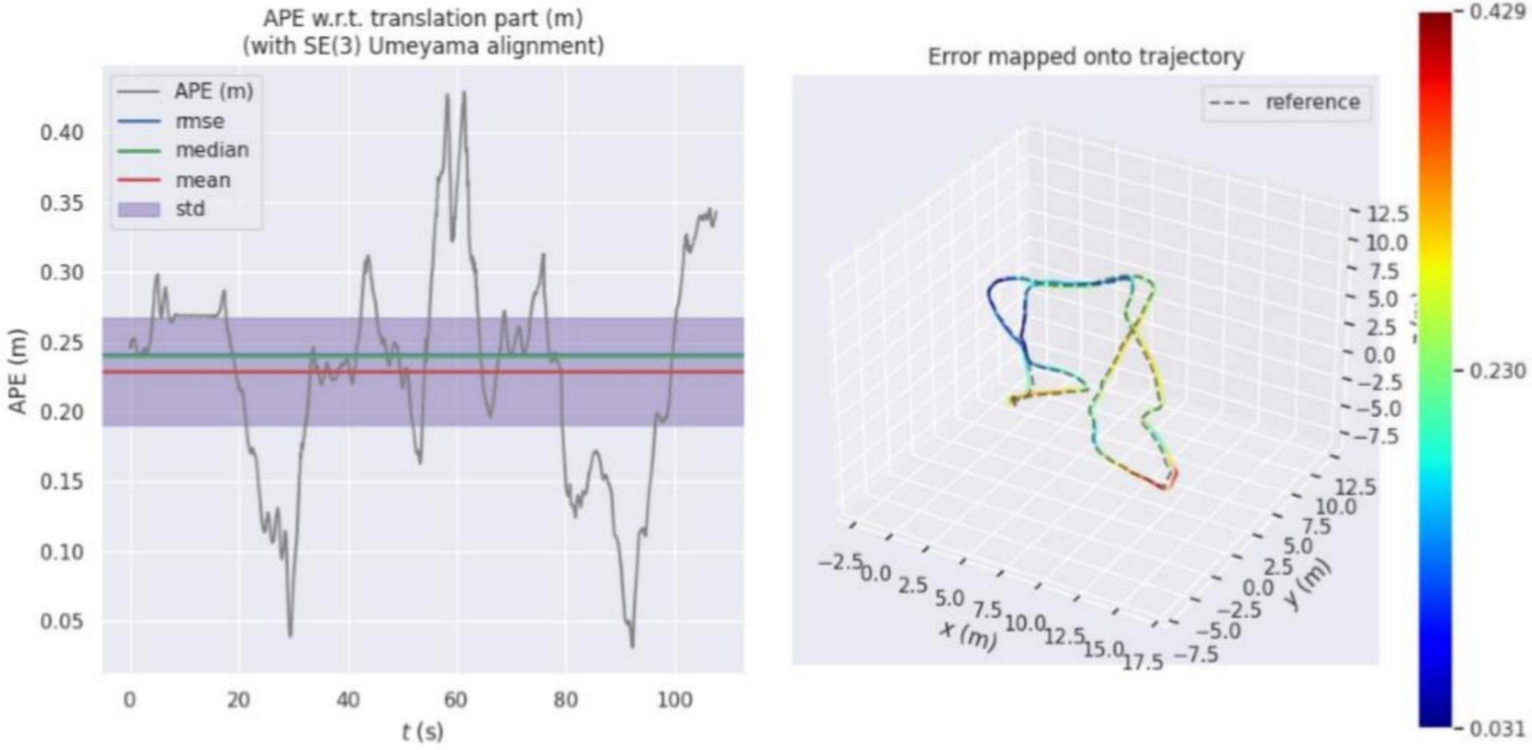
方法	初始化	求解方程	λ 变化比例
1	与 Hessian 矩阵无关	Hessian 对角线元素加上 λ *Hessian 对角线元素	固定
2	与 Hessian 矩阵有关	Hessian 对角线加上 λ	与 α 有关
3	与 Hessian 矩阵有关	Hessian 对角线加上 λ	与 ρ 有关

阻尼因子更新策略



Start
SetOrdering
MakeHessian
ComputeLambdaInit
MaxIter?
no
yes
Stop
SolveLinearSystem
UpdateStates
IsGoodStepInLM?
yes
no
MakeHessian
RollBackStates

阻尼因子更新策略



APE w.r.t. translation part (m)
(with SE(3) Umeyama alignment)
APE (m)
rmse
median
mean
std
APE (m)
t (s)
Error mapped onto trajectory
reference
x (m)
y (m)
0.429
0.230
0.031

Dog-Leg 算法

- 属于信赖域（Trust Region）类优化算法；
- Dog-Leg 算法在以当前点为中心，半径为Δ的区域（信赖域）内，将目标函数做了二阶近似；
- Dog-Leg 算法分别计算了最速下降法和高斯牛顿法的最优增量 δ_{sd} 和 δ_{gn}，并根据信赖域的大小来选择合适的增量。

【参考文献】Lourakis M , Argyros A A . Is Levenberg-Marquardt the Most Efficient Optimization Algorithm for Implementing Bundle Adjustment?[C] // 10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China. IEEE, 2005.

Dog-Leg 算法步骤

- 第一步：计算最速下降法和高斯牛顿法的最优增量 δ_{sd} 和 δ_{gn}

■ $\boldsymbol{\delta}_{\mathrm{sd}}$

- $F(x+\Delta x)=\frac{1}{2}||f(x+\Delta x)^{\top}f(x+\Delta x)||_2^2 \approx F(x)+J^{\top}f\Delta x+\frac{1}{2}\Delta x^{\top}J^{\top}J\Delta x, s.t. ||\Delta x|| \le \Delta$
- 增量方向：$d=F'(x)=(J^{\top}f)^{\top}$
- 增量步长：$\frac{\partial F(x-\alpha d)}{\partial \alpha}=0 \quad \alpha=\frac{(J^{\top}f)^{\top}J^{\top}f}{(J^{\top}f)^{\top}(J^{\top}J)J^{\top}f}$
- $\boldsymbol{\delta}_{\mathrm{sd}}$ $\delta_{sd}=\frac{-(J^{\top}f)^{\top}J^{\top}f}{(J^{\top}f)^{\top}(J^{\top}J)J^{\top}f}(J^{\top}f)^{\top}$ 对应代码$\frac{b^{\top}b}{b^{\top}Hb}b$

Dog-Leg 算法步骤

深蓝学院 shenlanxueyuan.com

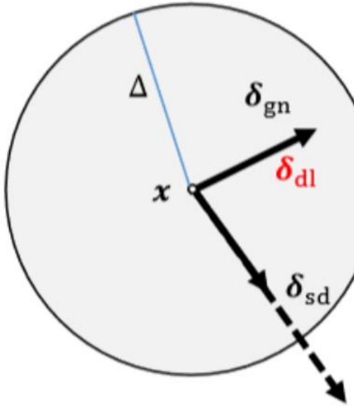
- 第一步：计算最速下降法和高斯牛顿法的最优增量 δ_{sd} 和 δ_{gn}

■ $\boldsymbol{\delta}_{\mathrm{gn}}$

- $F(x+\Delta x)=\frac{1}{2}||f(x+\Delta x)^{\top}f(x+\Delta x)||_2^2 \approx F(x)+J^{\top}f\Delta x+\frac{1}{2}\Delta x^{\top}J^{\top}J\Delta x, s.t. ||\Delta x|| \leq \Delta$
- $J^{\top}J\delta_{gn}=-J^{\top}f$ 对应代码 $\mathrm{H}x=b$

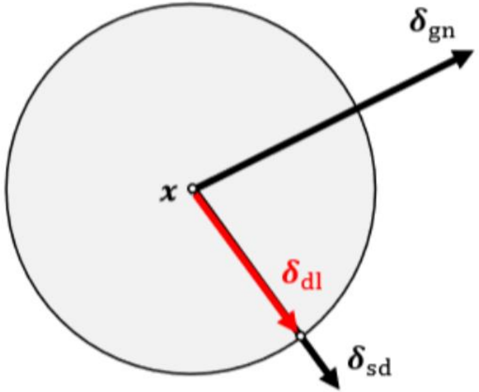
Dog-Leg 算法步骤

- 第二步：计算 Dog-Leg 算法的最优增量 δ_{dl}



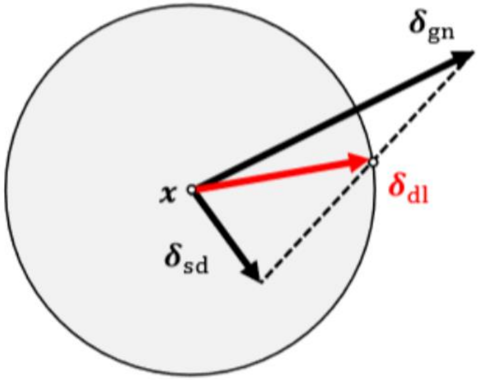
若$\|\boldsymbol{\delta}_{\mathrm{gn}}\| \leq \Delta, \|\boldsymbol{\delta}_{\mathrm{sd}}\| \in \mathrm{R}$

$$\boldsymbol{\delta}_{\mathrm{dl}} = \boldsymbol{\delta}_{\mathrm{gn}}$$



若$\|\boldsymbol{\delta}_{\mathrm{gn}}\| > \Delta, \|\boldsymbol{\delta}_{\mathrm{sd}}\| > \Delta$

$$\boldsymbol{\delta}_{\mathrm{dl}} = \frac{\boldsymbol{\delta}_{\mathrm{sd}}}{\|\boldsymbol{\delta}_{\mathrm{sd}}\|}\Delta$$



若$\|\boldsymbol{\delta}_{\mathrm{gn}}\| > \Delta, \|\boldsymbol{\delta}_{\mathrm{sd}}\| \leq \Delta$

$$\boldsymbol{\delta}_{\mathrm{dl}} = \boldsymbol{\delta}_{\mathrm{sd}} + \alpha(\boldsymbol{\delta}_{\mathrm{gn}} - \boldsymbol{\delta}_{\mathrm{sd}})$$

Dog-Leg 算法步骤

深蓝学院 shenlanxueyuan.com

- 第三步：更新 Dog-Leg 算法的信赖域半径 Δ

$$\rho = \frac{F(x+\Delta x) - F(x)}{m_k(x+\Delta x) - m_k(x)}$$

若近似效果不好，即 $\rho \le 0.25$，则缩小信赖域 $\Delta = 0.25 * \Delta$

若近似效果较好，即 $\rho \ge 0.75$，则增大信赖域 $\Delta = \max\left\{\Delta, 3\left\|\delta_{dl}\right\|\right\}$

否则信赖域不变

更快地构造海森矩阵

2、更快的 MakeHessian 矩阵

（a）OpenMP：最为便捷，资料较多：

http://supercomputingblog.com/openmp/openmp-tutorial-the-basics/

（b）SSE Instruction Set：英特尔公司有效增强 CPU 浮点运算能力的指令集：

http://supercomputingblog.com/optimization/getting-started-with-sse-programming/

（c）Nvidia CUDA：英伟达 GPU 的加速运算库，较为复杂：

https://cuda-tutorial.readthedocs.io/en/latest/tutorials/tutorial01/

感谢各位聆听！

Thanks for Listening