

第8周：ORB_SLAM2 课程课件

本周课程重点：

闭环检测线程

Sim3搜索匹配SearchBySim3

EdgeSE3ProjectXYZ 误差计算

EdgeSim3ProjectXYZ 的误差计算

EdgeInverseSim3ProjectXYZ 的误差计算

ComputeSim3计算Sim3

闭环矫正 CorrectLoop

第8周：ORB_SLAM2 课程课件

本课件是公众号 计算机视觉life 旗下课程 [《全网最详细的ORB-SLAM2精讲：原理推导+逐行代码分析》](#)

(点击可跳转课程详情) 的课程课件。谢谢各位学员的支持！

本课程对应的注释代码：https://github.com/electech6/ORBSLAM2_detailed_comments

由于源码注释和课件在持续更新，所以：

如视频课程中注释与上述GitHub中有不同，**以GitHub上最新源码为准。**

如视频课程中课件与本课件不同，**以本课件为准。**

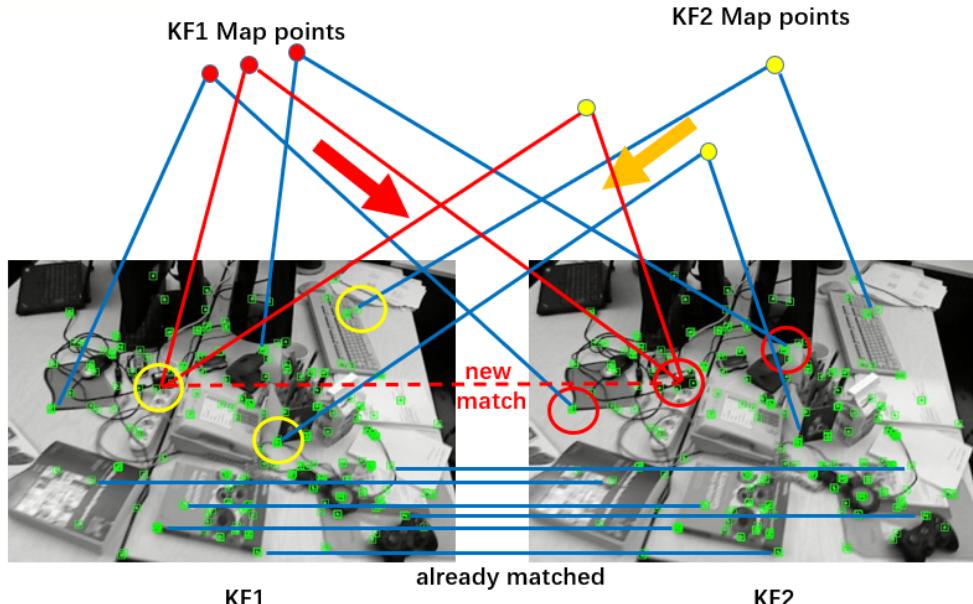
本周课程重点：

1. 掌握Sim3优化位姿的原理（尤其是顶点边的定义等）与代码（重要）。
2. 理解闭环矫正调整位姿和地图点的原理。

闭环检测线程

Sim3搜索匹配SearchBySim3

g2o优化Sim3时，地图点不加入优化。



EdgeSE3ProjectXYZ 误差计算

```
// 误差 = 观测 - 投影
_error = obs-cam_project(v1->estimate().map(v2->estimate()));
    const VertexSim3Expmap* v1;
// 其中:
// 相机2世界坐标系下的三维点坐标
    = static_cast<const VertexSim3Expmap*>(<vertices[1]>)
v2->estimate()

//map 函数是用R, t把某个坐标系下三维点变换到另一个坐标系下
Vector3d map(const Vector3d & xyz) const
{
    return _r*xyz + _t;
}

// 把相机2地图点用相机1变换矩阵变换到相机1坐标系下
v1->estimate().map(v2->estimate())

// 反投影到图像坐标系
Vector2d EdgeSE3ProjectXYZ::cam_project(const Vector3d & trans_xyz) const{
    Vector2d proj = project2d(trans_xyz);
    Vector2d res;
    res[0] = proj[0]*fx + cx;
    res[1] = proj[1]*fy + cy;
    return res;
}
```

EdgeSim3ProjectXYZ 的误差计算

```
// 误差 = 观测 - 投影
_error = obs-v1->cam_map1(project(v1->estimate().map(v2->estimate())));
// 其中:
// 相机2坐标系下的三维点坐标
```

```
v2->estimate()
```

```
//map 函数是用sim变换(r,t,s)把某个坐标系下三维变换到另一个坐标系下, 定义是
Vector3 map (const Vector3& xyz) const {
    return s*(r*xyz) + t;
}

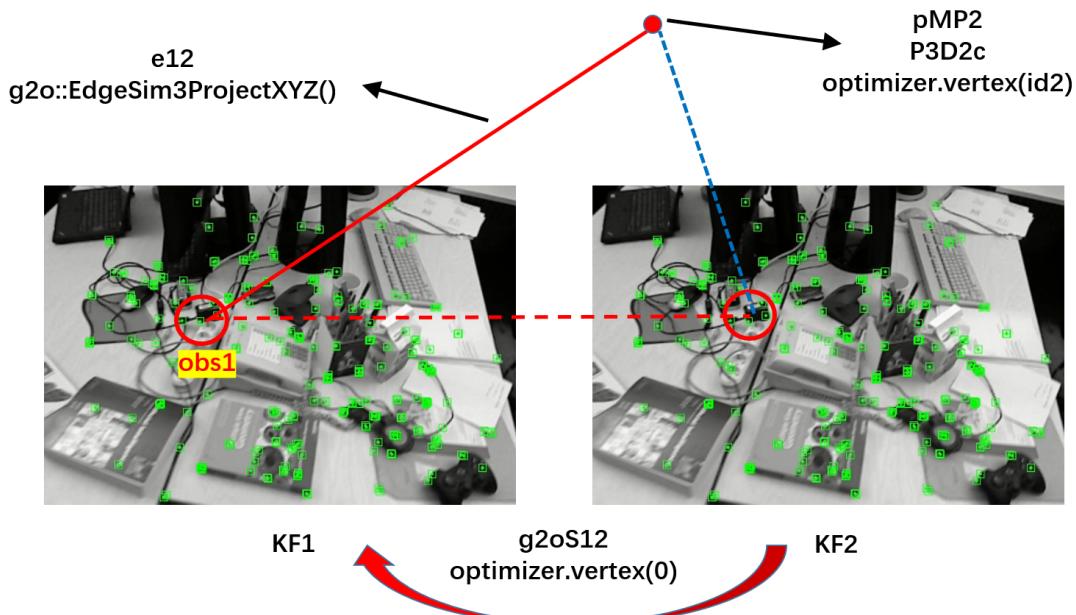
// 用v1估计的Sim12 变换把v2代表的相机2坐标系下三维点变换到相机1坐标系下
v1->estimate().map(v2->estimate())

// project 函数是把三维坐标归一化
Vector2d project(const Vector3d& v)
{
    Vector2d res;
    res(0) = v(0)/v(2);
    res(1) = v(1)/v(2);
    return res;
}

// cam_map1是用内参转化为图像坐标
Vector2d cam_map1(const Vector2d & v) const
{
    Vector2d res; 构建Sim3顶点
    res[0] = v[0]*_focal_length1[0] + _principle_point1[0];
    res[1] = v[1]*_focal_length1[1] + _principle_point1[1];
    return res;
}
```



计算机视觉life



EdgeInverseSim3ProjectXYZ 的误差计算

地图点可以互相提升

-致性.

// 误差 = 观测 - 投影

_error = obs->v1->cam_map2(project(v1->estimate().inverse().map(v2->estimate())));

// 相机1坐标系下的三维点坐标 (和上面不同)
v2->estimate()

// 用v1估计的sim12 变换的逆 (也就是sim21) 把v2代表的相机1坐标系下三维点变换到相机2坐标系下
v1->estimate().map(v2->estimate())

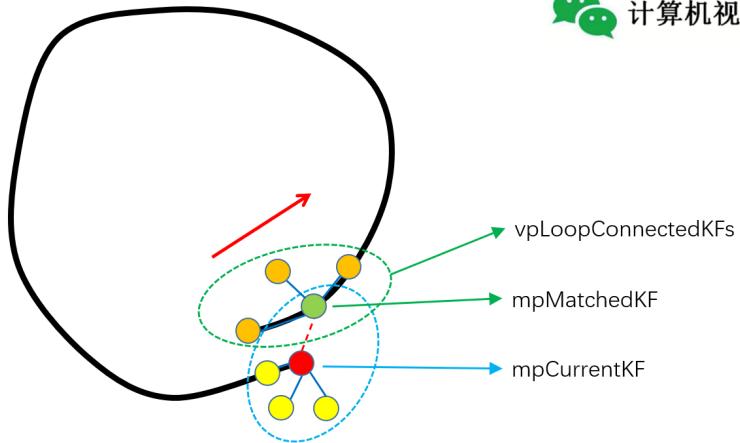
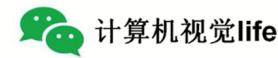
// 后面类似

这是 current frame 中
对应的地图点

这就是 Sim3 计算的 $\begin{bmatrix} sR_{12} & t_{12} \\ 0 & 1 \end{bmatrix}^{-1}$
 $= \begin{bmatrix} \frac{1}{s} R_2^T & -\frac{1}{s} R_2^T t_{12} \\ 0 & 1 \end{bmatrix}$

ComputeSim3计算Sim3

LoopClosing::ComputeSim3() 里查找vpLoopConnectedKFs, mpLoopMapPoints



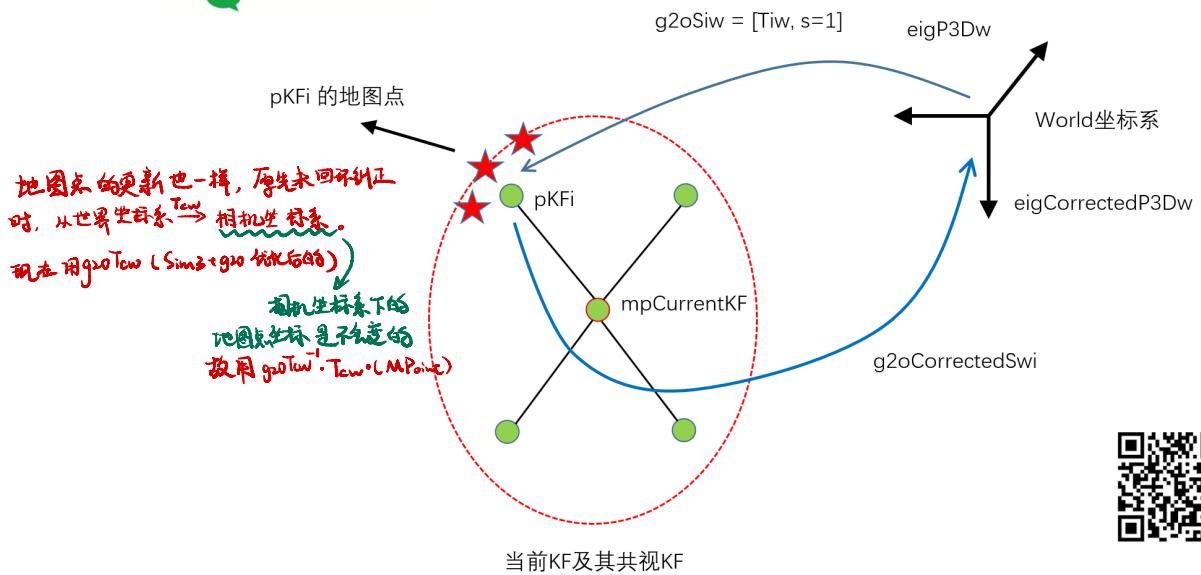
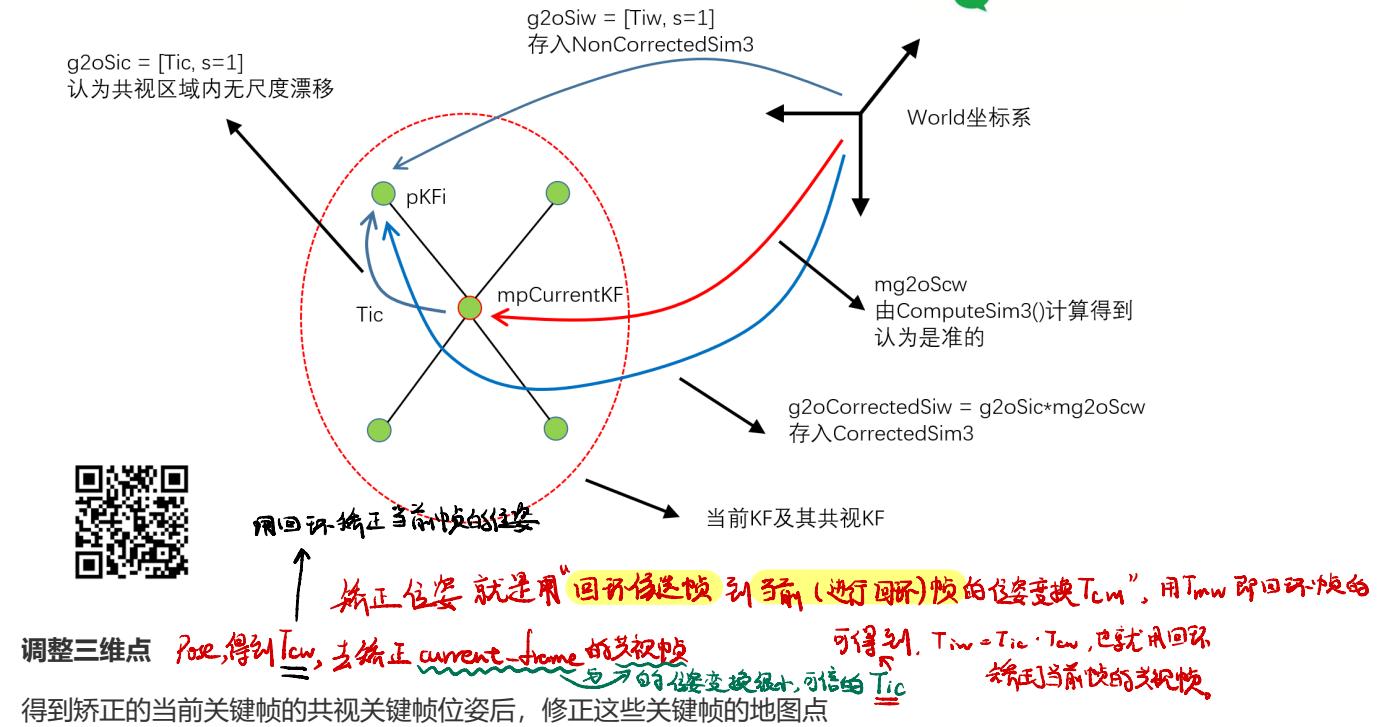
- 当前关键帧
- 当前关键帧的连接关键帧
- 闭环候选关键帧 (不和当前关键帧连接) 及其连接关键帧
- 闭环候选关键帧的连接关键帧



闭环矫正 CorrectLoop

调整位姿

通过mg2oScw (认为是准的) 来进行位姿传播, 得到当前关键帧的共视关键帧的世界坐标系下Sim3 位姿 (还没有修正)



[《全网最详细的ORB-SLAM2精讲：原理推导+逐行代码分析》](#) (点击可跳转课程详情)

全网最详细的ORB-SLAM2精讲

超实用的VSLAM解决方案

疑难点全面覆盖，带你深度掌握实际项目细节

原理：关键公式推导 + 走心示意图

代码：详细注释 + 逐行详解 + 源码勘误

主讲人：小六

中科院博士、多年VSLAM从业经验、计算机视觉life公众号创始人

长按或扫描二维码查看课程介绍和购买方式：

