SfM & SLAM Tutorial

提纲

1. **Introduction**

   1.1 References
   1.2 Terms
   1.3 Mathematical Basis

2. **Monocular RGB Camera Based**

   2.1 Overview
   2.2 Sparse Reconstruction (feature based)

   - Camera Calibration
   - Epipolar Geometry
   - Camera Pose Estimation
   - False Correspondences
   - Triangulation
   - Perspective n Points Problem
   - Bundle Adjustment
   - Image-Based Relocalization
   - PTAM (Parallel Tracking and Mapping)

   2.3 Dense Reconstruction (optical flow based)

   - Optical Flow
   - Disparity Map & Depth map
   - Regularized Monocular Depth Estimation
   - Visual Odemetry
   - Semi-direct Monocular Visual Odometry
   - DTAM
   - Semi-Dense SLAM
   - LSD-SLAM

3. **Stereo Camera Based**
4. **RGB-D Camera Based**

1.Introduction

- **SfM**

Structure from motion (SfM) 简单来说就是由图像序列（可能结合了局部运动信息）来估计场景三维结构的过程。比较具有代表性的项目有离线的如 Changchang Wu 的 VisualSFM， Noah Snavely 的 Bundler SFM；以及实时的如 Pollefeys 的 Visual modeling with a hand-held camera， Pradeep 的 MonoFusion 等。传统的 SfM 包含以下 4 个步骤：

1. Image feature extraction

2. Feature matching（例如 *Scalable Recognition with a Vocabulary Tree* 以及 *Fast and Accurate Image Matching with Cascade Hashing for 3D Reconstruction* 等方法）

3. Generate tracks

4. Bundle adjustment

- **SLAM**

SLAM 是 Robotics 中的一种定位算法，全称是 Simultaneous Localization and Mapping。

要让机器人知道自己在哪里（Localization），可以用 GPS，但是误差有点大而且还必须在室外。怎么才能让机器人在室内，或者更精确地知道自己的位置呢？SLAM 就是一种方法。

基本的想法是，如果我是一个机器人，我知道两个定点 1：(x1,y1)和定点 2：(x2,y2)，我现在可以看见定点 1，测量一下，离它 5 米，把头向右转过 30 度角后，可以看见离定点 2，测量一下，离它 10 米。然后解一个三角形，我就知道自己在那里了（Localization），位置准确度和测量精度有关（转角，距离），通常可以控制在很小的范围内。

但是问题又来了，每次我走几步，就得回头看看定点 1 和定点 2，才知道自己在哪里。这一步三回头的走法，真是一件不怎么优雅的事情。于是聪明勤劳勇敢的专家们想出来一个方法，在第一步的时候，一旦我知道自己在哪里，我就添加几个定点：定点 3，定点 4，定点 5……。因为知道自己的坐标，我可以测量出这些新定点离我多远，加上自己的坐标，就知道这些定点在哪里（Mapping）。以后的定位(Localization)就可以用这些新定点。多了这些定点后，以后再走路，哇身边都是定点，腰不酸腿不疼，知道自己在哪里了。

所以 SLAM 就是指，同时（Simultaneous）知道自己的位置（Localization）和（And…） 新的定点的位置（(feature) Mapping）。因为不管是测量距离，还是计算自己的转角，或记录行走的距离（利用 Odometer）都会有 noise，而直接计算 SLAM，noise 会叠加。所以一般 SLAM 要有一个 Kalman Filter 的过程。ICRA 10 的一篇文章就讨论了为什么需要 Filter. (ICRA 10: Why Filter?) .

举个例子，MonoSLAM 是 Andrew Davison 提出来的利用一个摄像头来做 SLAM 的方法，也叫 Real-Time Structure From Motion。在这里，定点变成了 visual feature，测量定点的位置转变为 match feature, then triangulate. 一个 2004 年的 demo 。当摄像头在空间里忘乎所以地移动时，MonoSLAM 都可以利用 feature matching，知道摄像头的位置，和那些 feature 的位置。demo 中的右图的黄线，是 camera 的 trajectory。而椭圆表示对于新加的 feature 的不确定性。

可以看到在 demo 中，虽然知道摄像头的位置，但是 mapping 的 feature 很少很稀疏，这样不好不强大。经过了 6 年，随着 CPU,GPU 性能的提升，Andrew Davison 这个组在 CVPR 10 搞出了 dense live MonoSLAM：利用 GPU 计算 PTAM，然后 Mesh 成 Surface。请看 demo。

- **区别与联系**

首先 SfM 和 vSLAM 基本讨论的是同一问题，不过 SFM 是 vision 方向的叫法，而 vSLAM 是 robotics 方向的叫法。vSLAM 所谓的 mapping 和 localization 在 vision 方向分别称作 structure 和 camera pose。但从出发点考虑的话，SFM 主要是要完成 3D reconstruction，而 vSLAM 则主要完成 localization。

从方法论的角度上考虑，传统的 SfM 是不要求 prediction 的，但是对于 vSLAM 而言 prediction 是必须的，因为 vSLAM 的终极目标是要 real-time navigation，而对于传统 SfM 而言，real-time 是不要求的。而传统的 vSLAM 也把主要精力放在 prediction 上面，而且是借助非 camera 的外界的手段来 predict，例如 acceleration sensor。而传统 SFM 则把精力放在 feature tracking 上面了。不过近些年来，SFM 开始利用图片间的 optical flow 做 prediction，而 vSLAM 则更加的注重了 feature tracking。所以就目前而言两个领域似有大融和趋势。

由上面的讨论可以看出，real time 的 SfM 其实和 SLAM 的研究内容已经非常接近了。

本文按照所使用的硬件设备分类，将分为三个部分去具体讨论：

- 基于单目摄像机的；
- 基于双目摄像机的；
- 基于 RGB-D 摄像机的。

目前着重整理基于单目摄像机的，另外另个部分会随着学习的深入陆续补充。

1.1 References

书籍：

1. *Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods*
2. *Multiple View Geometry in computer vision*
3. *Probabilistic Robotics*
4. *Numerical Recipes*
5. *Mastering OpenCV with Practical Computer Vision Projects*

视频课程：

1. University of Freiburg 的 Cyrill Stachniss 教授的 SLAM 课程（首推）
   https://www.youtube.com/playlist?list=PLgnQpQtFTOGQrZ4O5QzbIHgl3b1JHimN_
2. Rich Radke 教授的 Computer Vision and Visual Effects 课程（公式推导的非常详细，非常适合入门）
   https://www.youtube.com/watch?v=rE-hVtytT-I&list=PLuh62Q4Sv7BUJlKlt84HFqSWfW36MDd5a
3. 多视图几何的视频课程，如果看书时有哪些推导不理解可以参看
   https://www.youtube.com/watch?v=RDkwklFGMfo&list=PLTBdjV_4f-EJn6udZ34tht9EVIW7lbeo4
4. Technische Universität München 的 Jürgen Sturm 教授的 Autonomous Navigation for Flying Robots 课程

   https://www.edx.org/course/autonomous-navigation-flying-robots-tumx-autonavx-0

文档：

1. *Visual SLAM, Structure from Motion, Multiple View Stereo* by Yu Huang (对 SLAM 和 SfM 的总结非常到位，本文的主要参考文章)

Useful Sites：

1. http://www.mrpt.org/ （包含一些适合初学者的 Tutorial）
2. http://vision.in.tum.de/ （TUM 的计算机视觉小组非常厉害）
3. http://www.cvchina.info （有些文章写得很棒）

项目(代码)：

1. /Matlab/toolbox/vision/visiondemos 在 Matlab 比较新的版本中都附带有一些常见功能的实现 demo（例如相机标定、追踪、检测、稀

疏重建与稠密重建等），步骤说明也很详细，非常适合入门时学习。P.S. 用 visiondemos 命令可以查看所有 demo 合集。

2. Bundler-SFM（also see the site）
3. PTAM
4. LSD-SLAM
5. TheiaSfM (专注 SfM 的开源库)
6. 3DReconstruction
7. https://github.com/ieted/3DReconstruction-Primer （今后的学习过程中我会尽可能的将所学到的每个具体知识点的代码上传于此）

1.2 Terms

(此处主要补充一些出现频率较高、单纯从字面上比较难理解的技术词汇)

distortion correction Often we will want to undo the effects of image distortion:

- To get the correct image coordinates of a measured point that are then used in a subsequent 3D reconstruction. In this case the image does not have to be warped but feature extraction is performed on the original image. This might be beneficial as warping can introduce aliasing and generally implies a low-pass filtering step.
- Other tasks such as the extraction of straight lines or the grouping of features is better done on the warped images. This is also the case for simple stereo correlation methods that seeks to establish dense correspondences by searching along horizontal scan lines - such algorithms also require rectification.

This process straightens the lines that are bent by the radial distortion of the lens.

rectification The purpose of rectification is to transform the general stereo geometry to that of the standard stereo setup. In this simple geometry the image planes are parallel and coordinate systems aligned. Then the epipoles are at infinity and the epipolar lines are parallel and along the x-direction (scan lines). After rectification the problem of stereo matching reduces to a one-dimensional search along the rows of the images. The depth of a 3D point becomes inversely proportional to the disparity, i.e. the horizontal distance between corresponding image points.

(见 pdf '*Rectification and Distortion Correction*')

（待补充……）

1.3 Mathematical Basis

1. 首先是基本的微积分及线性代数，基本贯穿论文的始终；
2. 多视图几何（*Multiple View Geometry in computer vision* 可以称为是三维重建的圣书之一）；
3. 概率论（首推 *Probabilistic Robotics*）；
4. 数值计算（推荐 *Numerical Recipe*）。

(待细化、补充……)

2. Monocular RGB Camera Based

2.1 Overview

目前单目 tracking 的方法大致可以被分为两类：

- **feature based**
- **flow based**

前者是利用特征匹配进行，而后者则是利用光流场。目前很多的研究都倾向于将两种方法结合在一起（hybrid），取各自的长处。

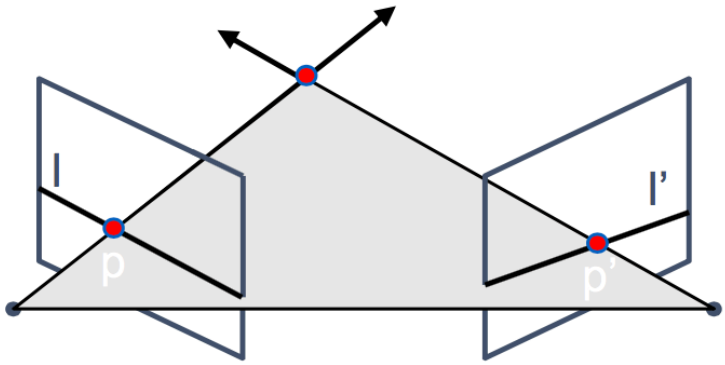
2.2 Sparse Reconstruction (feature based)

Camera Calibration

- 目标：找出相机的内参与外参（即找出像素坐标系到世界坐标系的变换关系，参见实验室 SVN 中/like/docs/calibration.pptx）
    - 内参:像素坐标系与相机坐标系之间的变换关系；
    - 外参:相机在世界坐标系中的位置及朝向。
- Roger Tsai 以及 Zhengyou Zhang 关于相机标定的算法都十分有名，分别利用了 3D 节点设定(Tsai)以及 2D 平面(Zhang)的方法。
- 标定技术
    - 用 3D 标定物进行标定
        - 已精准的知道 3D 标定物在空间中的几何特征
        - 标定物通常包含两到三个相互正交的平面
        - 将一个平面精确地平移一个已知量一个以完成相机标定（Tsai 的方法）
        - 优点：理论简单，精度高
        - 缺点：设置复杂而费力
    - 用 2D 平面标定板进行标定(Zhang's approach)
        - 需要从几个不同的角度观察标定板
        - 无需知道平面运动
        - 设置简单，使用最为广泛
    - 用 1D 线性标定(Zhang's approach)
        - 该方法相对较新；
        - 标定物是一组共线的点，例如两个距离已知的点，三个相互距离已知的共线点，以此类推；
        - 通过观察一个绕固定点移动的线来完成标定；
        - 可以一次标定多个相机
    - 自标定（不需要标定参照物）
        - 精度较低
    - 消失点检测法
        - 通过图像中的消失点(Vanishing Points)来完成相机标定

Epipolar Geometry

- 设 p 为左图像中的一点，p′为右图像中的一点



- 对极关系
  - 点 p 将投射到对极线 l′上，l′ = Fp
  - 点 p′将投射到对极线 l 上，l = p′F
- 上述矩阵 F 被称为
  - 本质矩阵(Essential Matrix)(当相机内参已知时)

$$p^T \varepsilon p' = 0 \; with \; \varepsilon = [t_\times]R$$

  - 基础矩阵(Fundamental Matrix)(当相机内参未知时，这种情况更为常见)

$$p^T F p' = 0 \; with \; F = K^{-T} \varepsilon K'^{-1}$$

    F 可由匹配点对求得
  - 每一个(p, p′)点对都可以写出一个关于 F 的线性方程
  - 8/5 点法可以用来计算 F/E

Camera Pose Estimation

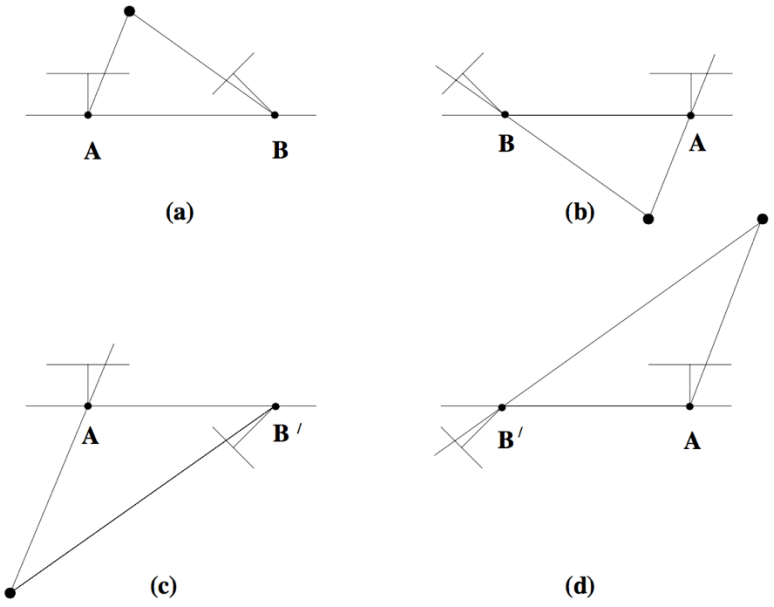
- 图像上关键点的匹配
- 基础矩阵的预测：8 点法或 7 点法
- 去除外点：RANSAC(或 M-estimation, LMedS, Hough transform)
- 如果内参已知，计算本质矩阵 E：5 点法
- 从 E 中提取旋转矩阵 R 和平移矩阵 t
- 设第一个相机的相机矩阵为$P_1 = [I|0], E = Udiag(1,1,0)V^T$

  对于第二个相机矩阵有 4 种可能：

$$P_2 = [UWV^T|+u_3]$$
$$P_2 = [UWV^T|-u_3]$$
$$P_2 = [UW^TV^T|+u_3]$$
$$P_2 = [UW^TV^T|-u_3]$$

  其中$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$，$u_3$为矩阵 U 的第 3 列。（具体推导可见多视图几何 p257-259）

  利用所得矩阵对图像关键点进行三角化测量（Triangulation），检验 4 种结果中哪一种符合所有 3D 点均在两个相机之前（如下图所示，只有(a)符合条件）



False Correspondences

通常来讲，基于特征的预测方法需要解决在匹配过程中因缺少几何约束而造成的错误的关键点匹配。解决该问题的相关的算法有：

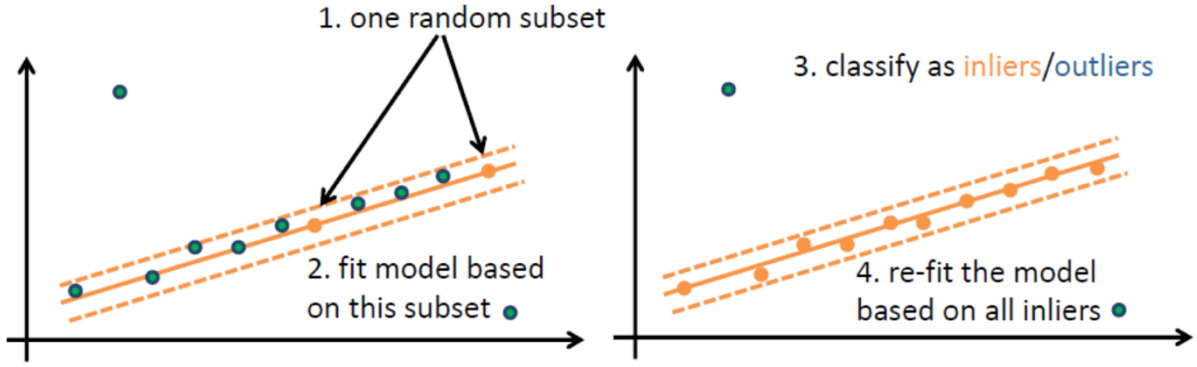
- **M-estimators** (Huber 1981),
- Least Median of Squares (**LMedS**) (Rousseeuw and Leroy 1987) and
- Random Sample Consensus (**RANSAC**) (Fischler and Bolles 1981)

RANSAC:

目标：将一个含有外点的数据集健壮的匹配出一个模型

算法：

1. 随机的从数据点中选取一个子集并实例化模型；
2. 使用这个模型来归类数据点，判断为内点还是外点；
3. 迭代步骤 1 和 2；
4. 选出最大的内点集合并为集合中的点重新预测匹配模型



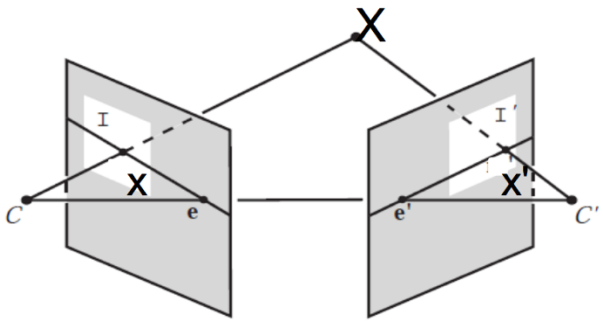
Triangulation

线性方法：

- 通常来讲，射线 C→x 与射线 C'→x'并不完全相交
- 该问题可以通过 SVD 分解找到一个系统方程的最小平方解

设 P,P'为相机矩阵，x, x'为匹配点对

- 对点和投影矩阵进行预处理
- 构造一个矩阵 A
- [U, S, V] = svd(A)
- X = V(:,end)



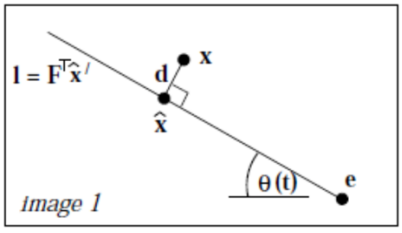
$x = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, x' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$,易知$x \times (PX) = 0$ 且 $x' \times (P'X) = 0$

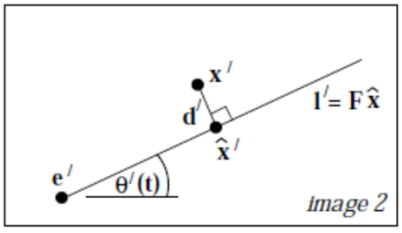
则 AX = 0, 其中$A = \begin{bmatrix} up_3^T - p_1^T \\ vup_3^T - p_2^T \\ u'p'^T_3 - p'^T_1 \\ v'p'^T_3 - p'^T_2 \end{bmatrix}$

非线性方法：

满足$x^T Fx = 0$ 的同时最小化投影误差

$C(X) = d(x, \hat{x})^2 + d(x', \hat{x}')^2$





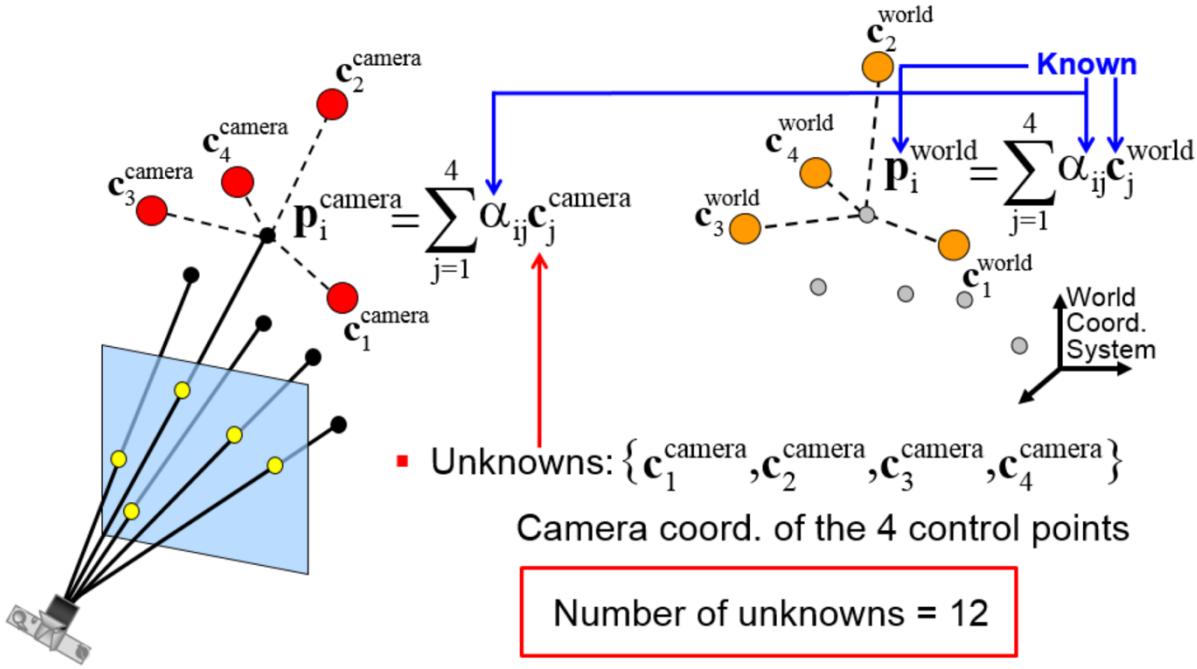
- 解是一个关于 t 的 6-degree 多项式，最小化$d(x, l(t))^2 + d(x', l'(t))^2$

Perspective n Point Problem

- 透视 n 点问题（PnP）是摄像机标定的一个基本问题（也称之为位置估计），它起源于摄像机标定，是根据一个 n 点对应的场景对象来决定摄像机的位置与方位。PnP 问题主要研究摄像机标定的外部参量，它可以提高摄像机标定的速度和精度，从而提高三维重建的效率。
- Efficient PnP:一种非迭代的方法

  首先，将 3D 点用 4 个控制点的权重和表示出来，接着

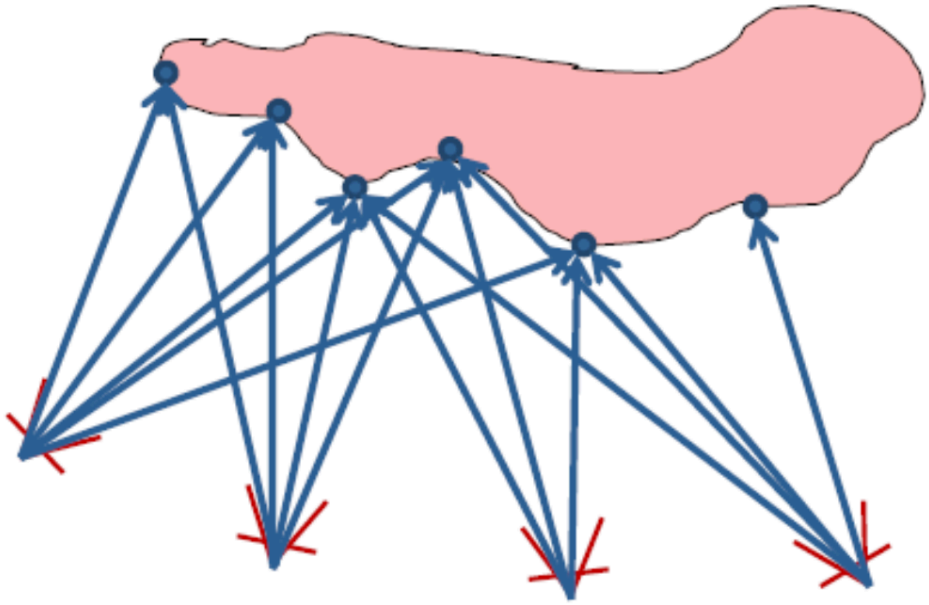
  1. 控制点的坐标是未知的（共 12 个未知数）
  2. 将 3D 点投影到图片上→用控制点坐标构建一个线性系统
  3. 将控制点的坐标用空特征向量的线性组合表示出来
  4. 权重（系数β_i）是新的未知量（不多于 4 个）
  5. 增加刚性约束来获得用β_i表示的二次方程
  6. 通过他们的数量解β_i（线性化或重线性化）



Bundle Adjustment

- Bundle Adjustment:优化 6DOF 相机姿态及 3D 特征点

$$E(P,X) = \sum_{i=1}^{m}\sum_{j=1}^{n} D(x_{ij}, P_i X_j)^2$$



  - 非线性；
  - 稀疏的；
  - 优化

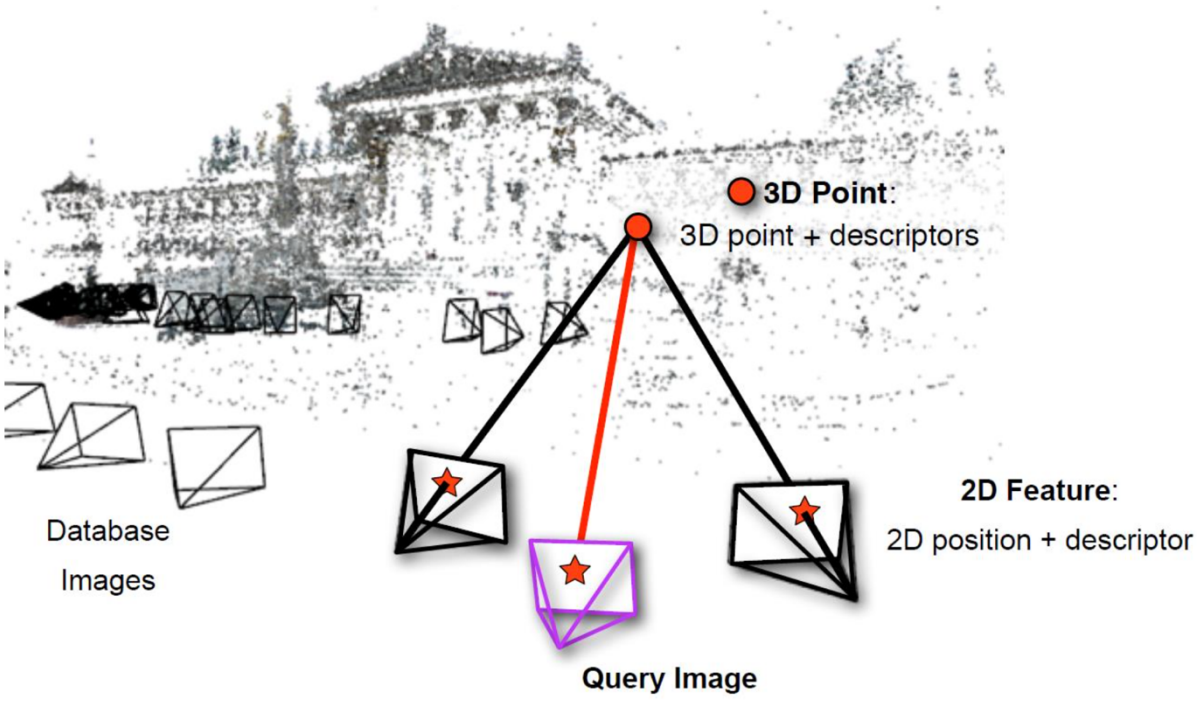
- 算法：
  - Schur Complement；
  - Levenberg-Marquardt iteration

Image-Based Relocalization

Pipeline:

- Key Frame and its Camera Pose Stored;
    - Frame id, image feature and camera pose.
- Extract Local Features;
    - Feature original or coded.
- Establish 2D-2D or 3D-to-2D Matches;
    - Single camera: 3d-to-2d;
    - Stereo or depth camera: 2d-to-2d;
- Camera Pose Estimation.
    - Refine by 3-d point estimation;
    - 2-d features along with their depths.

3D-to-2D Relocalization:



Feature Matching-based Relocalization:

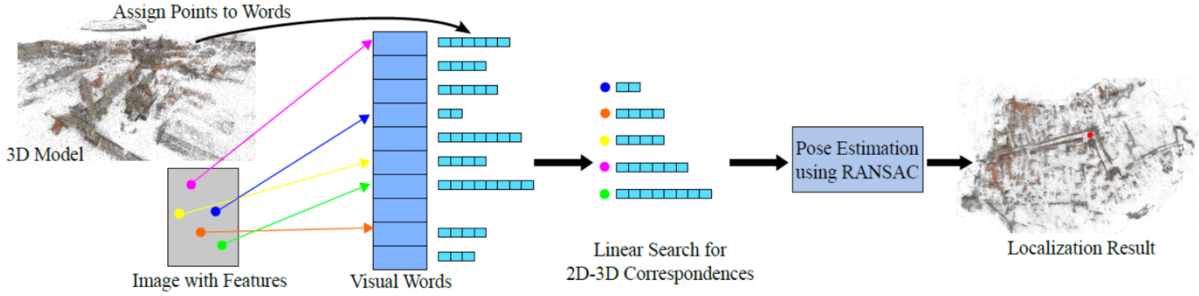
- Feature detection and descriptor for key frames or new frames which require relocalization:

    - Harris corner or FAST detection;
    - SURF (64-d vector), SIFT, BRIEF or ORB;
- Feature matching for pose recovery:
    - Brute force or ANN search;
    - Pose difference is small enough.
- Key frame selection:
    - Pose change enough big;

Randomized Tree-based ANN Search for Relocalization:

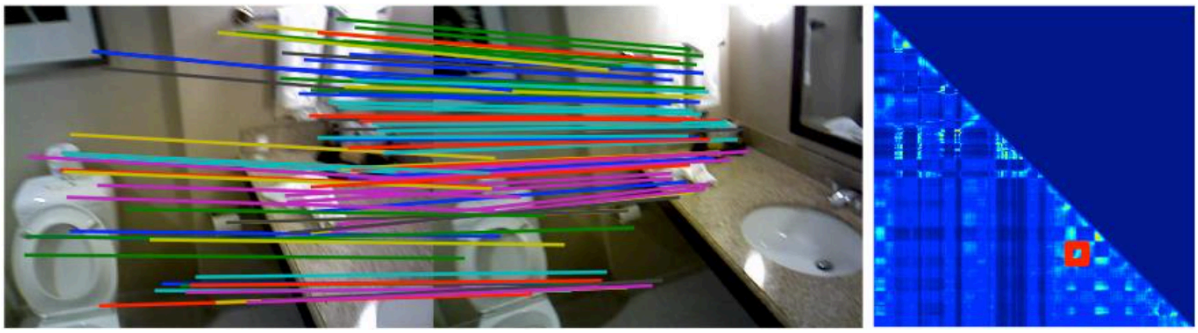
- The kd-tree data structure is based on a recursive subdivision of space into disjoint hyper-rectangular regions called cells;
- ANN (approximate NN) search: Limit number of neighboring k-d tree bins to explore;
- Randomised kd-tree forest: Split by picking from multiple (5) top dimensions with the highest variance in randomized trees, to increase the chances of finding nearby points;
- Best Bin First in K-d tree: a variant of normal K-d tree
    - A branch-and-bound technique that maintains an estimate of the smallest distance from the query point to any of the data points down all of the open paths (priority queue).
- Priority queue-based k-means tree: partition the data at each level into K distinct regions using k-means, then applying it also recursively to the points in each region;
    - The priority queue is sorted in increasing distance from the query point to the boundary of the branch being added to the queue.

Bag of Visual Words for Image-based Relocalization:



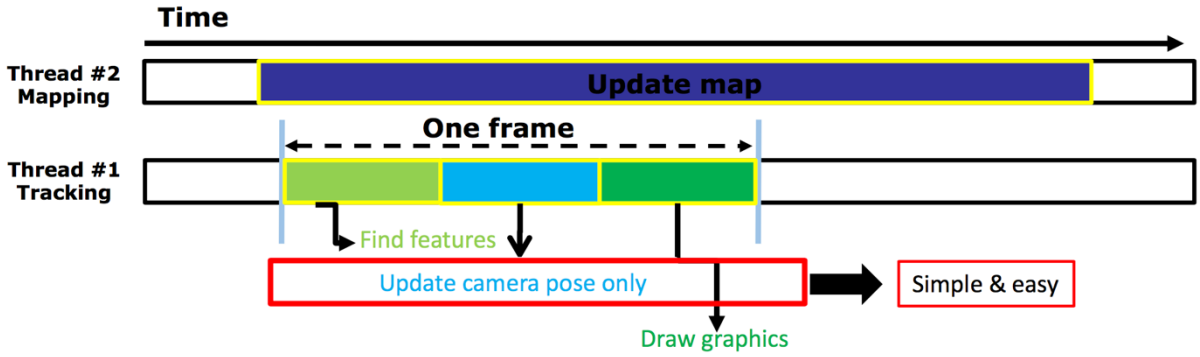
BoVW-based Relocalization in FAB-MAP:

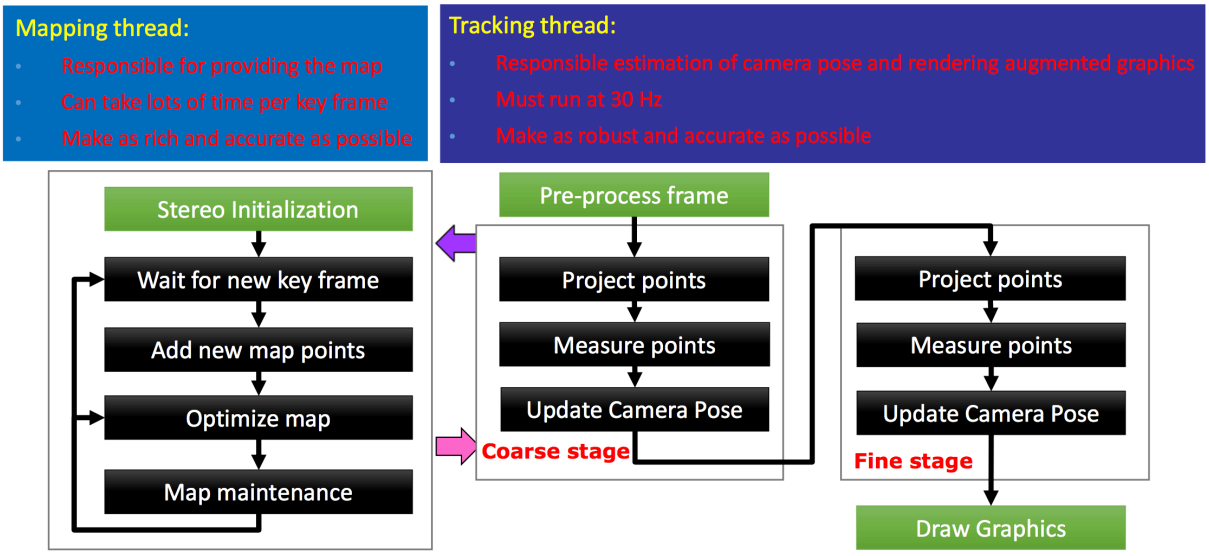
- Codebook building from training data:
  - Feature extraction: FAST corner detection + SURF/ORB feature descriptor;
  - Feature clustering for visual vocabulary: incremental K-means;
  - Construct co-occurrence statistics of visual words by Chow Liu tree.
    - A naive Bayes approximation by a maximum weight spanning tree.
- Visual pattern from key frames or new frames for relocalization:
  - Feature extraction: same as above;
  - Mapping descriptors to bags of words by ANN-based (kd-tree) methods;
- Pattern matching for pose recovery: descriptor mapped to codebook;
  - Sampling in training data or mean field optimization to find the best match.



PTAM

- 使用 map 中大量的点（几千个）；
- 不会对每一帧都更新 map，仅针对关键帧；
- 将 tracking 和 mapping 分到了两个线程去做；
- 为 tracking 和 mapping 创建金字塔等级；
  - 检测 FAST corner；
  - 使用 motion model 来更新相机姿态
- 有条件的增加关键帧/新的 map 点
  - baseline 要足够长且对极搜索要好；
- 用基于 batch 的方法优化 map：Bundle Adjustment
- Stereo initialization densely
  - 使用 five-point-pose 算法
- map 的维持依靠数据关联和外点的重复测试





2.3 Dense Reconstruction (optical flow based)

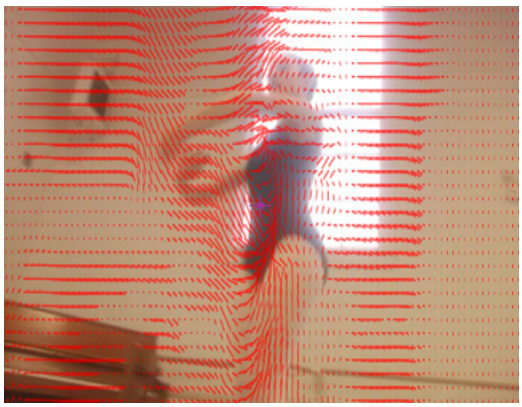
Optical Flow

光流（optic flow）是什么呢？其实这种视觉现象我们每天都在经历。从本质上说，光流就是你在这个运动着的世界里感觉到的明显的视觉运动。例如，当你坐在火车上，然后往窗外看。你可以看到树、地面、建筑等等，他们都在往后退。这个运动就是光流。而且，我们都会发现，他们的运动速度居然不一样？这就给我们提供了一个挺有意思的信息：通过不同目标的运动速度判断它们与我们的距离。一些比较远的目标，例如云、山，它们移动很慢，感觉就像静止一样。但一些离得比较近的物体，例如建筑和树，就比较快的往后退，然后离我们的距离越近，它们往后退的速度越快。

光流除了提供远近外，还可以提供角度信息。与我们的眼睛正对着的方向成 90 度方向运动的物体速度要比其他角度的快，当小到 0 度的时候，也就是物体朝着我们的方向直接撞过来，我们就是感受不到它的运动（光流）了，看起来好像是静止的。当它离我们越近，就越来越大（当然了，我们平时看到感觉还是有速度的，因为物体较大，它的边缘还是和我们人眼具有大于 0 的角度的）。

以上是对光流通俗的解释，其研究性的定义是 Gibson 在 1950 年首先提出来的。

> 光流是空间运动物体在观察成像平面上的像素运动的瞬时速度，是利用图像序列中像素在时间域上的变化以及相邻帧之间的相关性来找到上一帧跟当前帧之间存在的对应关系，从而计算出相邻帧之间物体的运动信息的一种方法。一般而言，光流是由于场景中前景目标本身的移动、相机的运动，或者两者的共同运动所产生的。



当人的眼睛观察运动物体时，物体的景象在人眼的视网膜上形成一系列连续变化的图像，这一系列连续变化的信息不断“流过”视网膜（即图像平

面），好像一种光的“流”，故称之为光流（optical flow）。光流表达了图像的变化，由于它包含了目标运动的信息，因此可被观察者用来确定目标的运动情况。

研究光流场的目的就是为了从图片序列中近似得到不能直接得到的运动场。运动场，其实就是物体在三维真实世界中的运动；光流场，是运动场在二维图像平面上（人的眼睛或者摄像头）的投影。

那通俗的讲就是通过一个图片序列，把每张图像中每个像素的运动速度和运动方向找出来就是光流场。那怎么找呢？咱们直观理解肯定是：第 t 帧的时候 A 点的位置是(x1, y1)，那么我们在第 t+1 帧的时候再找到 A 点，假如它的位置是(x2,y2)，那么我们就可以确定 A 点的运动了：(ux, vy) = (x2, y2) - (x1,y1)。那怎么知道第 t+1 帧的时候 A 点的位置呢？ 这就存在很多的光流计算方法了。

1981 年 Horn 和 Schunck 创造性地将二维速度场与灰度相联系，引入光流约束方程，得到光流计算的基本算法。人们基于不同的理论基础提出各种光流计算方法，算法性能各有不同。Barron 等人对多种光流计算技术进行了总结，按照理论基础与数学方法的区别把它们分成四种：基于梯度的方法、基于匹配的方法、基于能量的方法、基于相位的方法。近年来神经动力学方法也颇受学者重视。目前在 OpenCV 中主要实现了以下光流算法：

1）calcOpticalFlowPyrLK

通过金字塔 Lucas-Kanade 光流方法计算某些点集的光流（稀疏光流）。理解的话，可以参考这篇论文：”Pyramidal Implementation of the Lucas Kanade Feature TrackerDescription of the algorithm”

2）calcOpticalFlowFarneback

用 Gunnar Farneback 的算法计算稠密光流（即图像上所有像素点的光流都计算出来）。它的相关论文是："Two-Frame Motion Estimation Based on PolynomialExpansion"

3）CalcOpticalFlowBM

通过块匹配的方法来计算光流。

4）CalcOpticalFlowHS

用 Horn-Schunck 的算法计算稠密光流。相关论文好像是这篇：” Determining Optical Flow”

5）calcOpticalFlowSF

这一个是 2012 年欧洲视觉会议的一篇文章的实现："SimpleFlow: A Non-iterative, Sublinear Optical FlowAlgorithm"，工程网站是：http://graphics.berkeley.edu/papers/Tao-SAN-2012-05/ 在 OpenCV 新版本中有引入。

稠密光流需要使用某种插值方法在比较容易跟踪的像素之间进行插值以解决那些运动不明确的像素，所以它的计算开销是相当大的。而对于稀疏光流来说，在他计算时需要在被跟踪之前指定一组点（容易跟踪的点，例如角点），因此在使用 LK 方法之前我们需要配合使用 cvGoodFeatureToTrack() 来寻找角点，然后利用金字塔 LK 光流算法，对运动进行跟踪。但个人感觉，对于少纹理的目标，例如人手，LK 稀疏光流就比较容易跟丢。

至于他们的 API 的使用说明，我们直接参考 OpenCV 的官方手册就行：

http://www.opencv.org.cn/opencvdoc/2.3.2/html/modules/video/doc/motion_analysis_and_object_tracking.html - calcopticalflowfarnebackIJCV2011 有一篇文章，《A Database and Evaluation Methodology for Optical Flow》里面对主流的光流算法做了简要的介绍和对不同算法进行了评估。网址是：http://vision.middlebury.edu/flow/

感觉这个文章在光流算法的解说上非常好，条例很清晰。想了解光流的，推荐看这篇文章。另外，需要提到的一个问题是，光流场是图片中每个像素都有一个 x 方向和 y 方向的位移，所以在上面那些光流计算结束后得到的光流 flow 是个和原来图像大小相等的双通道图像。

光流学习代码：

- https://github.com/ieted/3DReconstruction-Primer
- http://people.csail.mit.edu/celiu/OpticalFlow/

Disparity Map & Depth Map

Disparity Map：

Disparity map refers to the apparent pixel difference or motion between a pair of stereo image. Objects that are close to you will appear to jump a significant distance while objects further away will move very little. That motion is the disparity.

In a pair of images derived from stereo cameras, you can measure the apparent motion in pixels for every point and make an intensity image out of the measurements.

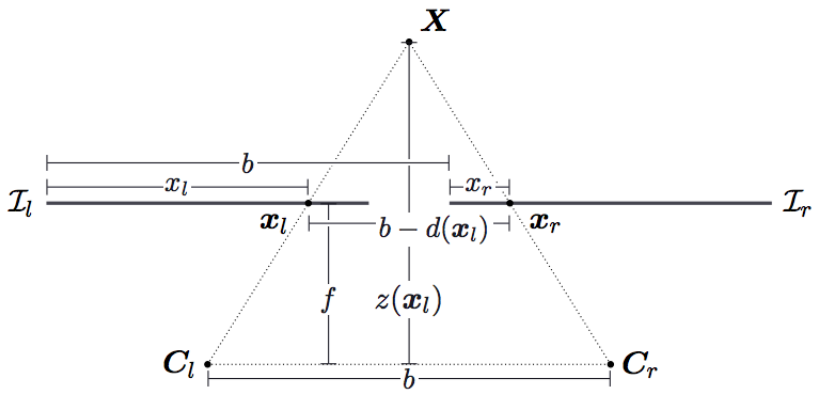
Depth Map：

In 3D computer graphics a depth map is an image or image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint.

*Why sometimes inverse depth map?

Features like the sun and clouds and other things that are very far off would have a distance estimate of infinity. This can cause a lot of problems. To get around it, the inverse of the distance is estimated. All of the infinities become zeroes which tend to cause fewer problems.

Disparity Map 与 Depth Map 的关系：



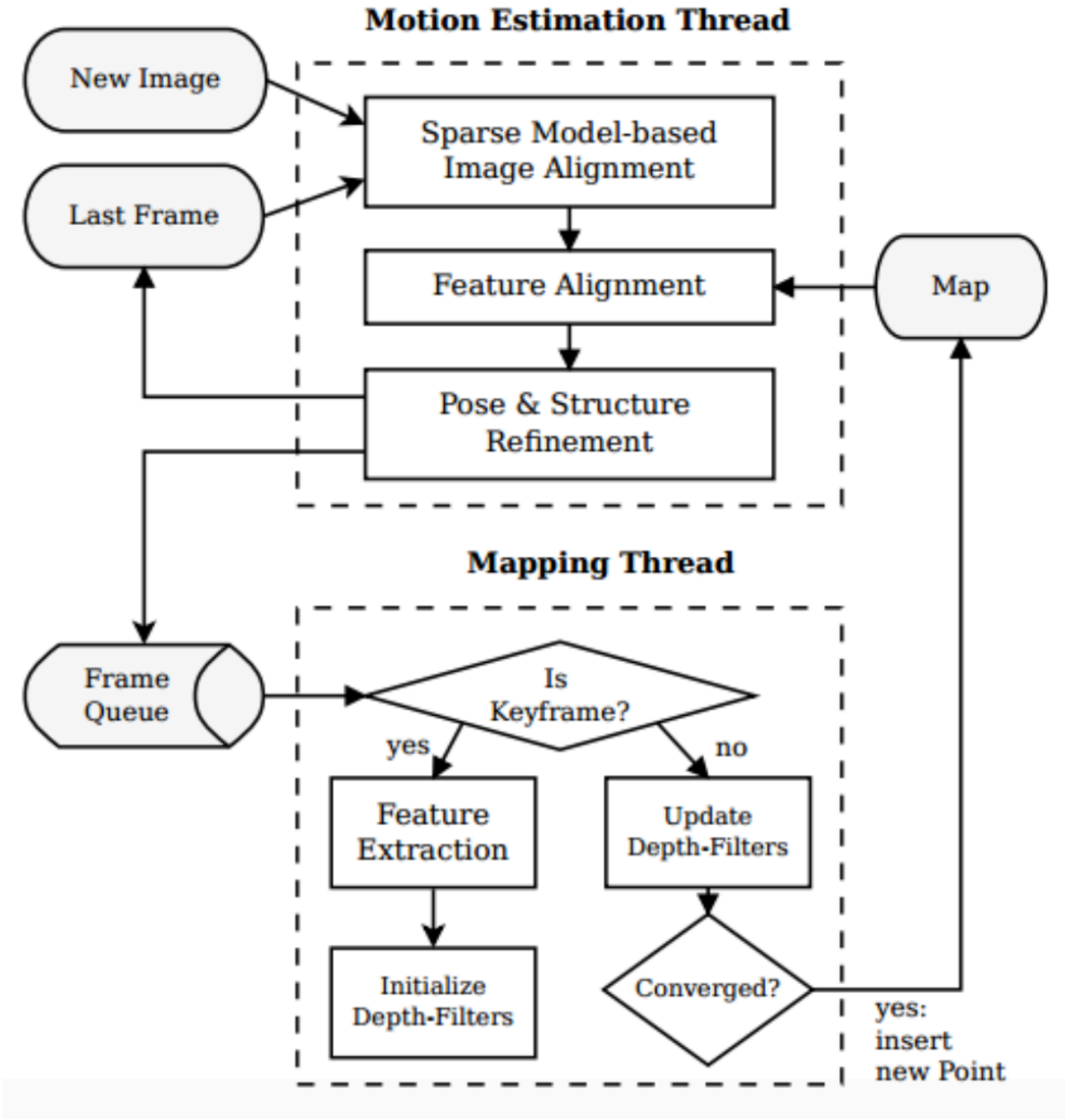
$$\frac{b - d(\boldsymbol{x}_l)}{z(\boldsymbol{x}_l) - f} = \frac{b}{z(\boldsymbol{x}_l)} \Longleftrightarrow z(\boldsymbol{x}_l) = \frac{fb}{d(\boldsymbol{x}_l)}$$

有上式可知，当两相机间的距离和焦距已知时，disparity 信息可以与 depth 信息互相转换。

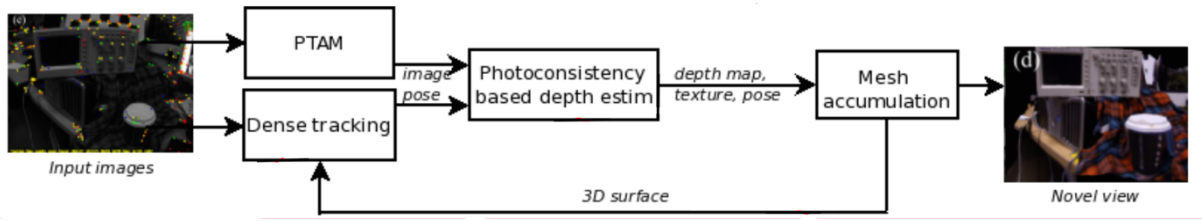
Disparity Map 学习代码：https://github.com/ieted/3DReconstruction-Primer

Regularized Monocular Depth Estimation

- Combine Bayesian estimation and convex optimization for processing;
    - Probabilistic depth map handles the uncertainty and allows sensor fusion.
- Depth estimated by triangulating a reference view and the last view;
    - Smoothness of depth map enforced by a regularization term (TV).
- Camera pose estimation: Semi-direct monocular visual Odometry (SVO);
    - Particle/Kalman filter or LS optimization (such as key frames-based or batch Bundle Adjustment) solution;
    - SVO: Feature correspondence from direct motion estimation with sparse depth map estimate (motion estimation and mapping together), still a sparse model-based alignment method, related to model-based dense method;
        - Initial guess by direct method and then followed by feature-based refinement with reprojection error;
        - A prob. depth filter initialized by sparse features, then posterior depth estimate by Bayesian rule and update sequentially.
- Scene reconstruction: update after initial estimation with two frames.
    - Region growing (low textured, sparse data sets) or occlusion-robust photo consistency (dense stereo, 3-d segment.);
    - Inverse depth: A 6-d vector for parallax, a relation btw disparity and depth.
        - The ray from the 1st camera position, the current camera orientation and the inverse depth;
        - Switch from inverse depth to XYZ (3-d point) for high parallax features.
    - Extend SVO with dense map estimate within Bayesian framework.


Motion Estimation Thread
New Image
Last Frame
Sparse Model-based Image Alignment
Feature Alignment
Map
Pose & Structure Refinement
Mapping Thread
Frame Queue
Is Keyframe?
yes
no
Feature Extraction
Update Depth-Filters
Initialize Depth-Filters
Converged?
yes: insert new Point

DTAM



在 Dense tracking 阶段：

Inputs:

- 3D texture model of the scene
- Pose at previous frame

Tracking as a registration problem

- First rotation estimation: previous is aligned on the current to estimate a coarse inter-frame rotation;
- Estimated pose is used to project the 3D model into 2.5D image;
- The 2.5D image is registered with the current frame to find the current camera pose;
- Two template matching problems for minimize SSD: direct and inverse compositional.

在 Photo consistency based depth estimation 阶段：

Principle:

- S depth hypothesis are considered for each pixel of the reference image I_r
- Each corresponding 3D point is projected onto a bundle of images I_m
- Keep the depth hypothesis that best respects the color consistency from the reference to the bundle of images

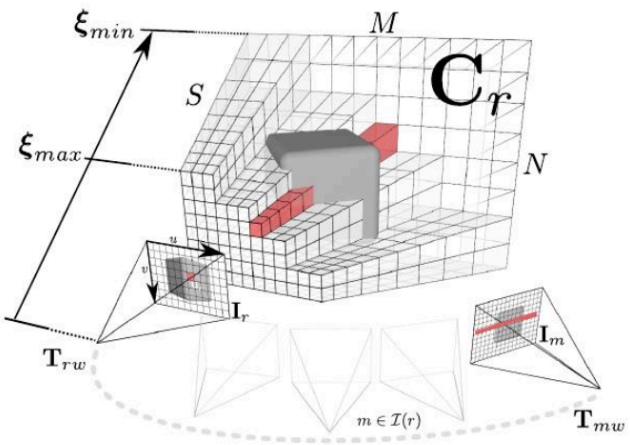
Formulation:

$$\mathbf{C}_r(\mathbf{u},d)=\frac{1}{|\mathcal{I}(r)|}\sum_{m\in\mathcal{I}(r)}\|\rho_r\left(\mathbf{I}_m,\mathbf{u},d\right)\|_1 \qquad \begin{aligned}\pi(\mathbf{x}_c)&=(x/z,y/z)^T\\ \pi^{-1}(\mathbf{u},d)&=\frac{1}{d}K^{-1}\dot{\mathbf{u}}\end{aligned}$$

- u, d: pixel position and depth hypothesis
- $|\mathcal{L}((r))|$: number of valid reprojection of the pixel in the bundle
- ρ_r: photometric error between reference and current image $\rho_r(I_m, u, d) = I_r(u) - I_m(\pi(KT_{mr}\pi^{-1}(u, d)))$

Problem (Depth Estimation): Uniform regions in reference image do not give discriminative enough photometric error;

Solution (Primal Dual): Assume that depth is smooth on uniform regions, use total variation approach where depth map is the functional to optimize, where photometric error defines the data term and the smoothness constraint defines the regularization.



Problem in SSD minimization:

- Align template T(x) with input I(x).

Formulation:

Find transform **W** (x; p) that best maps the pixels of the templates into the ones of the current image, minimize:

$$\sum_X [I(W(x;p)) - T(x)]^2$$

Formulation of a variational approach:

$$E_{\boldsymbol{\xi}} = \int_\Omega \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_\epsilon + \lambda \mathbf{C}(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})) \right\} d\mathbf{u}$$

$$g(\mathbf{u}) = e^{-\alpha \|\nabla \mathbf{I}_r(\mathbf{u})\|_2^\beta}$$

- First term as regularizer with Huber norm, second term as photometric error;
- Problem: optimizing this equation directly requires linearising of cost volume, expensive and cost volume has many local minima.

Approximation:

$$\mathbf{E}_{\boldsymbol{\xi},\boldsymbol{\alpha}} = \int_{\Omega} \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \frac{1}{2\theta} \left(\boldsymbol{\xi}(\mathbf{u}) - \boldsymbol{\alpha}(\mathbf{u})\right)^2 + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\alpha}(\mathbf{u})\right) \right\} \mathrm{d}\mathbf{u} \,.$$

- Introduce $\alpha(u)$ as an auxiliary variable, can optimized with heuristic search
- Second terms brings original and auxiliary variable together

Reformulation of regularization with primal dual method:

- Dual variable p is introduced to compute the TV norm:

$$TV(u) = \max \left\{ \int_{\Omega} \boldsymbol{p} \cdot \nabla u \, d\Omega : \|\boldsymbol{p}\| \leq 1 \right\} \text{ for } \nabla u \neq 0, \; \|\boldsymbol{p}\| = \left\| \frac{\nabla u}{|\nabla u|} \right\| = 1$$

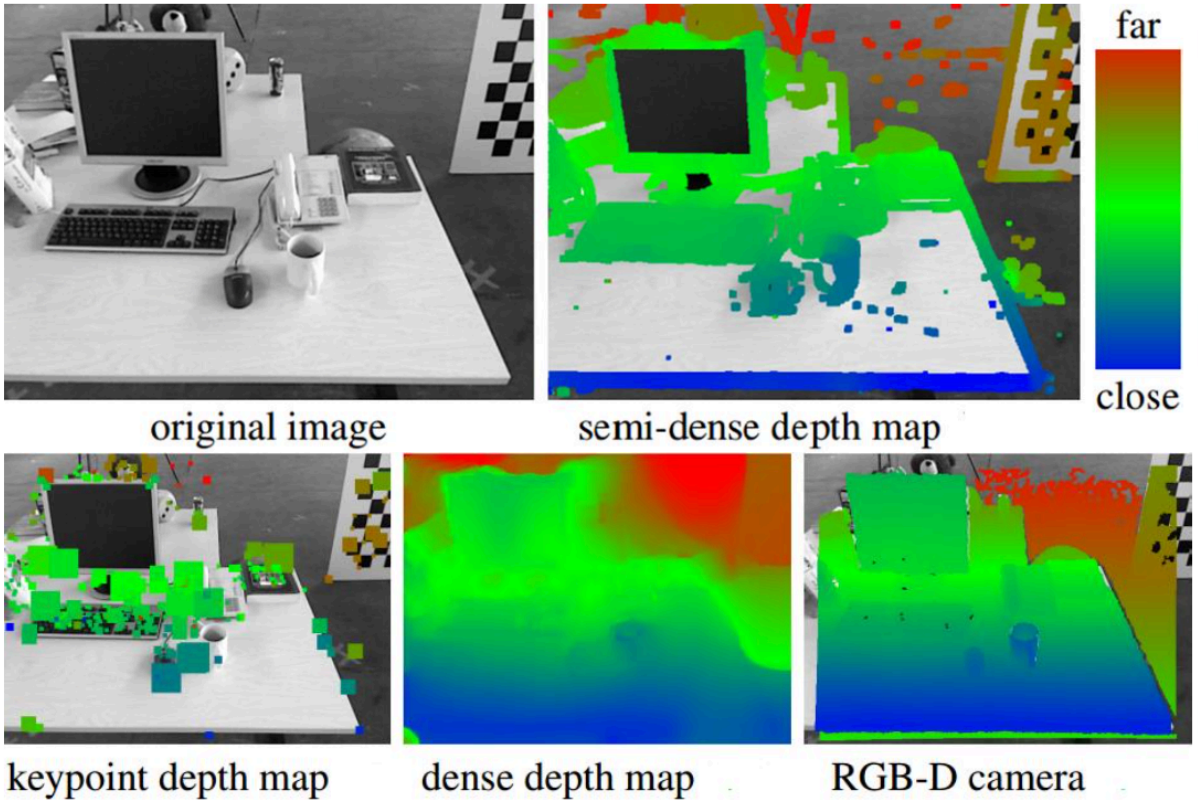
$$\boldsymbol{p} \cdot \nabla u = \frac{\nabla u}{|\nabla u|} \nabla u = |\nabla u|$$

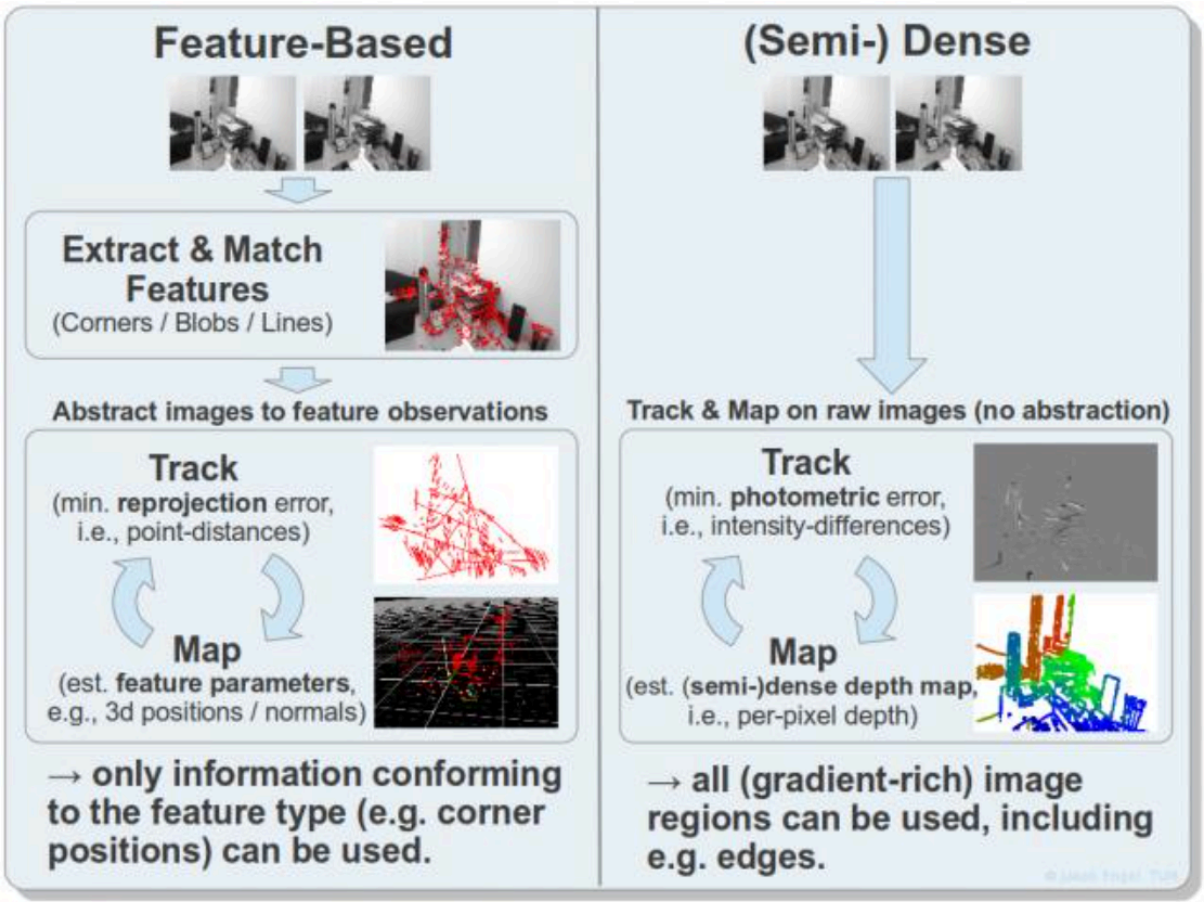
Visual Odometry

- Visual Odometry (V0): The process of estimating the ego-motion of an agent using only the input of a single or multiple camera;
- VO incrementally estimates pose of the agent through examination of the changes that motion induces on the images of its onboard cameras;
- VO: a special case of SfM (relative camera pose and 3-d structure);
    - VO focus on 3-d motion of the camera sequentially to recover the full trajectory;
- Visual SLAM vs. VO:
    - VO only for recovering path incrementally, pose after pose, and potentially optimizing only over the last n poses of the path (windowed bundle adjustment);
    - VO only with local consistency of trajectory, and SLAM with global consistency;
    - vSLAM needs understanding when a loop closure occurs and efficiently integrating this new constraint into the current map;
    - VO can be used as a building block for a complete SLAM.

Semi-Dense SLAM

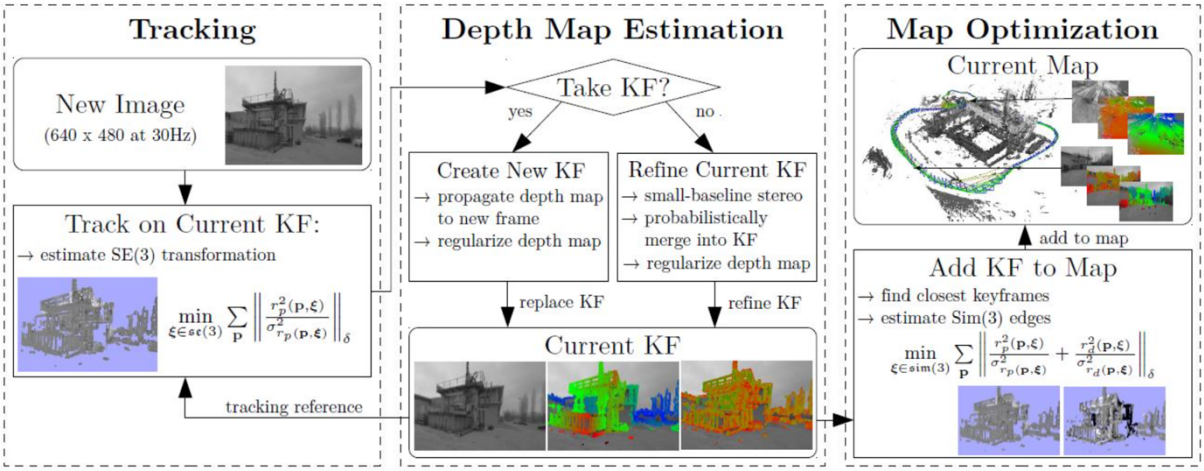
- Continuously estimate a semi-dense inverse depth map for the current frame, which in turn is used to track the motion of the camera using dense image alignment.
    - Estimate the depth of all pixels which have a non-negligible image gradient.
- Use the oldest frame, where the disparity range and the view angle within a certain threshold;
    - If a disparity search is unsuccessful, the pixel's "age" is increased, such that subsequent disparity searches use newer frames where the pixel is likely to be still visible.
- Each estimate is represented as a Gaussian probability distribution over the inverse depth;
    - Minimize photometric and geometric disparity errors with respect to pixel-to-inverse depth ratio.
- Propagate this information over time, update with new measurements as new images arrive.
    - Based on the inverse depth estimate for a pixel, the corresponding 3D point is calculated and projected into the new frame, providing an inverse depth estimate in the new frame;
    - The hypothesis is assigned to the closest integer pixel position – to eliminate discretization errors, the sub-pixel image location of the projected point is kept, and re-used for the next propagation step;
    - Assign inverse depth value the average of the surrounding inverse depths, weighted by respective inverse variance;
    - Keep track of validity of each inverse depth hypothesis to handle outliers (removed if needed).


far
close
original image
semi-dense depth map
keypoint depth map
dense depth map
RGB-D camera


Feature-Based
Extract & Match Features
(Corners / Blobs / Lines)
Abstract images to feature observations
Track
(min. reprojection error, i.e., point-distances)
Map
(est. feature parameters, e.g., 3d positions / normals)
→ only information conforming to the feature type (e.g. corner positions) can be used.
(Semi-) Dense
Track & Map on raw images (no abstraction)
Track
(min. photometric error, i.e., intensity-differences)
Map
(est. (semi-)dense depth map, i.e., per-pixel depth)
→ all (gradient-rich) image regions can be used, including e.g. edges.

Large Scale Direct SLAM

- Goal: to build large-scale, consistent maps of the environment;
- Initialization: a first keyframe with a random depth map and large variance, given sufficient translational camera movement in the first seconds, converge to a correct depth configuration after a couple of keyframe propagations;
- Pose estimation based on direct scale-aware image alignment on similarity transform;
- Keyframes are replaced when the camera moves too far away from the existing map;
- Once a frame is chosen as keyframe, its depth map is initialized by projecting points from the previous keyframe into it, followed by regularization and outlier removal;
- The depth map is scaled to have a mean inverse depth of one; then refined by filtering over a large number of pixelwise small-baseline stereo;
- 3D reconstruction in real-time as pose-graph of keyframes with semi-dense (gradient) depth;
- Probabilistically consistent incorporation of uncertainty of the estimated depth;
- Loop Closure is detected by appearance-based mapping (BovW with co-occurrence statistics) .



3. Stereo Camera Based

双目摄像机相较于单目摄像机提供了两相机间的相对位置信息，简化了求解的几何模型。

（待补充……）

4.RGB-D Camera Based

相比较传统的 RGB 相机，RGB-D 相机还可额外提供图像中像素的深度信息，利用这些深度信息可以用来稠密重建某一视角下的三维场景。

（待补充……）