点云匹配算法

作者：范天行
学号：11821030

1. 简介

在计算机视觉领域中，点云是一种常见的几何结构。比如，RGB-D三维重建中，为了重建出一个复杂结构（物体、房间）的模型，会从图像与深度图中恢复出三维的稠密点云，对一系列的点云计算位姿，进行局部、全局的优化、融合后，重建模型面片，最终形成三维模型。 在这个过程中点云位姿估计的精确与否直接影响了模型生成的正确性。一个好的模型需要点云进行精确的匹配，如果仅使用特征点，如ORB、SIFT等特征点算法去匹配特征点获得图像间的变换矩阵，并进行逆投影变换，将二维图像投影到三维空间中，从而获得图像间的三维变换矩阵，得到的精度是远远不够的。模型的匹配效果会非常差。

为了获得较高精度的点云匹配结果，专业学者们经过了广泛的研究，发明了许多的点云匹配算法。常见的算法有ICP[Besl92]（Iterative Closest Point），即迭代就近点法；以及基于高斯混合模型与EM算法的匹配算法，如FilterReg[Gao198]在速度与精度上都超过了基于feature的ICP，Lawin等人的工作[Lawin18]在Lidar扫描的数据集上得到了不依赖于点云密度的鲁棒结果。还有一些脱离了ICP的对应点-刚体变换迭代，以及EM算法的E步M步迭代的算法，比如FGR[zhou16]是一种使用FPFH（快速点特征直方图）进行快速全局点云匹配的算法，Super4pcs[Mellado14]在不使用ICP进行调整的情况下，使用仿射不变性等几何信息就可以得到非常好的结果。

在这篇文章中因为篇幅所限，只介绍三种基于ICP改进的点云匹配算法，并且默认读者都了解ICP。这三篇文章分别为：

- Colored Point Cloud Registration Revisited[Park17]
- Fast Descriptors and Correspondence Propagation[Lei17]
- SDRSAC[Le19]

在这篇文章中，我将介绍传统的ICP存在的三个问题，并分析这三种算法是从什么层面、什么角度解决这些问题的。

2.传统ICP算法存在的问题问题

传统的ICP算法，如Point to Point ICP[Besl92]、Point to Plane ICP[Chen92] 虽然简单优雅而有效，但其仍存在相当多的问题。 主要问题有如下：

1. 对初始矩阵敏感。
   ICP算法是非凸的，他有很多局部最优解。如果初始的刚体变换矩阵设置的不好，ICP算法很容易收敛到局部最优解，而非全局最优解。这个问题被科学家广泛研究，并提出了很多解决的方法。形成的

算法有GOICP[Yang16]、Super4pcs[Mellado14]等。这些算法跳出了使用局部信息，如特征点等方法，而是从全局的角度去看待这个问题，从而获得合适的初始刚体变换矩阵。

2. 对对应点集敏感。
   ICP算法对对应点集十分敏感。这个问题和1.提到的比较类似。因为有了合适的对应点集，就可以获得全局最优解；而有了走向全局最优解的变换矩阵，自然可以得到很好的对应点集。一些局部的方法也尝试去获得好的对应点集，如TrICP[chetverikov02]，在overlapping较小的数据集上，获得了比较好的效果。

3. 无法用到颜色信息。
   传统的ICP算法不使用颜色信息，这一点不符合在RGB-D三维重建的需求。RGB-D图片、有色雷达数据具有颜色信息，而点到点ICP完全无法使用。

3. 改进的ICP算法

3.1 Colored Point Cloud Registrarion Revisited

这是一篇ICCV2017的文章。J Park[Park17]提出了这种方法，在ICP的求解过程中加入了关于颜色的残差函数，从而使ICP算法有了求解颜色信息的能力。这种方法针对有色点云，引入了颜色信息，使原有的算法在RGB颜色的约束下，可以取得更好的效果。

3.1.1 算法流程与介绍

这篇文章使用的算法的原型可以看作是Point to Plane ICP，其通过修改Point to Plane ICP的能量函数，加入了对点云颜色的使用部分。
总体算法的能量函数可以表示为:

$$E(T) = (1-\sigma)E_C(T) + \sigma E_D(T)$$

其中$E_D(T)$就是前面所提到的Point to Plane ICP的能量函数，而$E_C(T)$是论文新引入的RGB颜色信息的能量函数，其表示形式为：

$$E_C(T) = \sum(C_q(p') - C(p)) = \sum(C_q(Rp+t) - C(p))$$

C表示点的颜色，而$C_q(p') = C_q(Rp+t)$为点p在经过刚体变换后，投影到点q所在的切平面后的对应颜色。具体来说，其形式为:

$$C_q(p') = C(q) + d_q^T f(p')$$

$$f(p') = p' - n_q(p'-q)^T n_q$$

$-$

$f(p)$为投影函数，因为$d_p^T n_p = 0$,其实也可以不用写这么复杂。
而梯度信息d_p可以使用最小二乘法进行估计，其计算方式如下:

$$L(d_p) = \sum (C_p(f(p_{neibors}) - p) - C(p_{neibors}))^2$$
$$= \sum (C(p) - C(p_{neibors}) + d_p^T (f(p_{neibors}) - p)$$

之后，使用线性最小二乘法就可以求得d_p的最优解。
最后，可以对

$$E(T) = (1 - \sigma) E_C(T) + \sigma E_D(T)$$

做一次非线性最小二乘，使用高斯牛顿法求解。解法与Point to Plane ICP一致，在这里就不赘述了。

3.1.2 创新与问题

这篇文章的算法已经被集成在Open3d的算法库中，在处理有色点云时，具有非常好的效果。文章中使用到的超参数σ在Open3d中被预设为了0.968；在论文中提到，其取值可以通过Grid Search得到。

这种方法在ICP中引入了颜色信息，但是其本质仍然是一种局部的方法。虽然在论文中，作者提到可以使用Coarse-to-Fine的方式，建立金字塔下采样，从底层金字塔开始，使用ICP计算粗略的刚体变换矩阵，并将结果作为上层金字塔的预估值，进行连续迭代，从而获取类似于全局ICP的结果，但是其精度在两个点云间旋转角度大于30°、平移距离大于30cm后飞速降低。因此若要使用这种方法，仍然需要对两个初始点云进行粗配准，再进行点云注册。

3.2 Fast Descriptors and Correspondence Propagation

这篇[Lei17]是T-IP2017年的文章，其介绍了较为快速的全局方法，用于进行ICP算法的粗配矩阵估计。无论是基于GMM的使用概率的点云匹配算法，还是ICP，都存在陷入局部最优解的问题，而传统的局部特征的描述子容易受到噪声的干扰，使匹配得到错误的结果。因此这篇文章尝试建立一种用于点云匹配的新式描述子，这种描述子需要有效且鲁棒。

3.2.1 基于特征值的描述子

基于以上原因，这篇文章介绍了一种基于特征值的描述子，其基础算式为：

$$C_l = \frac{1}{S_l} \sum_{x_i \in S_l} (x_i - x)(x_i - x)^T for x \in X$$

其中，S_l为点x的大小为r_l的邻域。论文推荐将L设置为4，求取四个不同大小下邻域的特征值，并将其组合形成描述子。在接下来的篇幅中，我将会讲述这个算法的通用过程。

可以发现，这个公式就是用来求点云中点的法向量的公式。这篇论文对于每个邻域，求这个公式的特征值$\lambda_1, \lambda_2, \lambda_3$,以及对应特征向量$n_1, n_2, n_3$。因为$\lambda$不随点云的刚体变换改变，$n_1, n_2$与点云的密度分布关系较大，而$n_3$作为点云的法向，具有其稳定性，所以文章选择$\lambda_1, \lambda_2, \lambda_3$以及$n_3$构建描述子。

λ不随刚体变换而改变是显然的，证明如下：

$$C_l^{'} = \frac{1}{S_l}\sum_{x_i \in S_l}(Rx_i - Rx)(Rx_i - Rx)^T$$

$$= \frac{1}{S_l}\sum_{x_i \in S_l} R(x_i - x)(x_i - x)^T R^T$$

$$= RC_l R^T$$

因为R是正交矩阵，所以$C_l^{'}$与C_l相似，因此特征值不变。
这篇论文随后将特征描述子表示为(N, D)的形式，其中:

$$N = (n_{3,1}, n_{3,2}, ...n_{3,L})^T$$

$$D = (\Delta s_1, \Delta s_2, ..., \Delta s_{L-1})^T$$

$$s_l = \frac{(\lambda_1, \lambda_2, \lambda_3)^T}{\lambda_1 + \lambda_2 + \lambda_3}, \Delta s_{l-1} = s_l - s_{l-1}$$

3.2.2 Corresponence Propagation

随后进行的，就是根据特征描述子求取初始变换矩阵的过程。论文使用了一种被其称为Correspondence Propagation的过程，整体流程如下：

1. 从初始的源点云X中选取一个点x,在目标点云Y中寻找其最优匹配点y,这个过程通过最近邻搜索匹配特征描述子D来实现。通过这样的步骤，就可以得到N组匹配点，被称为seed match。匹配完后的每一组seed match可以被表示为:

$$M = (x_i, y_j)$$

2. 之后，对于源点云X中的所有点x_k,求其与seed match中点x_i的距离d与角度θ,其公式为:

$$d = ||x_k - x_i||_2$$

$$\theta = (\theta_1, \theta_2, ..., \theta_l)^T$$

$$\theta_l = cos^{-1}(n_l^{x_i}, n_l^{x_k}), l = 1, 2, ..., L$$

同理，对目标点云Y中所有点求取距离d与角度θ。再进行点云X与Y的匹配。选取源点云中的点x_k,查看所有点云Y中的点，如果满足：

$$||d_{x_k} - d_{y_d}|| \leq \epsilon_1$$

则求:

$$\Delta\theta = \frac{1}{L}\sum_{i=1}^{L}|\theta_l^{x_k} - \theta_l^{y_d}| if all \theta \leq \epsilon_2$$

若不满足条件，则$\Delta\theta = infinity$。
之后，选取所有成功的匹配点中，$\Delta\theta$最小的一组匹配点，求其特征描述子D的2-范数,若:

$$||D_x - D_y|| \leq \epsilon_3$$

则将这个匹配加入M中，不然则舍去。

3. 重复2中的步骤，直到所有源点云X中的点x_k都被查看过，则得到了最终的匹配集合M_i。因为这样的seed match有N个，所以最终生成的匹配集合M_i也有N个。

3.2.3 初始匹配选择

接下来的步骤，就是从N组匹配集合M中选择最优的一组。其方式如下：

1. 对每个匹配集合M,使用RANSAC剔除其outliers，估计其刚体变换$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$
2. 根据刚体变换T,从点云X,Y中根据最近邻匹配法，选择ηN组匹配点，求这些匹配点的能量函数:

$$E = \sum_{(x_i,y_i)\in\eta N} ||Tx_i - y_i||_2^2$$

,
选择能量最小的一组刚体变换T，作为ICP初始的刚体变换。
3. 根据刚体变换T,进行TrICP的迭代，得到最终结果。

其实可以发现2.中的过程，其本身也仅是一次TrICP的迭代而已。

3.2.4 创新与问题

这篇文章通过求取特征值，实现了初始变换矩阵T的全局计算，是一种全局匹配算法。不过应该看到，这种方法的算法复杂度是比较高的，达到了$O(N^2 log(N))$。以及算法中使用了大量的超参数，如$\epsilon_1, \epsilon_2, \epsilon_3$,虽然论文中后续将他们都设置为了$\rho$，但这种设置方法我觉得是不太有道理的，在一定的特定场景下可能会失配。并且，论文的一些结果是基于使用TrICP得到的，并不能证明其选择的特征值对于点云注册的真正帮助。其使用的TrICP也并不是自适应求取η的TrICP，论文将η简单的设置为30%。

3.3 SDRSAC

SDRSAC[Le19]是CVPR2019的一篇文章，也是一种全局求解ICP的算法。其使用了半定松弛获得了一组条件，通过求解这个半定规划问题，获得一组对应点，使用对应点求解初始矩阵，再进行ICP。

3.3.1算法设计与实现

算法首先构建了一个对应矩阵$X \in N \times N$,N为每次RANSAC随机选取的两组点云P、Q中的点的数量，矩阵X中的值表示P_i与Q_j是否对应。

接下来，将X展开，形成一个N^2的向量$\hat{X}$，并定义矩阵$A \in N^2 \times N^2$。

其横坐标与纵坐标则分别表示了$\hat{X}$中的一组对应点，即选出了两组对应点，并为其赋值:

$$A_{ab,cd} = exp(-|d(p_a, p_c) - d(p_b, p_d)|) if |\delta(p_a, p_c) - \delta(q_b, q_d)| \leq \gamma$$

,否则:

$$A_{ab,cd} = 0$$

这也是使用了之前提到的，刚体变换中的距离不变性，为集合中距离一致的点对赋予更高的被选择权重，当然，若两组点对间距离相差过大，此方法会直接给其赋予值为0。

之后，SDRSAC求解一个如下的规划问题：

$$\begin{gathered} max_X \hat{X}^T A \hat{X} \\ s.t. X_{i,j} \in \{0, 1\} \\ \sum_{j=1}^{N} X_{i,j} \leq 1 \\ \sum_{i=1}^{N} X_{i,j} \leq 1 \\ \sum_{i=1}^{N} \sum_{j=1}^{N} X_{i,j} = m \end{gathered}$$

设$Y = \hat{X}\hat{X}^T$

这个问题的本质实在寻找一组m个点对，这组点对在其各自的点云中，两两间都满足$d(x_a, x_c) = d(y_b, y_d)$。这是一个很强的几何约束。

那么原问题则可变为：

$$\begin{gathered} max^{X,Y} tr(AY) \\ s.t. Y = \hat{X}\hat{X}^T \\ tr(Y) = m \\ 0 \leq X_{i,j} \leq 1 \end{gathered}$$

$$\sum_{j=1}^{N} Xi,j \le 1$$

$$\sum_{i=1}^{N} Xi,j \le 1$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} Xi,j = m$$

但这个问题因为$Y = \hat{X}\hat{X}^T$的存在，是非凸的，因此再对其进行一次半定松弛。

$$Y - \hat{X}\hat{X}^T \ge 0$$

就得到了我们最终的问题，这个问题可以使用半定规划来解。

3.3.2 算法中的细节

算法为了更快、更好的获得初始矩阵，对初始的半定松弛做了一些拉紧的操作，如:

$$Yab,cd = min(X_{ab}, X_{cd})$$

$$Yab,cd = 0 if Aab,cd = 0$$

在通过半定规划，求得松弛后的结果$\tilde{X}$后，选择$\tilde{X}$中最大的m个矩阵值，作为点云间的匹配，根据这组匹配计算初始刚体变换T,进行ICP。
为了使算法快速而鲁棒，这篇文章使用如下流程进行计算：

1. 从源点云与目标点云中各随机选择N个点。
2. 对这(N, N)个点做半定规划，求得一组对应点X。
3. 根据对应点集X,求一个初始变换矩阵T。
4. 将T作为初始矩阵，进行ICP，得到最终的变换矩阵$T^{'}$。
5. 求ICP的残差函数值r。
6. 继续1-5.的步骤，记录最小的残差r与对应的最终变换矩阵$T^{'}$

3.3.3 创新与问题

SDRSAC所表述的这个规划问题，其优势在于，每一组最终选择的控制点对，两两之间都被这个规划问题的函数所控制。也就是说控制点对间的联系是较强的。接下来，使用得到的控制点对，获取一个初始的矩阵变换，再使用ICP求解得到E^*。重复这个过程，直到获得的E^*满足要求。

4 总结

ICP算法简单而有效，但是因为其算法非凸并且依赖于对应点集的选取，容易陷入局部最优解，得到错误的结果。在这篇文章中，我介绍了三种较新的ICP优化方法，并详细描述了他们的求解过程。TrICP从两个点云间的覆盖程度入手，提出了基于覆盖程度与残差，限制对应点集大小，提升精确性的方法，Colored Point Cloud Registration Revisit引入了有色点云，使用颜色信息，加入ICP来提升精确性。SDRSAC则自己构建了一个半定规划问题，使用RANSAC随机选取两个集合中的一些点，在半定松弛求解其中的最优匹配，使用最优匹配初始化最初的矩阵变换。Fast Descriptor一文使用了基于特征值的描述子，实现了快速、有效、鲁棒的全局点云匹配。

5 参考文献

[Besl92] Method for registration of 3-D shapes,Paul J. Besl,Neil D. McKay

[Chen92] Object modelling by registration of multiple range images,Yang C,Medioni G

[Park17] Jaesik Park, Qian-Yi Zhou, Vladlen Koltun, The IEEE International Conference on Computer Vision (ICCV), 2017, pp. 143-152

[Chetverikov02] The Trimmed Iterative Closest Point algorithm,D. Chetverikov,D. Svirko,D. Stepanov,P. Krsek

[Yang16] Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration,Jiaolong Yang,Hongdong Li,Dylan Campbell,Yunde Jia，T-PAMI2016

[Mellado14] Super 4PCS Fast Global Pointcloud Registration via Smart Indexing,Nicolas Mellado,Dror Aiger,Niloy J. Mitra，SGP 2014

[Le19] SDRSAC: Semidefinite-Based Randomized Approach for Robust Point Cloud Registration without Correspondences,Huu Le,Thanh-Toan Do,Tuan Hoang, Ngai-Man Cheung,CVPR 2019

[Zhou16] Fast Global Registration,Qian-Yi Zhou,Jaesik Park,Vladlen Koltum,ECCV 2016

[Lei17] Fast Descriptors and Correspondence Propagation for Robust Global Point Cloud Registration,Huan Lei;Guang Jiang; Long Quan,T-IP 2017

[Gao19] FilterReg:Robust and Eifficient Probabilistic Point Set Registration using Gaussian Filter and Twist Parameterization,CVPR 2019

[Lawin18] Density Adaptive Point Set Registration,F.J.Lawin,Martin Danelljan,Fahad Shahbaz Khan,Per-Reik Forssen,CVPR 2018