

## 摘要

同步定位与地图构建 (Simultaneous Localization and Mapping, SLAM) 是机器人领域中一大关键技术, 具有广泛的应用前景。由于视觉传感器的独特优势, 视觉 SLAM 成为近几年的研究热点。点特征是视觉 SLAM 中最广泛使用的特征, 在特征提取、匹配和表示上都比较成熟。然而点特征对于环境依赖较大, 在纹理缺失等场景中表现不佳。在人造的结构化环境中, 存在着丰富的线特征, 与图像中的点特征起到了信息互补的作用。同时线特征是一种更高层次的特征, 利用线段特征构建的环境地图具有更直观的几何信息。

本文利用从双目相机图像中提取得到的点特征和线特征, 提出一种基于点线综合特征的视觉 SLAM 方法, 可用于在室内室外环境的机器人定位与姿态估计。

本文的主要研究成果包括:

1. 对线特征的提取、匹配过程进行改进, 以提高前端中数据关联的正确率。  
其中提取部分对图像中割裂的线段进行合并, 匹配部分采用匀速度模型和 Liang-Barsky 线段裁剪算法对相机观测进行预测。
2. 后端优化中, 综合利用点、线特征构建图模型, 并采用正交表示参数化空间直线, 推倒相应的雅克比矩阵的解析形式。
3. 设计并构建了一套完整的统一的综合点线特征的 SLAM 系统, 包括双目匹配、帧间匹配、局部构图, 以及基于点线特征的闭环检测。实验表明, 该系统具有良好地定位精度和地图重构精度。

**关键词:** 视觉 SLAM, 双目视觉, 点线特征, 关键帧, 图优化

## Abstract

Simultaneous Localization and Mapping (SLAM) is one of the key technology in the field of robotics, which has a wide range of applications. Due to the unique advantages of visual sensors, visual SLAM has become a research hotspot in recent years. Point feature is the most widely used feature in visual SLAM, and it is more mature in feature extraction, matching and representation. However, point features are more dependent on the environment, and are not performing well in textureless scenes and etc. In man-made structured environment, there are plenty of line features. In fact, line features and point features provide complementary information about the image. Meanwhile, the line feature is a higher level primitive, and the environment map constructed by line segments has more intuitive geometric information.

In this thesis, we propose a visual SLAM method based on point and line features extracted from the stereo camera, which can be used to locate and estimate the robot in indoor and outdoor environments. The main research work includes:

1. An improved extraction and matching method for line features is introduced to robustify data association by merging the broken line segments, and using the constant velocity model and Liang-Barsky line segment clipping algorithm to predict the camera observations.

2. In the back-end of the proposed visual SLAM, we construct the graph model using point and line features. And we employ the orthonormal (minimal) representation to parameterize lines and analytically compute the corresponding Jacobians.

3. We design and implement a complete visual SLAM system using both point and line features, which includes stereo matching, frame tracking, local mapping, bundle adjustment of both line feature and point feature, as well as point-line based loop detection. Extensive experimental results are presented to validate its performance.

**Keywords:** Visual Slam, Stereo Vision, Point and Line Features, Keyframe Based, Graph Optimization

## 目录

致谢.....	I
摘要.....	III
Abstract.....	V
目录.....	VII
1 绪论.....	1
1.1 课题研究背景.....	1
1.2 视觉 SLAM 的发展与现状 .....	2
1.2.1 视觉 SLAM 的分类 .....	3
1.2.2 基于点特征和线段特征的视觉 SLAM .....	5
1.3 本文的主要内容.....	7
2 相关背景知识.....	9
2.1 相机模型.....	9
2.2 李群与位姿变换.....	11
2.2.1 群.....	12
2.2.2 流形.....	13
2.2.3 李群与李代数.....	14
2.3 基于图优化 SLAM 的基本原理 .....	15
2.4 本章小结.....	19
3 视觉 SLAM 中的线特征 .....	21
3.1 图像中的线特征.....	21
3.1.1 线段特征的提取.....	22
3.1.2 改进的线特征提取方法.....	23
3.1.3 线段特征的描述.....	25
3.2 环境地图中的线.....	27
3.2.1 空间直线的参数化.....	27
3.2.2 空间直线的观测模型.....	30
3.3 点线综合的后端优化框架.....	30
3.3.1 点线综合的图模型.....	31

3.3.2 点线误差的雅可比矩阵.....	33
3.4 点线综合的视觉词典.....	36
3.5 本章小结.....	39
4 基于点线综合特征的双目 SLAM .....	41
4.1 系统框架.....	41
4.2 跟踪线程.....	42
4.2.1 图像预处理.....	43
4.2.2 环境地图初始化.....	44
4.2.3 特征匹配.....	45
4.2.4 位姿估计.....	48
4.2.5 关键帧判断.....	49
4.3 局部构图线程.....	49
4.3.1 地图管理.....	49
4.3.2 局部地图优化.....	51
4.4 闭环线程.....	51
4.4.1 闭环检测.....	52
4.4.2 闭环校正.....	53
4.5 本章小结.....	54
5 实验与分析.....	55
5.1 仿真实验.....	55
5.1.1 实验描述.....	55
5.1.2 实验结果与分析.....	56
5.2 点线综合的视觉词典实验.....	58
5.2.1 实验描述.....	58
5.2.2 实验结果与分析.....	58
5.3 点线综合的视觉 SLAM 实验 .....	61
5.3.1 实验描述.....	61
5.3.2 实验结果与分析.....	62
5.4 本章小结.....	67

6 总结与展望.....69

    6.1 总结.....69

    6.2 展望.....70

参考文献.....71

作者简介.....75

攻读学位期间取得的科研成果.....77

## 2 相关背景知识

### 2.1 相机模型

视觉 SLAM 中采用相机来获取场景中的信息，通过获取的位置、几何、亮度和色彩信息等再进行相应的处理。相机模型描述了三维世界的点是如何映射到二维平面上的，这个映射称之为投影。而将二维平面上的点映射回三维世界中，则称之为反投影，通常需要提供二维点对应的深度信息。

常用的相机模型是针孔相机模型，它描述了一束光线通过针孔之后，在针孔背面投影成像的关系。如图 2.1 所示，针孔模型由相机光心  $C$ 、主光轴  $Z_c$  和垂直于主光轴的成像平面  $\pi$  组成。光心到成像平面的距离为焦距  $f$ ，主光轴与成像平面的交点称为主点  $p$ 。

在针孔模型相机中定义了几个坐标系，分别为

- 1) 相机坐标系  $O - X_c Y_c Z_c$ ：以光心为坐标原点，主光轴为  $Z$  轴。
- 2) 图像坐标系  $O - xy$ ：在成像平面上，主点为坐标原点，水平线与竖直线分别为  $x$  轴和  $y$  轴。
- 3) 图像像素坐标系  $O - uv$  则以图像左上角为坐标原点。

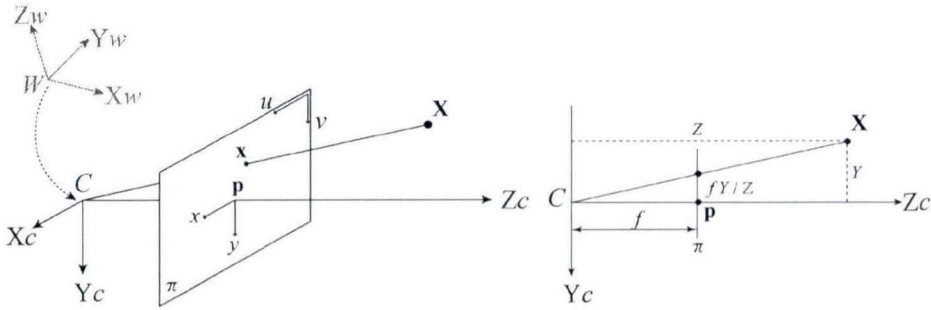


图 2.1 针孔相机模型

在相机坐标系下，3D 点  $\mathbf{X} = [X, Y, Z, 1]^T$  与光心的连线相交于成像平面，交点为  $\mathbf{x}$ ， $\mathbf{x}$  在图像上的坐标为  $[u, v, 1]^T$ ：

$$u = \frac{f_u X}{Z} + c_u \quad (2.1)$$

$$v = \frac{f_v Y}{Z} + c_v \quad (2.2)$$

其中， $f_u$ 、 $f_v$  为以像素为单位的焦距，等于焦距  $f$  乘以不同的尺度因子， $[c_u, c_v]^T$  为主点在图像像素坐标系下的坐标。为了方便推导，这里忽略了相机的畸变模型。

利用齐次坐标，将式(2.1)和式(2.2)以矩阵的形式重写：

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.3)$$

第一个  $3 \times 3$  的矩阵称为相机的内参数矩阵  $\mathbf{K}$ ，内参数矩阵可以张正友标定法获得。若式(2.3)中的 3D 点不以相机坐标系为参考坐标系，则需要添加一个旋转矩阵和平移向量，使得 3D 点回到相机参考坐标系中。因此更一般的形式为：

$$s\mathbf{x} = \mathbf{K} \begin{bmatrix} \mathbf{R}_{cw} & \mathbf{t}_{cw} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{X}_w = \mathbf{K} \mathbf{T}_{cw} \mathbf{X}_w \quad (2.4)$$

其中， $\mathbf{X}_w$  的下标表示该 3D 点以世界坐标系为参考坐标系， $\mathbf{R}_{cw}$  和  $\mathbf{t}_{cw}$  分别表示从世界坐标系到相机坐标系的旋转矩阵和平移向量。

在已知图像上  $\mathbf{x}$  点坐标的情况下，并不能知道所对应 3D 点  $\mathbf{X}_c$  的确切位置，但点  $\mathbf{X}_c$  位于光心  $C$  和  $\mathbf{x}$  的射线上。当给定其在  $Z$  轴的坐标，即深度  $d$  后，能唯一确定  $\mathbf{X}_c$  的坐标，这是通过反投影完成的：

$$\mathbf{X}_c = d\mathbf{K}^{-1}\mathbf{x} \quad (2.5)$$

双目相机则通过增加的第二个视角，可以进行双目匹配计算视差，从而估计像素点的深度。最简单的双目相机模型由两个参数相同的针孔相机组成，其主光轴相互平行，且成像平面共面，光心  $O_L$ 、 $O_R$  共线于  $x^c$  轴，如图 2.2 所示。

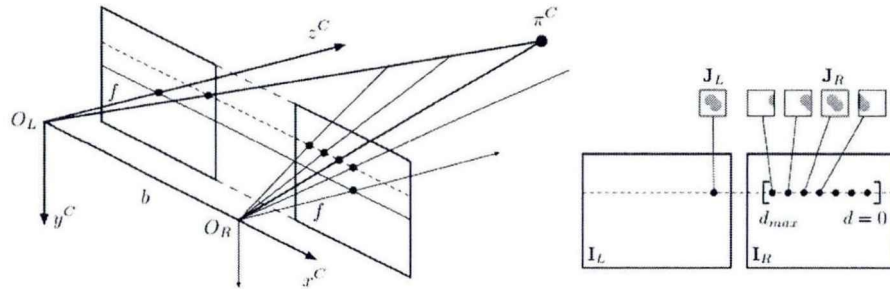


图 2.2 双目相机模型<sup>[41]</sup>

光心之间的距离  $b$  称为基线，通常以左相机坐标系作为双目相机坐标系。双目视觉中，需要找到左图中某个像素点在右图中的位置，即双目匹配。由于两相机视角、光照不同等影响，对所有的像素点进行双目匹配是很困难的，而采用图像中的特征点则相对容易。如图 2.2 的右图中，左图像有一特征点， $J_L$  为以特征点为中心的图像块，虚线为该特征点对应的极线。通过在极线上的搜索，利用

SAD 或 NCC<sup>[42]</sup>等方法计算两个图像块的相似程度，可以在右图像中找出和 $\mathbf{J}_L$ 相似的图像块 $\mathbf{J}_R$ ， $\mathbf{J}_R$ 对应的特征点即匹配点。特征点与匹配点的横坐标差即视差，它描述了空间点离相机的远近。

假设匹配点在左右图像中的坐标分别为 $\mathbf{m} = [u_l, v]^T$ 和 $\mathbf{m}' = (u_r, v)^T$ 。视差 $d = u_l - u_r$ ，将 $\mathbf{m}$ 和 $\mathbf{m}'$ 进行三角化后，其三维点 $\mathbf{M}$ 在左相机坐标系下的坐标为：

$$\begin{cases} X = \frac{b(u_l - c_u)}{d} \\ Y = \frac{b(v - c_v)}{d} \\ Z = \frac{bf}{d} \end{cases} \quad (2.6)$$

在实际应用中，左右相机之间存在相对旋转和平移关系，双目相机系统的参数包括内参数矩阵和外参数和畸变参数，也可以使用张正友标定方法同时获得<sup>[43]</sup>。

## 2.2 李群与位姿变换

在同步定位与地图构建的过程中，需要确定相机每一时刻相对于参考坐标系中的位姿，即通过平移运动与旋转运动来描述。参考坐标系一般是静止的，也可以称为世界坐标系，在 SLAM 中一般以相机第一帧的位姿作为参考坐标系。如图 2.3 所示，其中 $W$ 为世界坐标系， $C$ 为相机坐标系，相机在世界坐标系中的位姿可以由变换矩阵 $\mathbf{T}$ 表示。

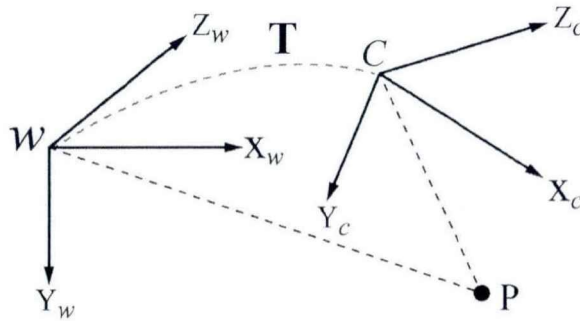


图 2.3 坐标变换

旋转运动的描述方式多种多样，包括旋转矩阵、旋转向量、欧拉角、单位四元数等若干种方式。旋转矩阵自身是带有约束的，旋转矩阵为正交矩阵且其行列式为 1；欧拉角具有奇异性，不适和插值和迭代，往往只用于人机交互；单位四元数同样存在模为 1 的约束。将它们作为优化变量时，会引入额外的约束，使优化问题变得复杂。而通过李群李代数间的转换关系，可以将位姿估计变成无约束



的优化问题,简化求解方式。同时李群与李代数具有以下特征:可加(composed)、可逆(inverted)、可微(differentiated)、可插值(interpolated),并且可以表示位姿的不确定性。

在介绍李群与李代数前,首先介绍群和流形的概念。

### 2.2.1 群

群(Group)由一个非空集合和一个运算组成。我们把集合记作  $\mathcal{G}$ , 运算记作  $\cdot$ 。对于集合中的元素  $a, b, c \in \mathcal{G}$  需要满足以下条件:

- 1) 封闭性:  $a \cdot b \in \mathcal{G}$
- 2) 结合律:  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- 3) 单位元: 存在唯一的元素  $e \in \mathcal{G}$ , 使得  $e \cdot a = a \cdot e = a$
- 4) 逆元: 群中的任意元素均存在逆元  $a^{-1} \in \mathcal{G}$ , 使得  $a \cdot a^{-1} = a^{-1} \cdot a = e$

若一个群满足交换律,则称该群为阿贝尔群。群在物理学和数学中经常出现,常见的群包括整数的加法  $(\mathbb{Z}, +)$ , 去掉 0 后的有理数的乘法  $(\mathbb{Q} \setminus \{0\}, \cdot)$  等等。除此之外,矩阵也可以构成群,例如  $n \times n$  可逆矩阵与乘法构成群称为一般线性群  $GL(n)$ , 其单位元为  $\mathbf{I}_n$ 。若一般线性群中  $n \times n$  可逆矩阵的秩行列式为 1, 则称该群为特殊正交群  $SL(n)$ ; 若一般线性群中  $n \times n$  可逆矩阵为正交矩阵, 则称该群为正交群  $O(n)$ , 正交矩阵的行列式必为 +1 或 -1。

$$GL(n) = \{\mathbf{G} \in \mathbb{R}^{n \times n}: \det(\mathbf{G}) \neq 0\} \quad (2.7)$$

$$SL(n) = \{\mathbf{S} \in \mathbb{R}^{n \times n}: \det(\mathbf{S}) = 1\} \quad (2.8)$$

$$O(n) = \{\mathbf{O} \in \mathbb{R}^{n \times n}: \mathbf{O}\mathbf{O}^T = \mathbf{O}^T\mathbf{O} = \mathbf{I}_{n \times n}\} \quad (2.9)$$

作为  $SL(n)$  与  $O(n)$  的共同的子群, 则构成我们最为关注的特殊正交群  $SO(n)$ , 如表示三维空间旋转的  $SO(3)$ 。这几个群之间的关系如图 2.4 所示。

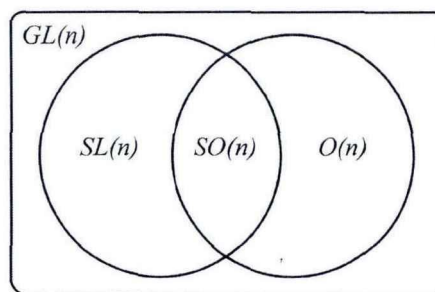


图 2.4 特殊正交群与其他李群的关系

### 2.2.2 流形

流形 (manifold) 是局部具有欧几里得空间性质的空间, 是欧几里得空间中的曲线、曲面等概念的推广。如图 2.5 所示,  $x$  为球体上的一点, 平面  $T_x \mathcal{M}$  为  $x$  处的切平面, 即流形对应的切空间。对于球面上任意一个局部点  $x$ , 其局部特性可由过该点的切平面来近似表示, 而该平面具有欧几里得空间的性质, 但对于整个球面而言, 其结构是非欧的。

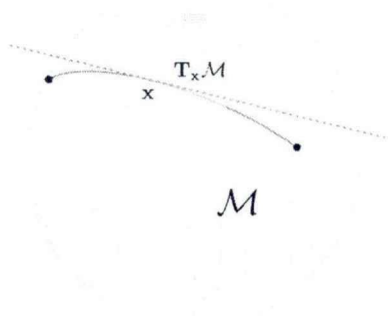


图 2.5 流形  $\mathcal{M}$  与其切空间  $T_x \mathcal{M}$

最简单的流形是拓扑流形, 拓扑流形要求该拓扑空间为豪斯多夫空间, 并且第二可数。对于拓扑流形上的每一点  $p \in \mathcal{M}$ , 要求可以找到一个开集  $V \subseteq \mathcal{M}$  包含  $p$ , 另一个开集  $U \subseteq \mathbb{R}^n$ , 使得  $V$  和  $U$  为拓扑同胚。换句话说, 存在一个一对一的映射  $\varphi: V \rightarrow U$ 。开集  $U$  加上同胚映射  $\varphi$  记为坐标卡  $\{U, \varphi\}$ , 坐标卡全体叫做一个图册。可以想象, 给定  $\mathcal{M}$  中任意一点, 会在某个坐标卡中, 通过映射作用把它映射到欧式空间。若对于  $\mathcal{M}$  的两个坐标卡  $\{U_\alpha, \varphi_\alpha\}$  和  $\{U_\beta, \varphi_\beta\}$ ,  $U_\alpha \cap U_\beta \neq \emptyset$ , 这两种映射相差一个光滑映射, 即  $\varphi_\beta \circ \varphi_\alpha^{-1}: \varphi_\alpha(U_\alpha \cap U_\beta) \rightarrow \varphi_\beta(U_\alpha \cap U_\beta)$  是光滑的 (无穷次可微), 则称  $\mathcal{M}$  为光滑流形, 如图 2.6 所示。

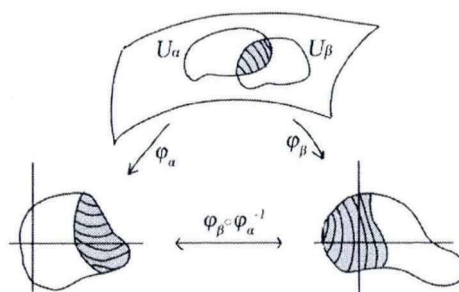


图 2.6 光滑流形

### 2.2.3 李群与李代数

李群是连续和光滑的流形，具有群和光滑流形的特性。李群是可微分的群，微分的基本想法就是在无穷小的层面上的线性化。因此李群的结构对应着它的线性化所得的一种“无穷小群”的结构，这个“无穷小群”即为李代数。李群上的李代数，就是流形上对应于单位元那个点上的切空间，刻画了李群在单位元的附近的局部性质。假设 $V$ 是数域 $F$ 上的向量空间，如果 $V$ 上有一个二元运算 $[\cdot, \cdot]$ ，如果满足双线性、自反性、雅可比等价三个条件；则称之为李代数。相对于李群的流形结构而言，李代数是性质更为简单的线性矢量空间。

在计算机视觉中，常常需要进行坐标系变换或将一个向量从一个坐标系转换到另一个坐标系等操作，这些变换需要用到三维旋转矩阵构成的特殊正交群 $SO(3)$ 和变换矩阵构成的特殊欧氏群 $SE(3)$ 。

$SO(3)$ 和 $SE(3)$ 的定义如下：

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} | RR^T = I, \det(R) = 1\} \quad (2.10)$$

$$SE(3) = \left\{ T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, R \in SO(3), t \in \mathbb{R}^3 \right\} \quad (2.11)$$

这里的 $R$ 为旋转矩阵， $t$ 为平移向量。 $SO(3)$ 和 $SE(3)$ 对应的李代数为 $so(3)$ 和 $se(3)$ 。

$$so(3) = \{\Phi = \phi^\wedge \in \mathbb{R}^{3 \times 3}, \phi \in \mathbb{R}^3\} \quad (2.12)$$

$$se(3) = \left\{ \Xi = \xi^\wedge = \begin{bmatrix} \phi^\wedge & \rho \\ 0^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4}, \xi = \begin{bmatrix} \phi \\ \rho \end{bmatrix} \in \mathbb{R}^6 \right\} \quad (2.13)$$

其中， $\Phi$ 为反对称矩阵，由3维向量 $\phi = [\phi_1, \phi_2, \phi_3]^T$ 生成。 $\xi$ 为6维向量，前3维为旋转，记为 $\phi$ ，后3维为平移，记为 $\rho$ 。同时使用 $\wedge$ 和 $\vee$ 符号来指代“从向量到矩阵”和“从矩阵到向量”的变换关系：

$$\phi^\wedge = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} = [\phi]_\times \quad (2.14)$$

$$\Phi^\vee = \phi, \Xi^\vee = \xi \quad (2.15)$$

记号 $[\cdot]_\times$ 与叉乘有关，可以将两向量叉乘写成乘法的形式：

$$[a]_\times \cdot b = a \times b \quad (2.16)$$

连接李群与李代数的纽带是指数映射和对数映射，如图2.7所示。指数映射将李代数中的元素映射到李群中，对数映射则反之。任意矩阵的指数映射可以写成一个泰勒展开(2.17)，但是只有在收敛的情况下才会有结果，其结果仍是一个

矩阵。

$$\exp(\mathbf{A}) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{A}^n \quad (2.17)$$

对于 $SO(3)$ ，指数映射(2.18)和对数映射(2.19)都具有闭式的方程。 $SO(3)$ 的指数映射即罗德里格斯变换公式，说明 $\mathfrak{so}(3)$ 中三个参数具体的意义，即三维空间中的任意旋转可以由旋转向量 $\boldsymbol{\phi}$ 表示，绕固定轴（单位向量） $\mathbf{u}_{\boldsymbol{\phi}} = \frac{\boldsymbol{\phi}}{\theta}$  旋转一定的角度 $\theta = \|\boldsymbol{\phi}\|_2$ 产生一个旋转。同样，给出 $SE(3)$ 的指数映射(2.20)和对数映射(2.21)。

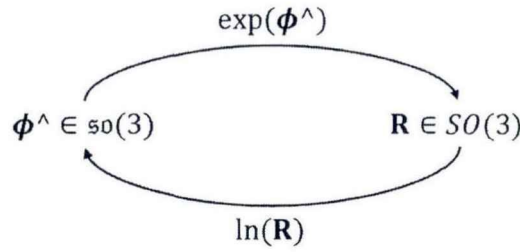


图 2.7  $SO(3)$ 的指数映射与对数映射

$$\mathbf{R} = \exp(\boldsymbol{\phi}^\wedge) = \mathbf{I} + \frac{\sin(\theta)}{\theta} [\boldsymbol{\phi}]_\times + \frac{1 - \cos(\theta)}{\theta} [\boldsymbol{\phi}]_\times^2, \quad \theta = \|\boldsymbol{\phi}\|_2 \quad (2.18)$$

$$\boldsymbol{\phi}^\wedge = \ln(\mathbf{R}) = \frac{\theta}{2 \sin(\theta)} (\mathbf{R} - \mathbf{R}^T), \quad \theta = \arccos \frac{\text{trace}(\mathbf{R}) - 1}{2} \quad (2.19)$$

$$\mathbf{T} = \exp(\boldsymbol{\xi}^\wedge) = \begin{bmatrix} \exp(\boldsymbol{\phi}^\wedge) & \mathbf{J}\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (2.20)$$

$$\boldsymbol{\xi}^\wedge = \ln(\mathbf{T}) = \begin{bmatrix} \ln(\mathbf{R})^T & (\mathbf{J}^{-1}\mathbf{t})^T \end{bmatrix}^T \quad (2.21)$$

其中 $\mathbf{J} = \mathbf{I} + \frac{1 - \cos(\theta)}{\theta^2} [\boldsymbol{\phi}]_\times + \frac{\theta - \sin(\theta)}{\theta^3} [\boldsymbol{\phi}]_\times^2$ 。

当李代数趋近于无穷小时，可以得到以下几条性质：

$$\mathbf{R} = \mathbf{I} + [\boldsymbol{\phi}]_\times, \quad \boldsymbol{\phi} \rightarrow \mathbf{0} \quad (2.22)$$

$$\mathbf{J} = \mathbf{I}, \quad \boldsymbol{\phi} \rightarrow \mathbf{0} \quad (2.23)$$

$$\mathbf{T} = \mathbf{I} + \begin{bmatrix} [\boldsymbol{\phi}]_\times & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}, \quad \boldsymbol{\xi} \rightarrow \mathbf{0} \quad (2.24)$$

## 2.3 基于图优化 SLAM 的基本原理

通常可以将 SLAM 分为前端和后端两部分，如图 2.8 所示。不管是基于特征还是基于直接跟踪方法的 SLAM 系统，都可以该框架表示。前端主要对传感器采集的数据进行处理，提取计算机能认知的特征作为观测，建立观测与环境地图

路标之间的对应关系，构建图模型中的顶点和边；后端主要通过图优化推断出全局一致的轨迹和环境地图。

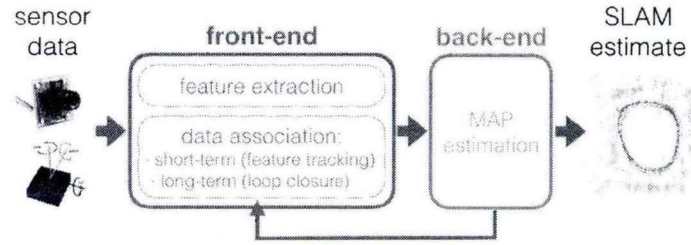


图 2.8 SLAM 框架<sup>[44]</sup>

前端的数据关联分为短期和长期两部分。其中短期的数据关联指的是相邻帧之间的特征跟踪匹配，再根据匹配关系进行运动估计和地图构建；长期的数据关联指的是闭环，系统通过传感器的观测判断是否返回了之前访问过的环境。

后端解决的问题，就是在给定带有噪声的观测下，估计整个系统的状态，以及状态估计不确定性，这里的状态既包括相机的位姿，也包含地图中的路标，即最大后验概率估计（Maximum-a-Posteriori, MAP）问题。

定义机器人在一段时间内的位姿集合  $\mathbf{X} = \{\mathbf{x}_t\}$  和环境地图中的路标集合  $\mathbf{L} = \{\mathbf{l}_i\}$ ，以及传感器的观测集合  $\mathbf{Z} = \{\mathbf{z}_k\}$ 。 $\mathbf{X}$  和  $\mathbf{L}$  的集合  $\mathcal{X}$  构成 SLAM 需要估计的状态变量。对应的需要定义传感器的观测模型  $\mathbf{z}_k = h_k(\mathbf{x}_k) + \epsilon_k$ ，将环境地图、位姿和观测三者联系起来。其中  $h_k(\cdot)$  为生成观测  $\mathbf{z}_k$  的传感器的观测模型， $\epsilon_k$  为观测噪声， $\mathbf{x}_k$  为  $\mathbf{X}$  的子集。最大后验概率估计，即求解：

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmax}} P(\mathbf{X}|\mathbf{Z}) \quad (2.25)$$

也就是寻找一组位姿和路标，最符合传感器的观测。为了估计状态变量的条件分布，利用贝叶斯法则，有：

$$P(\mathbf{X}|\mathbf{Z}) = \frac{P(\mathbf{Z}|\mathbf{X})P(\mathbf{X})}{P(\mathbf{Z})} \propto P(\mathbf{Z}|\mathbf{X})P(\mathbf{X}) \quad (2.26)$$

贝叶斯法则左侧通常称为后验概率。它右侧的  $P(\mathbf{Z}|\mathbf{X})$  称为似然，另一部分  $P(\mathbf{X})$  称为先验。直接求后验分布是困难的，但是求一个状态的最优估计，使得在该状态下，后验概率最大化则是可行的。当先验知识不存在时，后一项也可以直接去掉。因此最终需要求解：

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmax}} P(\mathbf{Z}|\mathbf{X}) \quad (2.27)$$

最大后验概率转换为了极大似然估计问题。在观测为高斯分布的假设下，极

大似然会有更简单的形式。对于模型  $\mathbf{z}_k = h_k(\mathbf{x}_k) + \boldsymbol{\epsilon}_k$ ，若  $\boldsymbol{\epsilon}_k$  服从高斯分布  $N(0, \boldsymbol{\Sigma}_k)$ ，则

$$P(\mathbf{Z}_k | \mathbf{x}_k) \propto \exp\left(-\frac{1}{2} \|\mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k\|_{\boldsymbol{\Omega}_k}^2\right) \quad (2.28)$$

$\boldsymbol{\Omega}_k = \boldsymbol{\Sigma}_k^{-1}$ ，称为信息矩阵。假设观测之间相互独立，对(2.27)式进行展开并求负对数：

$$\begin{aligned} \mathbf{x}^* &= \underset{\mathbf{x}}{\operatorname{argmin}} -\log(\prod P(\mathbf{Z}_k | \mathbf{x}_k)) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} \sum \frac{1}{2} \|\mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k\|_{\boldsymbol{\Omega}_k}^2 \end{aligned} \quad (2.29)$$

这就得到了一个总体意义下的无约束的最小二乘问题，它的最优解等价于状态的极大似然估计。通常可以用图模型来表达上面问题，因此之为图优化问题。常见的图模型包括马尔科夫随机场、动态贝叶斯网络或因子图等。在因子图中，状态变量用圆表示，观测信息以黑色的小块表示。其中蓝色的圆表示机器人的位姿，绿色的圆表示环境地图中的路标。在状态变量之前存在着不同的约束，黄色的可以是相机对空间点的观测约束，黑色的可以是轮式里程计的约束，红色的可以是闭环检测的约束。

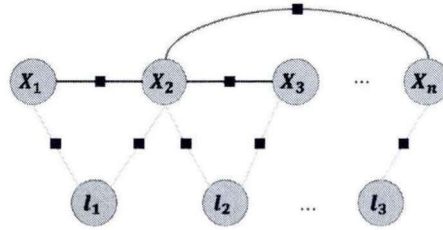


图 2.9 因子图模型

由于测量方程  $h_k()$  一般是非线性的函数，需要使用高斯牛顿法或 LM 法等对该非线性最小二乘问题进行求解，其思想是在当前估计量附近对定义的代价函数迭代地线性化求解。以高斯牛顿法为例，令观测误差  $\mathbf{e}(\mathbf{x}_k, \mathbf{z}_k) = \mathbf{h}_k(\mathbf{x}_k) - \mathbf{z}_k$ ，在初值  $\hat{\mathbf{x}}_k$  附近进行一阶泰勒展开：

$$\mathbf{e}(\mathbf{x}_k, \mathbf{z}_k) \approx \mathbf{h}_k(\hat{\mathbf{x}}_k) - \mathbf{z}_k + \left. \frac{\partial \mathbf{e}}{\partial \mathbf{x}_k} \right|_{\hat{\mathbf{x}}_k} \delta \mathbf{x}_k \quad (2.30)$$

$\delta \mathbf{x}_k$  为在线性点附近定义的增量，则(2.30)式可以等效为求解最优的增量：

$$\delta \mathbf{x}^* = \underset{\delta \mathbf{x}_k}{\operatorname{argmin}} \sum \frac{1}{2} \|\mathbf{J}_k \delta \mathbf{x}_k - \mathbf{b}_k\|_{\boldsymbol{\Omega}_k}^2 = \underset{\delta \mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{J} \delta \mathbf{x} - \mathbf{b}\|_{\boldsymbol{\Omega}}^2 \quad (2.31)$$

其中  $J_k = \left. \frac{\partial e}{\partial \mathbf{x}_k} \right|_{\hat{\mathbf{x}}_k}$  为误差函数在线性点  $\hat{\mathbf{x}}_k$  的雅可比矩阵,  $\mathbf{b}_k = \mathbf{z}_k - h_k(\hat{\mathbf{x}}_k)$  为残差。通过一阶近似, 可转化为线性的最小二乘问题, 对增量  $\delta \mathbf{x}$  求导并等于 0 即可得到正规方程:

$$(\mathbf{J}^T \mathbf{\Omega} \mathbf{J}) \delta \mathbf{x}^* = \mathbf{J}^T \mathbf{\Omega} \mathbf{b} \quad (2.32)$$

最后通过  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \delta \mathbf{x}^*$  迭代更新直到收敛。在 SLAM 问题中  $(\mathbf{J}^T \mathbf{\Omega} \mathbf{J})$  通常是一个稀疏的矩阵, 可以采用舒尔补以及增量式平滑算法 iSAM<sup>[45]</sup> 等技巧对正规方程进行快速求解。

在上面的推导中, 假设了  $\mathbf{x}$  是存在于欧式空间的, 因此可以使用 “+” 进行迭代更新。对于非欧式空间的变量, 并不能直接使用 “+” 对进行更新, 因为它们对 “+” 并不封闭, 一种做法是对  $\hat{\mathbf{x}} + \delta \mathbf{x}^*$  进行强制性约束, 另一种做法需要利用到前面介绍的李群与李代数。由于李代数表示无穷小增量且存在于线性矢量空间, 可以定义一个广义的  $\oplus$ , 采用  $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} \oplus \delta \mathbf{x}^*$  作为更新。同时雅可比矩阵修改为  $J_k = \frac{\partial e(\mathbf{x}_k \oplus \delta \mathbf{x}, \mathbf{z}_k)}{\partial \delta \mathbf{x}}$ , 由于李群具有光滑流形的性质, 该雅可比具有解析形式。通常, 基于流形的优化具有更低的计算复杂度以及更好的数值稳定性。

对于李群中的  $SE(3)$ , 定义  $\oplus$ :

$$\begin{aligned} \oplus: SE(3) \times se(3) &\rightarrow SE(3) \\ \exp(\xi^\wedge) \mathbf{T} &\rightarrow \mathbf{T} \end{aligned} \quad (2.33)$$

图 2.10 清晰地展示了李代数  $se(3)$  作为增量的具体意义, 图中黑色长方体所在的位置为初始位置, 通过对长方体的 8 个端点进行坐标变换  $\exp(\xi_i^\wedge) \mathbf{X} \rightarrow \mathbf{X}$ , 得到更新后的位置, 即图中的黑色长方体。

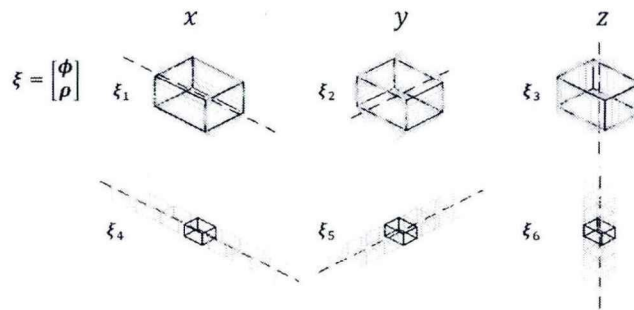


图 2.10 微小增量  $se(3)$  的意义<sup>[46]</sup>

通过对前端、后端的分析，对于实现本文基于点线综合特征的视觉 SLAM，有以下几个问题需要解决。

- 1) 线特征的检测、描述以及数据关联。
- 2) 空间直线的参数化。最大后验概率中，状态变量 $\mathbf{x}$ 的表达称为参数化问题，如欧拉角、四元数、 $SO(3)$ 等均为旋转矩阵的参数化形式，在 2.2 节中已经介绍了这几种参数化形式。由于空间直线具有 4 个自由度，需要选择一种合适的参数化方法来表达，并根据参数化形式定义运算符 $\oplus$ 。
- 3) 空间直线的观测模型。空间直线作为环境地图中的元素，需要定义观测量和观测方程。观测方程与观测量的定义具有一定关系，对后端的计算效率和迭代次数都有一定的影响。
- 4) 优化的规模。前面提到的最大后验概率中， $\mathbf{x}$ 包含了环境地图中的路标和位姿等状态变量，如果将所有时刻的位姿和全部的路标作为优化的量，则意味着 $\mathbf{x}$ 维度极高，不能满足实时性的需求。为了加快求解，可以降低 $\mathbf{x}$ 的维度，例如放弃路标，仅对位姿进行优化，或者对局部地图内的路标和位姿进行优化等等，然而这些方法不可避免地丢失了一些信息，虽然提升了处理的速度，但降低了精度。因此需要在不同的地方定义合适的优化规模。

## 2.4 本章小结

本章介绍了视觉 SLAM 中涉及到的基础知识。首先介绍了单目和双目相机模型，为后文观测模型的引入做理论准备。然后引入群和流形的概念，介绍了表示位姿变换的李群和李代数。最后介绍了基于图优化 SLAM 的基本构成，重点叙述了后端中非线性最小二乘问题的求解，并结合前端、后端对实现本文的 SLAM 视觉方法中存在的问题进行分析。



### 3 视觉 SLAM 中的线特征

在马尔的视觉计算理论<sup>[47]</sup>中提到,通过视觉这一过程能够对外部世界进行描述。其核心问题是要从图像的结构推理出外部世界的结构,最后达到认识外部世界。图像推理是对图像进行一系列处理,从而得到物体形状乃至更高层次的过程。根据层次的高低,马尔将其分为三个阶段,分别为初始简图、2.5 维简图和三维模型。初始简图即图像中强度变化明显处的位置和分布情况,例如角点、斑点、边缘、线段和曲线等等,通过这些基元可以对图像进行简单地描述。因此特征点检测、边缘检测是计算机视觉中较为基础的问题,其目的是构建可认知的初始简图。对于图像中亮度变化明显、位置相邻且方向相近的点集,它们构成了特殊的边缘——线段。

在视觉 SLAM 中,点特征和线特征分别代表两类不同的信息来源,点特征主要是图像中的角点、斑点,线特征主要是图像中的边缘。在不同场景中,点线特征的丰富程度是不一样的。同时线特征赋予了 SLAM 天然的光照和视角不变性,作为 SLAM 的一部分提供了更加直观的环境地图,如图 3.1 所示,右图是在一段走廊里的场景,可以辨别出墙的边角、门和天花板的灯等特征。直线相对于点是一种大尺度的特征,可以快速地扩充环境地图。同时也可以利用直线构建更高级基元,例如平面特征等等。因此将这两类特征进行结合,可以有效地提高系统的精度和鲁棒性。

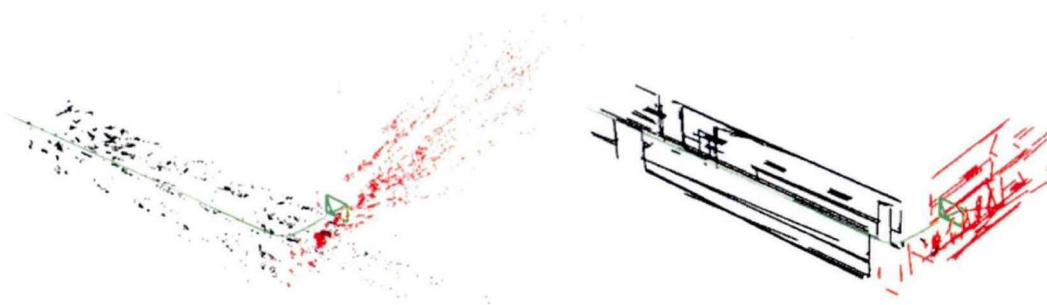


图 3.1 点特征和线特征构建的环境地图

#### 3.1 图像中的线特征

视觉 SLAM 前端首要的工作就是进行特征的提取,再对提取的特征进行数据关联。一个好的特征要求具有可重复性、可区分性和准确性等性质。点特征的

提取和描述已经比较成熟，这里不在赘述，下面将叙述图像中线特征的提取与描述。

### 3.1.1 线段特征的提取

传统的线段检测算法一般采用 Canny 等边缘检测算法计算提取图像的边缘信息，接着利用霍夫变换提取直线，最后通过分割的方法找到直线的端点。这种方法实时性不高，并且在边缘密集处会产生误检，不能用于视觉 SLAM 系统。

LSD (Line Segment Detector) 是一种能在线性时间内得到亚像素级精度的局部提取线段的算法<sup>[48]</sup>，其核心思路是将梯度方向相近的像素点合并。LSD 首先计算对图像进行降采样和高斯滤波，去除噪声的干扰，并计算图像的梯度幅值与梯度方向，所有像素点的梯度构成梯度场，如图 3.2 所示。接着合并梯度场里方向近似相同的像素点，构成一个区域，该区域被称为线段支持域，在图中分别用 3 种颜色标记出来。

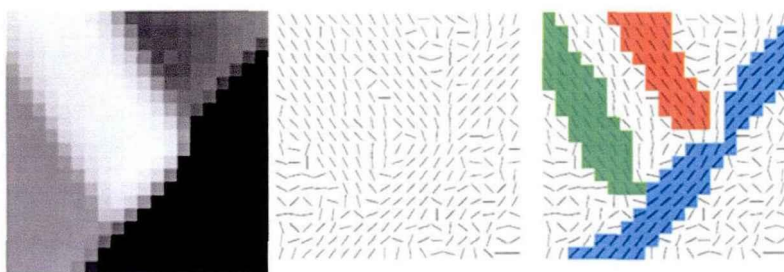


图 3.2 LSD 中的线段支持域

每一个线段支持域可能对应着一条线段，需要进一步做检验。具体方法如下：对线段支持域生成一个最小外接矩形，统计最小外接矩形的主方向。若区域内某一像素点的梯度方向与主方向的角度差在设定阈值内，那么这个点被称作支持点。利用最小外接矩形内支持点的比例来判定这个线段支持域是否是一条线段。

LSD 的优点是速度快、不需要调整额外的参数，并且 LSD 提取的线段具有主方向，在进行线段匹配时可以排除一些误匹配对。然而，LSD 也存在着一些缺点。

- 1) 在区域增长步骤中设置了每个像素点是否使用的标志，因此每个像素点只能隶属于一条直线，当图像上两直线相交时至少有一条直线被分为两段。
- 2) 降采样和高斯滤波是为了减少图像噪声和锯齿的影响，保证提取线段的

连续性和稳定性，然而在尺寸缩小的图中检测出来的线段，会损失一定的精度。

- 3) 对于遮挡、局部模糊等情况，由于区域增长算法的特性，原本的一条线段会割裂成多段。

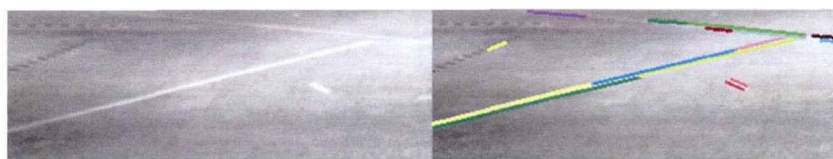


图 3.3 LSD 检测的线段

图 3.3 展示了使用 LSD 检测直线的效果，左图清晰地显示出地面上贴有白色的线条，右图中彩色的表示检测到线段，由于图像模糊和噪声的影响，一长条的直线被分割为很多段。这样会使得线特征在 SLAM 中的匹配并不是一对一的，而是多对多的。并且分割的点在邻近的帧中是不同的，导致了线特征匹配难度增大。基于上述情况，本文采用合并线段的方式对 LSD 提取的线段进行处理，将线段方向相似的邻近线段进行合并，以减少这种情况的发生。

### 3.1.2 改进的线特征提取方法

对于 LSD 提取的每一个线段特征，可以区分出线段的起始点和终点，起始点指向终点的向量右边区域比左边区域的平均灰度值要大，该向量与水平线的夹角即为主方向，范围为 $(-\pi, \pi]$ 。图 3.3 中白条处的两条线段虽然平行，但这两条线段主方向相差 180 度。仅由主方向来判断是不够的，同时合并的直线需要相邻近，这里定义线段中点到另一条线段的距离  $d$  和两线段端点与端点之间的最小距离  $l$ ，若  $d$  和  $l$  均小于给定值，则认为两线段相邻近。最后还需要通过描述子的距离进行最后的判断，即线段在图中的局部特征相似。描述子部分在下一章会详细叙述。

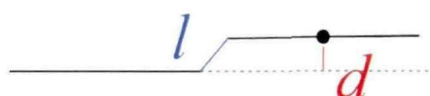


图 3.4 相邻近的线段

改进后的线段特征提取算法如表 3.1 所示：

表 3.1 改进的线段特征提取方法

---

输入：灰度图，线段长度阈值 $s_{TH}$ ，主方向阈值 $ang_{TH}$ 距离阈值 $d_{TH}$ 和 $l_{TH}$ ，描述子距离阈值 $des_{TH}$

---

输出：特征线段集合

---

开始：

使用 LSD 提取线段特征 $kl = \{kl_i\}_{i=1...m}$

计算所有线段特征对应的描述子 $des = \{des_i\}_{i=1...m}$

$i = m$

**While**  $i \neq 0$  **do**

    选取线段特征 $kl_i$

**While**  $kl$  不为空 **do**

        选取线段特征 $kl_j$ ,  $i \neq j$ 。计算主方向差 $\text{dist}(ang_i, ang_j)$ 、

        点线距离 $d$ 、端点距离 $l$ 、描述子距离 $t$

**If**  $\text{dist}(ang_i, ang_j) < ang_{TH} \ \&\& \ d < d_{TH} \ \&\& \ l < l_{TH} \ \&\&$

$t < des_{TH}$  **THEN**

            1) 使用最小二乘法合并线段

            2) 重新计算 $kl_i^*$ 的描述子

**End**

$i = i - 1$

**For**  $i = 0:n$

    计算合并后线段长度 $s$ ，若 $s < s_{TH}$ ，移除该特征线段

**END**

---

经过处理后，可以看到原本断开的线段已经合并为同一根线段，可以在一定程度上解决线段被割裂的问题。

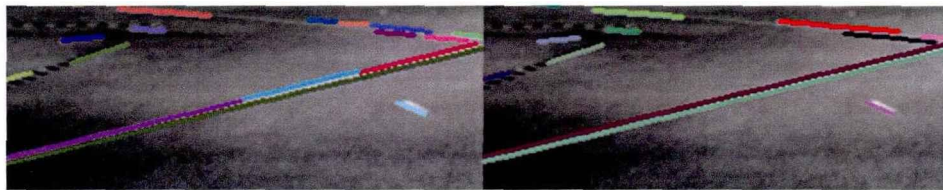


图 3.5 合并后的线段特征

### 3.1.3 线段特征的描述

在计算机视觉中，特征匹配是场景重建、模式识别、视觉 SLAM 等应用的基础。线特征的匹配比较困难的，主要包括：线段端点位置不准确、线段发生割裂、不具备强有力的几何约束（如极线几何约束）和在纹理缺失处不具有较强的辨识度等原因。Zhiheng Wang 等将线段分割为多个子区域，采用和 SIFT 类似的方法，根据区域内的梯度均值和标准差建立了 MSLD (mean-standard deviation line descriptor) 描述子<sup>[49]</sup>，该描述子具有旋转、光照不变性等优点，同时与线段的长短无关。Lilian Zhang 等在 MSLD 的基础上进行改进提出了 LBD 描述子<sup>[50]</sup>，引入了全局和局部的高斯权重系数。相比于 MSLD，具有更优的匹配效果，计算速度更快。

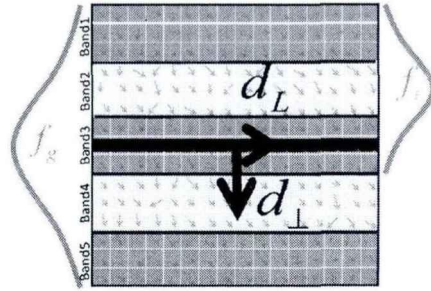


图 3.6 LBD 特征描述中的线段支持域

LBD 描述子首先在线段处建立一个矩形，该矩形称为线段支持域 LSR (Line Support Region)，并定义了方向  $d_L$  和方向  $d_\perp$ 。用  $m$  表示线段支持域的条带数目、 $w$  表示条带的像素宽度，如图 3.6 所示。通过全局高斯函数  $f_g$  作用于 LSR 的每一列，降低距离线段较远的像素梯度对描述子的影响。通过局部高斯函数  $f_l$  作用于相邻条带的每一列，降低条带间的边界效应。

基于相邻条带的梯度可以计算出每个条带  $B_j$  对应的特征向量  $BD_j$ ，将所有条带的特征向量进行合并，即形成 LBD 描述子：

$$\text{LBD} = (BD_1^T, BD_2^T \dots BD_m^T)^T \quad (3.1)$$

将  $B_j$  与相邻条带  $B_{j-1}$ 、 $B_{j+1}$  的每一行的局部梯度  $\mathbf{g}' = (g'_{d_\perp}, g'_{d_L})$  进行分别求和。其中对于第  $k$  行，有：

$$v1_j^k = \lambda \sum_{g'_{d_\perp} > 0} g'_{d_\perp}, v2_j^k = \lambda \sum_{g'_{d_\perp} < 0} g'_{d_\perp}, \quad (3.2)$$

$$v3_j^k = \lambda \sum_{g'_{a_L} > 0} g'_{a_L}, v4_j^k = \lambda \sum_{g'_{a_L} < 0} g'_{a_L}$$

其中,  $\lambda = f_g(k)f_l(k)$  为高斯系数, 将每一行的和放在一起, 构成特征向量  $BD_j$  对应的特征描述矩阵  $BDM_j$ :

$$BDM_j = \begin{pmatrix} v1_j^1 & v1_j^2 & \dots & v1_j^n \\ v2_j^1 & v2_j^2 & \dots & v2_j^n \\ v3_j^1 & v3_j^2 & \dots & v3_j^n \\ v4_j^1 & v4_j^2 & \dots & v4_j^n \end{pmatrix} \in \mathbb{R}^{4 \times n} \quad (3.3)$$

$$n = \begin{cases} 2w, j = 1 \text{ or } m \\ 3w, \text{ else} \end{cases}$$

最后计算特征描述矩阵  $BDM_j$  的均值方差  $M_j^T$  和均值向量  $S_j^T$ , 即得到特征向量  $BD_j = (M_j^T, S_j^T)^T \in \mathbb{R}^8$ , 得到最后的 LBD 特征:

$$LBD = (M_1^T, S_1^T, M_2^T, S_2^T \dots M_m^T, S_m^T)^T \in \mathbb{R}^{8m} \quad (3.4)$$

根据作者 Lilian Zhang 的实验分析, 当  $m = 9, w = 7$  时效果最好, 此时 LBD 描述子是一个 72 维浮点型的特征向量。在实现闭环检测时, 需要反复大量地计算特征向量之间的距离, 对于实时性要求较高的视觉 SLAM 而言, 这种描述方法是不适用的。为了提升计算效率, 需要将其转化为二进制的描述子。与 Brief 描述子类似, 二进制的 LBD 特征向量由许多个 0 和 1 组成, 这里的 0 和 1 编码了 72 维浮点型的特征向量中元素之间的大小关系。如图 3.7 所示, 如果第  $p$  个元素比第  $q$  个元素大, 则取 1, 反之取 0。按照一定的顺序取出 256 对, 最后就得到一个 256 维仅由 0、1 组成的向量, 本文采用的是 OpenCV 的实现。在采用二进制的 LBD 后, 可以使用汉明距离作为两个特征向量之间的距离度量, 通常只需要对这两个二进制字符串进行异或和求和操作, 大大提高了匹配的效率。

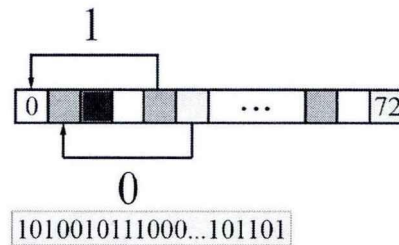


图 3.7 二进制 LBD 描述子



### 3.2 环境地图中的线

在后端优化中,需要对空间直线进行参数化,并设定观测方程。合适的参数化和观测方程对后端的计算复杂度和优化的精度具有一定的影响。下面将介绍空间直线的参数化以及观测方程。

#### 3.2.1 空间直线的参数化

空间上的点可以用中 $\mathbb{R}^3$ 的坐标表示:  $\tilde{\mathbf{X}} = [x, y, z]^T$ , 即该点的欧式坐标。由欧式空间和无穷远平面的并集所形成的扩展空间称为三维射影空间 $\mathbb{P}^3$ 。

1) 点的齐次坐标:  $\mathbf{X} = [x, y, z, w]^T$

2) 平面的齐次坐标:  $\boldsymbol{\pi} = [\pi_1, \pi_2, \pi_3, \pi_4]^T$

空间直线具有 4 个自由度,不像空间点和平面那样可以简单地用四维向量来表示。常见的有几种表示方法,如点表示、面表示。点表示的直观理解即两空间点确定一条直线,将这两空间点的齐次坐标作为 $\mathbf{W}$ 的行向量构成 $2 \times 4$ 矩阵:

$$\mathbf{W} = \begin{bmatrix} \mathbf{X}_1^T \\ \mathbf{X}_2^T \end{bmatrix} \quad (3.5)$$

然而这是一种非紧凑的表示,具有多余的 4 个参数,在优化问题中会引入额外的约束,在表达点线、线面相交以及相机投影时不太便利。这里引入其他两种空间直线的表示,在后面的文章中将采用这两种表示。

普吕克坐标由两空间点确定,其齐次坐标分别为 $\mathbf{X}_1 = (x_1, y_1, z_1, w_1)^T$ 和 $\mathbf{X}_2 = (x_2, y_2, z_2, w_2)^T$ 。首先定义普吕克矩阵:

$$\mathbf{L} = \mathbf{X}_2 \mathbf{X}_1^T - \mathbf{X}_1 \mathbf{X}_2^T \in \mathbb{R}^{4 \times 4} \quad (3.6)$$

普吕克矩阵具有特殊的性质,它是一个反对称矩阵,对角线上的元素均为 0,并且其行列式 $\det(\mathbf{L}) = 0$ 。若取同一空间直线上任意两对空间点,它们对应的普吕克坐标只相差一个系数,即 $\mathbf{L} = \alpha \mathbf{L}'$ 。因为反对称矩阵 $\mathbf{L}$ 中有 6 个非零元素,但仅有 5 个比率是有意义的,另外由于 $\det(\mathbf{L}) = 0$ 的正交约束,所以也反映出直线的自由度为 4。普吕克坐标即由普吕克矩阵中 6 个非零的元素,将其按照一定的顺序排列成的一个六维向量,这里采用 Bartoli 论文<sup>[51]</sup>中的记号。

$$\mathcal{L} = \begin{bmatrix} \tilde{\mathbf{X}}_1 \times \tilde{\mathbf{X}}_2 \\ w_1 \tilde{\mathbf{X}}_2 - w_2 \tilde{\mathbf{X}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} \in \mathbb{P}^5 \subset \mathbb{R}^6 \quad (3.7)$$

$\mathcal{L}$  中的元素与普吕克矩阵  $\mathbf{L}$  中的非零元素对应, 其中  $\mathbf{v}$  是直线的方向向量, 矩  $\mathbf{n}$  是过该直线且通过原点的平面的法向量, 并且垂直于  $\mathbf{v}$ , 即  $\mathbf{n}^T \mathbf{v} = 0$ , 如图 2.11 所示。原点到空间直线的垂直距离为  $\|\mathbf{n}\|/\|\mathbf{v}\|$ , 垂足  $\mathbf{d}$  的齐次坐标为  $(\mathbf{v} \times \mathbf{n}, \mathbf{v}^T \mathbf{v})$ 。

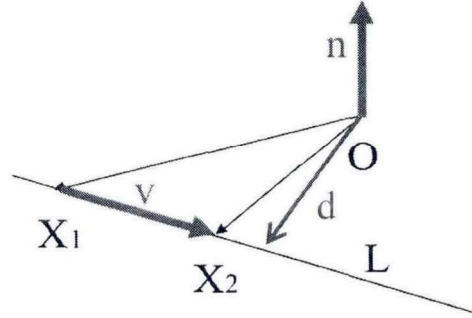


图 2.11 普吕克坐标

普吕克坐标与普吕克矩阵的关系如下:

$$\mathbf{L} = \begin{bmatrix} [\mathbf{n}]_{\times} & \mathbf{v} \\ -\mathbf{v}^T & 0 \end{bmatrix} \quad (3.8)$$

普吕克坐标在表达坐标变换和重投影时具有线性的形式, 其中坐标变换的公式为:

$$\begin{bmatrix} \mathbf{n}_c \\ \mathbf{v}_c \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cw} & [\mathbf{t}_{cw}]_{\times} \mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix} \begin{bmatrix} \mathbf{n}_w \\ \mathbf{v}_w \end{bmatrix} \quad (3.9)$$

$$\begin{bmatrix} \mathbf{n}_w \\ \mathbf{v}_w \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{cw}^T & -\mathbf{R}_{cw}^T [\mathbf{t}_{cw}]_{\times} \\ \mathbf{0} & \mathbf{R}_{cw}^T \end{bmatrix} \begin{bmatrix} \mathbf{n}_c \\ \mathbf{v}_c \end{bmatrix} \quad (3.10)$$

将空间直线投影到当前相机图像中, 得到其投影直线方程为:

$$\mathbf{l}_c = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & f_u c_v & f_u f_v \end{bmatrix} \mathbf{n}_c \quad (3.11)$$

然而这种 6 参数的形式大于空间直线自由度, 且存在正交约束, 因此普吕克坐标是一种过参数化的表示。在后端的图优化中, 多余的参数使得在迭代优化中增加了计算代价, 并且会降低系统的数值稳定性, 在迭代更新后需要强制满足普吕克的正交约束。因此 Bartoli<sup>[33]</sup>提出了一种 4 参数的最小化表示方法, 称为正交表示。

设普吕克坐标为  $\mathcal{L} = [\mathbf{n}^T, \mathbf{v}^T]^T$ , 其正交表示为  $(\mathbf{U}, \mathbf{W}) \in SO(3) \times SO(2)$ , 通过对矩阵  $[\mathbf{n} | \mathbf{v}]$  进行 QR 分解即可获得。由于存在正交约束  $\mathbf{n}^T \mathbf{v} = 0$ , 因此其分解的结果比较特殊:



$$\text{QR}([\mathbf{n}|\mathbf{v}]) = \mathbf{U} \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \\ 0 & 0 \end{bmatrix}, \quad (3.12)$$

$$\mathbf{W} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix}$$

其中  $w_1, w_2 > 0$ , 对应的  $\theta \in (0, \frac{\pi}{2}]$ 。事实上从普吕克坐标到正交表示并不需要进行 QR 分解, 可以一步写出:

$$[\mathbf{n}|\mathbf{v}] = \begin{bmatrix} \frac{\mathbf{n}}{\|\mathbf{n}\|} & \frac{\mathbf{v}}{\|\mathbf{v}\|} & \frac{\mathbf{n} \times \mathbf{v}}{\|\mathbf{n} \times \mathbf{v}\|} \end{bmatrix} \begin{bmatrix} \|\mathbf{n}\| & 0 \\ 0 & \|\mathbf{v}\| \\ 0 & 0 \end{bmatrix} \quad (3.13)$$

正交表示可以很容易地转为普吕克坐标:

$$\mathcal{L} = \begin{bmatrix} w_1 \mathbf{u}_1 \\ w_2 \mathbf{u}_2 \end{bmatrix} \quad (3.14)$$

其中  $\mathbf{u}_i$  表示矩阵  $\mathbf{U}$  的第  $i$  列。

空间直线的正交表示属于积流形 (product manifold), 因此可以利用切空间上的元素定义其增量。设增量为  $\delta_{\theta} = [\theta^T, \theta]^T$ , 使用前三个元素更新  $\mathbf{U}$ , 最后一个参数更新  $\mathbf{W}$ , 即可定义  $(\mathbf{U}^*, \mathbf{W}^*) = (\mathbf{U}, \mathbf{W}) \oplus \delta_{\theta}$ , 其中

$$\mathbf{U}^* = \exp(\hat{\theta}) \mathbf{U} \quad (3.15)$$

$$\mathbf{W}^* = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \mathbf{W} \quad (3.16)$$

$\delta$  中的 4 个参数具有具体的几何意义,  $\mathbf{W}$  隐含了  $\|\mathbf{n}\|/\|\mathbf{v}\|$  的比值, 即原点到空间直线的垂直距离  $d$ , 因此第 4 个参数  $\theta$  调节空间直线离原点的距离, 如图 3.8 中灰色的直线簇所示。在距离固定的情况下,  $\theta$  调节空间直线绕 3 个轴旋转, 在图中分别用橙色、绿色和蓝色的直线簇表示, 其中黑色直线为更新前的空间直线。

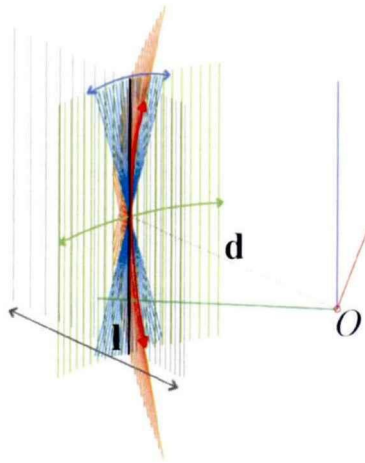


图 3.8 正交表示中参数更新的意义

然而正交表示并不能像普吕克坐标一样很好地表示坐标变换和重投影,因此在本文中除了在后端优化中采用正交表示,其他情况下均采用普吕克坐标。

### 3.2.2 空间直线的观测模型

空间直线的观测模型存在多种定义方式,可以通过 3D-3D 的方式定义空间直线与匹配空间直线的误差,也可以通过 3D-2D 的方式定义空间直线与匹配图像直线的误差,通常采用后一种方法精度更高<sup>[9]</sup>。

3D-2D 的方式首先将空间直线重投影到图像上,然后计算投影直线  $\mathbf{l}'$  和图像中的匹配线段  $\mathbf{l}$  的误差,图像中的线特征也可以用两个端点表示,分别为  $\mathbf{x}_s$  和  $\mathbf{x}_e$ 。Bartoli 列出了几种误差的定义方式<sup>[51]</sup>,如无几何意义的线性误差  $\mathbf{e} = \mathbf{l} \times \mathbf{l}'$ ,代数距离定义的线性误差  $\mathbf{e} = [\mathbf{x}_s^T \mathbf{l}', \mathbf{x}_e^T \mathbf{l}']^T$  等,并分析了各种误差的时间效率和优化后的精度。本文采用几何距离定义的非线性误差,其精度优于以上两种方法:

$$\mathbf{e} = \left[ \frac{\mathbf{x}_s^T \mathbf{l}'}{\sqrt{l_1'^2 + l_2'^2}}, \frac{\mathbf{x}_e^T \mathbf{l}'}{\sqrt{l_1'^2 + l_2'^2}} \right]^T \quad (3.17)$$

其中  $\mathbf{l}' = [l_1', l_2', l_3']^T$ 。由于孔径问题以及在连续帧的图像中空间直线端点不准确,只有匹配线段垂直于投影直线这部分才能提供有效信息,而平行于投影直线方向可能存在很大的误差,因此使用该模型作为观测模型是比较合理的。

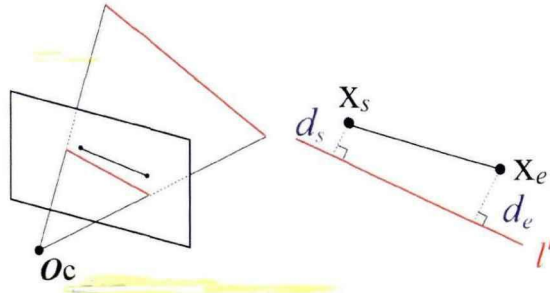


图 3.9 空间直线的误差定义

### 3.3 点线综合的后端优化框架

在以图优化框架的视觉 SLAM 算法里,BA (Bundle Adjustment) 起到了核心作用。BA 指的是,在给定多个 3D 点以及从不同视角观测到这些 3D 点的多组图片的情况下,同时提炼出最优的相机参数(内参数和外参数)和 3D 点的坐标。在近几年视觉 SLAM 理论的研究中,BA 算法不仅具有很高的精度,也开始具备良好的实时性,能够用于在线计算的 SLAM 场景中。而在 21 世纪早期,虽

然计算机视觉领域的研究者已经开始利用 BA 进行重构, 但 SLAM 的研究者通常认为包含大量特征点和相机位姿的 BA 计算量过大, 不适合实时计算。直到近十年来, 人们逐渐认识到 SLAM 问题中 BA 的稀疏特性, 才使它能够在实时的场景中使用<sup>[52]</sup>。

### 3.3.1 点线综合的图模型

BA 同样是一个状态估计问题, 在 2.3 小节已经进行了完整的叙述。在基于点特征的 SLAM 中, 以相机的位姿和三维点的坐标作为图模型的顶点, 定义三维点的重投影误差作为图模型中的边。加入了线特征后, 可以相应地增加图模型中的顶点和边, 构建点线综合的图模型。

为了描述的完备性, 这里再回顾一下点、线的观测模型, 并给出其数学表达。

假设图模型中相机的位姿为  $\mathbf{T}_{cw}$ , 表示世界坐标系相对于相机坐标系的位姿。

三维点  $\mathbf{X}_w$  与位姿的约束通过重投影误差来定义, 首先将  $\mathbf{X}_w$  转换到相机坐标系  $\mathbf{X}_c$ , 然后根据相机模型计算  $\mathbf{X}_c$  在图像上的像素坐标  $\mathbf{x}'$ , 最后计算投影点  $\mathbf{x}'$  与匹配特征点  $\mathbf{x}$  的误差。

$$\mathbf{X}_c = \mathbf{R}_{cw}\mathbf{X}_w + \mathbf{t}_{cw} \quad (3.18)$$

$$\mathbf{x}' = \mathbf{n}(\mathbf{K}\mathbf{X}_c) \quad (3.19)$$

$$\mathbf{e}_p = \mathbf{x} - \mathbf{x}' \quad (3.20)$$

其中  $\mathbf{n}(\cdot)$  表示将齐次坐标转换为非齐次坐标。

空间直线  $\mathcal{L}_w$  与位姿的约束由上一小节中的观测模型定义, 和三维点的处理方式相同, 首先将普吕克坐标  $\mathcal{L}_w$  转换到相机坐标系  $\mathcal{L}_c$ , 然后根据相机模型计算  $\mathcal{L}_c$  在图像上的投影直线  $\mathbf{l}'$ , 最后匹配线段  $\mathbf{z}$  两端点  $\mathbf{x}_s$ 、 $\mathbf{x}_e$  到投影直线  $\mathbf{l}'$  的距离。

$$\mathcal{L}_c = \begin{bmatrix} \mathbf{R}_{cw} & [\mathbf{t}_{cw}]_{\times} \mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix} \mathcal{L}_w \quad (3.21)$$

$$\mathbf{l}' = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & f_u c_v & f_u f_v \end{bmatrix} \mathbf{n}_c \quad (3.22)$$

$$\mathbf{e}_l = d(\mathbf{z}, \mathbf{l}') = \left[ \frac{\mathbf{x}_s^T \mathbf{l}'}{\sqrt{l_1^2 + l_2^2}}, \frac{\mathbf{x}_e^T \mathbf{l}'}{\sqrt{l_1^2 + l_2^2}} \right]^T \quad (3.23)$$

表 3.2 总结三维点和空间直线的坐标变换和投影方程, 采用了齐次坐标的表示使得式子更加简洁, 其中第三行表示将空间特征投影到双目的右图像中, 可以

构建双目的观测误差。

表 3.2 空间点和直线的相关操作

	空间点	空间直线
坐标变换	$\mathbf{X}_c = \mathbf{T}_{cw}\mathbf{X}_w$	$\mathcal{L}_c = \mathcal{H}_{cw}\mathcal{L}_w$
投影	$\mathbf{x} = \mathbf{K}\mathbf{X}_c$	$\mathbf{l} = \mathcal{K}\mathbf{n}_c$
投影（右相机）	$\mathbf{x}_R = \mathbf{K}(\mathbf{X}_c - [b, 0, 0]^T)$	$\mathbf{l}_R = \mathcal{K}(\mathbf{n}_c - [b, 0, 0]^T \times \mathbf{v}_c)$

将三维点和空间直线进行重投影后的结果如图 3.10 所示，图中黑色的点和线表示当前相机的观测，红色的为重投影后的点和直线，这里仅以左相机的投影为例。

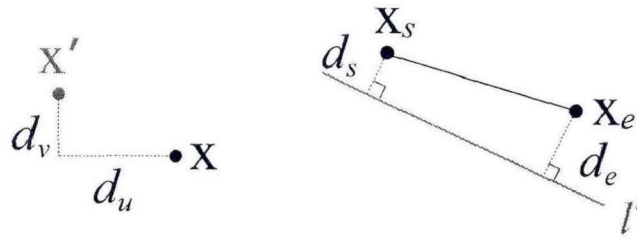


图 3.10 点和线的重投影误差

由于噪声的存在，投影的点和线并不会和观测重合，可以得到观测的误差。因此在点线综合的图模型中，以相机的位姿 $\mathbf{T}_{kw}$ 、三维点的坐标 $\mathbf{X}_{w,i}$ 和空间直线的普吕克坐标 $\mathcal{L}_{w,j}$ 作为的顶点，根据前端数据关联的结果建立两种类型的边：位姿—三维点边和位姿—空间直线边。这两种类型边的误差为：

$$\mathbf{ep}_{k,i} = \mathbf{x}_{k,i} - \mathbf{n}(\mathbf{K}\mathbf{T}_{kw}\mathbf{X}_{w,i}) \quad (3.24)$$

$$\mathbf{el}_{k,j} = \mathbf{d}(\mathbf{z}_{k,j}, \mathcal{K}\mathcal{H}_{cw}\mathcal{L}_{w,j}) \quad (3.25)$$

在假设观测误差为高斯分布的情况下，可以得到最后的代价函数 $\mathcal{C}$ ，即求解待估计状态量使得所有三维点和空间直线的重投影误差的和最小，如图 3.11 所示。

$$\mathcal{C} = \sum_{k,i} \rho_p(\mathbf{ep}_{k,i}^T \Sigma \mathbf{p}_{k,i}^{-1} \mathbf{ep}_{k,i}) + \sum_{k,j} \rho_l(\mathbf{el}_{k,j}^T \Sigma \mathbf{l}_{k,j}^{-1} \mathbf{el}_{k,j}) \quad (3.26)$$

其中 $\Sigma \mathbf{p}$ 、 $\Sigma \mathbf{l}$ 表示点、线的观测协方差， $\rho_p$ 、 $\rho_l$ 为 Huber 鲁棒性代价函数。由于最小化的是误差项的二范数平方和，其增长速度是误差的平方，若出现误匹配等会导致系统向错误的值进行优化，因此引入 Huber 函数可以降低代价函数中的

异常项。

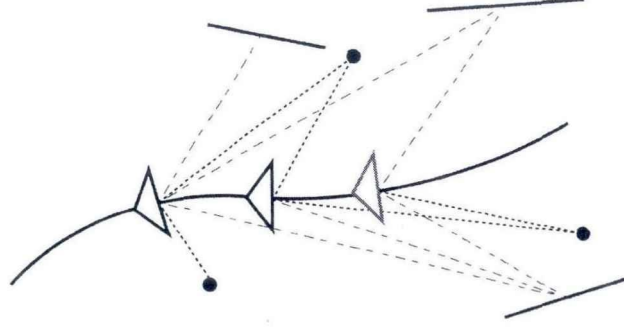


图 3.11 点线综合的图模型

### 3.3.2 点线误差的雅可比矩阵

在求解 BA 的过程中，不管是采用高斯牛顿还是 LM 算法，都需反复地要在当前估计值附近对代价函数进行线性展开，即求解误差函数关于状态变量的雅可比矩阵  $J = \frac{\partial e}{\partial x} \Big|_{\hat{x}}$ 。

通常可以使用有限差分法的方式求解，对于误差函数  $e(x)$ ，若  $x$  为流形中的元素， $\delta$  为切空间上的微小增量，则

$$J = \frac{e(x \oplus \delta) - e(x \oplus (-\delta))}{2\delta} \quad (3.27)$$

由于需要不停地迭代求解，采用数值方法精度不高且效率较低，最佳的方法是能够直接得到雅可比矩阵的解析形式。对于三维点的重投影误差的雅可比矩阵，在诸多文献上都有提到<sup>[46, 53]</sup>。对于空间直线重投影误差的雅可比矩阵，目前并没有相关文献。

首先给出三维点的重投影误差(3.20)关于位姿增量  $\delta_\xi$  和三维点坐标  $X_w$  的雅可比矩阵。对于坐标变换  $X_c = R_{cw}X_w + t_{cw}$  部分，通过在  $T_{cw}$  上叠加一个微小的扰动  $\exp(\delta_\xi^{\wedge})$  进行求解，在不影响推导过程和结果的前提下，将  $X$  的非其次坐标和齐次坐标进行混用：

$$\begin{aligned} \frac{\partial X_c}{\partial \delta_\xi} &= \lim_{\delta_\xi \rightarrow 0} \frac{\exp(\delta_\xi^{\wedge}) T_{cw} X_w - T_{cw} X_w}{\delta_\xi} \\ &\approx \lim_{\delta_\xi \rightarrow 0} \frac{\left( I + \begin{bmatrix} [\delta_\phi]_{\times} & \delta_\rho \\ 0^T & 0 \end{bmatrix} \right) T_{cw} X_w - T_{cw} X_w}{\delta_\xi} \end{aligned} \quad (3.28)$$

$$\begin{aligned}
&= \lim_{\partial \delta_\xi \rightarrow 0} \frac{\begin{bmatrix} [\delta_\phi]_\times & \delta_\rho \\ \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{cw} \mathbf{X}_w + \mathbf{t}_{cw} \\ 1 \end{bmatrix}}{\delta_\xi} \\
&= \lim_{\partial \delta_\xi \rightarrow 0} \frac{[\delta_\phi]_\times \mathbf{X}_c + \delta_\rho}{\delta_\xi} \\
&= [-[\mathbf{X}_c]_\times \quad \mathbf{I}]_{3 \times 6} \\
\frac{\partial \mathbf{X}_c}{\partial \mathbf{X}_w} &= \lim_{\partial \delta_\xi \rightarrow 0} \frac{\partial \exp(\delta_\xi) \mathbf{T}_{cw} \mathbf{X}_w}{\partial \mathbf{X}_w} = \mathbf{R}_{cw}
\end{aligned} \tag{3.29}$$

对于相机投影  $\mathbf{x}' = \mathbf{n}(\mathbf{K}\mathbf{X}_c)$  部分:

$$\frac{\partial \mathbf{x}'}{\partial \mathbf{X}_c} = \begin{bmatrix} f_u/Z_c & 0 & -f_u X_c/Z_c^2 \\ 0 & f_v/Z_c & -f_v Y_c/Z_c^2 \end{bmatrix}_{2 \times 3} \tag{3.30}$$

其中,  $\mathbf{X}_c = [X_c, Y_c, Z_c]^T$ , 最后通过链式法则可以求得重投影误差  $\mathbf{e}_p = \mathbf{x} - \mathbf{x}'$  关于位姿和三维点的雅可比矩阵:

$$\mathbf{J}_{p_\xi} = \frac{\partial \mathbf{e}_p}{\partial \delta_\xi} = -\frac{\partial \mathbf{x}'}{\partial \mathbf{X}_c} \frac{\partial \mathbf{X}_c}{\partial \delta_\xi} \tag{3.31}$$

$$\mathbf{J}_{p_x} = \frac{\partial \mathbf{e}_p}{\partial \mathbf{X}_w} = -\frac{\partial \mathbf{x}'}{\partial \mathbf{X}_c} \frac{\partial \mathbf{X}_c}{\partial \mathbf{X}_w} \tag{3.32}$$

下面将重点叙述空间直线重投影误差的雅可比矩阵的求解。需要求解直线重投影误差(3.23)关于位姿增量  $\delta_\xi$  和空间直线正交表示增量  $\delta_\theta$  的雅可比矩阵。

对于  $\mathbf{e}_l = d(\mathbf{z}, \mathbf{l}')$  部分, 设两端点的坐标为  $\mathbf{x}_s = [u_1, v_1, 1]^T$ 、 $\mathbf{x}_e = [u_2, v_2, 1]^T$ , 误差对投影直线  $\mathbf{l}' = [l_1, l_2, l_3]^T$  的雅可比矩阵为

$$\begin{aligned}
\frac{\partial \mathbf{e}_l}{\partial \mathbf{l}'} &= \begin{bmatrix} \frac{u_1 l_2^2 - l_1 l_2 v_1 - l_1 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{v_1 l_1^2 - l_1 l_2 u_1 - l_2 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{1}{\sqrt{l_1^2 + l_2^2}} \\ \frac{u_2 l_2^2 - l_1 l_2 v_2 - l_1 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{v_2 l_1^2 - l_1 l_2 v_2 - l_2 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} & \frac{1}{\sqrt{l_1^2 + l_2^2}} \end{bmatrix} \\
&= \frac{1}{l_n} \begin{bmatrix} u_1 - \frac{l_1 e_1}{l_n^2} & v_1 - \frac{l_2 e_1}{l_n^2} & 1 \\ u_2 - \frac{l_1 e_2}{l_n^2} & v_2 - \frac{l_2 e_2}{l_n^2} & 1 \end{bmatrix}_{2 \times 3}
\end{aligned} \tag{3.33}$$

其中,  $e_1 = \mathbf{x}_s^T \mathbf{l}'$ ,  $e_2 = \mathbf{x}_e^T \mathbf{l}'$ ,  $l_n = \sqrt{l_1^2 + l_2^2}$ 。上面式子中第二个等号中简化部分

的推导如下，对于第一行第一列的元素，有

$$\begin{aligned} \frac{u_1 l_2^2 - l_1 l_2 v_1 - l_1 l_3}{(l_1^2 + l_2^2)^{\frac{3}{2}}} &= \frac{(u_1 l_2^2 + u_1 l_1^2) - (u_1 l_1^2 + l_1 l_2 v_1 + l_1 l_3)}{(l_1^2 + l_2^2)^{\frac{3}{2}}} \\ &= \frac{1}{l_n} (u_1 - \frac{l_1 e_1}{l_n^2}) \end{aligned} \quad (3.34)$$

对于相机投影  $\mathbf{l}' = \mathcal{K} \mathbf{n}_c$  部分，投影直线  $\mathbf{l}'$  对普吕克坐标  $\mathcal{L}_c$  雅克比矩阵为

$$\frac{\partial \mathbf{l}'}{\partial \mathcal{L}_c} = \frac{\partial \mathcal{K} \mathbf{n}_c}{\partial \mathcal{L}_c} = [\mathcal{K} \quad \mathbf{0}]_{3 \times 6} \quad (3.35)$$

对于坐标变换  $\mathcal{L}_c = \mathcal{H}_{cw} \mathcal{L}_w$  部分，分别对位姿增量  $\delta_\xi$  和空间直线正交表示增量  $\delta_\theta$

求雅克比矩阵，令  $\mathcal{L}_w$  的正交表示为  $\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{bmatrix}$ ， $\mathbf{W} = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix}$ 。

$$\frac{\partial \mathcal{L}_c}{\partial \mathcal{L}_w} = \frac{\partial \mathcal{H}_{cw} \mathcal{L}_w}{\partial \mathcal{L}_w} = \mathcal{H}_{cw} \quad (3.36)$$

$$\frac{\partial \mathcal{L}_w}{\partial \delta_\theta} = \begin{bmatrix} -[w_1 \mathbf{u}_1]_\times & -w_2 \mathbf{u}_1 \\ -[w_2 \mathbf{u}_2]_\times & -w_1 \mathbf{u}_2 \end{bmatrix}_{6 \times 4} \quad (3.37)$$

$\frac{\partial \mathcal{L}_c}{\partial \delta_\xi}$  的求解比较复杂，在此将其分解为两部分，分别对  $\delta_\xi$  的前 3 个量和后 3 个量

求解雅克比矩阵。首先对于位移  $\delta_\rho$ ，令旋转部分  $\delta_\phi = \mathbf{0}$ ，则  $\delta_\xi = \begin{bmatrix} \mathbf{0} \\ \delta_\rho \end{bmatrix}$ ，先求出更新后的空间直线  $\mathcal{L}_c^*$

$$\mathbf{T}^* = \exp(\delta_\xi^\wedge) \mathbf{T}_{cw} \approx \begin{bmatrix} \mathbf{I} & \delta_\rho \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{T}_{cw}$$

$$\mathbf{R}^* = \mathbf{R}_{cw}, \quad \mathbf{t}^* = \delta_\rho + \mathbf{t}_{cw}$$

$$\mathcal{H}_{cw}^* = \begin{bmatrix} \mathbf{R}_{cw} & [\delta_\rho + \mathbf{t}_{cw}]_\times \mathbf{R}_{cw} \\ \mathbf{0} & \mathbf{R}_{cw} \end{bmatrix}$$

$$\mathcal{L}_c^* = \mathcal{H}_{cw}^* \mathcal{L}_w = \begin{bmatrix} \mathbf{R}_{cw} \mathbf{n}_w + [\delta_\rho + \mathbf{t}_{cw}]_\times \mathbf{R}_{cw} \mathbf{v}_w \\ \mathbf{R}_{cw} \mathbf{v}_w \end{bmatrix} \quad (3.38)$$

再对  $\delta_\rho$  求偏导，移除不相关的项可得

$$\begin{aligned} \frac{\partial \mathcal{L}_c^*}{\partial \delta_\rho} &= \begin{bmatrix} \frac{[\delta_\rho + \mathbf{t}_{cw}]_\times \mathbf{R}_{cw} \mathbf{v}_w}{\partial \delta_\rho} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} -\frac{[\mathbf{R}_{cw} \mathbf{v}_w]_\times \delta_\rho}{\partial \delta_\rho} \\ \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} -[\mathbf{R}_{cw} \mathbf{v}_w]_\times \\ \mathbf{0} \end{bmatrix}_{6 \times 3} \end{aligned}$$

$$\begin{aligned} a \times b &= -b \times a \\ [p + \mathbf{t}_{cw}]^\wedge \mathbf{R}_{cw}^* \mathbf{v}_w &= [p + \mathbf{t}_{cw}]^\wedge \mathbf{v}_c \\ &= -[\mathbf{v}_c]^\wedge (p + \mathbf{t}_{cw}) \\ &= -[\mathbf{R}_{cw}^* \mathbf{v}_w]^\wedge (p + \mathbf{t}_{cw}) \\ &\text{因为 } \mathbf{t}_{cw} \text{ 与 } p \text{ 不相关，所以偏导为 } 0, \\ &\text{可以移除此项，即} \\ &= -[\mathbf{R}_{cw}^* \mathbf{v}_w]^\wedge p \end{aligned} \quad (3.39)$$

其次对于旋转  $\delta_\phi$ ，令  $\delta_\rho = \mathbf{0}$ ，则  $\delta_\xi = \begin{bmatrix} \delta_\phi \\ \mathbf{0} \end{bmatrix}$ ，同样先求出更新后的空间

直线  $\mathcal{L}_c^*$ ，为了使式子简明这里去掉了  $\mathbf{R}$  和  $\mathbf{t}$  的下标

$$\begin{aligned}\mathbf{T}^* &= \exp(\delta_\xi^{\wedge}) \mathbf{T}_{cw} \approx \left( \mathbf{I} + \begin{bmatrix} [\delta_\phi]_{\times} & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \right) \mathbf{T}_{cw} \\ \mathbf{R}^* &= (\mathbf{I} + [\delta_\phi]_{\times}) \mathbf{R}_{cw}, \quad \mathbf{t}^* = (\mathbf{I} + [\delta_\phi]_{\times}) \mathbf{t}_{cw} \\ \mathcal{H}_{cw}^* &= \begin{bmatrix} (\mathbf{I} + [\delta_\phi]_{\times}) \mathbf{R} & (\mathbf{I} + [\delta_\phi]_{\times}) [\mathbf{t}]_{\times} \mathbf{R} \\ \mathbf{0} & (\mathbf{I} + [\delta_\phi]_{\times}) \mathbf{R} \end{bmatrix} \\ \mathcal{L}_c^* &= \mathcal{H}_{cw}^* \mathcal{L}_w = \begin{bmatrix} (\mathbf{I} + [\delta_\phi]_{\times}) \mathbf{R} \mathbf{n} + (\mathbf{I} + [\delta_\phi]_{\times}) [\mathbf{t}]_{\times} \mathbf{R} \mathbf{v} \\ (\mathbf{I} + [\delta_\phi]_{\times}) \mathbf{R} \mathbf{v} \end{bmatrix}\end{aligned}\quad (3.40)$$

上面的式子中用到了  $(\mathbf{R}\mathbf{a}) \times (\mathbf{R}\mathbf{b}) = \mathbf{R}(\mathbf{a} \times \mathbf{b})$ ,  $\mathbf{R} \in SO(3)$  这条性质。

再对  $\delta_\phi$  求偏导，移除不相关的项可得

$$\begin{aligned}\frac{\partial \mathcal{L}_c^*}{\partial \delta_\phi} &= \begin{bmatrix} \frac{\partial [\delta_\phi]_{\times} \mathbf{R} \mathbf{n}}{\partial \delta_\phi} + \frac{\partial [\delta_\phi]_{\times} [\mathbf{t}]_{\times} \mathbf{R} \mathbf{v}}{\partial \delta_\phi} \\ \frac{\partial [\delta_\phi]_{\times} \mathbf{R} \mathbf{v}}{\partial \delta_\phi} \end{bmatrix} = \begin{bmatrix} -\frac{[\mathbf{R} \mathbf{n}]_{\times} \delta_\phi}{\partial \delta_\phi} - \frac{\partial [[\mathbf{t}]_{\times} \mathbf{R} \mathbf{v}]_{\times} \delta_\phi}{\partial \delta_\phi} \\ -\frac{[\mathbf{R} \mathbf{v}]_{\times} \delta_\phi}{\partial \delta_\phi} \end{bmatrix} \\ &= \begin{bmatrix} -[\mathbf{R} \mathbf{n}]_{\times} - [[\mathbf{t}]_{\times} \mathbf{R} \mathbf{v}]_{\times} \\ -[\mathbf{R} \mathbf{v}]_{\times} \end{bmatrix}_{6 \times 3}\end{aligned}\quad (3.41)$$

因此

$$\frac{\partial \mathcal{L}_c}{\partial \delta_\xi} = \frac{\partial \mathcal{L}_c^*}{\partial \delta_\xi} = \begin{bmatrix} -[\mathbf{R}_{cw} \mathbf{n}_w]_{\times} - [[\mathbf{t}_{cw}]_{\times} \mathbf{R}_{cw} \mathbf{v}_w]_{\times} & -[\mathbf{R}_{cw} \mathbf{v}_w]_{\times} \\ -[\mathbf{R}_{cw} \mathbf{v}_w]_{\times} & \mathbf{0} \end{bmatrix}_{6 \times 6}\quad (3.42)$$

最后通过链式法则可以求得重投影误差  $\mathbf{e}_l$  关于位姿和空间点的雅克比矩阵：

$$\mathbf{J}l_\xi = \frac{\partial \mathbf{e}_l}{\partial \delta_\xi} = \frac{\partial \mathbf{e}_l}{\partial \mathbf{l}'} \frac{\partial \mathbf{l}'}{\partial \mathcal{L}_c} \frac{\partial \mathcal{L}_c}{\partial \delta_\xi}\quad (3.43)$$

$$\mathbf{J}l_\theta = \frac{\partial \mathbf{e}_l}{\partial \delta_\theta} = \frac{\partial \mathbf{e}_l}{\partial \mathbf{l}'} \frac{\partial \mathbf{l}'}{\partial \mathcal{L}_c} \frac{\partial \mathcal{L}_c}{\partial \mathcal{L}_w} \frac{\partial \mathcal{L}_w}{\partial \delta_\theta}\quad (3.44)$$

### 3.4 点线综合的视觉词典

SLAM 中的闭环算法指的是通过一定的算法检测出是否访问过同一个地方。闭环检测是 SLAM 前端中一个重要的部分，由于短期的数据关联仅考虑相邻时间上的关联，之前产生的误差将不可避免地累计到下一个时刻，使得整个 SLAM 会出现累积误差，无法构建全局一致的轨迹和地图。而闭环检测通过识别相识场景，增加图模型的约束，可以很好地消除累积误差，保证地图的一致性。需要强



## 作者简历

谢晓佳：男，1991年8月20日生于广东深圳。2010年9月至2014年6月就读于电子科技大学机械电子工程学院，获得工学学士学位。2014年9月推荐免试进入浙江大学控制科学与工程学系智能系统与控制研究所学习，师从刘勇教授，攻读控制科学与工程硕士学位。主要研究包括：SLAM和多传感器融合。

## 攻读学位期间取得的科研成果

1. Robust Visual SLAM with Point and Line Features, 投稿于 Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. 目前正在审稿中。