摘 要

三维物体识别一直是计算机视觉的一个重要方向，如何快速而准确的识别出物体的形状与方位是研究的关键问题之一。工程中存在诸多实际应用,并可进行有效简化。例如，在可视化导航应用中，两个相邻位置的变换通常很小或者近似知道。但是对于复杂物体，特别是自由形态物体，由于描述的复杂性以及分割的低可靠性使得问题变得尤为困难。本文讨论的基于模型几何学原理的匹配算法正是为满足这一需求而提出的。匹配过程中，物体是用任意形状的曲线表示，即实际中可能存在的任意空间曲线，并将曲线用序列点集的形式表示。算法直接对用深度图建立的物体表面三维数据坐标进行处理，而不需要对物体的特征进行假设与分割。算法的思想是在迭代最近点匹配算法(ICP)的基础上将一条曲线上的点与另一条曲线上的最近点迭代匹配，利用最小二乘法的优化算法估计相应点的三维变换，从而不断减小两条曲线（序列点）间的平均距离。仿真试验表明，这是一种高效、精确的三维物体识别与配准的估计方法。

论文对近十年来三维物体识别系统必须解决的传感器类型、三维物体表达方法和匹配策略等三个方面问题进行了综述，并对主要方法进行了分类和总结。提出了一种精确而有效的估计三维物体运动参数的数学方法—对偶四元数法，并通过实验检验了对偶四元数法的性能、精度和速度。论文将对偶四元数法应用到三维物体识别的匹配算法中，利用最小二乘的思想提出了一种基于迭代最近点的匹配算法。通过对实际应用中的几个问题的讨论，明确了算法需要解决的关键技术和主要运算步骤。通过模拟数据仿真试验，验证了算法的鲁棒性和有效性。

关键词：计算机视觉；ICP 算法；几何匹配；对偶四元数法

ABSTRACT

3D object recognition is a challenge problem in computer vision, in which the most important task is how to identify the shape and the pose of the object rapidly and accurately. There are many practical applications in project. In visual navigation, for example, the motion between successive positions is usually either small or approximately known. But regarding to complex object especially free-form object, it is difficult to achieve proper registration result because of the complexity of representation and low segmentation reliability. The matching algorithm described in this report that is based on geometric principle of model meets this need. In process of recognition, objects are represented by free-form curves, i.e., arbitrary space curves of the type found in practice. A curve is available in the form of a set of chained points. The algorithm deals with 3D object coordinate directly without need of an explicit object description in terms of primitives. The idea of algorithm is lied in iteratively matching points on one curve to the closest points on the other, which based on iteratively closest point (ICP) algorithm. A least-squares technique is used to estimate 3D motion from the point correspondences, which reduces the average distance between curves in the two sets. The results of emulation experiment with synthetic and real data indicate that it is an efficient and accurate estimation method of 3D object recognition and registration.

The report summarizes three problems including sensor types, 3D object representation methods and matching strategies during 10 years, which need to be solved by 3D object recognition system, furthermore, it classifies and sums up those of main methods. Then, we work out an accurate and efficient mathematics method, dual quaternion algorithm, which can be used to estimate 3D object motion parameters, and validate the algorithm's performance, precision and speed by experiments. Last, we apply the dual quaternion algorithm to matching algorithm of 3D object recognition and make use of least-squares technique to propose a matching method based on iterative closest point. The key technologies and

main operation steps of algorithm are confirmed by demonstrating several problems in practical applications. We apply emulation experiment with synthetic and real data to prove the validity and robust of our method.

Key Word: Computer Vision, ICP Algorithm, Geometric Matching, Dual Quaternion Algorithm.

哈尔滨工程大学
学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：_____

日期：　　年　　月　　日

第1章 绪论

1.1 引言

计算机视觉是在图像处理的基础上发展出来的新兴学科，它从信息处理的层次研究信息的认识过程，研究视觉信息处理的计算机理论、表达与计算方法。从视觉计算理论出发，可以将计算机视觉研究分为自上而下的三个阶段[1]，即视觉信息从最初的原始数据（二维数据）到最终的三维环境表达所经历的三个阶段。第一阶段是构成所谓的“要素图”或“基元图”(primary sketch)，它由二维图像中的边缘点、直线段、曲线、顶点、文理等基本几何元素或特征组成；第二阶段，称为环境的 2.5 维表述，即部分不完整的三维信息描述；从“计算”语言上讲，就是重建三维物体在观察者为中心的坐标系下的三维物体形状。事实上，从各个不同角度观察物体获得的信息都是不完整的。因此，2.5 维物体表述必须进一步处理以得到物体的完整三维描述，而且必须是物体本身某一固定坐标下的描述，这一阶段称为三维阶段。

计算机视觉处理的目的是由从景物图像中提取的信息实现对景物的描述。一般可以把计算机视觉大体分为对二维数据的处理和对三维数据的处理两个大的方向。对二维数据的处理通常按如下方式进行：从图像中提取诸如边缘这样的特征，或者将图像分割成区域，得到一个有用特性值标记的图像特征组成的表示图，然后对其作分组处理已得到有关信息。对三维数据的处理过程要复杂的多，由于三维物体数据描述的复杂性和物体特征分割的低可靠性使三维数据的处理过程很难直接使用传统的对二维数据的一般处理方法。然而在很多情况下，我们需要实现对三维数据的处理，例如机器人行为、计算机虚拟现实模拟、工业制造中三维物体数据的获得与描述等，其中应用最广，涉及计算机处理领域最多的是一种被称为反向工程(Reverse Engineering)的方法，即对已经存在的实物建立计算机几何模型。反向工程的主要应用领域包括计算机图形学、机器视觉以及医学成像等。反向工程一般涉及数据获取、预处理、数据分割与融合、CAD 模型建立等领域。一般流程如图 1.1 所示。

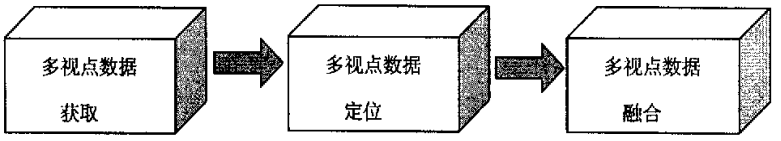


图 1.1 反向工程的流程图

1.2 课题的研究背景、目的和意义

计算机视觉系统的目标是解释已有的“视觉”数据，并使用这些解释去完成任务。在自动导航、自动检测、装配任务与医学图像分析等领域都存在这种应用的需求，三维物体识别是计算机视觉系统的基本功能，目前已成为该领域的研究热点。

计算机视觉系统一般可分为两大类[1]：一类是深度重建的计算机视觉系统，另一类是基于知识的计算机视觉系统。前者主要是从图像中得到物体的形状特征，这些特征包括立体视觉、运动、灰度、纹理等，而后得到 2.5 维图，即在以观察者为中心的坐标系中，把可见表面的朝向，大致轮廓及光的不连续轮廓表达清楚，最终和三维模型匹配，其识别工作是在三维域中进行的。第二类则是根据图像特征，通过感知机构得到图像中物体的二维特征结构群，它包括能代表图像中物体的大部分信息，而且在较大的观察范围内特征是不变的，而后通过基于知识的匹配过程在模型中找到对应的结构，和三维模型进行匹配，一个较为完整的计算机视觉系统必须包括图像获取，低层次的边界检测、分割、物体的建模、匹配和识别几个部分。

三维物体识别是计算机视觉的重要组成部分，它的任务是识别出图像中有什么类型的物体，并反映出物体在图像所反映的场景中的位置和方向，如果在识别过程中要利用有关物体的先验知识，则称为基于模型的三维物体识别。在基于模型的三维物体识别中，一般要通过两个阶段的处理，以完成对图像中 3D 物体的识别和定位[2]，即首先通过传感器获取的场景输入数据来得到场景的表达，然后将它与数据库中存储的物体表达相匹配，具体来说，三维物体识别系统必须解决如下几个问题[3]：（1）获取原始数据的传感器类型；（2）三维物体模型表达（构

造）方法；（3）描述原始数据和物体模型的工具；（4）从输入数据获得的物体描述和物体模型描述之间的匹配方法。其中，传感器决定了输入数据的分辨率、准确性和类型（2D 和 3D 信息）；而模型则提供了视觉系统的先验知识；表达则用于表述采集的数据和物体模型，是计算机视觉中的一个关键论题；另外，知识表达还决定了匹配策略、匹配策略的鲁棒性和系统效率，并且在匹配过程中表达还用来计算物体的不同特征。由于匹配策略是在识别系统运行时执行的，因此它必须解决数据和模型表述之间的二义性[4]。一旦确定了正确的匹配，则根据物体模型，既可以将已定位物体的方向和变换计算出来，最终完成物体识别的任务。

如上所述，匹配策略的设计是整个计算机视觉系统设计的关键技术和重要环节，也是本课题研究的重点。本课题就是针对三维物体识别中的匹配方法进行理论和实验研究。

1.3 国内外研究动态及课题引出

1.3.1 三维物体匹配策略

对三维物体进行空间定位与识别是计算机视觉应用研究领域中极为活跃的课题之一，国内外已有不少学者在这方面作了许多工作，并取得一定成就。根据 3D 物体识别过程，从输入图像中得到物体表达后，就需要对场景中存在的物体进行识别和定位。这种识别是通过将场景中得到的物体特征和物体模型中的特征进行匹配来完成的，而各种识别方法的不同之处不仅在于使用不同的物体表达方法，而且也在于如何完成场景特征和模型特征之间的匹配、如何从场景模型特征匹配中得到一致解释和如何从一致解释中得到姿态估计。针对这一问题，国际上也有许多学者进行了大量的研究工作，如今三维物体识别的主要匹配方法有[5]：①假设检验法；②解释树；③基于图的识别方法；④霍夫姿态聚类；⑤几何哈西表；⑥迭代模型拟合技术；⑦基于神经网络的匹配方法。这些方法的具体形式将在第二章中作详细描述。

本课题选取迭代模型拟合法，这类方法的一个明显特征是在计算上的迭代过程。当三维物体用参数进行表达描述，并在该参数既能确定形状，也能确定物体姿态的情况下，可以使用迭代模型拟合的方法。此时，物体识别和姿态估计可简

化成估算图像数据中的模型（姿态）参数，且可与存储的模型参数表达相匹配，即通过迭代的方法求解一个约束线性方程或者非线性方程系统，并选择与场景有最小平方和误差的解，作为估计参数。比较典型的是 Besl、Chen 和 Zhang 等提出的最临近点匹配算法，其目的是为了找到两幅深度图之间的变换关系，具体实现过程是通过迭代的方法找到平均平方误差最小的变换，也称作 ICP 算法（Iterative closest points），该方法广泛用于 3D 物体的识别。

将 ICP 算法应用到 3D 物体识别中必须考虑到几个关键问题，分别是：初值和全局最优化、匹配点的选择、对应关系的确定、刚性变换、迭代终止条件、鲁棒性、快速收敛性和精度评估等，它们之间是相互影响的。其中前四个是 ICP 算法的主要步骤，但是它们的执行都影响后三个问题。之前已有许多学者针对 ICP 算法的基本技术进行改进，如 Zhang 用鲁棒核代替误差函数加强了基本技术；Chen 和 Medioni 假设模型点能由曲面法线提供，它使得点对点的距离被点对切平面的距离所代替。这两种扩展在基本算法上都提高了收敛性，而且没有大幅度增加计算量。为了将 ICP 算法更有效的应用到 3D 物体识别中，本文提出的迭代匹配算法以实现对 ICP 算法的几个主要步骤进行改进，包括伪点匹配、最大距离容限、对偶四元数法计算变换矩阵等，在不同程度上提高了匹配性能。

ICP 算法的最基本前提或者说假设条件就是测量值无误差，但是这在实际中是不存在的，而且由于野值点（Outliers）的存在，可能使得 ICP 算法不能正常工作，所以如何提高算法的鲁棒性是 ICP 算法在目标识别应用中需要研究的方向之一。再有，ICP 算法被证明是单调收敛到局部最小的，但是如何让 ICP 算法快速地收敛到全局最优也是 ICP 算法的改进方向之一。在增强算法的鲁棒性的同时必然增加算法的复杂性和计算时间，鲁棒性和快速收敛性是相互矛盾的，因此在实际应用中应该综合考虑系统的情况来折衷选择算法的鲁棒性和快速性。本文针对这一问题提出了一种改进算法，即 LM-ICP 算法[6]，将非线性最优化算法（Levenberg-Marquardt）引入到 ICP 算法中，通过非线性最优化直接最小化模型数据的拟合误差，提高算法的收敛速度。使用该方法来生成一个鲁棒误差函数，这个函数大大增加了所给样本的收敛半径，而且在速度上没有很大的损失。因此从整体上可以改进算法的鲁棒性同时又提高快速收敛性。

1.3.2 几何匹配技术

几何匹配在计算机视觉和机器人视觉中是一个普遍存在而难于解决的问题，主要有两个方面的应用：目标识别和可视化导航。对于目标识别问题是将观测数据与代表不同目标的预存模型相匹配；可视化导航问题是匹配不同时刻的动态场景观测数据以便再现目标的运动和解释场景。1985 年以前的绝大部分工作主要集中在多面体目标[7]，通常使用几何元素，如，点、线和平面，与我们生活的真实世界相比，这当然非常有限。最近，曲线目标已经引起很多计算机视觉研究者的注意。本文采用曲线特别是任意形状的曲线即现实中发现的任意空间曲线来表示三维物体目标。

任意形状的曲线用序列点集合表示。对任意形状的曲线已经有文献提出了几种匹配技术。第一类技术是测量曲率的极值，然后用于匹配[8]，但是曲率极值的精确定位是非常困难的，特别是曲线非常平滑的时候。曲线的很小变化就改变了曲率极值的数值及其在曲线上的位置，因而，基于曲率极值的匹配对噪声极为敏感。第二类技术是将曲线变换为一个局部序列旋转和平移不变特征（即曲率和挠率），曲线的匹配问题就简化为一维串匹配问题[9,10,11]，由于使用更多的信息，这类方法可能比第一类方法更具鲁棒性，但是这些方法仍然受噪声干扰的影响，因为它们是用曲线的弧长来采样获得点集，这些弧长本身对噪声是敏感的。

上述方法是利用全局匹配准则的思想来处理任意形状曲线的两个点集，它们因大的运动或变换而不同，这种处理大运动的能力对于目标识别来说通常是非常重要的。在很多其它的应用中，如可视化导航，相继两帧（框架）中曲线的运动通常要么较小（因为目标的最大速度是有限的，而采样频率可以很高），要么在一个合理的精度内是已知的（因为一个移动的载体通常安装几种仪表如计程仪和惯性系统，可以提供这些运动信息），对后一种情况，第一帧可以首先应用给定的估计产生中间帧，之后认为中间帧与第一帧之间的运动很小。在运动很小的前提下就可以应用迭代匹配算法对第二帧进行估计。

本文提出的匹配方法正是针对小运动的三维物体进行配准，即直接对物体三维数据进行处理运算，以获得三维物体的识别与定位。

1.4 论文主要研究内容

完成对图像中 3D 物体的识别和定位，一般要通过两个阶段的处理，即首先通过传感器获取的场景输入数据来得到场景的表达，然后将它与数据库中存储的物体表达相匹配。由于本课题的重点在于匹配算法的研究与验证，因此对于三维物体识别中的数据获取和三维物体模型的构造方法只在第二章的综述中有所介绍，不对其进行深究。

本文主要研究三维物体数据的匹配方法。这种方法是一种基于模型的几何学匹配方法，使用最小二乘法估计数据间的刚体变换获得最佳旋转矩阵和最佳平移向量。物体与模型间的比较是通过迭代最近点匹配算法(ICP)的思想实现的，简单原理如图 1.2 所示。

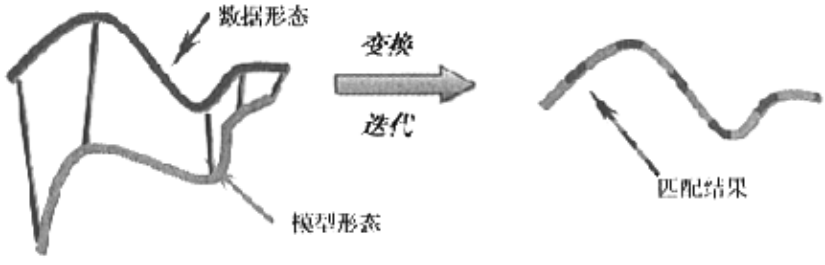


图 1.2 匹配算法简单原理图

虽然本文的算法应用了 ICP 算法的思想，但是与通常意义上的 ICP 算法比较，还有所不同。主要表现在，匹配过程中每一个步骤的具体实现方法都有不同的考虑，并且有所改进。包括伪点匹配、最大距离容限、迭代终止条件、k-D 树寻找最近点、引入加权系数和对偶四元数法计算刚体变换参数等。论文主要研究内容及结构安排如下：

第二章围绕着三维物体识别系统必须解决的传感器类型、三维物体表达方法和匹配策略等三个方面问题，对近十年来的研究成果进行了综述，并对主要方法进行了分类和总结。

第三章提出一种新的算法，用对偶四元数来解决物体定位问题。物体定位与物体识别是计算机视觉中一脉相承的问题，都是要用最小二乘法的最优技术估计

刚体变换参数**T**。这一章首先介绍了算法的理论基础，包括对偶数、四元数、对偶四元数，及刚体变换的对偶四元数表示法与相似矩阵表示法之间的转换，以及刚体变换中各个参数的求取方法。其次将物体定位参数的估计作为最优化问题，通过最小化一个与方向和位置误差总和有关的单值函数解出物体的方位。其中，物体数据与模型数据是用对偶四元数表示，则计算物体位置和方向的方法就是要选择对偶四元数使用其表示的误差函数最小化。最后通过实验检验了对偶四元数算法的性能、精度和速度。

第四章研究了三维物体识别中的匹配方法，提出一种基于迭代最近点的匹配算法。物体数据与模型数据都用 3D 点集合表示，算法通过迭代的方法找到平均平方误差最小的变换实现物体与模型的匹配。首先研究算法匹配过程中的四个步骤：寻找最近点、伪点匹配、匹配更新、计算运动，其中寻找最近点是整个过程中最耗时的步骤，本文选用 k-D 树的方法大大加速了这一过程；伪点匹配中对参加匹配的点对引入了两个限制条件，最大距离容限和方向的一致性。匹配更新中将最大距离容限在每次迭代过程中进行更新，它是在距离统计的基础上适当地设置的，因此算法更具鲁棒性。计算运动是从更新的匹配中计算两图像间的运动，选用对偶四元数法计算刚体的旋转和平移。将运动应用到物体数据中的所有点和它们的切线上，完成数据更新。算法引入了迭代终止条件：平移误差和旋转误差，在程序满足终止条件时终止迭代过程。接下来讨论了实际中应该考虑的问题包括搜索最近点，参数**D**的选择，和不确定性，这几方面细化了算法使其更有效的应用到实际中。最后模拟数据进行了仿真试验，分析实验结果验证了算法的鲁棒性和有效性。

第五章总结 ICP 算法的派生形式，针对算法的收敛性和鲁棒性简单介绍了一种改进算法，即 LM-ICP 算法，将非线性最优化算法（Levenberg-Marquardt 算法）引入到 ICP 算法中，通过非线性最优化直接最小化模型数据的拟合误差，从而提高算法的收敛速度。

第 2 章 三维物体的表述与匹配方法

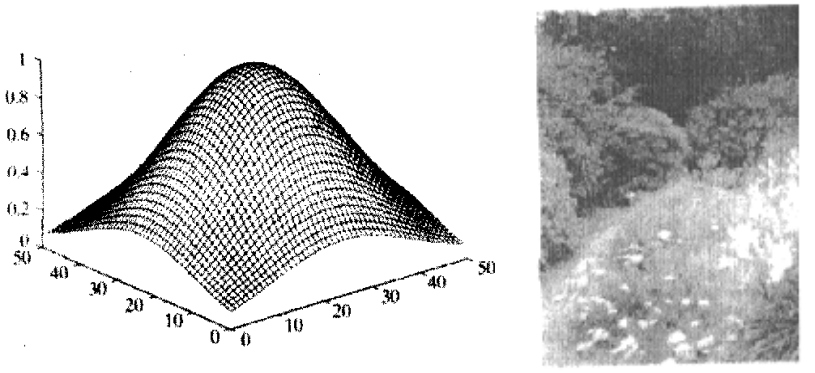
2.1 引言

计算机视觉的根本目的是由从景物图像数据中提取景物特征的描述，并利用这些描述去完成特定的任务，如船舶的自动导航、自动操纵避碰、机械零件装配任务或医学图像分析等，物体表达是匹配和识别的基础。

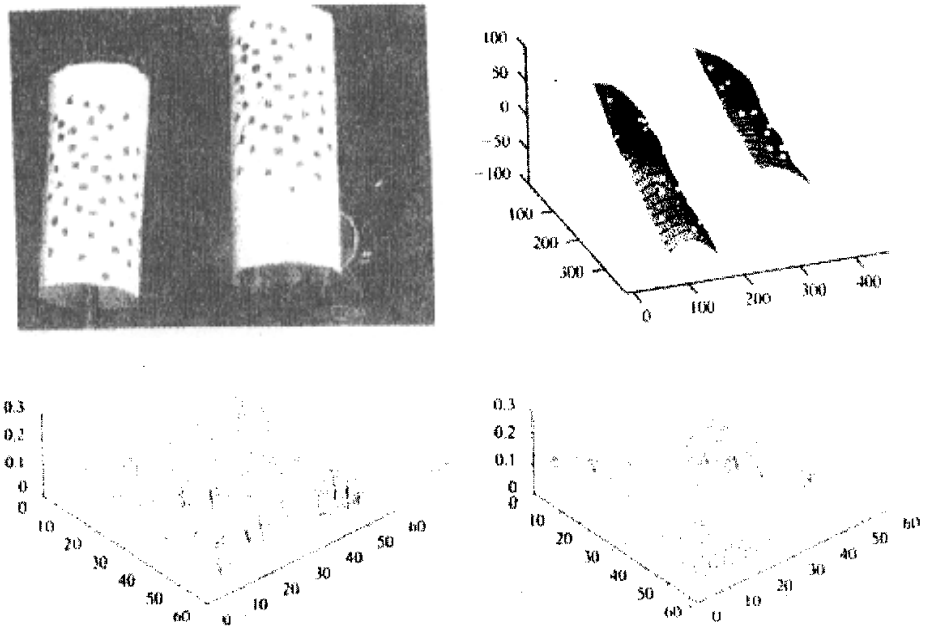
计算机视觉系统的处理过程要通过两个阶段，现将图像中的三维物体的识别和定位[2]，即通过传感器获取的物体输入数据获得物体的表达；然后将它与模型库中存储的物体表达相匹配。具体来说，识别系统必须解决如下几个问题[3]：(1)获取原始数据的传感器类型；(2)三维物体模型构造及构造方法；(3)描述原始数据和物体模型的数字化工具；(4)输入的物体描述数据和模型的描述数据之间的匹配方法。其中，传感器决定了输入数据的分辨率、准确性和类型(2D 或 3D 信息)；而模型是视觉系统的先验知识；物体的表达则用于描述采集的数据和物体模型之间的相互关系，是计算机视觉中需要解决的一个关键问题；同时，表达决定了匹配的实现方法及这些方法的鲁棒性和完成的效率，并且在匹配过程中还必须用表达来描述物体的特征、衡量配准的精度。由于匹配是在物体识别之前进行，因此，匹配必须解决测量数据和模型描述之间的二义性。一旦确定了正确的匹配，就可以根据模型库中的模型实现物体的识别，最终完成导航或避碰等要求的任务。

2.2 获取测量数据的传感器

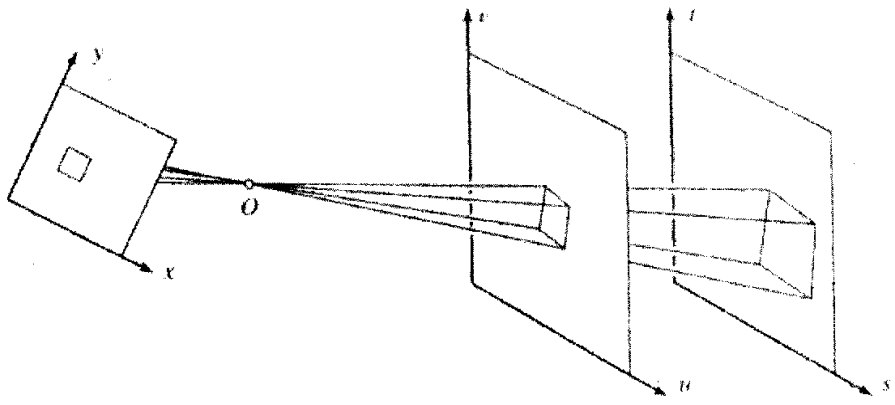
目前，获取测量数据常常采用两类传感器：一类是灰度传感器(Intensity sensor)，得到的是物体每个象素点对应的一个亮度测量；另一类是深度传感器(Range sensor)，其测量数据是象素点对应于从传感器到物体表面的距离。灰度传感器获得的数据受环境光照强度、物体表面反射特性和纹理特征等多种因素的影响，不是于采用几何特征的物体模型匹配，而深度图像能够明确表达三维物体的几何特征，更适合根据物体几何信息进行匹配的三维物体识别，因此得到了广泛的应用。图 2.2 为深度图像与灰度图像获取方法的比较示意图。



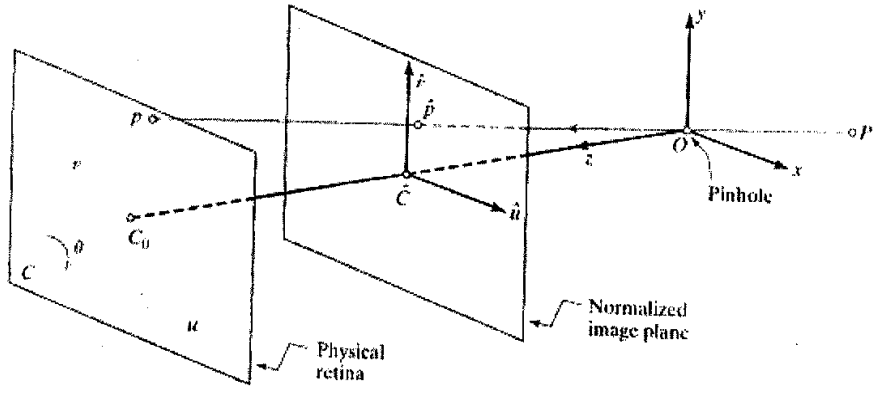
(a) 获取一幅灰度图像



(b) 获取一幅灰度图像



(c) 获取一幅深度图像



(d) 获取一幅深度图像

图 2.2 (a)、(b)、(c)、(d)为深度图像与灰度图像获取方法的比较示意图

2.3 三维物体表达方法

可以利用物体的其它信息如颜色、纹理、材质等来改善识别系统的性能，本

文只讨论物体的几何表达方法，而且具备下述条件[3]：(1)明确性(没有两个物体有相同的表达)；(2)唯一性(使用同一表达方法的每个物体必须只有一种描述)；(3)不敏感性(物体在部分被遮挡的情况下会缺少部分数据)；(4)方便匹配和存储。

2.3.1 表达方法的分类

三维物体表达方法可以分为以物体为中心的(Object centered)和以观察者为中心的(Viewer centered)两类。其中，以物体为中心的表达方法是在物体本身坐标系中描述物体，并使用与观察者视点无关的特征(如拐角、孔洞、边等等)来描述物体；以观察者为中心的表达方法则依赖于从某一个或多个视角来确定物体的外观，并使用与视点相关的特征(例如，遮挡的边缘、轮廓和形状等)。

根据描述三维物体所使用的几何特征不同，可以按照图 2.1 将表达方法从低到高分为如下 6 类：①基于三维点的表达方法，主要使用物体的深度方向的数据、表面点的法线方向和曲/挠率大小等信息；②基于凸点的表达方法，利用物体的顶点、拐角点和曲率极大、极小值点等信息；③基于轮廓的表达方法，主要是用物体的边缘信息；④基于表面的表达方法，主要使用物体的表面信息，例如平面、球面、二次曲面以及面与面之间的连接关系等；⑤基于体的表达方法，主要使用体素、椭球体、超级椭球体等体描述符；⑥基于零件的表达方法，主要分析物体的基本组成部分，例如体元（geon）方法。

2.3.2 表达方法的特点

下面分别介绍几类有代表性的三维物体表达方法，并具体论述各种方法的主要特点[5]。

(1) 基于基本表面特征的方法

物体表达为面、边和顶点的序列是在图像投影中与视角无关的特征，如 Grimson 和 Lozano-peres 曾使用边、直线段和方向矢量来表达多面体，由于使用多面体来表达逼近曲面物体的表面，需要大量的存储空间，所以可能使用平面和二次曲面方程来描述曲面表面或根据表面点的高斯曲率和平均曲率将表面片分成峰、凹、鞍)等等一些主要的形状更为合适。

同时 Stein 和 Medioni 提出了一种结构表达方法，这种方法根据表面法矢量

的分布来确定边和局部表面，因此它能够处理任意形状的物体表面；另外，Chitra Dorai 提出了 3D 自由形态物体的 COSMOS 表达方法，这种方法是先定义表面小区域上的归一化形状索引特征，然后在全部表面上进行统计，以得到形状谱，再根据形状索引特征得到最大一致形状表面片，两者联合起来则构成对任意形状物体的表达。

一般来说，基于局部边界和表面等基本表面特征的方法，对获取数据中的噪声敏感，并且准确性要依赖于是否能够从输入图像可靠地提取和描述物体特征。

(2)基于表面不连续性的表达(Discontinuity-based representation)法

用物体表面的不连续性来表达物体，如用物体的屋脊和阶越边缘来构造边缘链接图，其中，节点表示连接关系；边表示连接到一起的边缘；而 Chen 和 Stockman 则综合使用了表面和断点的信息，来标记边缘，并定义了飞边(wing)的概念，即每条飞边是一个面、边缘、面的三元组；另外，Chen 和 Stockman 还使用二维轮廓特征来描述 3D 任意形状物体。由于基于表面不连续性的表达方法是利用物体的边缘信息，因此降低了存储空间的需求，提高了高层处理的效率，但在大多数情况下，这种表达是不完整的，因为它丢失了某些表面信息特性。

(3)拟合表面参数表达法

将物体表面用参数方程拟合。如 Umasuthan 分析了高阶代数方程某些参数具备的不变特性，并用它们来表达 3D 物体的表面片；李松涛提出了一种双二次变量正交多项式的曲面拟合方法，该方法可根据拟合得到的代数方程参数来计算物体表面特征；另外使用 B 样条表面也可拟合 3D 表面，但表面拟合与识别的范围约束和收敛性都是需要彻底研究的领域。

另外，超二次曲面特征最早由 Barr 引入计算机视觉领域，而从深度图像获取超二次曲面表达也是通过对输入数据进行隐含方程(implicit equation)拟合得到的。周林还提出了一种扩展超二次曲面的表达方法，它比超二次曲面具有更强的描述能力。

在这类方法中，物体遮挡再次成为难以解决的问题，因为由物体部分外观生成的代数多项式与由物体全部外观生成的多项式可能不同，而且它首先需要进行区域分割，然后通过表面拟合才能得到代数方程的参数，这样得到的参数准确性将依赖于分割的结果。

(4)基于定位的表达（Orientation-based representation）法。

对物体表面上的任一点，都可以在高斯球(单位球)上找到与之表面方向相同的一点，这样就可以得到物体表面上的点到高斯球的映射。如果该物体表面是一凸性物体的表面，则上述映射是一一映射的，但高斯球表示的缺点是存在二义性，它不能保存物体的变换和尺寸信息。一些改进方法是将物体的表面法矢量根据支撑函数映射到单位球上，例如广义高斯图像(Generalized Gaussian image, GGI)由于存储了单位球上直接相邻点之间的连接信息，因此确保了对所有物体表达的唯一性；又如 Ghos 和 Kumar 对支撑函数进行了讨论，并证明了一些看似差异很大的表达方法，其本质上都是相同类型的支撑函数表达，同时也讨论了表达非凸物体的支撑函数。

基于定位的表达方法主要描述物体的表面信息，虽然它是一种全局的表达方法，但从识别的角度看，它们太烦琐，且不能处理物体有被部分遮挡的情况，而且，从由多个物体的场景图像生成的高斯图中分割出单独属于不同物体的区域，也是非常困难的问题。

(5)基于网格的表达法

网格表达法就是图形学中使用的三维任意形态物体描述方法，它使用多边形来表示物体的形状，常用的是四边形和三角形格网，由于其数据量大，因而存储、传输和运算都比较困难，很难直接用在三维物体识别中，因此经常要转换为其它表达。在这方面 Johnson 作了较有成效的研究，它将格网中某个基准顶点与其它顶点的几何位置关系转换成一个二维图像，称为 spin-image,它可用来描述物体的三维特征，进而实现识别和定位等工作。虽然引入了简化算法，但使用 spin-image 的操作计算还是太复杂。

(6)体素(voxel)法

体素表达主要用于描述物体的体特征，它类似于像素在平面中代表一小块面积，而体素在空间中则代表一小块体积，它将物体描述成一个互不重叠的立方体的集合，这些体素(立方体)分布在三维方形网格里紧密排列，并填充物体所占的空间。素表达并不适用于 3D 物体识别，主要用于物体的表面重建和建模上。

(7)八叉树(octree)法

八叉树(octree)法以分层的方式来描述物体，它是一种每个节点有 8 个分支(octant)的树型结构，且它的根节点是一个能够完全包住物体的立方体，分层描述

时它将物体所占的空间逐层迭代地划分为 8 个分量，迭代的停止条件是每个最终的子立方体在某种特征上是同质的，如 Chien 等提出了使用 3 对相互垂直的深度图像来生成物体 octree 模型的方案[26]；而 Li 则提出了一种利用任意视角深度图像来构造 octree 的方法，它能够描述物体的凹面，虽然 octree 表达能描述物体的全局性的体特征，但它不是一种精确的表示法，其描述物体的近似程度取决于分割的精度。

(8)构造实体几何法(Constructive solid geometry, CSG)法

构造实体几何法是用系统定义的简单几何形体，经过并、交、差等布尔组合，来构造出所需要的复杂实体，而每个物体表示乘一棵树，树的叶节点是快、楔、柱等简单物体或它们的变形，其它节点是构造物体的布尔操作。一般构造实体几何法经常用在构造物体的模型上，特别是在 CAD/CAM 系统中，而用在物体识别的时候则较少。这种表达方法的缺点是不能精确的表达物体，即表达不唯一，且建造时所用的简单刚体知识对识别系统来说，效果不好，因为识别系统种只有物体表面是可视的，而物体的组成部件可能被遮挡。

(9)体基元(geon)法

体基元表达可以作为物体的高层次表达方法。Biederman 提出人脑一定程度上可使用物体的组成部分来识别物体，而且可以用二维线条来描述物体组成部分的形状，因此他根据 4 种二维属性，定义了 24 类体基元，以用来描述物体的组成部分。Nguyen 和 Levine 应用了 Biederman 提出的体基元种类，并以物体的边缘连接图作为系统输入，然后将边缘连接图分解为子图，每个子图代表物体的一个组成部分，并对应于一个体基元，而体基元之间的连接关系由物体的凹边缘和 T 型连接确定。这种基本体基元的表达还可以使用超二次曲面和广义柱的方法。体基元表达方法的缺点是基元特性集合不一致，缺乏对这些集合存在的必要性、充分性和正确性的论证。另外，从物体的单幅外观表现来计算所有这些基元特性也非常困难。

(10)扫描(sweeping)法

扫描法表达是将形状描述成一个 2D 函数在 3D 空间中的扫描，其扫描的方式有平移和旋转[28]两种。这类方法中用得最广的是广义柱(Generalized cylinder)，而最普通的广义柱是一个圆沿着与圆平面垂直的轴平移所得到的柱体，而且如果圆的半径是轴位置的线性函数，则平移的结果是广义锥。为描述更多类型的形状，

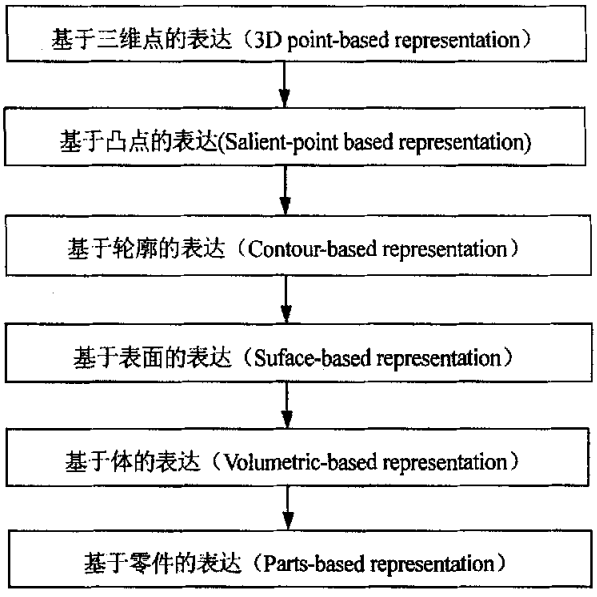


图 2.1　三维物体表达方法分类

人们定义了多种广义柱，例如，直线均匀广义柱(Straight homogeneous generalized cylinders)和曲面刚体旋转(Curved solid revolution)。广义柱的缺点是表达可能不唯一，因为为了表达数局，其起始的横界面可能以不同方式定义，因而致使相同的数据点有不同的表达方法。

(11)以观察者为中心的表达法

以物体为中心的表达方法，其目的都是在以一种本质的方式来表达物体的全部形状；而以观察者为中心的表达方法则是将物体表达成一个 2D 外观的集合，而不是单一的模型。其中外观图方法(The aspect graph)就是将三维物体的所有二维外观，分别组成有一定意义的外观聚类集合。它是将物体的视点空间，分割成一些具有相同外观表现(appearance)的区域，称为 aspect,并由发生在两类相邻观察(view)之间的视觉事件（visual event）分开，当物体的轮廓拓扑发生变化时，就产生一个视觉事件。基于外观的表达是采取了一个连接图的方式，其图中每个节点对应一个外观，每条边即对应视觉事件。

大多数以观察者为中心的表达，其主要缺点是描述物体不简洁，因而如果物体由几个参数方程描述，则以观察者为中心的表达就明显不合适，但是，在描述那些形状不能由简单分析方式或一套方程来获得的复杂物体时，这种方法就扮演了一个重要的角色。

表 2.1 是对以上这些方法的总结。

2.4 匹配方法

从输入图像中得到物体表达后，就需要对场景中存在的物体进行匹配和定位，并依此完成物体的识别。物体表达方法不同决定了识别方法不同，完成物体测量特征和模型特征之间的匹配方法也不尽相同、如何从物体模型特征中得到一致解释和如何从一致解释中得到物体形状和姿态的估计是匹配所要完成的任务。

通常一个识别系统可能会联合使用多种匹配方法。

(1) 假设验证法

假设验证法是基于模型知识，并采用了自低向上(bottom-up)与自上而下(top-down)相结合的策略，这样可以大幅度的减少图像低层次处理与识别的计算时间[1]。

该方法是先将假设的物体模型特征和部分场景特征联合起来以得到一个超限定(over-constrained)线性方程或非线性方程，然后通过解此方程组，以得到标志匹配质量平均误差最小的解，并将作为从物体参考坐标系到场景参考坐标系的变换。验证的过程是将假设模型的其它特征变换到场景中，再利用原场景图像来检查这些特征的位置是否正确，从而确定假设的正确性。通过配准的识别方法(Recognition-by-Alignment)就非常贴切的使用了这一思路。

假设验证法主要存在两个问题：一是在场景复杂的情况下，由于生成的假设很多，因而验证计算的代价会很大[8]；另一个是测量数据中的噪声会影响姿态估计最终结果，降低匹配的质量[9]。现今研究人员针对这些困难已作了大量工作，如 Gandhi and Camps 先由一小部分特征得到方位的估计，之后选择待匹配的特征，其目的是为了找到能够使姿态验证中数据不确定性最小的模型点的子集。Jurie 也提出了一种快速、鲁棒的假设验证方法[8]，即先根据高斯误差模型在姿态空间内设置一定范围，并计算变换后的姿态再此范围内特征的可能性估计，然后

选取具有最大概率的物体作为匹配结果的方法，这种配准方法是相当费时的。Kenji 和 Grimson 指出在最差情况下，配准方法计算复杂度是多项式级的[10]，因而为提高配准速度，他们提出了以相关特征组为中心的配准方法。现在这种假设验证法常与其它匹配方法相结合来完成识别工作，例如，王新华曾使用假设验证法与特征关系图相结合来完成识别[11]；Yi 和 Chelberg 则将其与几何哈西表相结合来完成识别[12]。

(2) 解释树(Interpretation tree, IT)法

构成解释树的节点常用于表示场景特征和模型特征见可能的匹配，且树上从根到叶子的每条路径都表示相关问题的一个可能解。这种方法就是为了找到一条路径以便得到传感器数据和模型特征之间的一致相关。但是基上并不需要搜索整个树来得到一致的解释，可根据局部几何约束，例如，特征之间的角度和距离，既可以丢弃和修建一些模型特征和场景特征之间不一致的匹配。当找到一个完整的路径后，就可以计算出物体整体的变换，以确定和验证物体的姿态位置。该算法的整体控制结构是采用顺序假设-测试加回溯的方式，Grimson 很好的研究并公式化了解释树的方法[13]。Ikeuchi 使用解释树搜索的方法将场景中的物体分类成为物体模型存储的一个外观[14]，然后在这些外观组中估计场景物体的姿态；Umasuthan 和 Wallace 又用待树多项式参数的不变量作为 3D 物体表面特征，且在模型库中以 k-d 树的结构来存储这些代数不变量[15]；Grimson 在假设了一些条件后，得到的解释树搜索复杂度为 $O(n^2)$[16]，其中，n 是模型和单物体场景中的场景特征数，但在较差情况下(例如多物体场景中)，复杂度会呈指数上升。

(3) 基于图的识别方法(Graph-based recognition)

特征关系图(Attributed-Relational Graphs, ARG)中的节点对应于场景和模型中的主要特征，而节点之间的弧则表示两个特征之间的关系。这里，场景和模型物体都用 ARG 图来描述，且它们之间的匹配均使用图论的匹配技术(例如图论中的子图同构(isomorphism)、单一同态(monomorphicsm)和同胚(homomorphism)等方法)来实现。

使用相关图的识别非常困难，因为图匹配算法是 NP 完备的，而对特征关系图的扩展则是特征超图表达(Attributed hypergraph representation, AHR)[17],它包含的超边和超节点本身就是 ARG。由于应用 AHR 方法可以使用分层匹配策略，因此复杂度大大降低，如 Kao 等就使用超级关系图(Super relational graph, SRG)来识

别边界表达(Brep)的三维物体特征[18]，且 SRG 是一种无方向图，其图的节点代表一个面，其节点之间由两种连接，分别对应它们定义的两种面与面之间的关系，王新华则定义了 3 种面之间的关系(邻接关系、临接交面，关系和平行关系)，并构造了 3 种关系下的关连图，且分别进行匹配识别[11]。而 Dorai 又将她定义的最大一致形状片(Constant-shape maxium patch, CSMP)组织成一个无向图，并使用分层的图同构方法来识别有深度数据表达的三维物体[19]。这种基于图的识别方法的主要困难在于难以处理物体遮挡问题和图匹配的大运算量问题。

(4) 姿态聚类法(Pose cluster)

该法也称作广义霍夫变换。他是根据局部几何约束(例如角度和距离度量)来减少不正确的匹配，即先从每一个成功的匹配计算出一个基和变换，然后将其存储成霍夫空间(变换参数空间)的一个点，并进行累加[20]。在霍夫空间中最大的聚类点就确定了场景中物体字体的一致假设，如 Grimson 和 Huttenlocher 曾使用“统计占有模型”(Statistical occupancy model)分析了霍夫聚类的敏感性[21]，他们得出的结论是：在霍夫累加器中最大误差与场景中物体的杂乱程度无关，且参数空间中误差极大值出现的概率依赖于传感器的噪声、遮挡和量化效果。

(5) 几何哈西表法(Geometric hashing)

也称索引表(indexing)法，这种方法是将搜索空间设定为由待匹配特征作为索引值组成的查找表，该表中还包含对应物体模型的索引值，这样确定特征相关和数据库搜索过程则转变成表查找的过程。不同几何哈西表法的区别在于表中索引值类型和表的排列方法的不同。Stein 和 Medioni 就使用一种自己定义的溅射(splash)特征和 3D 曲线来分别描述物体的表面和边缘，并对这两种特征进行编码，以作为索引值[22]；Flynn 和 Jain 则根据同时可见的表面片三元组来生成表面-表面的约束关系，并将其作为索引键值[23]。为提高识别效率，人们还引入了统计方法[12,24,25]，如 Wheeler 和 Ikeuchi 从每个物体的多幅深度图像中离线的编译图像和物体特征，然后使用马尔可夫随机场来表示假设之间的相似性，并选择可能性大的假设[25]。Yi 和 Chelberg 则使用局部表面组(LSG, Local surface group)作为索引值，并将 LSG 特征对各个模型的后验概率也存储在索引表中，来综合考虑特征的检测准确性概率和后验概率，然后选择可能性大的物体优先验证[12]。

(6) 迭代模型拟合法

这类方法的一个明显特征是在计算上的迭代过程。当三维物体用参数进行表

达描述，并在该参数既能确定形状，也能确定物体姿态的情况下，可以使用迭代模型拟合的方法。此时，物体识别和姿态估计可简化成估算图像数据中的模型(姿态)参数，且可与存储的模型参数表达相匹配，即通过迭代的方法解一个过约束线性方程或者非线性方程系统，并选择与场景有最小平方和误差的解，来作为估计参数[26]，如 Besl 提出了最临近点匹配算法[27]，其目的是为了找到两幅深度图之间的变换关系，其具体实现过程是通过迭代的方法找到平均误差最小的变换。该法也称作 ICP 算法(Iterative closest point),后来该法广泛地应用到 3D 物体识别中[28,29]。

(7) 基于神经网络的方法

由于神经网络有很好的分类能力和并行处理能力，因此有些研究人员将其运用到 3D 物体识别中。如 Chen 和 Lin 曾使用一个单层 Hopfield 网络来识别 CSG 法表达的 3D 物体[30]，但该系统只能识别表面为平面的物体和单物体；又如 Zha 等同样也使用单层 Hopfield 网络完成识别工作[31]，他们的方法是将物体模型表示成基于表面片的描述(Patch-based description),并定义能量函数，其最小化时的网络节点状态即为识别结果，但与文献[30]不同的是，该方法可以识别场景中的多个物体。Ham 和 Park 也曾使用表面类型、聚特征、表面面积和相邻表面关系来描述 3D 物体[32]，并使用隐马尔可夫模型(Hidden markov model, HMM)来匹配场景和模型物体表面，然后将得到匹配表面的概率值输入到三层 BP 神经网络，以最终识别出物体。由于 HMM 是单独训练每一个物体模型，所以这个系统比较适用于模型库经常更新的情况。与传统的方法相比，基于神经网络的识别方法具有能并行运算、处理速度高和鲁棒性好等优点。

2.5 本章小结

本章围绕着三维物体识别系统必须解决的传感器类型、三维物体表达方法和匹配策略等 3 个方面问题，对近 10 年来的研究成果进行了综述，并对主要方法进行了分类和总结。通过以上分析和综述可见，在过去的 10 年里，人们已经达到了对不同主题的深刻理解，并完成了对过去系统的巨大改进，其中，首先，深度传感器的精度和数据采集速度得到了大幅提高，然而价格还是居高不下；其次，以研究出大量新的知识表达方法和匹配策略，且每种都适合于不同的表面形状和传

感器假设，没有哪一种方法能适应所有的应用，视觉系统知识表达的选择不能独立于现实应用。

第 3 章 对偶四元数法估计三维方位参数

3.1 引言

物体方位参数的估计是计算机视觉的重要组成部分，通常需要计算物体坐标系和基准坐标系之间的一个 4×4 的相似变换矩阵 T,

$$T=\begin{bmatrix} & R & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-1}$$

这里矩阵 R 是3×3的旋转矩阵它表示物体的方向，t 是3×1的平移向量，它代表物体的位置。

综合冗余的感知特征进行物体定位可以提高定位精度，最小二乘法常常用于从这些冗余的特征中找到变换矩阵的最优估计。变换矩阵的最优估计有两种方法，一种方法是先确定物体的最优方向，然后以它为基础确定物体的位置。也就是说，平移向量是最优旋转矩阵 R 和其它量测量的一个函数。这种方法的问题是在计算平移向量时由于来自先前的计算和测量误差，可能存在积累误差。比如，SVD 算法中平移向量 $\mathbf{t}$ 是从 $R,\bar{\mathbf{p}}_i,\mathbf{p}_i$，等等计算出来的，$\mathbf{t}=f\left(R,\bar{\mathbf{p}}_i,\mathbf{p}_i\right)$，这里 $\bar{\mathbf{p}}_i$ 和 $\mathbf{p}_i$ 分别为 3-D 空间的测量点集合和对应的模型点集合。因为 $\bar{\mathbf{p}}_i$ 和 R 有误差，平移向量的误差结果将由于误差传播而被混合。第二种方法就是分别计算两个最优解，一个是方向，另一个是位置，此法并不是很有效。两种方法的通性是都产生方向和位置分开确定的问题。因为变换矩阵本身可以简单的分解成两个部分：一个旋转子矩阵和一个位置向量。两种方法的不同之处在于优化的途径不同：第一种方法只是把相似变换矩阵的旋转部分最优化，然后平移部分是由其推导来的，而第二种方法把旋转部分和平移部分都分别最优化了。

本文采用对偶四元数法提出了一种有效的算法。该方法通过最小化一个与方向和位置误差总和有关的单值函数解算物体的方位。算法需要输入的数据包括物体表面的测量点，从物体上测量的单位方向向量和它们相对应的模型特征。比如点特征包括物体角点（制高点），遮盖点，或者球心。又如单位向量特征包括表面法线，边缘方向向量，轴方向向量，或者曲面法线。

3.2 对偶四元数

这里介绍对偶四元数的定义，性质及其物理解释。说明了位置的对偶四元数表示法和相似变换表示法是如何相互转换的。

3.2.1 对偶数及其性质

对偶数$\hat{a}=a+\varepsilon b$可以定义成由两个普通的实数a和b及服从特殊乘法规律$\varepsilon^2=0$的ε构成。两个实数a和b分别叫做对偶数的实部和对偶部分。

对偶数的加法、减法、乘法可以定义成这种形式

$$\begin{aligned}&(a+\varepsilon b)+(c+\varepsilon d)=(a+c)+\varepsilon(b+d)\\&(a+\varepsilon b)-(c+\varepsilon d)=(a-c)+\varepsilon(b-d)\\&(a+\varepsilon b)(c+\varepsilon d)=ac+\varepsilon(ad+bc)\end{aligned}\tag{3-2}$$

上世纪初,德国几何学家 E.Study (1862-1930)首次考虑了对偶数[45]。在他的研究中他用对偶数表示对偶角，这个角可以测量空间两条斜线的相对位置。对偶角定义为

$$\hat{\theta}=\theta+\varepsilon d\tag{3-3}$$

这里d是两条三维空间线的距离，θ使它们方向间的夹角。

对偶数一些重要性质如下

1. 对偶数$\hat{a}$与它的共轭数$\bar{\hat{a}}=a-\varepsilon b$的乘积是

$$\hat{a}\bar{\hat{a}}=a^2\tag{3-4}$$

2. 对偶数的模是 $$|\hat{a}|=a\tag{3-5}$$

它可以是负的。

3. 由于$\varepsilon^2=0$。对偶数函数有一个很简单的泰勒级数展开形式

$$f(a+\varepsilon b)=f(a)+\varepsilon bf'(a)\tag{3-6}$$

4. 对于对偶角，有$\hat{\theta}$

$$\sin(\hat{\theta})=\sin(\theta+\varepsilon d)=\sin(\theta)+\varepsilon d\cos(\theta)\tag{3-7}$$

$$\cos(\hat{\theta})=\cos(\theta+\varepsilon d)=\cos(\theta)-\varepsilon d\sin(\theta)\tag{3-8}$$

在上面的性质中，性质 1 和 2 可以由两个对偶数的乘法定义直接推导出来。性质 3 我们只需要知道对偶数函数$f(a+\varepsilon b)$像复数域里的通常函数一样可以展

开成一个正规的泰勒级数，只要函数 $f(\hat{a})$ 在圆 $|\hat{a}-c|<R(R>0)$ 内是可分解的，其中 c 是圆心，那么 $f(\hat{a})$ 就可以在这个圆内展开成泰勒级数，

$$f(\hat{a})=\sum_{n=0}^{\infty}u_n(\hat{a}-c)^n \tag{3-9}$$

这里 $u_n=f^{(n)}(c)/n!$ 级数是唯一的。如果我们在点 a 展开 $f(\hat{a})$，泰勒级数将有这种形式

$$f(a+\varepsilon b)=f(a)+\varepsilon bf'(a)+\varepsilon^2\frac{b^2}{2}f''(a)+\cdots \tag{3-10}$$

因为 $\varepsilon^2=0$，所有大于 ε 平方的项在泰勒级数里都将是零。最后一个性质我们只需将 $\sin(\hat{\theta})$ 和 $\cos(\hat{\theta})$ 代入对应的泰勒级数就可得到。

对偶特性的思想可以延伸到定义对偶向量、对偶四元数和对偶矩阵。这些对偶特性可以使两个不同的性质在许多方面合并成一个性质。例如，对偶向量可以定义成 3-D 空间任意直线。直线的方向和位置可以表示成

$$\hat{\mathbf{n}}=\mathbf{n}+\varepsilon\mathbf{p}\times\mathbf{n} \tag{3-11}$$

这里 $\mathbf{n}$ 是直线的单位方向向量，$\mathbf{p}$ 是直线上任一点的位置向量。另一个例子是，对偶四元数可以定义成表示两坐标系之间的任意变换，这将在方程(3-18)中给出。

3.2.2 对偶四元数及其性质

四元数是四个元素的向量，它可以理解成由一个 3×1 的向量部分和一个标量部分组成。例如，四元数 $\boldsymbol{q}$ 是

$$\boldsymbol{q}=\begin{bmatrix}q_1\\q_2\\q_3\\q_4\end{bmatrix}=\begin{bmatrix}\mathbf{q}\\q_4\end{bmatrix} \tag{3-12}$$

在本文的符号中，四元数都是由黑斜体字母表示，像 $\boldsymbol{q}$；而 3×1 的向量由黑体罗马字母表示，像 $\mathbf{q}$。

四元数的组成还可有表示为

$$\boldsymbol{q}=\begin{bmatrix}\sin(\theta/2)\mathbf{n}\\\cos(\theta/2)\end{bmatrix} \tag{3-13}$$

如图 3.1 所示，向量 **n** 是坐标系绕其旋转的单位向量，θ 是绕 **n** 旋转的角度。对应的旋转矩阵可以表示成

$$\boldsymbol{R}=\left(q_4^2-\mathbf{q}^T\mathbf{q}\right)\boldsymbol{I}+2\mathbf{q}\mathbf{q}^T+2q_4\boldsymbol{K}(\mathbf{q}) \tag{3-14}$$

这里 $\boldsymbol{K}$ 是反对称矩阵

$$\boldsymbol{K}(\mathbf{q})=\begin{bmatrix}0 & -q_3 & q_2\\ q_3 & 0 & -q_1\\ -q_2 & q_1 & 0\end{bmatrix} \tag{3-15}$$

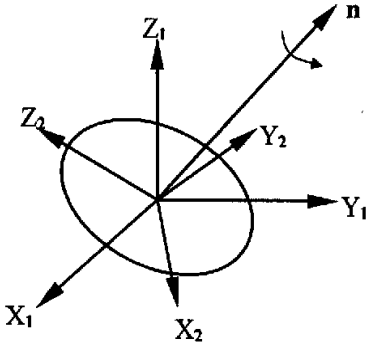


图 3.1 四元数表示的旋转

把方程扩展成包括位置和方向的表达形式只需将方程中的全部四元数变成对偶四元数[34,46]：

$$\hat{\boldsymbol{q}}=\begin{bmatrix}\hat{q}_1\\ \hat{q}_2\\ \hat{q}_3\\ \hat{q}_4\end{bmatrix}=\begin{bmatrix}\hat{\mathbf{q}}\\ \hat{q}_4\end{bmatrix} \tag{3-16}$$

对偶四元数由两部分组成，

$$\hat{\boldsymbol{q}}=\boldsymbol{r}+\varepsilon\boldsymbol{s} \tag{3-17}$$

这里 $\boldsymbol{r}$ 和 $\boldsymbol{s}$ 都是纯四元数分别称作实部和对偶部分。

对偶四元数和纯四元数具有一个相似的表示，

$$\hat{\boldsymbol{q}} = \begin{bmatrix} \sin\left(\hat{\theta}/2\right)\hat{\mathbf{n}} \\ \cos\left(\hat{\theta}/2\right) \end{bmatrix} \tag{3-18}$$

这里对偶向量$\hat{\mathbf{n}}$表示坐标系绕其旋转和平移的 3-D 空间曲线，$\hat{\theta}$是旋转和平移的对偶角。

对偶向量$\hat{\mathbf{n}}$和对偶角$\hat{\theta}$是

$$\hat{\mathbf{n}} = \mathbf{n} + \varepsilon \mathbf{p} \times \mathbf{n} \tag{3-19}$$

$$\hat{\theta} = \theta + \varepsilon d, \tag{3-20}$$

这里 **n** 是单位向量说明了旋转轴的方向也说明了平移的方向；旋转是关于具有方向 **n** 并过点 **p** 的线旋转了角θ；d是沿着 **n** 说明的方向平移的距离，传统的坐标系变换是由平移向量 **t**,旋转轴 **n** 和旋转角θ来说明的。首先平移原始坐标系 **t** 然后关于 **n** 旋转角θ形成新的坐标系。当然，平移和旋转是可以颠倒的。用对偶四元数表示法可以形成相同的变换，首先将原始坐标系沿着 **n** 的方向平移了距离 d 然后关于具有单位向量 **n** 的直线旋转了角θ，**n** 有方向并过点 **p**。如图 3.2 所示。

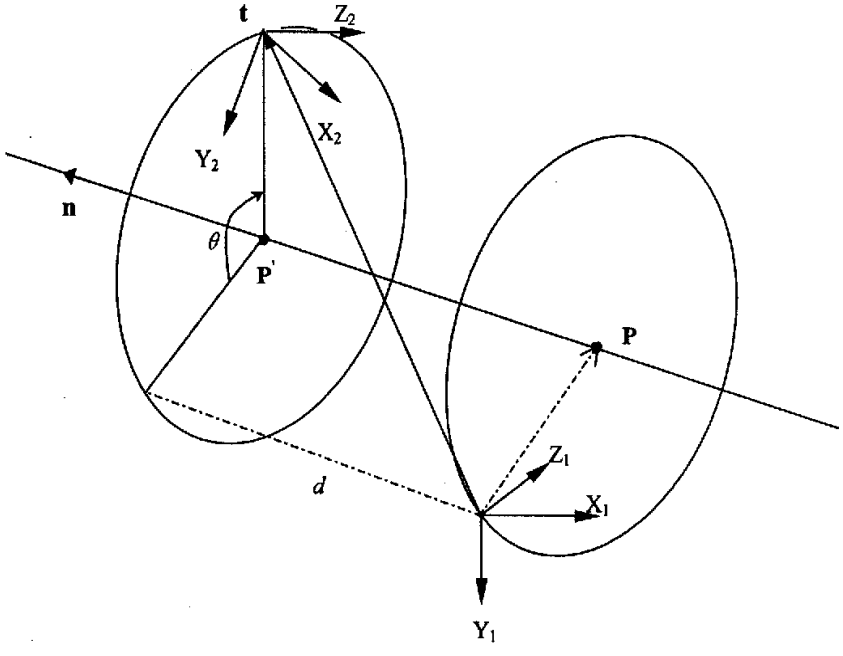


图 3.2 对偶四元数表示的旋转和平移

把方程(3-19)和(3-20)代入 (3-18), 化简得

$$\boldsymbol{r}=\begin{bmatrix}\sin(\theta/2)\mathbf{n}\\ \cos(\theta/2)\end{bmatrix} \tag{3-21}$$

$$\boldsymbol{s}=\begin{bmatrix}(d/2)\cos(\theta/2)\mathbf{n}+\sin(\theta/2)(\mathbf{p}\times\mathbf{n})\\ -(d/2)\sin(\theta/2)\end{bmatrix} \tag{3-22}$$

对偶四元数有八个元素，而表示 3-D 物体变换的独立变量的最少数目是六个，这就意味着对偶四元数表示法中的八个元素有两个不是独立的。实际上任何对偶四元数的组成都可以从方程(3-21)和(3-22)看出来。如果它们由方程(3-18)-(3-20)定义，则满足下列两个限制条件：

$$\boldsymbol{r}^T\boldsymbol{r}=1 \tag{3-23}$$

$$\boldsymbol{r}^T\boldsymbol{s}=0 \tag{3-24}$$

两个重要的四元数矩阵函数 $\boldsymbol{Q}(r)$和 $\boldsymbol{W}(r)$定义为

$$\boldsymbol{Q}(\boldsymbol{r})=\begin{bmatrix}r_4\boldsymbol{I}+\boldsymbol{K}(\mathbf{r}) & \mathbf{r}\\ -\mathbf{r}^T & r_4\end{bmatrix} \tag{3-25}$$

$$\boldsymbol{W}(\boldsymbol{r})=\begin{bmatrix}r_4\boldsymbol{I}-\boldsymbol{K}(\mathbf{r}) & \mathbf{r}\\ -\mathbf{r}^T & r_4\end{bmatrix} \tag{3-26}$$

这里 $\boldsymbol{K}(\mathbf{r})$是反对称矩阵如方程(3-15)中定义。

3.2.3 对偶四元数与相似变换间的转换

3.2.3.1 给定对偶四元数计算相似变换

方程(3-21)表明对偶四元数的实部 $\boldsymbol{r}$ 与方程(3-13)中定义的具有相同的形式。于是，旋转矩阵 $\boldsymbol{R}$ 就可以用对偶四元数的形式写出

$$\boldsymbol{R}=\left(r_4^2-\mathbf{r}^T\mathbf{r}\right)\boldsymbol{I}+2\mathbf{r}\mathbf{r}^T+2r_4\boldsymbol{K}(\mathbf{r}) \tag{3-27}$$

或者

$$\begin{bmatrix}\boldsymbol{R} & \mathbf{0}\\ \mathbf{0}^T & 1\end{bmatrix}=\boldsymbol{W}(\boldsymbol{r})^T\boldsymbol{Q}(\boldsymbol{r}) \tag{3-28}$$

位置向量以对偶四元数的形式写出

$$t = W(r)^T s \tag{3-29}$$

这里t是平移向量$\mathbf{t}$的平移四元数定义为

$$t = \frac{1}{2}\begin{bmatrix}\mathbf{t}\\0\end{bmatrix} \tag{3-30}$$

3.2.3.2 给定相似变换计算对偶四元数

给出一个由旋转矩阵R和平移向量$\mathbf{t}$组成的相似变换T, 可以计算对应的r和s

$$r_4 = (1/2)\sqrt{R_{11}+R_{22}+R_{33}+1} \tag{3-31}$$

这里R_{ij}表示矩阵R的第ij元素。r_4的值等于零表示旋转了 180 度。如果r_4的值不是零，那么

$$\mathbf{r} = \frac{1}{4r_4}\begin{bmatrix}R_{32}-R_{23}\\R_{13}-R_{31}\\R_{21}-R_{12}\end{bmatrix} \tag{3-32}$$

如果r_4是零，那么

$$\mathbf{r}\mathbf{r}^T = (1/2)(R+I) \tag{3-33}$$

所以r可以由任意非零的$(1/2)(R+I)$的值确定，令它为$\mathbf{a}$。因此

$$\mathbf{r} = \pm\frac{\mathbf{a}}{\|\mathbf{a}\|} \tag{3-34}$$

旋转角是 180 度正负号都可行。

r的值确定了，s的值就可以由方程(3-29)计算出来，

$$s = W(r)t \tag{3-35}$$

因此相似变换矩阵与相应的对偶四元数之间可以相互转换。

3.2.3.3 由给定的对偶四元数求取变换的元组$(\mathbf{n},\theta,d,\mathbf{p})$

给定对偶四元数，旋转子矩阵就可以按照前面的描述推导出来，旋转轴$\mathbf{n}$和旋转角θ也可以从矩阵中提取出来。平移距离d可以从方程(3-22)推导出来

$$d=\frac{2s_4}{\sin(\theta/2)} \tag{3-36}$$

方程(3-22)也可以用来推导 $\mathbf{p}$。由方程(3-22)得

$$\mathbf{s}=(d/2)\cos(\theta/2)\mathbf{n}+\sin(\theta/2)(\mathbf{p}\times\mathbf{n}) \tag{3-37}$$

可以写成

$$\mathbf{p}\times\mathbf{n}=\frac{\mathbf{s}-(d/2)\cos(\theta/2)\mathbf{n}}{\sin(\theta/2)} \tag{3-38}$$

因为 $\mathbf{p}\times\mathbf{n}=\boldsymbol{K}(-\mathbf{n})\mathbf{p}$，于是

$$\boldsymbol{K}(-\mathbf{n})\mathbf{p}=\frac{\mathbf{s}-(d/2)\cos(\theta/2)\mathbf{n}}{\sin(\theta/2)} \tag{3-39}$$

这里只有 $\mathbf{p}$ 未知。

矩阵 $\boldsymbol{K}(-\mathbf{n})$ 的秩是 2，因此矩阵零空间的维数是 1。方程(3-39)通解的形式是

$$\mathbf{p}=\mathbf{p}_0+\alpha\mathbf{n} \tag{3-40}$$

这里 $\mathbf{p}_0$ 是零空间里的任意向量。

也就是说，$\mathbf{p}$ 的解不是唯一的。通常选择一个垂直于 $\mathbf{n}$ 的向量作为期望解。

一个可能的解是

$$\mathbf{p}=\frac{\alpha}{n_1}\begin{bmatrix}0\\-n_3\\n_2\end{bmatrix} \tag{3-41}$$

这里 $n_1\neq 0$。

3.3 问题的提出和解决方案

3.3.1 问题的提出

正如已经提到的，要考虑传感器测量的两种类型：物体上点的位置和物体的单位向量如单位法线，边缘方向向量等。通过分析我们定义四元数表示这些特性。用 $\mathbf{p}$ 表示物体表面点的位置向量。定义位置四元数如下

$$p=\frac{1}{2}\begin{bmatrix}\mathbf{p}\\0\end{bmatrix} \tag{3-42}$$

$\mathbf{n}$ 表示从物体提取的单位向量。定义方向四元数为

$$n=\begin{bmatrix}\mathbf{n}\\0\end{bmatrix} \tag{3-43}$$

为了确定物体的位置和方向，我们测量与模型对应的 k 个单位向量，把它们储存在方向四元数 $\overline{n}_i$ 中。上面的符号表示测量值。同样地我们测量物体上的 l 个点把它们储存在位置四元数 $\overline{p}_i$ 中。

对应每一个测量点 $\overline{p}_i$，有一个关于物体坐标系描述的点 $\mathbf{p}_i^0$ 的数据库。如果 $\mathbf{t}$ 和 $\boldsymbol{R}$ 是待确定的变换矩阵的平移和旋转部分，模型点将变换到位置 $\mathbf{p}_i$，

$$\mathbf{p}_i=\mathbf{t}+\boldsymbol{R}\mathbf{p}_i^0 \tag{3-44}$$

如果这些由位置四元数 p_i^0 和 p_i 和一个对偶四元数表示的模型点用于表示变换参数，则由方程(3-28),(3-29),(3-44)得

$$p_i=W(r)^T s+W(r)^T Q(r)p_i^0 \tag{3-45}$$

相同的道理，对模型的方向四元数 n_i 和 n_i^0，有

$$n_i=W(r)^T Q(r)n_i^0 \tag{3-46}$$

计算物体位置和方向的方法就是要确定 r 和 s 使 $\overline{p}_i$ 和 p_i 及 $\overline{n}_i$ 和 n_i 之间的误差最小化。也就是说，选择 r 和 s 来最小化误差函数

$$E=\sum_{i=1}^{k}\alpha_i(n_i-\overline{n}_i)^2+\sum_{i=1}^{l}\beta_i(p_i-\overline{p}_i)^2 \tag{3-47}$$

这里 α_i 和 β_i 是恒正的加权系数。

分别考虑每一个量：

$$(n_i-\overline{n}_i)^2=2\left(1-r^T Q(\overline{n}_i)^T W(n_i^0)r\right) \tag{3-48}$$ (3-49)

$$(p_i-\overline{p}_i)^2=s^T s+2s^T\left(W(p_i^0)-Q(\overline{p}_i)\right)r-2r^T Q(\overline{p}_i)^T W(p_i^0)r+\left((p_i^0)^T p_i^0+\overline{p}_i^T\overline{p}_i\right)$$

这样，误差函数可以写成 r 和 s 的二次函数，

$$E=r^T\mathbf{C}_1 r+s^T\mathbf{C}_2 s+s^T\mathbf{C}_3 r+\text{constant} \tag{3-50}$$

这里

$$\mathbf{C}_1=-2\sum_{i=1}^{k}\alpha_i Q(\overline{n}_i)^T W(n_i^0)-2\sum_{i=1}^{l}\beta_i Q(\overline{p}_i)^T W(p_i^0) \tag{3-51}$$

$$\mathbf{C}_2=\left(\sum_{i=1}^{l}\beta_i\right)I \tag{3-52}$$

$$\mathbf{C}_3 = 2\sum_{i=1}^{l}\beta_i\left(W\left(\boldsymbol{p}_i^0\right)-Q(\overline{\boldsymbol{p}}_i)\right) \tag{3-53}$$

$$\text{constant} = 2\sum_{i=1}^{k}\alpha_i + \sum_{i=1}^{l}\beta_i\left(\left(\boldsymbol{p}_i^0\right)^T\boldsymbol{p}_i^0 + \overline{\boldsymbol{p}}_i^T\overline{\boldsymbol{p}}_i\right) \tag{3-54}$$

计算$\boldsymbol{r}$和$\boldsymbol{s}$最小化这个误差函数服从限制条件

$$\boldsymbol{r}^T\boldsymbol{r} = 1 \tag{3-55}$$

$$\boldsymbol{s}^T\boldsymbol{r} = 0 \tag{3-56}$$

3.3.2 问题的解决方案

最优的对偶数定位四元数是通过附加约束方程到误差方程中，然后最小化不代约束的结果函数而得到的

$$\overline{E} = \boldsymbol{r}^T\mathbf{C}_1\boldsymbol{r} + \boldsymbol{s}^T\mathbf{C}_2\boldsymbol{s} + \boldsymbol{s}^T\mathbf{C}_3\boldsymbol{r} + \text{constant} + \lambda_1\left(\boldsymbol{r}^T\boldsymbol{r}-1\right) + \lambda_2\left(\boldsymbol{s}^T\boldsymbol{r}\right)$$

这里λ_1和λ_2是拉格朗日乘数。取偏导数有

$$\frac{\partial\overline{E}}{\partial\boldsymbol{r}} = \left(\mathbf{C}_1+\mathbf{C}_1^T\right)\boldsymbol{r} + \mathbf{C}_3^T\boldsymbol{s} + 2\lambda_1\boldsymbol{r} + \lambda_2\boldsymbol{s} = 0 \tag{3-57}$$

$$\frac{\partial\overline{E}}{\partial\boldsymbol{r}} = \left(\mathbf{C}_2+\mathbf{C}_2^T\right)\boldsymbol{s} + \mathbf{C}_3\boldsymbol{r} + \lambda_2\boldsymbol{r} = 0 \tag{3-58}$$

于是，解方程(3-55),(3-56),(3-57)和(3-58)求$\boldsymbol{r}$和$\boldsymbol{s}$给出物体方位的最优解。

先求解λ_2，用$\boldsymbol{r}$乘以方程(3-58有

$$\lambda_2 = -\boldsymbol{r}^T\mathbf{C}_3\boldsymbol{r} \tag{3-59}$$

因为$\mathbf{C}_3$是反对称阵，　　$\lambda_2 = 0$ (3-60)

现在可以把$\boldsymbol{s}$作为$\boldsymbol{r}$的一个函数有方程(3-58)得

$$\boldsymbol{s} = -\left(\mathbf{C}_2+\mathbf{C}_2^T\right)^{-1}\mathbf{C}_3\boldsymbol{r} \tag{3-61}$$

将方程(3-60)和(3-61)代入方程(3-57)有

$$A\boldsymbol{r} = \lambda_1\boldsymbol{r} \tag{3-62}$$

其中

$$A = \frac{1}{2}\left(\mathbf{C}_3^T\left(\mathbf{C}_2+\mathbf{C}_2^T\right)^{-1}\mathbf{C}_3 - \mathbf{C}_1 - \mathbf{C}_1^T\right) \tag{3-63}$$

因而，四元数$\boldsymbol{r}$是矩阵A的一个特征向量，λ_1是对应的特征值。通常方程有四个

解。因为 A 是实对称阵，则它的所有特征值和特征向量都是实数并且不同特征值对应的特征向量正交。期望值的确定还要参考误差方程(3-50).

用 $\boldsymbol{r}^T$ 乘以方程(3-57)有

$$\boldsymbol{r}^T\mathbf{C}_1\boldsymbol{r}=(1/2)\boldsymbol{r}^T\left(\mathbf{C}_1+\mathbf{C}_1^T\right)\boldsymbol{r}=-(1/2)\boldsymbol{s}^T\mathbf{C}_3\boldsymbol{r}-\lambda_1 \tag{3-64}$$

用 $\boldsymbol{s}^T$ 乘以方程(3-58)有

$$\boldsymbol{s}^T\mathbf{C}_2\boldsymbol{s}=(1/2)\boldsymbol{s}^T\left(\mathbf{C}_2+\mathbf{C}_2^T\right)\boldsymbol{s}=-(1/2)\boldsymbol{s}^T\mathbf{C}_3\boldsymbol{r} \tag{3-65}$$

代入方程(3-50)得

$$E=\text{constant}-\lambda_1 \tag{3-66}$$

因而，选择最大正特征值对应的特征向量误差最小。

求出了 $\boldsymbol{r}$ 后再代回方程(3-61)得到 $\boldsymbol{s}$ 从而完成物体方位的求解。

为了给出以上推导过程更清晰的描述，下面将总结最优对偶四元数定位算法(DQ 算法)。

从算法中可以看出第 2-4 步的执行时间基本恒定，而第一步的执行时间与测量向量的数量之间是一个线性关系。因此，算法在时间复杂度上是 $O(n)$。

DQ 方位确定算法

输入： k 个测量点 $\overline{\boldsymbol{p}}_i^0$ l 个测量单位向量 $\overline{\boldsymbol{n}}_i^0$ 的集合；对应的模型点 $\boldsymbol{p}_i$ 和向量 $\boldsymbol{n}_i$，也要试探地选择加权系数 α_i 和 β_i 反映数据点的可靠性。

输出： 估计变换矩阵 $\boldsymbol{T}$。

步骤 1. 计算矩阵 $\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3$：

$$\mathbf{C}_1=-2\sum_{i=1}^{k}\alpha_i\boldsymbol{Q}(\overline{\boldsymbol{n}}_i)^T\boldsymbol{W}\left(\boldsymbol{n}_i^0\right)-2\sum_{i=1}^{l}\beta_i\boldsymbol{Q}(\overline{\boldsymbol{p}}_i)^T\boldsymbol{W}\left(\boldsymbol{p}_i^0\right)$$

$$\mathbf{C}_2=\left(\sum_{i=1}^{l}\beta_i\right)\boldsymbol{I}$$

$$\mathbf{C}_3=2\sum_{i=1}^{l}\beta_i\left(\boldsymbol{W}\left(\boldsymbol{p}_i^0\right)-\boldsymbol{Q}(\overline{\boldsymbol{p}}_i)\right)$$

步骤 2. 计算 4×4 对称矩阵 A：

$$A=\frac{1}{2}\left(\mathbf{C}_3^T\left(\mathbf{C}_2+\mathbf{C}_2^T\right)^{-1}\mathbf{C}_3-\mathbf{C}_1-\mathbf{C}_1^T\right)$$

步骤 3. 计算矩阵 A 的最大正特征值对应的特征向量 $\boldsymbol{r}$，由 $\boldsymbol{r}$ 推导出 $\boldsymbol{s}$。

步骤 4. 由 $\boldsymbol{s}$ 和 $\boldsymbol{r}$ 计算矩阵 T。

3.4 仿真结果

为了检验 DQ 算法的性能、精度和速度，我们在计算机上用 Matlab 进行仿真。仿真数据取自模型“桌子”（见图 3.3），它有 28 个顶点，42 个表面和 40 条边。桌子的尺寸是302×116×131单位。选择 SVD 算法作为样本算法与我们的 DQ 算法比较精度和性能。

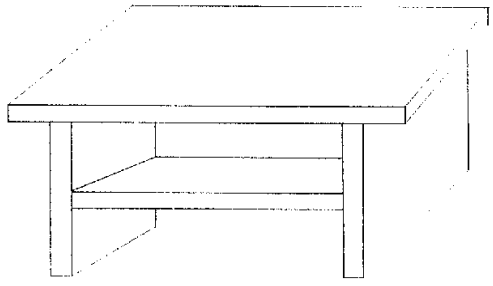


图 3.3 物体“桌子”

由于 SVD 算法只接收 3-D 点作为它的输入，为了比较精度，两个算法只输入样本点。算法选用 5 个、10 个、20 个和 30 个点作为输入数据进行测试。每次测试对点重复 25 次不同的选择，例如，对 5 个、10 个、20 个和 30 个点，每组都运行 25 个不同的集合，每个集合都运行 20 次，每次都将随机误差加入样本值（见下面的讨论）。在我们的仿真试验中，所需 3-D 点 $\boldsymbol{p}_i^0$ 的数量是在每次实验开始时从桌子的顶点中随机选择的。然后，首先绕着过原点方向向量为(3.0,4.0,6.0)的轴旋转36°角，再平移(7,8,13)，最后将每个变换后的点的坐标加入均值为 0 标准差为 0.5 的高斯随机噪声，则产生了相应的测量点 $\bar{\boldsymbol{p}}_i$。这些测量点和模型点用来计算估计方向和平移参数。为了简化仿真过程，所有的加权系数 α_i 和 β_i 都设为 1。变换后的方向和平移参数的标准偏差是由这 20 次实验计算出来的。表 3.3 列出了仿真结果。所有算法都是由 Matlab 程序语言编写。由于 Matlab 具有强大的数学计算功能，因此编写程序调用 Matlab 精确计算的函数包，用来执行所有矩阵的计算还有 SVD（奇异值分解）计算和特征值计算，可以得到准确的仿真结果。

表 3.3 变换参数标准偏差的比较

相关点的数量	使用方法							
	SVD 算法				对偶四元数法			
	x	y	z	θ	x	y	z	θ
5	1.434	3.013	1.190	0.147	0.461	0.277	0.509	0.147
10	1.133	2.373	0.843	0.046	0.133	0.215	0.169	0.046
20	0.296	0.607	0.254	0.040	0.102	0.187	0.108	0.040
30	0.171	0.246	0.125	0.037	0.115	0.115	0.087	0.037

从表 3.3 可以看出无论仿真过程中使用多少个点两种算法均产生相同的旋转误差，这是预料之中的。而对平移误差，在所有情况下 DQ 算法与 SVD 算法相比都展示了更好的性能。甚至在 30 个点时应该提供一个很好的估计的情况下，DQ 算法与 SVD 算法相比，在平移参数的计算上平均精度提高了 20%。

3.5 本章小结

本章提出用对偶四元数来解决物体方位问题。初步的仿真试验结果可以看出，定位问题可以转化为构造最优化问题来求解。与过去的方法相比，以前的方法直接得到位置向量 $\mathbf{p}$ 而不是得到对偶四元数中的位置组成 $\boldsymbol{s}$。为了解出 $\boldsymbol{s}$ 而不是 $\mathbf{p}$ 我们需要构造一个二次函数，用标准的特征值-特征向量的数学包简单的求解。

算法计算结果有效而且很准确。所需要的量测量没有增加。输入到算法中的应该是特征点和特征单位向量。

第 4 章 基于迭代算法的三维物体匹配

4.1 引言

本文采用一种基于几何学原理的匹配方法。在这种匹配方法中，物体与模型之间的匹配是通过迭代最近点(Iterative Closest Point)实现的。物体表面由深度图(range image)描述，而模型数据可以用三维点集、三角形集或其它几何原型描述。算法用迭代的方法直接针对 3D 数据实现物体与模型的匹配。在每一次迭代运算中，首先获得物体数据在模型上对应的最近点，然后用最小方差估计的方法计算物体与模型间的 6 自由度空间刚体变换。

图 4.1 和图 4.2 给出了 ICP 算法的基本步骤。在图 4.1 中，上面的点为进行第 k 步变换以前的数据形态点集 $\mathbf{P}_k$，$\mathbf{P}_k$ 中的每一个点在模型（下面的点集）中寻找最近匹配点，并寻找使得目标函数 $d_k = \frac{1}{N_p}\sum_{i=1}^{N_p}\left\|\mathbf{y}_{i,k} - \boldsymbol{R}\mathbf{p}_{i,k} - \mathbf{t}_{i,k}\right\|^2$ 最小的最优变换，将最优变换应用到点集 $\mathbf{P}_k$ 上，得到了变换后的点集 $\mathbf{P}_{k+1}$，如图 4.2 所示，然后 $\mathbf{P}_{k+1}$ 再在模型形态点集中寻找最近匹配点，进行下一循环。

ICP 算法的匹配结果收敛于局部最优解，通过设定不同的迭代旋转初始值（如果大部分测量数据覆盖于模型数据上）和平移初始值（如果仅有少量测量数据覆盖于模型数据上）可以获得全局最优解。ICP 算法直接对用深度图建立的物体表面的三维数据坐标进行处理，而不需要对物体的特征进行假设与分割，这使得此方法对于不同形式物体的识别结果表现出很高的可靠性，被广泛的运用于三维物体识别中。

将 ICP 算法应用到 3D 物体识别中必须考虑到几个关键问题，分别是：初值和全局最优化、匹配点的选择、对应关系的确定、刚性变换、迭代终止条件、鲁棒性、快速收敛性和精度评估等，它们之间是相互影响的。其中前四个是 ICP 算法的主要步骤，但是它们的执行都影响后三个关键技术。本文虽然采用 ICP 算法的思想，但匹配过程中每一个步骤的具体实现方法都有不同的考虑，并且有所改进，包括伪点匹配、最大距离容限、对偶四元数法计算变换矩阵等，都不同程度

的提高了匹配性能。本章将抛开通常意义上的 ICP 算法的基本步骤，详细描述本文所采用的迭代匹配算法的完整工作过程。

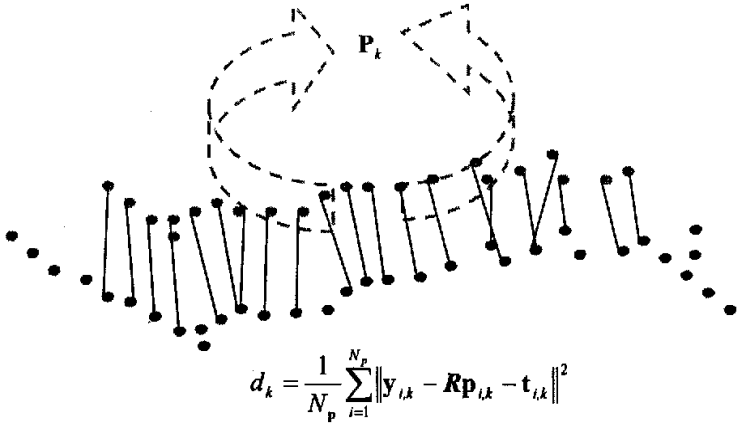


图 4.1 第 k 步更新前的情况

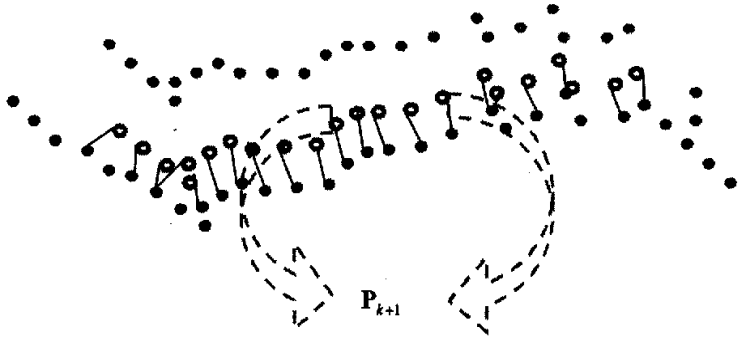


图 4.2 第 k 步更新后的情况

4.2 最近点迭代匹配算法

设定物体图像与模型图像之间的距离很近，在物体图像中的曲线与模型图像中的对应曲线很近。通过将物体图像上的点与在模型图像中曲线上与之最近的点相匹配，我们可以在两图像中找到将曲线变得更近的运动（也就是说，使两条曲线间距离变得更小），反复应用这个过程，算法就可以给出越来越好的运动估计。

一个三维（空间）曲线段C是一个向量函数$\mathbf{x}:[a,b]\to\mathbf{R}^3$，这里$a$和$b$是标量。在计算机视觉应用中，空间曲线的数据可以用3-D点集的形式表示，这些点要么来自立体算法[50]，要么来自空间图像传感器[51]。如果知道曲线的形式，则可以通过拟合的办法获得曲线的描述$\mathbf{x}$，称为二次曲线点数据。在此，我们直接使用序列点，也就是说我们仅关心任意形状的空间曲线，而不考虑特定的曲线元素。对曲线来说使用序列点等同于分段线性近似。设$\mathbf{x}_{i,j}(j=1,\ldots,N_i)$是曲线$\boldsymbol{C}_i$上的$N_i$序列点，通过增大$N_i$和减小距离$\|\mathbf{x}_{i,j}-\mathbf{x}_{i,j+1}\|$可以使近似误差任意小。在所有的点$\mathbf{x}_{i,j}$处，计算正切方向$\mathbf{u}_{i,j}$，它将用于匹配过程。算法中不必知道精确的正切方向，使用简单的估计：

$$\mathbf{u}_{i,j}=\left(\mathbf{x}_{i,j+1}-\mathbf{x}_{i,j-1}\right)\Big/\left\|\mathbf{x}_{i,j+1}-\mathbf{x}_{i,j-1}\right\|$$

起始点和终止点的切线方向为：

$$\mathbf{u}_{i,1}=\left(\mathbf{x}_{i,2}-\mathbf{x}_{i,1}\right)\Big/\left\|\mathbf{x}_{i,2}-\mathbf{x}_{i,1}\right\|$$
$$\mathbf{u}_{i,N_i}=\left(\mathbf{x}_{i,N_i}-\mathbf{x}_{i,N_i-1}\right)\Big/\left\|\mathbf{x}_{i,N_i}-\mathbf{x}_{i,N_i-1}\right\|$$

给定物体与模型的两幅3-D图像，两图像都包含一系列的曲线。设$\boldsymbol{C}_i(i=1,\ldots,m)$和$\boldsymbol{C}'_k(k=1,\ldots,n)$分别是两图像内的曲线，再设$\mathbf{x}_{i,j}(j=1,\ldots,N_1)$和$\mathbf{x}'_{k,l}(l=1,\ldots,N_k)$分别为曲线$\boldsymbol{C}_i$和$\boldsymbol{C}'_j$上的点。目的是为了找到两图像之间的运动。即$\boldsymbol{R}$（旋转）和$\mathbf{t}$（平移）使下列判据最小。

$$F(\boldsymbol{R},\mathbf{t})=\sum_{i=1}^{m}\sum_{j=1}^{N_i}p_{i,j}d^2(\boldsymbol{R}\mathbf{x}_{i,j}+\mathbf{t},\boldsymbol{C}'_k)+\sum_{k=1}^{n}\sum_{l=1}^{N_k}q_{k,l}d^2(\boldsymbol{R}^T\mathbf{x}_{k,l}-\boldsymbol{R}^T\mathbf{t},\boldsymbol{C}_i)\quad(4\text{-}1)$$

其中$d(\mathbf{x},\boldsymbol{C})$表示点$\mathbf{x}$到曲线$\boldsymbol{C}$的距离（将在下面定义），如果点$\mathbf{x}_{i,j}$（对应的$\mathbf{x}'_{k,l}$）能匹配到模型中的曲线$\boldsymbol{C}'_k$上的点（对应的物体中的曲线$\boldsymbol{C}_i$），那么

$p_{i,j}$(对应的$q_{k,l}$)取1，否则取0。当然，等式（3-1）最小的同时，下式必然最大

$$\sum_{i=1}^{m}\sum_{j=1}^{N_i} p_{i,j} + \sum_{k=1}^{n}\sum_{l=1}^{N_k} q_{k,l}$$

如果不是，只有当所有的i,j,k,l都满足$p_{i,j}=q_{k,l}=0$时，才能获得方程（4-1）的通解。

上面的判据是对称的，某种程度上两个图像中的任何一个都不优于另外一个。为简化计算，用方程（4-1）右端的第一部分加上$\sum_{i=1}^{m}\sum_{j=1}^{N_i} p_{i,j}$的最大化。换句话说，就是下面的目标函数最小，

$$F(\boldsymbol{R},\mathbf{t}) = \frac{1}{\sum_{i=1}^{m}\sum_{j=1}^{N_i} p_{i,j}} \sum_{i=1}^{m}\sum_{j=1}^{N_i} p_{i,j}\, d^2(\boldsymbol{R}\mathbf{x}_{i,j} + \mathbf{t}, \boldsymbol{C}_k') \tag{4-2}$$

这个修改只对最终运动估计的精度产生微小的影响。虽然从迭代次数的意义上讲使算法的收敛性稍有变差，但却加快了算法的运算速度。

另外，我们假设两图像之间的运动很小或者近似知道。在后一种情况下，我们可以首先用两图像之间的运动近似估计物体图像产生中间图像，然后认为中间图像和模型图像之间的运动很小。

下面描述一种迭代算法，该算法通过匹配物体数据中的点实现曲线对准。最小二乘估计减少了两图像间的平均距离。由于物体中的点及其在模型中的最近点并不一定对应空间中的同一个点，所以这种迭代要进行很多次。

4.2.1 寻找最近点

如果$\boldsymbol{C}_k'$是参数曲线$\left(\mathbf{x}_k':[a,b]\to\mathbf{R}^3\right)$，定义点$\mathbf{x}$和曲线$\boldsymbol{C}_k'$之间的距离$d(\mathbf{x},\boldsymbol{C}_k')$为

$$d(\mathbf{x},\boldsymbol{C}_k') = \min_{u\in[a,b]} d(\mathbf{x},\mathbf{x}_k'(u)) \tag{4-3}$$

其中$d(\mathbf{x}_1,\mathbf{x}_2)$是点$\mathbf{x}_1$和点$\mathbf{x}_2$之间的欧几里德距离，即$d(\mathbf{x}_1,\mathbf{x}_2)=\|\mathbf{x}_1-\mathbf{x}_2\|$，这里，$\boldsymbol{C}_k'$以点序列$\mathbf{x}_{k,l}'(l=1,\ldots,N_k)$的一个子集给出。则简单的定义如下：

$$d(\mathbf{x},C_k')=\min_{l\in\{1,\ldots,N_k\}}d(\mathbf{x},\mathbf{x}_{k,l}') \tag{4-4}$$

对物体图像中给定的点$\mathbf{x}$，在模型图像中的最近点$\mathbf{y}$满足

$$d(\mathbf{x},\mathbf{y})=\min_{k\in\{1,\ldots,n\}}d(\mathbf{x},C_k')=\min_{k\in\{1,\ldots,n\}}\min_{l\in\{1,\ldots,N_k\}}d(\mathbf{x},\mathbf{x}_{k,l}')$$

最坏情况下，找到最近点的计算量是$O(N_k^n)$，其中N_k^n是模型数据中点的总数。物体数据中全部点的总的计算量是$O(N_i^m N_k^n)$，其中N_i^m为物体数据中点的总数。使用$k-$D树能大大加速这一过程。

4.2.2 伪点匹配

对于每个点$\mathbf{x}$，我们总能找到最近点$\mathbf{y}$。但是，由于传感器的分辨力所限，在两图像中存在许多虚假的点，或者由于在物体图像中可见的点因为传感器或目标运动而在模型图像中不可见，所以从某种程度上讲，$\mathbf{x}$与$\mathbf{y}$的配对可能没有意义。可以附加一些限制条件剔除这些虚假的点对。例如，沿一条曲线的距离连续性，它类似于立体匹配上的图形连续性[52,53]，对剔除错误的匹配很有效。但为了保持算法的最简单形式，有必要引入下述两种简单的限制条件：

第一个限制条件是最大距离容限，如果点$\mathbf{x}_{i,j}$与它的最近点$\mathbf{y}_{i,j}$之间的距离，即$d(\mathbf{x}_{i,j},\mathbf{y}_{i,j})$大于最大距离容限$D_{\max}$，那么取方程（4-2）中$p_{i,j}=0$，也就是说，在模型数据中找不到合理的点与$\mathbf{x}_{i,j}$配对。这个限制条件很容易证明，因为我们知道两图像间运动很小，因此合理配对的两点之间的距离不能太大。这里的$D_{\max}$是在每一次迭代过程中，通过分析距离统计表以鲁棒性的方式适当设置的。

第二个限制条件是方向的一致性。容易看出，点$\mathbf{x}$的切线和它最近点$\mathbf{y}$的切线的夹角不能超过两图像的旋转角[54]。因此，让两个配对点之间切线的夹角不超过一个预设的值Θ，Θ为两图像之间期望的最大转角，这个限制条件在运动相对较大时尤其有效。

4.2.3 匹配更新

用$D_{\max}^{\mathrm{I}}$表示第I次迭代过程中的最大距离容限。此时，保留物体数据中（应

用了前面的恢复运动后）与其最近点的距离小于$D_{\max}^{I}$的点及其最近点和两点的距离。设$\{\mathbf{x}_i\},\{\mathbf{y}_i\},\{d_i\}$分别为伪点匹配后得到的原始点集合，最近点集合，原始点与最近点间的距离集合。设N为集合的基数。计算距离的均值μ和方差σ

$$\mu=\frac{1}{N}\sum_{i=1}^{N}d_i$$

$$\sigma=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i-\mu)^2}$$

根据μ值的不同，适当的设定最大距离容限$D_{\max}^{I}$如下：

```
if  μ<D                    /* 校准非常好*/
    D_max^I = μ+3σ ;
else if  μ<3D              /*校准仍然很好*/
    D_max^I = μ+2σ ;
else if  μ<6D              /*校准不是太差*/
    D_max^I = μ+σ ;
else                       /*校准很差*/
    D_max^I = ξ
end
```

其中，ξ是所有距离的中值，即小于ξ的d_i的数量和大于ξ的数量大致相等。

这里用新的$D_{\max}^{I}$更新以前完成的匹配：如果$\mathbf{x}_i$和$\mathbf{y}_i$的距离大于$D_{\max}^{I}$，这个点对就被剔除。剩下的点对用来计算如下所述的两图像之间的运动。

由于$D_{\max}$是在距离统计的基础上适当地设置的，所以该算法对相对大的运动和粗糙的数据具有很好的鲁棒性。例如，当校准真的很差时，只有一半的原始匹配被保留下来，既使其中保留了几个错误匹配，使用最小二乘法仍能得到一个合理的运动估计，这足以使算法收敛到正确的结果。

4.2.4 变换的计算

本文中用$\{\mathbf{x}_i\}$和$\{\mathbf{y}_i\}$分别表示 3-D 点集合及与其合理配对的最近点集合，设N为点对的个数。因为通常N远大于 3（三个点是计算唯一刚性运动的最少的点数），所以有必要设计一个程序通过最小化下面的均方目标函数来计算变换。

$$F(\boldsymbol{R},\mathbf{t})=\frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{R}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_i\right\| \tag{4-5}$$

任何最优化方法，如最速下降法、变化梯度法或更复杂的方法，都可以用来找到最小二乘的旋转和平移。本文选用对偶四元数法计算刚体的旋转和平移，下面简述用对偶四元数法计算变换。

四元数 $\boldsymbol{q}$ 可以理解为四维向量 $[q_1,q_2,q_3,q_4]^T$ 或者数对 $(\mathbf{q},q_4)$，其中 $\mathbf{q}=[q_1,q_2,q_3]^T$。对偶四元数 $\hat{\boldsymbol{q}}$ 由两个四元数 $\boldsymbol{r}$ 和 $\boldsymbol{s}$ 组成，即

$$\hat{\boldsymbol{q}}=\boldsymbol{r}+\varepsilon\boldsymbol{s} \tag{4-6}$$

其中 ε 遵循特殊的乘法规则 $\varepsilon^2=0$。两个重要的四元数矩阵函数定义如下

$$\boldsymbol{Q}(\boldsymbol{q})=\begin{bmatrix} q_4\boldsymbol{I}+\boldsymbol{K}(\mathbf{q}) & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \tag{4-7}$$

$$\boldsymbol{W}(\boldsymbol{q})=\begin{bmatrix} q_4\boldsymbol{I}-\boldsymbol{K}(\mathbf{q}) & \mathbf{q} \\ -\mathbf{q}^T & q_4 \end{bmatrix} \tag{4-8}$$

其中 $\boldsymbol{I}$ 是单位矩阵，$\boldsymbol{K}(\mathbf{q})$ 是反对称矩阵，定义为：

$$\boldsymbol{K}(\mathbf{q})=\begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}$$

一个 3-D 刚性运动可表示为满足下面两个限制条件的对偶四元数 $\hat{\boldsymbol{q}}$：

$$\boldsymbol{r}^T\boldsymbol{r}=1 \quad 和 \quad \boldsymbol{r}^T\boldsymbol{s}=0 \tag{4-9}$$

于是还有六个独立的参数表示 3-D 运动。旋转矩阵 $\boldsymbol{R}$ 可以表示为

$$\boldsymbol{R}=\left(\mathrm{r}_4^2-\mathbf{r}^T\mathbf{r}\right)\boldsymbol{I}+2\mathbf{r}\mathbf{r}^T+2\mathrm{r}_4\boldsymbol{K}(\mathbf{r}) \tag{4-10}$$

平移向量 $\mathbf{t}$ 以对偶四元数的形式写出

$$\boldsymbol{t}=\boldsymbol{W}(\boldsymbol{r})^T\boldsymbol{s} \tag{4-11}$$

$\mathbf{t}$ 是四元数 $\boldsymbol{t}$ 的矢量部分，$\boldsymbol{t}$ 的标量部分恒等于零。

把 3D 向量 $\mathbf{x}$ 表示为四元数的形式 $(\mathbf{x},0)$，则

$$\mathbf{x}_i=\boldsymbol{R}\mathbf{x}_i^0+\mathbf{t}$$

$$\boldsymbol{x}_i=\boldsymbol{W}(\boldsymbol{r})^T\boldsymbol{s}+\boldsymbol{W}(\boldsymbol{r})^T\boldsymbol{Q}(\boldsymbol{r})\boldsymbol{x}_i^0$$

于是，目标函数方程(4-5)可以写成四元数 $\boldsymbol{r}$ 和 $\boldsymbol{s}$ 的二次函数：

$$F=\frac{1}{N}\left[\boldsymbol{r}^T\mathbf{C}_1\boldsymbol{r}+N\boldsymbol{s}^T\boldsymbol{s}+\boldsymbol{s}^T\mathbf{C}_2\boldsymbol{r}+\text{const.}\right] \tag{4-12}$$

其中：
$$\mathbf{C}_1 = -2\sum_{i=1}^{N} \boldsymbol{Q}(y_i)^T \boldsymbol{W}(x_i) = -2\sum_{i=1}^{N} \begin{bmatrix} \boldsymbol{K}(\mathbf{y})\boldsymbol{K}(\mathbf{x}) + \mathbf{y}\mathbf{x}^T & -\boldsymbol{K}(\mathbf{y})\mathbf{x} \\ -\mathbf{y}^T\boldsymbol{K}(\mathbf{x}) & \mathbf{y}^T\mathbf{x} \end{bmatrix} \quad (4\text{-}13)$$

$$\mathbf{C}_2 = 2\sum_{i=1}^{N} [\boldsymbol{W}(x_i) - \boldsymbol{Q}(\mathbf{y}_i)] = 2\sum_{i=1}^{N} \begin{bmatrix} -\boldsymbol{K}(\mathbf{x}) - \boldsymbol{K}(\mathbf{y}) & \mathbf{x}-\mathbf{y} \\ -(\mathbf{x}-\mathbf{y})^T & 0 \end{bmatrix} \quad (4\text{-}14)$$

$$\text{const.} = \sum_{i=1}^{N} \left(x_i^T x_i + y_i^T y_i \right) \quad (4\text{-}15)$$

联立方程（4-9）的限制条件，最佳的对偶四元数通过最小化目标函数得到

$$\overline{F} = \frac{1}{N}\left[r^T \mathbf{C}_1 r + N s^T s + s^T \mathbf{C}_2 r + \text{const.} + \lambda_1 \left(r^T r - 1 \right) + \lambda_2 \left(s^T r \right) \right] \quad (4\text{-}16)$$

其中λ_1和λ_2是拉格朗日乘数。上式取偏导有

$$\frac{\partial \overline{F}}{\partial r} = \frac{1}{N}\left[\left(\mathbf{C}_1 + \mathbf{C}_1^T \right) r + \mathbf{C}_2^T s + 2\lambda_1 r + \lambda_2 s \right] = 0 \quad (4\text{-}17)$$

$$\frac{\partial \overline{F}}{\partial r} = \frac{1}{N}\left[2Ns + \mathbf{C}_2 r + \lambda_2 r \right] = 0 \quad (4\text{-}18)$$

用r乘以方程(4-18)得$\lambda_2 = -r^T \mathbf{C}_2 r = 0$，因为$\mathbf{C}_2$是反对称阵 ，因此得到$s$

$$s = -\frac{1}{2N}\mathbf{C}_2 r \quad (4\text{-}19)$$

将结果带入方程（4-17）得

$$Ar = \lambda_1 r \quad (4\text{-}20)$$

其中：

$$A = \frac{1}{2}\left[\frac{1}{2N}\mathbf{C}_2^T \mathbf{C}_2 - \mathbf{C}_1 - \mathbf{C}_1^T \right] \quad (4\text{-}21)$$

因此四元数r是矩阵A的特征向量，λ_1为相应的特征值。将上述结果代入方程（4-16）得

$$\overline{F} = \frac{1}{N}\left(\text{const.} - \lambda_1 \right) \quad (4\text{-}22)$$

如果我们选择最大特征值对应的特征向量，那么误差最小。

计算出r后，旋转矩阵R可以从方程（4-10）中计算出来，对偶四元数s可从方程（4-19）算出，平移向量$\mathbf{t}$可从方程（4-11）解出。

其他有效的算法包括单位四元数法[55]和奇异值分解法，用单位四元数法和对

偶四元数法几乎得到同样的变换估计。而对偶四元数法的一个优点是矩阵$\mathbf{C}_1$和$\mathbf{C}_2$能增量的计算。

4.2.5 算法小结

迭代匹配算法的步骤总结如下：

- 输入：物体与模型的3-D图分别包含m条曲线C_i和n条曲线C_k'，曲线是3D点序列$\mathbf{x}$的集合，即输入物体数据点集合和模型数据点集合。
- 输出：两图像间的最优变换。
- 程序：

a. 初始化

设置$D_{\max}^0$，它表示每一个物体数据中与其在模型数据中的最近点的距离大于$D_{\max}^0$的点都在第一次迭代中被去掉。

b. 处理过程

（1）计算物体数据中每一个点的切线；

（2）计算模型数据中每一个点的切线；

（3）建立$k-\mathrm{D}$树表示模型图像的描述。

c. 迭代计算

（1）找到满足距离和方向限制条件的最近点；

（2）通过距离统计表更新匹配；

（3）从更新的匹配中计算两图像间的运动；

（4）把变换应用到物体数据中的所有点和它们的切线上。

需要说明的是，物体图像中的原始点与模型图像的点之间的变换是计算出来的，因此，算法给出的最终变换就代表物体与模型之间的变换。迭代终止条件定义为两次连续迭代间的变换估计的变化。第I次迭代中的平移变化定义为

$$\delta t = \frac{\|\mathbf{t}_I - \mathbf{t}_{I-1}\|}{\|\mathbf{t}_I\|}$$

为了测量旋转的变化，我们引入四元数的三角表示法，

$$r = \begin{bmatrix} \sin(\theta/2)\mathbf{n} \\ \cos(\theta/2) \end{bmatrix}$$

其中$\mathbf{n}$是旋转轴的单位向量，它说明了旋转轴的方向也说明了平移的方向。θ是绕旋转轴转过的角度。定义第I次迭代中旋转的变化为

$$\delta\mathbf{n}=\frac{\left\|\mathbf{n}_{I}-\mathbf{n}_{I-1}\right\|}{\left\|\mathbf{n}_{I}\right\|}$$

4.3 算法的关键技术

4.3.1 最近点搜索

搜索最近点的计算量是$O(N)$，其中$N=N_k^n$是模型数据中所有点的数目。存在一些方法可以加速搜索过程，包括桶形技术和$k-$D树(k-dimensional binary search tree)。本文选用$k-$D方法。

$k-$D树是一维空间中的对分扩展到k维空间[57]，这里$k=3$。下面建立3-D树，首先，选择一个过数据点$\mathbf{p}$且平行于yz平面的平面，将整个空间分成两个成直角的平行六面体，使分割双方的数据点大致相等，得到左右两个子(son)集。然后，每个子集再被平行于xz的平面分割，同样要求分割双方的数量大致相等。得到两个"孙"子(grandson)集。再选择一个平行于xy的平面继续分割，让每次分割平面的方向按照yz，xz和xy平面的顺序变化。分割过程直到直角平行六面体不包含任何点为止；相应的节点是树的叶子。$k-$D可以用$O(N)$存储量在$O(N\log N)$时间内建立起来，它们都是最优的。

现在，我们研究用3-D树搜索最近点。实际上，对于物体图像上的给定点$\mathbf{x}$，我们不去寻找他在第二帧上的最近点，而是，寻找所有到$\mathbf{x}$的距离小于最大距离容限$D_{\max}$的点。因此，对于每一点$\mathbf{x}$都有一个按距离增加顺序排列的候选点列表，列表可能为空。这给使用者采用更复杂的方法提供了弹性空间，也就是说，在伪点匹配时放宽了界限。搜索算法是递归的过程。形式上，3-D树T的节点v有两项$(P(v),t(v))$表示。点$P(v)$是将空间分成两部分的点。参数$t(v)$取0，1，2分别表示分割平面平行于平面yz，xz和xy。算法积累的返回点存在程序之外的列表U中，初始设为空。下面是寻找$\mathbf{x}$的最近点的伪代码：

- 输入：点$\mathbf{x}$，3-D树T和最大距离容限$D_{\max}$。
- 输出：列表U包含所有到$\mathbf{x}$的距离在$D_{\max}$之内的点。

- 程序伪码

$-\mathbf{if}(v == \text{leaf})\mathbf{return};$
$-c_1 = \mathbf{x}[t(v)];$
$-c_2 = P(v)[t(v)];/*c_2$用来分割空间$*/$
$-\mathbf{if}(|c_1 - c_2| \le D_{\max})\mathbf{then}\mathbf{if}(\|\mathbf{x} - P(v)\| \le D_{\max})\mathbf{then}U \Leftarrow P(v);$
$-\mathbf{if}(c_1 - D_{\max} < c_2)\mathbf{then}\mathbf{SEARCH}(\text{leafson}(v), \mathbf{x}, D_{\max});$
$-\mathbf{if}(c_2 - D_{\max} < c_1)\mathbf{then}\mathbf{SEARCH}(\text{rightson}(v), \mathbf{x}, D_{\max});$

使用 3-D 树的方法最差的计算量是$O(N^{2/3})$。搜索时间主要依赖于$D_{\max}$。当$D_{\max}$很小时，搜索可以很快完成。在每次迭代过程中更新$D_{\max}$，经过几次迭代后它就变得很小了。

4.3.2 参数D的选择

需要使用者提供的参数只有**D**，它标志着什么时候两图像之间的匹配可以认为是好的。换句话说，匹配好的时候**D**的值应该对应所期望的平均距离。

D值对算法的收敛性也有影响。如果**D**过小，那么算法收敛就需要较多次的迭代，因为许多好的匹配会在匹配更新时被去掉。另一方面，如果**D**的值过大，那么算法可能无法收敛到正确的结果，因为很多错误的匹配无法被去掉。因此，出于谨慎。**D**最好选一个小的值。

设$\overline{D}$为模型图像上连续点之间的平均距离，即

$$\overline{D} = \frac{\sum_{k=1}^{n}\sum_{l=1}^{N_k-1}\|\mathbf{x}_{k,l} - \mathbf{x}_{k,l+1}\|}{\sum_{k=1}^{n}(N_k - 1)}$$

认为完美的对准如图 4.3 所示。物体图像上的点用“*”表示，模型图像上的点用“点”表示。假设叉位于两个点的中间。那么在这种情况下，两点集合间的距离均值$\mu = \overline{D}/2$。因此，我们可以认为当校准不好时$\mu > \overline{D}/2$。这里取$\mathbf{D} = \overline{D}$就能得到满意的结果。

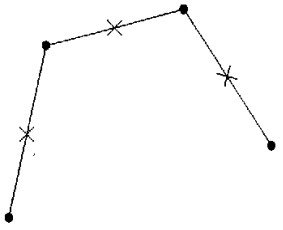


图 4.3 对准较好时如何选择 **D**

4.3.3 不确定性

因为计算的易处理性和合理的近似，3-D点的不确定性经常建立成高斯模型。也就是用3-D位置矢量和3×3方差矩阵来表示。4.2.4描述的变换计算的算法是很有效的。但是，它是假设每一点有相同的不确定性。而且，把不确定性考虑周全是很困难的。这里只考虑部分的不确定性。事实上，可以给两图像中的每一个点对配上一个加权系数。不对式（4-5）最小化，而是通过对下面函数的最小化计算 $\boldsymbol{R}$ 和 $\mathbf{t}$

$$F(\boldsymbol{R},\mathbf{t})=\frac{1}{N}\sum_{i=1}^{N}w_i\left\|\boldsymbol{R}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_i\right\| \tag{4-23}$$

其中 w_i 是点对 $\mathbf{x}_i$ 和 $\mathbf{y}_i$ 之间的正的加权系数。用对偶四元数的表示方法，目标函数 $F(\boldsymbol{R},\mathbf{t})$ 可写成 $\boldsymbol{r}$ 和 $\boldsymbol{s}$ 的二次函数：

$$F=\frac{1}{N}\left[\boldsymbol{r}^T\mathbf{C}_1\boldsymbol{r}+W\boldsymbol{s}^T\boldsymbol{s}+\boldsymbol{s}^T\mathbf{C}_2\boldsymbol{r}+\text{const.}\right] \tag{4-24}$$

其中：

$$\mathbf{C}_1=-2\sum_{i=1}^{N}w_i\,\boldsymbol{Q}(\boldsymbol{y}_i)^T\boldsymbol{W}(\boldsymbol{x}_i) \tag{4-25}$$

$$\mathbf{C}_2=2\sum_{i=1}^{N}w_i\left[\boldsymbol{W}(\boldsymbol{x}_i)-\boldsymbol{Q}(\mathbf{y}_i)\right] \tag{4-26}$$

$$W=\sum_{i=1}^{N}w_i \tag{4-27}$$

$$\text{const.}=\sum_{i=1}^{N}w_i\left(\boldsymbol{x}_i^T\boldsymbol{x}_i+\boldsymbol{y}_i^T\boldsymbol{y}_i\right) \tag{4-28}$$

我们发现方程（4-24）和方程（4-12）相似。四元数$\boldsymbol{r}$是矩阵$\boldsymbol{A}$最大特征值对应的特征向量

$$\boldsymbol{A}=\frac{1}{2}\left[\frac{1}{2W}\mathbf{C}_2^T\mathbf{C}_2-\mathbf{C}_1-\mathbf{C}_1^T\right]$$

那么四元数$\boldsymbol{s}$为

$$\boldsymbol{s}=-\frac{1}{2W}\mathbf{C}_2\boldsymbol{r}$$

加权系数w_i应该与$\boldsymbol{R}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_i$的不确定性相关。设$\Lambda_{x_i}$，$\Lambda_{y_i}$和$\Lambda_i$是$\mathbf{x}_i$，$\mathbf{y}_i$和$\boldsymbol{R}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_i$的协方差矩阵。$\Lambda_{x_i}$和$\Lambda_{y_i}$由感测系统给出，如立体成像。$\Lambda_i$如下式作为近似值计算

$$\Lambda_i=\boldsymbol{R}\Lambda_{x_i}\boldsymbol{R}^T+\Lambda_{y_i}$$

其中$\boldsymbol{R}$是之前迭代过程中计算出来旋转矩阵。Λ_i的迹粗略的表示$\boldsymbol{R}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_i$不确定性的大小。因而我们选择

$$w_i=\frac{1}{\mathrm{tr}(\Lambda_i)}=\frac{1}{\mathrm{tr}(\Lambda_{x_i})+\mathrm{tr}(\Lambda_{y_i})}$$

由此看出，加权系数独立于旋转之外。

4.3.4 由粗到精的策略

正如下面将要给出的那样，在最初的几次迭代时，算法的收敛较快，而在接近局部最小后变慢，而且还发现，在最初的几次迭代时，需要较多的搜索时间，因为开始时搜索空间更大。由于整个搜索时间与模型中的数据点数成线性关系，自然就要应用由粗到精的策略，在最初的几次迭代期间，可以使用粗样本（也就是每五个点）去代替曲线上的所有点。当算法几乎收敛时，使用全部可用点使之能够获得精确的估计。

4.4 仿真结果

上述算法采用 Matlab6.5 编程实现。实验中没有考虑测量的不确定性，也就是说全部实验数据的加权系数都预设为 1。程序在奔Ⅳ2.8GHz 个人机器上运行，所有的评价都是在同一机器上运行给出的。

本节分别列举了三组模拟数据来仿真迭代算法的匹配过程，包括任意 3-D 曲线、规则形状物体、任意形状物体。

4.4.1 模拟数据

第一组：模型曲线设为 $\mathbf{M}=\left[\mathbf{u}^2 \quad 5\mathbf{u}\sin(\mathbf{u})+10\mathbf{u}\cos(1.5\mathbf{u}) \quad \mathbf{u}\right]^T$，$\mathbf{u}$ 在 $4\,pi$ 范围内每隔 0.05 取一个点，因此模型集合有 252 个点。将模型集合沿着向量 $[3 \quad 4 \quad 6]$ 的方向旋转 36° 再平移 $[7 \quad 18 \quad -9]$ 产生模拟的测量曲线集合 $\mathbf{X}$，集合 $\mathbf{X}$ 同样有 252 个点。这样我们得到两个无噪声图像，如图 4.4 所示。将匹配算法应用到曲线数据上，经过 18 次迭代算法收敛，耗时 5.8 秒。匹配结果如图 4.5 所示。若将中间匹配过程显示图上，则得到图 4.6 所示的结果。

第二组：以正弦圆柱面物体为模型，即 $\mathbf{M}=\mathbf{cylinder}(2+\sin(\mathbf{t}))$，$\mathbf{t}$ 在 $2\,pi$ 范围内每个 $pi/10$ 取一个点，则模型集合有 $21\times 21=441$ 点。同样将模型集合沿着向量 $[3 \quad 4 \quad 6]$ 的方向旋转 36° 再平移 $[1 \quad 5 \quad -3]$ 产生模拟的物体集合。这样我们到两个物噪声的模拟图像，如图 4.7 所示。将匹配算法应用到 3-D 物体数据上，经过 21 次迭代算法收敛，耗时 6.2 秒。匹配结果如图 4.8 所示。若将中间匹配过程显示图上，则得到图 4.9 所示的结果。

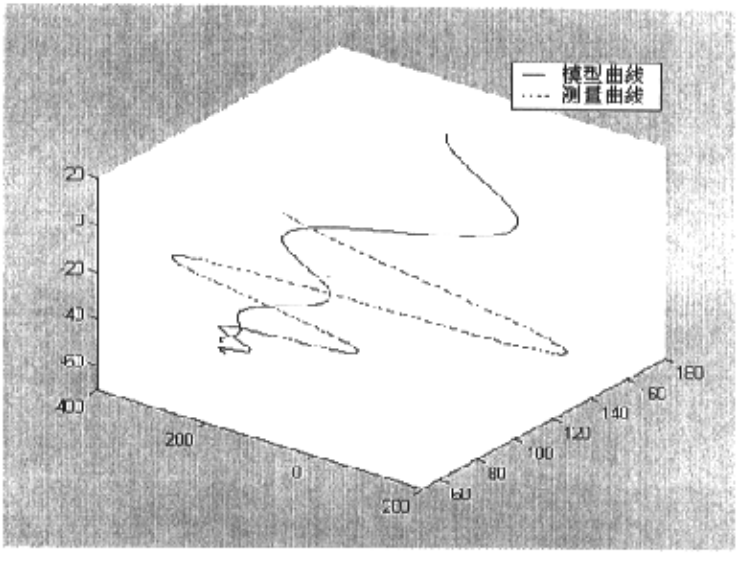


图 4.4 模型曲线与测量曲线原始状态

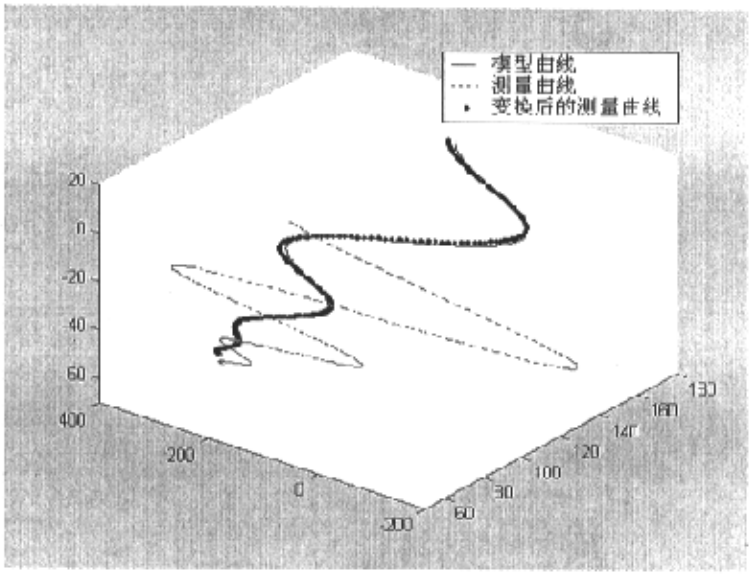


图 4.5 测量曲线经过迭代算法被匹配到模型曲线上

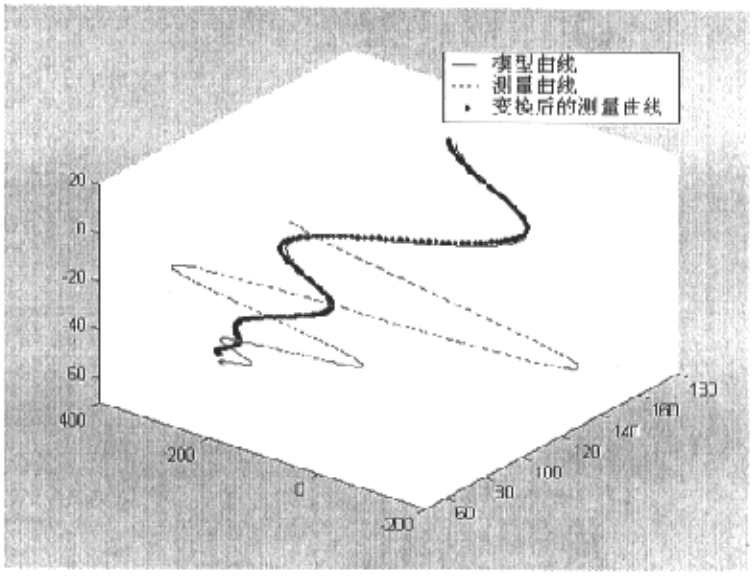


图 4.6 测量曲线向模型曲线运动的过程

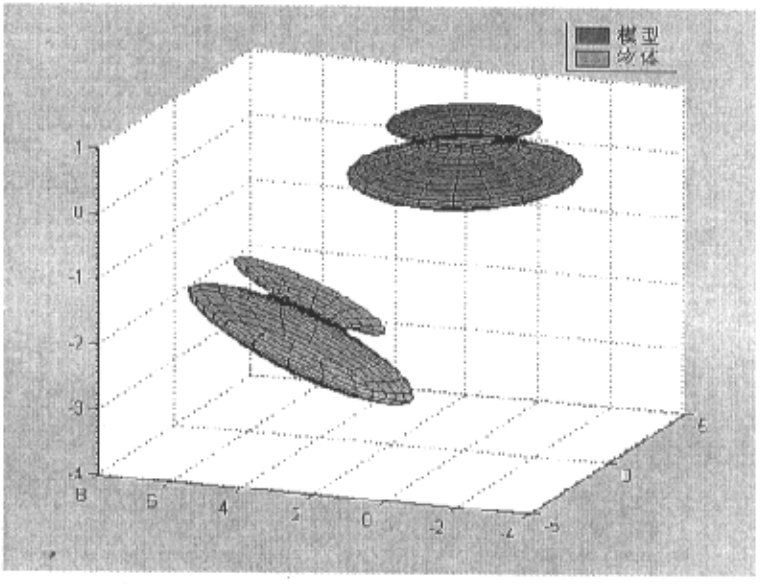


图 4.7 模型与物体的原始状态

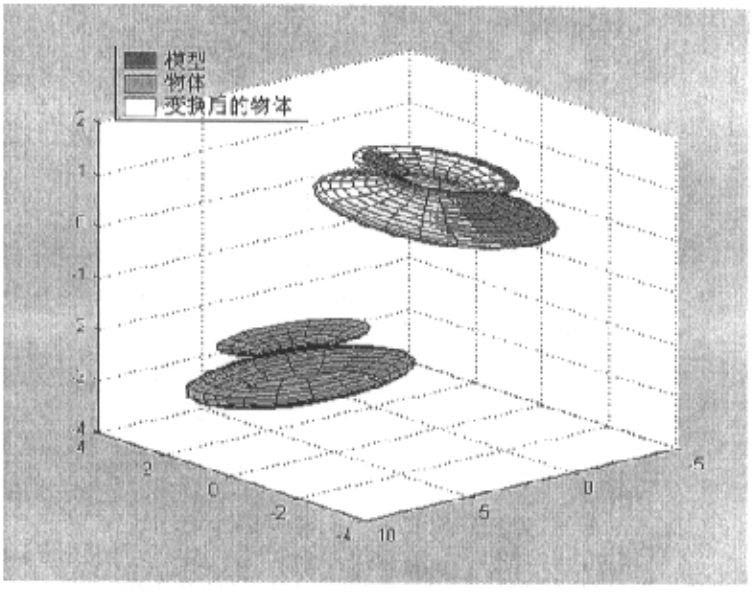


图 4.8 物体经过迭代算法匹配到模型

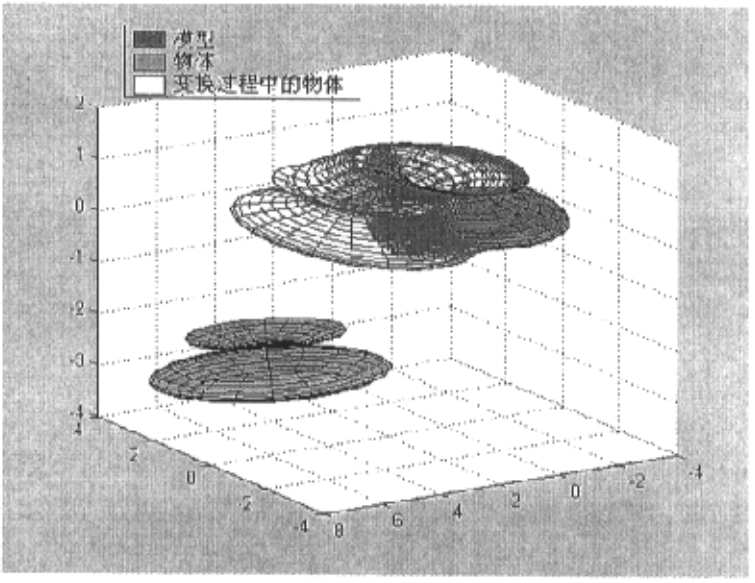


图 4.9 物体向模型运动的过程

第三组：模型数据来自于一个“兔子”模型，将能表现兔子外形特征的所有采样点都以 3-D 点坐标的形式表示于 3-D 坐标系内，如在兔子耳朵取的采样点为 $(-0.06325,0.0359793,0.0420873)$，$(-0.06275,0.0360343,0.0425949)$等等，共取采样点 40256 个，因此模型集合包括 40256 个点。以同样的方法将兔子模型沿着向量$[3\quad 4\quad 6]$的方向旋转 36° 再平移$[0.07\quad 0.15\quad -0.1]$后得到物体集合，实际应用中物体应单独采样，这里不这么做是因为变换后的物体采样点可能在模型采样点中无法找到对应点。在最后结论中给出分析。因此物体集合也包含 40256 个采样点。由于采样误差和精度误差存在，这组数据更接近实际 3-D 物体，所以更能表现出算法在实践应用中的性能。两个原始图像如图 4.10 所示。将匹配算法应用到 3-D 兔子数据上，经过 54 次迭代算法收敛，耗时 1075.6 秒。匹配结果如图 4.11，可见集合很大的情况下耗时也会很多，符合前面结论。若将中间匹配过程显示图上，则得到图 4.12 所示的仿真结果图，为了清晰起见其中只显示了两个中间图像。

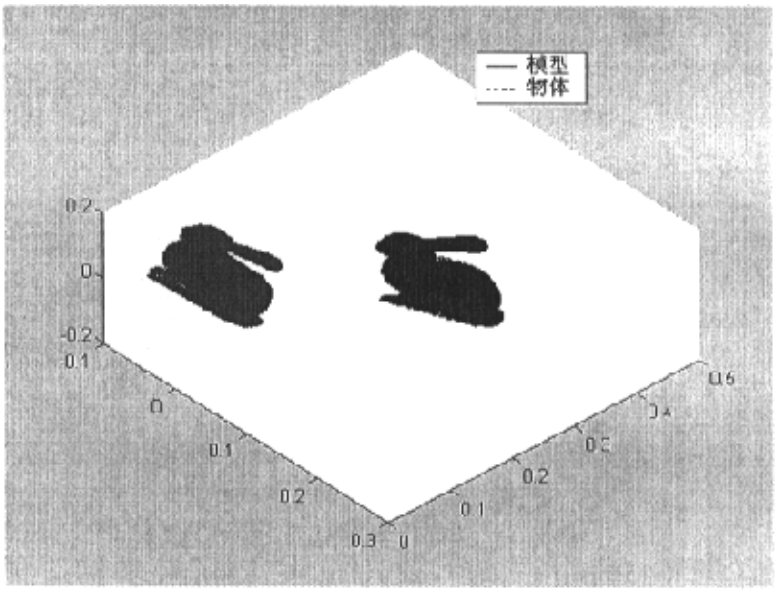


图 4.10 兔子模型与物体的原始状态

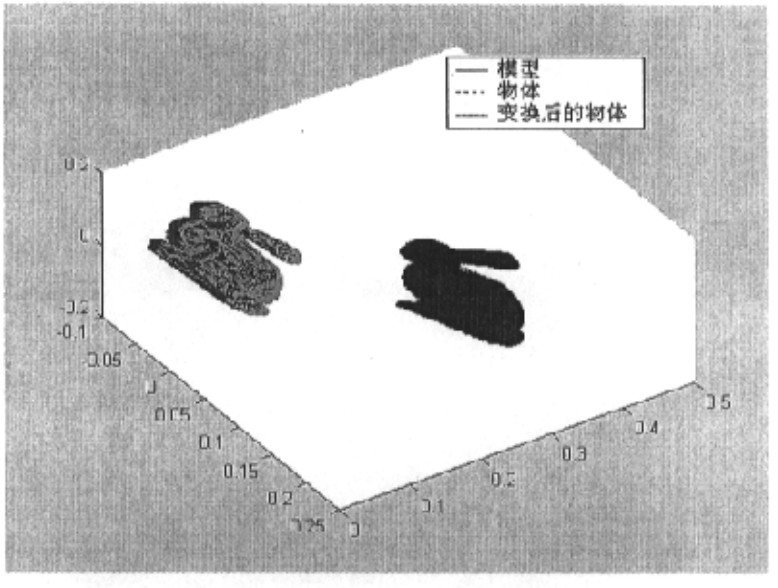


图 4.11 物体兔子匹配到模型兔子上

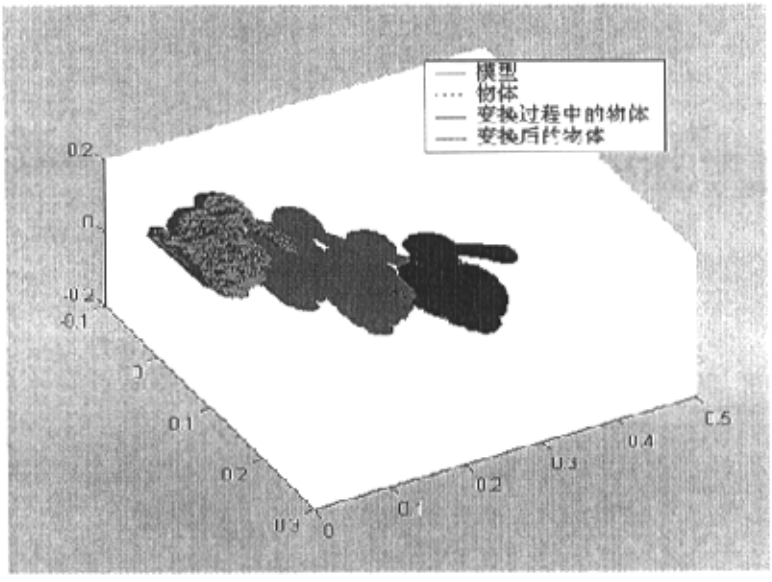


图 4.12 物体兔子向模型兔子运动的过程

4.4.2 算法的鲁棒性和有效性分析

本节以任意 3-D 曲线为例描述和分析一下算法的鲁棒性和有效性，使用上一节中同样的模拟数据，但是加入不同水平的噪声。下面给出的所有结果都是十次实验的平均值。

对于每一点在各坐标上都加上均值为 0，标准方差为 0 到 20 的高斯噪声，为了考核运动估计的精度，定义旋转误差为：

$$e_{\mathrm{r}} = \|\mathbf{r} - \hat{\mathbf{r}}\| / \|\mathbf{r}\| \times 100\% \tag{4-29}$$

这里$\mathbf{r}$和$\hat{\mathbf{r}}$分别是真实的和估计的旋转参数，平移误差为：

$$e_{\mathrm{t}} = \|\mathbf{t} - \hat{\mathbf{t}}\| / \|\mathbf{t}\| \times 100\% \tag{4-30}$$

这里$\mathbf{t}$和$\hat{\mathbf{t}}$分别是真实的和估计的平移参数。

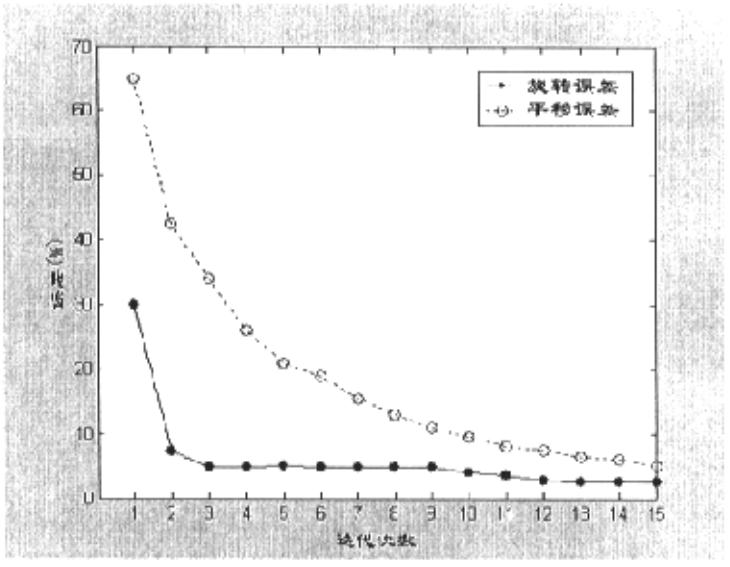


图 4.13 标准方差等于 2 时旋转误差和平移误差相对迭代次数的变化

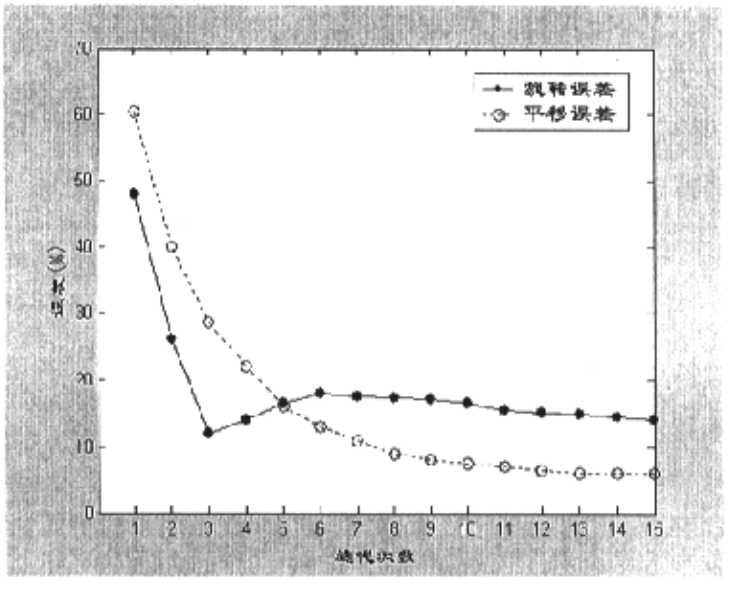


图 4.14 标准方差等于 8 时旋转误差和平移误差相对迭代次数的变化

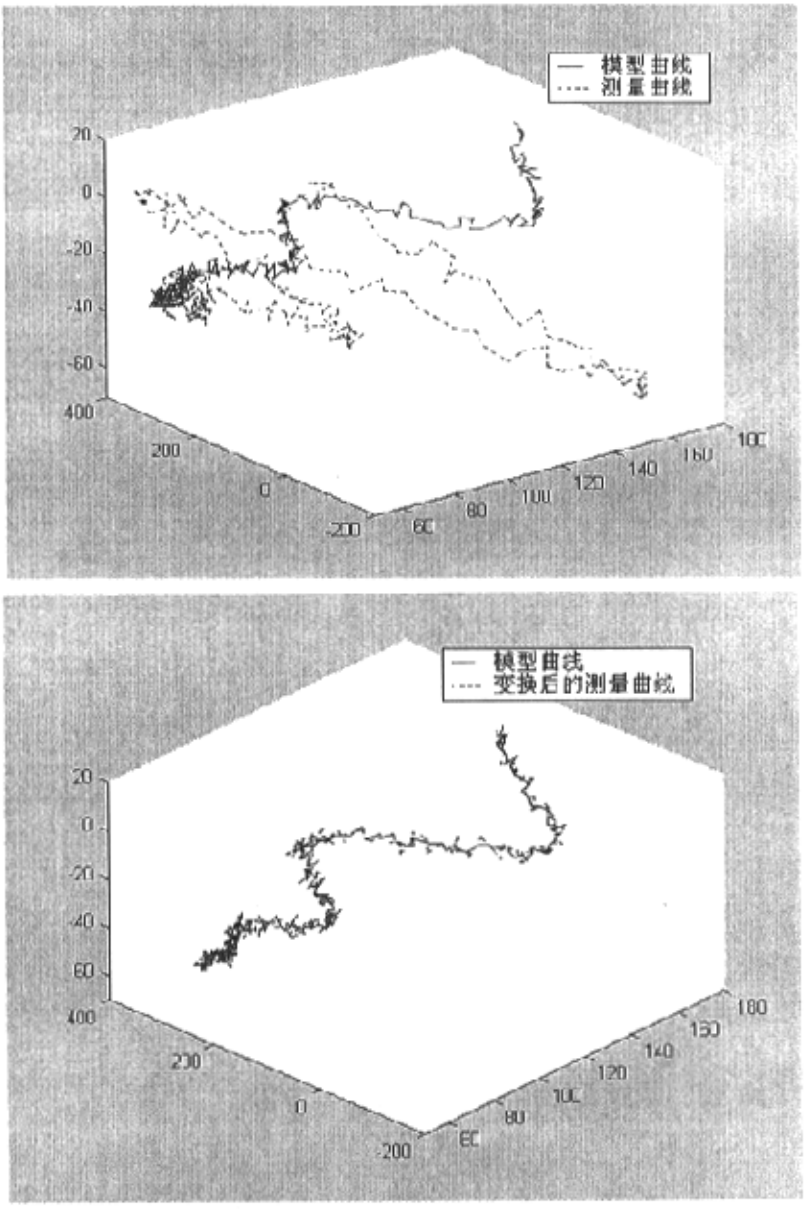


图 4.15 标准方差等于 16 时两条噪声曲线匹配前和匹配后的情况

现在以标准方差等于2和8的情况为例，观察旋转误差和平移误差相对迭代次数的变化情况。如图4.13和4.14。从这些结果中可以看出：

• 平移误差的减少几乎是单调的，而旋转误差的形状更为复杂。

• 噪声对旋转参数的影响比对平移参数的影响更强，当噪声较小时，总体上旋转误差小于平移误差，当噪声强烈时情况相反。

上述现象是由于测量法和旋转参数之间是非线性关系，而与平移参数之间是线性关系。

为了直观的展示加入噪声的影响和算法的能力，图4.15给出了算法中数据的各个坐标加入均值为0标准方差为16的高斯噪声后的仿真结果。即使曲线中有这么大的噪声，它们之间的匹配也是相当不错的。

表4.1旋转和平移误差的测量值及执行时间。其中的数值都是平均十次试验的结果。这里迭代次数为15，得到下述结论：

• 旋转和平移误差随着增加的噪声值增加而增加，正如所料。

• 测量中的噪声对旋转参数的影响大于对平移参数的影响。

• 算法具有鲁棒性，即使数据被严重破坏也能产生合理的运动估计。

• 执行时间也随着加入数据的噪声的增加而增加，因为当噪声很大时 $D_{\max}$ 的值也很大，搜索必须在大范围内进行。

表 4.1 模拟数据实验结果总结

标准方差	0	2	4	6	8	10	12	14	16	18	20
旋转误差	2.24	2.13	4.63	9.64	13.71	14.35	20.47	18.09	23.87	37.04	33.21
平移误差	1.77	4.36	4.55	4.84	5.72	7.83	8.94	9.89	17.15	22.01	27.17
执行时间	6.37	6.92	8.68	9.36	11.23	11.97	12.79	13.55	16.54	16.67	17.42

4.5 本章小结

本章描述了3-D物体的匹配方法，算法中的3-D物体用空间曲线上的序列点表示，并且假设物体与模型之间的运动很小或近似已知。讨论了实际应用中要考虑的关键技术，通过仿真实验验证算法的主要特性，得出了相关结论。

第5章 迭代算法的派生形式

5.1 关于本文算法的几点讨论

本文提出的算法还有几个需要讨论的方面。首先是此算法如何处理大的运动。由于采用了局部最优的匹配判据，算法收敛到最小，最适合应用于小运动或运动近似已知且需要精确估计运动的情况。在大运动的情况下，算法可以采取两种不同的方法。第一种方法是先应用在上一章引言中介绍的全局方法获得一个估计，然后应用本文给出的方法再优化。第二种方法是用采样6-D运动空间获得一个初始对准集合，然后将算法应用于每一个初始对准。相对于全局最小误差的最终估计作为最佳估计。这种方法在文献[27]中已有所描述。

其次是如何处理多目标运动。在动态环境中，通常不止一个运动目标，应用运动信息将目标可靠分段的算法非常重要，目前为止在该方向上的研究工作还很少。在文献[62]中提出了一个处理多目标运动的框架，分为两个等级，第一级是一帧一帧地处理3-D特征及其运动估计，对每个特征，处理过程完全是并行的，第二级是基于运动参数的相似性将特征分组成为目标，来自单个目标的特征应该具有相同的运动参数。文献[63]中，使用的特征是3-D线段，试验表明，该方法是灵活有效的。如果用3-D曲线代替3-D线段，并且对每条曲线估计3-D运动，总体方案仍然可用。

最后还要讨论一下此算法与之前算法的比较。与Besl和McKay[27]提出的用于3-D物体匹配ICP算法相比，我们应用了同样的思想：将一个集合中的点与另一个集合中的最近点迭代匹配，主要差别在于匹配判据。参照方程（4-2），在我们的算法中，$p_{i,j}$可以取1或0，依据是在第一个集合中的点和第二个集合中的点的匹配是否合理。这由最大距离容限D_{max}决定，它是通过分析距离统计表动态建立的，因此，我们的算法有能力处理下述情况：

（1） 粗糙的数据

粗糙的数据在匹配中会自动剔除，因此它不会影响最终的估计。

（2） 显现和消失

曲线在一个集合中出现在另一个集合中消失。这种情况常出现在避碰过程

中，目标可能离开观察区域。

（3）遮挡。一个目标可能部分或全部挡住其他的目标。这常发生在目标识别和导航的情况下。

ICP算法的 $p_{i,j}$ 总是取1，因此只能处理第一个集合是第二个集合的子集的情况，对于上述的情况无能为力。其它就是基本步骤上的区别，如对偶四元数法计算3-D运动，可以部分的考虑数据点的不确定性，而ICP使用奇异值分解法。

5.2 迭代算法的派生形式

综合 ICP 算法的派生形式，按照对鲁棒性、收敛性和配准精度的影响来分主要有以下几个方面 [64]：

- 从某一个或两个网格中选择一些点构成点集合；
- 寻找模型形态中对应的匹配点；
- 对应点对之间的相应加权值；
- 基于某些原则拒绝某些不满足条件的点对；
- 给定基于点对的误差机制；
- 最小化误差机制。

5.2.1 从数据形态和模型形态中选择测量点

匹配对准算法中数据形态包括很多测量点，从这么多测量点中选择用于匹配的点是 ICP 算法的第一步也是很重要的一步，这些匹配点的分布和数量影响算法的鲁棒性和快速收敛性，通常用到的选择策略有：

（1）总是取所有的点，此策略的计算量很大，尤其当点的个数很多的情况下更是如此，而且这种方法基本上不具有鲁棒性；

（2）现有点按一定规则均匀采样，这种策略的计算量可以根据需要确定采样点的个数，相对 Besl 提出的取所有点的方法鲁棒性得到了一定的增强；

（3）对现有点随机采样（在每一次迭代过程中有不同的采样点），计算量和前一种差不多，但是鲁棒性却有所增强；

（4）选择斜率变化较大的点，利用完全采样颜色或强度以辅助配准，这种方法需要模型形态和数据形态有比较大的斜率点，也即是要求特征明显的点，而

且在地形匹配等某些匹配对准中很少用颜色作为辅助对准的匹配单元；

（5）选择那些所选点之间的向量分布足够大的点。这一策略的目的是使得某些场景在只具备很少特征是正确配准。像前面给出的各种选择策略通常只是选择这些特征的一部分，这就可能导致不能确定正确变换的某些元素。它比传统的基于特征的方法计算量小，但是鲁棒性也较差。

5.2.2 寻找对应点对

前面我们提到过，寻找数据形态和模型形态之间的对应关系是图像配准方法需要解决的问题。ICP 算法提出的基本对应关系是寻找数据形态和模型形态之间的最近匹配点，对于每一个已经选择的测量点，由于最近匹配点定义的方式不一致，所以方法也不同，通常有以下几种：

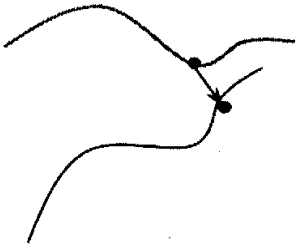


图 5.1 最近点

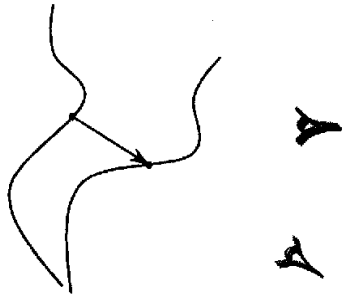


图 5.2 最近法向量相交点

（1）从模型形态中寻找最近距离点，如图 5.1 所示。这种方法当数据形态和模型形态靠得很近且相对平滑时工作的很好，而且此方法可以采用 k-d 树或最近点高速缓存的方法来加快它的计算速度；

（2）寻找起始于原点，方向为终点曲面在原点处的法向量的交点，称为最近法向量相交点（Normal shooting），如图 5.2 所示；

（3）将原点投影到模型形态上，从目标网络的距离相机的视觉看的点。这也称为“反对准”（reverse calibration）；

（4）将原点投影到目标网络上，然后在目标距离图像中搜寻。这一搜寻可以是基于点对点距离，点对直线距离，或者强度匹配或者颜色；

上面的任何方法，都限制在匹配点与原点是基于某一机制的匹配。所采用的

匹配机制有基于颜色和法向量之间的角度。

5.2.3 对应点对之间的权值

虽然我们只讨论了部分的不确定性（4.3.3）即权值为常值情况下的变换矩阵的求解方法式（4-23）~（4-28），但是权值选择的方法对算法的鲁棒性影响也不一样，考虑有三种加权的方法：

（1）常数权值，这种方法比较简单，但是它的鲁棒性很差，因为它不对可能的野值点加以限制；

（2）点与点之间距离越大，给定的权值越小。这一方法和把大于某个门限值的点对舍弃的方法类似，但是又避免了后一种方法的不连续性。这种方法采用的权值为

$$Weight = 1 - \frac{Dist(p_1, p_2)}{Dist_{\max}}$$

（3）基于法向量之间的匹配情况的加权方法：

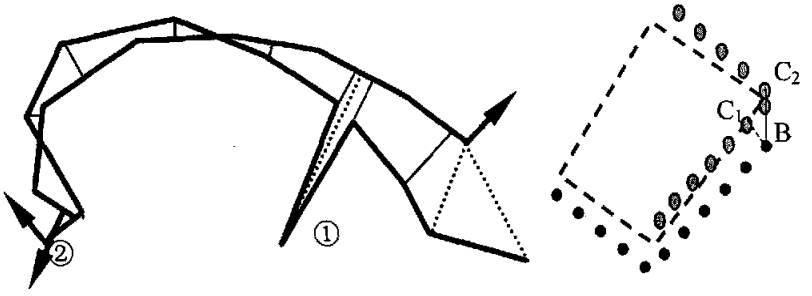
$$Weight = n_1 \cdot n_2$$

通常，权值对计算速度和收敛速率影响较小，但是对鲁棒性影响较大，所以权值的选取应当综合考虑其它的因素。

5.2.4 点对的剔除方法

Besl 和 McKay[27]提出的 ICP 算法给出了一个简单的方法：对数据点集的每一个点在模型点集中寻找最近点，但是实际中有可能由于量化的问题以及很多别的原因使得数据形态点集中的点与最近点并不能构成对应关系，这就非常严重地影响了对准匹配的最后结果。事实上，第 5.2.2 节给出的各种寻找对应点的方法都是给定的寻找正确对应关系的试探法，这种试探法经常失败，于是许多研究者[提出了附加的试探法排除那些有明显错误的匹配对，这就是通常所说的剔除方法。

这一方法与前面的加权方法中完全拒绝某一些匹配对是相关的。这一步骤的目的是消除野值点，因为野值点可能对最小二乘方法有很大影响。下面给出了几种剔除策略(几乎所有的策略都用到了舍弃离得非常远的点对的启发式方法)，几种剔除方法的图例示于图 5.3 中：



(a) 舍弃某些点对的启发式方法：①距离太大，②向量变化太大

(b) 线段边缘的点导致许多虚假配对

图 5.3 剔除点对的启发式方法

• Turk 和 Levoy 排除超过某一给定距离值（用户设定）或者距离太大的对应点，如图 5.3(a)中①所示。而且他们还排除距离图像的轮廓上的点，以及那些在另一个图像中不可见的点；

• Pulli 排除那些对应点对的法向量差别太大的点，如图 5.3(a)中②所示；

• Pulli 提出拒绝基于某一机制的最差的 $n\%$ 个点对，通常是指距离最大的 $n\%$，这种方法无论有没有野值点，都要排除一些距离相对较大的点对；

• Masuda 提出拒绝那些点与点之间距离大于距离的标准偏差一定倍数的点对，，倍数的大小可以由用户根据需要自行决定；

• 一些学者还提出了认为匹配点对是对称的。要实现对称可以将第一个图像中的点在第二个图像中寻找对应点，然后又对第二个图像中的点在第一个图像中寻找对应点，最好将正好匹配上的点对合并起来就可以。

• Dorai 采用拒绝那些相邻点不连续点对的方法。这一方法认为如果两个相邻点对 (p_1,q_1) 和 (p_2,q_2) 距离之间的差值 $|Dist(p_1,p_2)-Dist(q_1,q_2)|$ 大于某一个门限时拒绝这一点对。

• Eggert 等对遮断边界上的点寻找点对提出了一种更聪明的试探法，每一个边界上的点不是与真正的最近点的进行配对，而是与一个隐藏的表面上的最近点进行配对。图 4.8(b)给出了一个盒子的两种观测视角来对 Eggert 的方法进行描述，黑色的点是从下面的角扫描得到的，灰色的点是从右边的角扫描得到的。最右边的黑色点 B 是边界点，所以它应该对应于盒子的角。前面的许多方法都是寻找最近点 C_1 点，该方法却找到被遮断部分的最近点 C_2 点。

最后一种方案剔除了网格边缘的点对，当数据形态的真实值不完全在模型形态中的情况下对于避免错误匹配特别有用。因为它的计算量通常很低且在大多数应用中它的缺点很少，所以在动态图像处理过程中通常都推荐使用此方法。野值点拒绝方法虽然对于确定正确匹配时的精度和可靠性有影响，但是它们通常不加快收敛速度。

5.2.5 误差度量及其变换方法

迭代算法设置什么样的目标函数是非常关键的一步，也是最终作为最优化的评判标准。通常用到的误差度量方法有：

（1） 相关点对之间的距离平方和。对于这种形式的误差机制，存在确定最小化误差的刚性变换的闭环解。已经提出来的求解方法有基于奇异值分解 SVD 方法，四元数，正交矩阵，以及对偶四元数等；Eggert 等还估计了这些方法的数字精度和可靠性[66]，得出的结论是它们之间的差别很小；

（2）采用前面的“点一点”距离平方和机制，并考虑点对之间的距离和颜色之间的差别；

（3）所有源点到包含目标点和到目标法向量的方向垂线的平面之间的距离平方和。这种“点一平面”情况下不能获得闭环解。最小二乘解可以通过求解非线性的方法来解决（比如 Levenberg-Marquardt 方法），或者简化为线性问题（也就是，假定旋转增量很小，于是$\sin\theta$简化为θ，$\cos\theta$简化为 1）。

通常有以下几种方法来形成对准的搜索过程：

（1）利用当前的变换重复产生一系列相关点，并且寻找最小化误差机制的新变换。

（2）在上面的迭代最小化中，加上在变换空间的外推法以加快收敛速度。

（3）Szymon 用到的外推算法是基于 Besl 和 McKay 描述的方法，并增加了两个小的变动以提高有效性和减少超调：

当采用二次外推法且抛物线开放式下降时，Szymon 采用最大 x 截距来代替抛物线的极值。给外推数量乘以一个阻尼系数，在执行中任意设置为 $\frac{1}{2}$。

（4）执行迭代最小化开始于在初始位置加上随机扰动，然后选择最好的匹配结果。这避免了误差方程的虚假的局部最小，尤其在应用点－点误差机制时。

（5）执行迭代最小化，采用不同的随机选择点的子集，然后利用鲁棒性机制（最小中值方差 LeMes）选择最优结果。

（6）利用模拟退火法的随机搜索变换方法。

后面三种方法增强了算法的鲁棒性的同时增加了算法的复杂性，收敛速度也比较慢，所以鲁棒性和快速收敛性是相互矛盾的。在实际应用中应该综合考虑系统的情况来折衷选择算法的鲁棒性和快速性。

5.3 LM-ICP 算法

LM-ICP 算法的目的是通过非线性最小化的方法直接最小化模型数据的拟合误差，ICP 算法中距离误差函数用 $\boldsymbol{E}$ 表示。LM 算法是最优过程，它尤其适合象 $\boldsymbol{E}$ 这样的函数。误差函数 $\boldsymbol{E}(\mathbf{a})$ 可以写成 N_d 个残差的和的形式为

$$\boldsymbol{E}(\mathbf{a})=\sum_{i=1}^{N_d}\boldsymbol{E}_i^2(\mathbf{a}),\quad \boldsymbol{E}_i(\mathbf{a})=\sqrt{\omega_i}\min_j \varepsilon\left(\left\|\mathbf{m}_j-\boldsymbol{T}(\mathbf{a};\mathbf{d}_i)\right\|\right)$$

$\mathbf{e}(\mathbf{a})=\{\boldsymbol{E}_i(\mathbf{a})\}_{i=1}^{N_d}$ 为残差向量，其中 $\boldsymbol{E}(\mathbf{a})=\|\mathbf{e}(\mathbf{a})\|^2$。

LM 算法对函数最小化时合并了梯度下降法和高斯-牛顿法。每次迭代的目的是对现有的 $\mathbf{a}_k$ 选择一次更新，设为 $\mathbf{x}$，这样使 $\mathbf{a}_{k+1}=\mathbf{a}_k+\mathbf{x}$ 来减小误差函数 $\boldsymbol{E}(\mathbf{a})$。围绕 $\mathbf{a}$ 展开 $\boldsymbol{E}(\mathbf{a}+\mathbf{x})$ 得到

$$\boldsymbol{E}(\mathbf{a}+\mathbf{x})=\boldsymbol{E}(\mathbf{a})+(\nabla\boldsymbol{E}(\mathbf{a})\cdot\mathbf{x})+\frac{1}{2!}\left((\nabla^2\boldsymbol{E}(\mathbf{a})\cdot\mathbf{x})\cdot\mathbf{x}\right)+\text{h.o.t}$$

根据 $\mathbf{e}$ 表示，则

$$\boldsymbol{E}(\mathbf{a})=\mathbf{e}^{\mathrm{T}}\mathbf{e}$$
$$\nabla\boldsymbol{E}(\mathbf{a})=2(\nabla\mathbf{e})^{\mathrm{T}}\mathbf{e}$$
$$\nabla^2\boldsymbol{E}(\mathbf{a})=2(\nabla^2\mathbf{e})\mathbf{e}+2(\nabla\mathbf{e})^{\mathrm{T}}\nabla\mathbf{e}$$

用 $N_d\times p$ 的雅可比矩阵 J 表示 $\nabla\mathbf{e}$，ij^{th} 输入项为 $J_{ij}=\dfrac{\partial\boldsymbol{E}_i}{\partial\mathbf{a}_j}$，根据高斯-牛顿近似，也就是忽略 $(\nabla^2\mathbf{e})\mathbf{e}$，得到

$$\boldsymbol{E}(\mathbf{a}+\mathbf{x})\approx\mathbf{e}^{\mathrm{T}}\mathbf{e}+\mathbf{x}^{\mathrm{T}}J^{\mathrm{T}}\mathbf{e}+\mathbf{x}^{\mathrm{T}}J^{\mathrm{T}}J\mathbf{x}$$

每次迭代的任务就是决定步长 $\mathbf{x}$ 使 $\boldsymbol{E}(\mathbf{a}+\mathbf{x})$ 最小。用刚刚导出的 $\boldsymbol{E}$ 的近似值，对 $\mathbf{x}$

求微分并使结果为0，则

$$\nabla_{\mathbf{x}} E(\mathbf{a}+\mathbf{x}) = J^{\mathrm{T}}\mathbf{e} + J^{\mathrm{T}}J\mathbf{x} = 0$$

解方程求$\mathbf{x}$，得到高斯-牛顿更新，给算法高斯-牛顿ICP的一次迭代：

1. 计算残差向量$\mathbf{e}(\mathbf{a}_k)$，和根据$\mathbf{a}$导出J的$N_d \times p$矩阵。
2. 计算更新$\mathbf{x} = -(J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}}\mathbf{e}$
3. 使$\mathbf{a}_{k+1} = \mathbf{a}_k + \mathbf{x}$

当然，上述方法并不保证采用的步长会使误差$E(\mathbf{a}_{k+1})$减小。它是否减小依赖于对$\mathbf{a}_k$的泰勒级数展开的第二项的精确性，和高斯-牛顿近似值的有效性。但是，可以看出接近最小值时它们的近似值通常是好的，收敛是快速可靠的。

进行比较，用一些以前对准算法中用到的加速梯度下降法代替第二步，有

2. 计算更新$\mathbf{x} = -\lambda^{-1}J^{\mathrm{T}}\mathbf{e}$

这里λ的值控制沿梯度方向运行的距离。对于小的λ，迭代沿着下降的方向移动很长的距离，而大的λ就意味着小的步长。与高斯-牛顿法相比，梯度下降法在λ足够大的时候的确保证了E的减小，但是，它的收敛性在接近最优时是相当慢的。

LM算法为了在所有地区都获得好的性能，它用一种相对简单的方法合并了两种更新。步骤2代替为

2. 计算更新$\mathbf{x} = -(J^{\mathrm{T}}J + \lambda \mathrm{I})^{-1}J^{\mathrm{T}}\mathbf{e}$

现在大的λ对应小的、可靠的梯度下降步长，而小的λ允许在最小值附近快速收敛。好的LM执行的作用是在每次迭代后变化λ使其保证甚至在高斯-牛顿近似值不好的地方也能快速推进。

因此得出了LM-ICP算法，程序伪码为

function a = **lmicp** ($\{\mathrm{m}_j\}_{j=1}^{N_m}$, $\{\mathrm{d}_i\}_{i=1}^{N_d}$, a_0)
Set λ to an initial value —*see Press[17], p.684*
Set $\mathrm{a} = \mathrm{a}_0$
repeat
  Compute $\mathrm{e}_k = \mathrm{e}(\mathrm{a})$ —*One closest-point computation*
  Compute J —*p closest-point computations*
  Modify λ until $\mathrm{a}_k = \mathrm{a} - (\mathrm{J}^{\mathrm{T}}\mathrm{J} + \lambda\mathrm{I})^{-1}\mathrm{J}^{\mathrm{T}}\mathrm{e}_k$ reduces the error $\|\mathrm{e}(\mathrm{a}_k)\|^2$.
    —*One or more closest-point computations*
  Set $\mathrm{a} = \mathrm{a}_k$
until λ is large —*So only a small gradient descent step reduced the error.*

接下来介绍LM加强ICP鲁棒性的方法。标准的ICP要扩大收敛区间还要保

证鲁棒估计是很困难的，而用 LM-ICP，修改误差函数以包括鲁棒核却是很容易的事。两种常用的鲁棒核如下：

$$\text{Lorentzian: } \varepsilon(r)=\log\left(1+\frac{r^2}{\sigma}\right) \quad \text{Huber: } \varepsilon(r)=\begin{cases} r^2 & r<\sigma \\ 2\sigma|r|-\sigma^2 & \text{otherwise} \end{cases}$$

使用这些核的 LM-ICP 可与用 Winsorised 余数的 ICP 相匹敌，它是使 ICP 具有鲁棒性最普通的方法。用 Huber 核的 LMICP 的收敛区间比 Winsorised ICP 的收敛区间大两倍。

5.4 本章小结

本章讨论了 ICP 算法的派生形式，本文的算法也算是其派生形式之一。总结了 ICP 算法的各种派生形式及其特点，对比了鲁棒性、快速收敛性和对准精度等主要指标，并具体描述了 LM-ICP 算法的基本思想。

结　论

物体识别一直是计算机视觉的一个重要方向，如何快速而准确的识别出物体的形状与方位是研究的主要课题。本文提出的算法对一些简单物体实现了较为稳定的匹配，主要完成了以下工作。

1. 分析总结了 3-D 物体识别系统中的传感器类型、三维物体表达方法和匹配方法等问题。
2. 提出了一种有效计算 3-D 物体刚性变换参数的方位确定算法，即对偶四元数法。分析了该算法的理论基础及用于 3-D 物体定位参数估计的解决方法，并通过实验检验了该算法的性能、精度和速度。
3. 提出了采用 k-D 树的方法加速了搜索最近点的过程的思想；用在距离统计的基础上设置的最大距离容限限制伪点匹配；选用对偶四元数法计算刚体的旋转和平移；引入迭代终止条件等实际应用中需要考虑的几个主要问题，使算法可以更有效的应用到 3-D 物体识别。
4. 总结 ICP 算法的派生形式及其性能上的主要差别，提出了 LM-ICP 算法的基本思想。

3-D 物体识别是诸多工程领域都需要解决的基本问题之一，在目标识别、导航、医疗等领域都具有广阔的应用前景。因此，在未来一段时间内还有以下几个方面的工作需要深入研究：

1. 物体表面点的三维坐标描述。很明显，表现物体特征的采样点越多越密物体的形状表述越精确，但是，对于形状复杂的自由形态物体用采样点表述物体形状，必然存在某些物体上的采样点在模型点中找不到与之对应的最近点或者找到错误的对应点，导致匹配效果变差。
2. 尺度归一化及坐标系问题。初始情况下，需要将物体与模型的尺度归一化，并放置在同一坐标系内，坐标系的原点设置要保证物体形状的有效识别，这些问题涉及传感器类型、图像表示方法等诸多因素。
3. 初始值的获取问题。目前的算法都对初始值范围作了限定，即物体相对模型变化很小，或者粗略已知方位的初始值（坐标和方向）。事实上，这些先验知识如何获取还需要深入研究，这也在很大程度上决定算法是收敛到

局部最优还是全局最优。

本文最后搜集了许多 ICP 算法的派生形式，它们都是针对不同的实际需要对 ICP 算法的改进。在 3-D 物体识别的进一步研究中可以借鉴。我们正在考虑将算法扩展到曲面匹配，它属于立体视觉中的立体匹配问题。

参考文献

[1] 马颂德，张正友. 计算机视觉—计算理论与算法基础. 北京：科学出版社.1998.

[2] Jain A K, Flynn P J (Eds). 3D object recognition systems [M]. Amsterdam: Elsevier Science Publishers B.V. .1993.

[3] Arman F, Aggarwal J K. Model-based object recognition in dense range images [J] A review. ACM Computing Surveys, 1993, 25(1): 5~43P.

[4] Suk M, Bhandarkav S M. Three-dimensional object recognition from range images. Tokyo: Springer-Verlag, 1992

[5] 李庆，周曼丽，柳健. 三维物体识别研究进展. 中国图像图形学报. Vol. 5(A), No.12. Dec 2000.

[6] Hoover A. The space envelope representation for 3D scenes. PhD. Dissertation. Univ. of South Florida. 1996.

[7] 李松涛. 深度图像处理方法的研究[博士论文]. 北京：清华大学. 1999.

[8] Jurie F. Robust hypothesis verification: application to model-based object recognition. PR. 1999,32: 1069~1081P.

[9] Gandhi T, Camps O. Robust feature selection for object recognition using uncertain 2d image data. In: IEEE Conference on Computer Vision and Pattern Recognition, Seattle Washington. 1994:281~287P.

[10] Nagao K, Grimson W E L. Object recognition by alignment using invariant projections of planar surfaces. MIT: AI Lab Tech. Repot. 1994.

[11] 王新华. 三维工业物体识别与测量方法的研究[博士论文]. 武汉：武汉测绘科技大学. 1997.

[12] Yi J H, Chelberg D M. Model-based 3D object recognition using Bayesian indexing. CVIU. 1998,69(1): 87~105P.

[13] Grimson W E L, Peres T L. Localizing overlapping parts by searching the interpretation tree. ITPAMI. 1991,13(9): 920~934P.

[14] Ikeuchi K, Hong K S. Determining linear shape change: Toward automatic generation of object recognition programs. CVGIP. 1991,53(2): 154~170P.

[15] Umasuthan M, Wallace A M. Model indexing and object recognition using 3D viewpoint invariance. PR. 1997,30(9): 1415~1434P.

[16] Grimson W E L, Huttenlocher D P. On the verification of hypothesized matches in model-based recognition. ITPAMI. 1991,13(12): 1201~1213P.

[17] Ki A et al. Recognition and shape synthesis of 3D objects based on attributed hyper graphs. ITPAMI. 1995,17(12): 1228~1233P.

[18] Kao C Y, Kumara S R T, Kasturi R. Extraction of 3D object features from SAD boundary representation using the super relation graph method. ITPAMI. 1995,17(12): 1228~1233P.

[19] Dorai Ch. COSMOS: A framework for representation and recognition of 3D free-form objects. PhD. Dissertation, Michigan State Univ. 1997.

[20] Stockman G C. Object recognition and localization via pose clustering. CVGIP. 1987,40:361~387P.

[21] Grimson W E L, Huttenlocher D P. On the sensitivity of the bough transform for object recognition. ITPAMI. 1990,12(3): 255~274P.

[22] Stein F, Medioni G. Structural indexing: Efficient 3-D object recognition. ITPAMI. 1992,14(2): 125~145P.

[23] Flynn P J, Jain A K. 3D object recognition using invariant feature indexing of interpretation tables. CVGIP: IU. 1992,55(2): 119~129P.

[24] Wheeler M, Ikeuchi K. Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition. ITPAMI. 1995,17(3): 125~145P.

[25] Chen J L, Stockman G C. 3D free-form object recognition using indexing by contour features. CVIU. 1998: 535~540P.

[26] Solina F, Bajcsy R. Recovery of parametric models from range images: The case for super quadrics with global deformations. ITPAMI. 1990,12(2): 131~147P.

[27] Besl P, Mckay N. A method for registration of 3-D shapes. ITPAMI. 1992,14(2): 239~256P.

[28] Zhang Z. Iterative point matching for registration of free-form curves and surfaces. IJCV. 1994,13(2): 119~152P.

[29] 方昭江. 基于模型的三维物体匹配方法研究[硕士论文]. 武汉：华中理工大学

图像所. 1999.

[30] Chen T W, Lin W Ch. A neural network approach to CSG-based 3D object recognition. ITPAMI. 1994,16(7): 719~726P.

[31] Zha H B, Nanamegi H, Nagata T. Recognition 3-D objects by using a Hopfield-style optimization algorithm for matching patch-based descriptions. PR. 1998,31(6): 727~741P.

[32] Ham Y K, Park R H. 3D object recognition in range images using hidden Markov models and neural networks. PR. 1999,32:729~742P.

[33] K S Arun, T S Huang, S D Blostein. Least-squares fitting of two 3-D point sets. IEEE Trans. Pattern Anal. Mach. Intelligence PAMI-9. 1987,698~700P.

[34] W K Clifford. Preliminary sketch of bi-quaternion. J. London Math. Soc. 4, 1873, 381~395P.

[35] O D Faugeras and M Herbert. A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces. In Proceedings, Eighth International Conference on Artificial Intelligence. August 1983., pp 996~1002P.

[36] W Eric, L Grimson, Tomas Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. Int. J. Robotics Res. Fall 1984,3~35P.

[37] Steven J Gordon, Warren P Seering. Real-time part position sensing. IEEE Trans. Pattern Anal. Mach. Intelligence PAMI-10. 1988.

[38] Kristjan T Gunnarsson, Optimal Part Localization by Database Matching with Sparse and Dense Data. PhD. Dissertation. Dept. of Mechanical Engineering. Carnegie Mellon Univ. May 1987.

[39] Wahba Grac. A least squares estimate of satellite attitude. Problem 65.1. SIAM Rev. July 1966. 384~386P.

[40] J Keat. Analysis of Least-Squares Attitude Determination Routine DOAOP. Technical Report. Comp. Sci. Corp. CSC/TM-77/6034. February 1977.

[41] S Linnainmaa, D Harwood, L S Davis. Pose determination of a three-dimensional object using triangle pairs. IEEE Trans. Pattern Anal. Mach. Intelligence PAMI-10. 1988.634~648P.

[42] Edward Pervin, Jon A Webb. Quaternion in Computer Vision and robotics.

Technical Report. Dept. of Computer Science. Carnegie-Mellon University. CMU-CS-82-150. 1982.

[43] W S Rutkowski, R Benton, E W Kent. Model-driven determination of object pose for a visually served robot. In IEEE International Conference on Robotics and Automation. Raleith. North Carolina. 1987, pp 1419~1428.

[44] E Study. Geometric der Dynamen. Leipzig. 1903.

[45] I M Yaglom. Complex Numbers in Geometry. Academic Press. New York. 1968.

[46] A T Yang, F Freudenstein. Application of dual-number quaternion algebra to the analysis of spatial mechanisms. Trans. ASME. June 1964,300~308P.

[47] 曾接贤，陈震，储珺，黎明. 基于形态图表示的三维物体识别算法. 计算机应用. Vol.23, No.2, Feb 2003.

[48] 郑南宁，胡超，王庆元. 基于几何模型的三维物体识别算法的研究. 模式识别与人工智能. Vol.4, No.4, Sep 1991.

[49] 张宗华，彭翔，胡小唐. ICP 方法匹配深度图像的实现. 天津大学学报. Vol.35, No.5, Sep 2002.

[50] L Robert, O Faugeras. Curve-based stereo: Figural continuity and curvature. In Proc. IEEE Conf. Computer. Vision Pattern Recog. (Maui, Hawaii), pp 57~62. June 1991.

[51] R E Sampson. 3D range sensor-phase shift detection. Computer. No 20, pp 23~24. 1987

[52] J E W Mayhew, J P Frisby. P Frisby. Psychophysical and computational studies towards a theory of human stereopsis. Artif. Intell. Vol. 17, pp 349~385. 1981.

[53] S Pollard, J Mayhew, J Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. Perception, vol.14, pp 449~470. 1985.

[54] Z Zhang, O Faugeras, N Ayache. Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. In Proc. Second Int'l Conf. Comput. Vision (Tampa, FL), pp 177~186. IEEE. Dec 1988.

[55] O Faugeras, M Hebert. The representation recognition, and location of 3D shapes from range data. Int'l J. Robotics Res. Vol.5, no.3, pp 27~52. 1986.

[56] M W Walker, K Shao, R A Volz. Estimation 3-D location parameters using dual

number quaternion. CVGIP: Image Understanding. Vol.54, pp 358~367. Nov 1991.

[57] F Preparata, M Shamos. Computational Geometry. An Introduction. New York: Springer. Berlin. Heidelberg. 1986.

[58] 张斌,朱正中,葛成辉,吴佑寿.一种基于边缘轮廓信息的多源遥感图像子象素匹配定位方法.中国图像图形学报.2000,(5):830-835 页.

[59] 唐泽圣. 三维数据场可视化. 清华大学出版社. 1999.

[60] 李志林,朱庆.数字高程模型.武汉测绘科技大学出版社.2000.

[61] Z. Zhang and O. Faugeras, "Determining motion from 3D line segments: A comparative study," Image and Vision Computing, vol. 9, pp.10-19, February 1991.

[62] K. Arun, T. Huang and S. Blostein, "Least-squared fitting of two 3-D point sets," IEEE Trans. PAMI, vol. 9, pp. 698-700, September 1987.

[63] Z. Zhang and O. Faugeras, "Three-dimensional motion computation and object segmentation in a long sequence of stereo frames," Int'l J. Comput. Vision, March 1992。

[64] 刘承香. 水下潜器的海底地形匹配辅助定位技术研究 .哈尔滨工程大学博士学位论文. 2003.

[65] 孙枫, 袁赣南, 张晓红. 组合导航系统. 哈尔滨工程大学出版社. 哈尔滨, 1996.

[66] Eggert, D. W., Lorusso, A., and Fisher, R. B. Estimating3-D Rigid Body Transformations: A Comparison of Four Major Algorithms. MVA, Vol. 9, No. 5/6, 1997.

[67] Andrew W. Fizgibbon. Robust registration of 2D and 3Dpoint sets. In Proc. BMVC, pages 411-420, Sep 2002.

[68] David A. Forsyth and Jean Ponce. Computer Vision --- A Modern Approach. 北京：清华大学出版社. 2002, 10.

[69] 何光渝. Visual C++ 常用数值算法集. 北京：科学出版社. 2002.

[70] 勒中鑫. 数字图像信息处理. 北京：国防工业出版社. 2003, 1.

攻读硕士学位期间发表的论文和取得的科研成果

[1] Qu Hao, Liu Fan-ming. Methods for Registration of 3D Point Sets: An Evaluation ICCSE 2003

[2] 刘繁明，屈昊. ICP 算法的鲁棒性改进. 仪器仪表学报. 2004 年第 4 期(增).

致 谢

衷心感谢我的导师孙枫教授在我攻读硕士学位期间在学习和生活上对我的帮助和关怀以及在课题研究过程中的悉心指导。孙老师思想活跃、思路开阔、知识渊博，为人和蔼可亲、热情大方，工作认真负责、勤勉刻苦，做学问严谨求实、勇于创新，这些都是我学习的榜样，使我受用终身。

在整个课题的研究过程中，得到了407教研室刘繁明研究员全面、具体的指导。刘繁明老师给出了很多有益建议和热情帮助，刘繁明老师朝气蓬勃、求实奋进、勇于开拓、敢于创新，他是我的良师益友，他的敬业精神和工作能力是我一生的楷模。

在此论文完成之际，特向刘老师表示衷心的感谢！

衷心感谢407教研室其他老师的指导和帮助，他们严谨的实验态度、开拓灵活的思维对我能很好的完成课题起到了很大的帮助，在此十分感谢老师所给予的指导与帮助。

衷心感谢课题组成员们：成怡、赵旭、赵亚凤、顾兵、吴云峰，与这些优秀的研究生们共同的学习和研究使我学到了很多的东西，与他们相处的两年时间是我人生路上令人难忘的一段时光。

最后，感谢自动化学院领导和老师对作者热情的关心和帮助，感谢我的母校哈尔滨工程大学多年来对我的教育和培养。感谢所有关心和帮助过我的人们。