

摘 要

随着科技的日新月异，越来越多的高科技产品深入人们的日常生活之中。生活中难免会遇到设备使用方法不明、设备故障等情况。而普通人往往很难应对可能遇到的所有情况，且相关方面的专家也不能保证随叫随到，因此远程指导系统有着巨大的市场需求。而目前大多数远程指导系统仅局限于对二维平面进行相关操作，这种方式对于一些没有相关经验的人员来说不太直观。也有部分结合了增强现实技术的远程指导系统能起到不错的效果，但这些系统往往需要搭配昂贵的设备进行使用。

因此，本文设计并实现了一个结合增强现实技术并能够部署在移动端使用的在线远程指导系统。该系统通过结合增强现实技术，使用户可以采取结合物体 3D 模型对所遇到的问题进行详细阐述，专家也可以通过 AR 特效与 3D 模型相结合的方式对用户进行专业指导。该系统通过将物体与指导动作可视化的方式实现远程指导，大大地提高了远程指导的效率。增强现实技术需要实时获得指定物体的与相机间的位姿关系，因此增强实现技术的应用还需要基于实例级别的刚体 6D 位姿估计算法的帮助。

通常将刚体 6D 位姿估计算法分为定位算法与跟踪算法两种。所谓其中定位算法往往拥有较高的精度和较慢的推断速度，这对于具有实时性的增强现实技术而言是无法接受的。而跟踪算法满足实时性的要求，但目前基于实例级别的 6D 位姿跟踪算法基本上采用 RGB-D 数据作为输入，这显然对系统的使用设备有着极高的局限性。因此本文结合知识蒸馏的思想，提出了一种使用 RGB 图像作为输入数据的面向增强现实应用的定位—跟踪主从网络。通过对定位算法的网络进行知识蒸馏，得到一个参数量少的网络用于刚体的 6D 位姿跟踪。为了网络的训练与测试，本文构建了一个由四个物体数十个场景组成的 RGB 图像数据集。

本文基于 Unity3D、Visual Studio Code 与 Xcode 对系统前端、端到端通信与系统功能进行了实现。由于跟踪需要给予初始位姿并且是一个累计误差的过程，因此本文结合 C++ 与 Socket 技术搭建了一个用于部署定位网络模型的服务器。通过用户端与服务器端的信息传输，起到给予初始位姿与矫正跟踪误差的作用。

通过实验证明，本文提出的主从网络在刚体 6D 位姿估计上有着较高的精度，并且跟踪网络在 PC 端上平均推断速度可达 129FPS，在移动端上平均推断速度可达 25FPS，已充分满足实时性的要求。

关键词：知识蒸馏；远程指导系统；增强现实；刚体 6D 位姿估计

Abstract

With the rapid development of technology, an increasing number of high-tech products have penetrated into people's daily lives. In life, it is inevitable to encounter the unclear use method of equipment, equipment failure and other situations. However, it is often difficult for ordinary people to cope with all the situations they may encounter, and relevant experts are not guaranteed to be on call, so there is a huge market demand for remote guidance systems. And at present most remote guidance systems limited is associated with the two-dimensional plane, this way for some without the personnel of relevant experience is not very intuitive. There are also some remote guidance systems that incorporate augmented reality to good effect, but these systems often require expensive equipment to use.

Therefore, this paper designs and implements an online remote coaching system that incorporates augmented reality technology and can be deployed for use on mobile. The system works through the use of augmented reality technology. The user can combine the object 3D model to elaborate the problem encountered. Experts can also conduct professional guidance to users through the combination of AR special effects and 3D models. The system enables remote guidance by visualizing objects and guided movements, greatly improving the efficiency of remote guidance. Augmented reality technology requires real-time acquisition of the postural relationship between a given object and the camera, so the application of augmented reality technology also requires the help of the rigid object 6D pose estimation algorithm based on the instance level.

The rigid object 6D pose estimation algorithms are usually divided into two types of algorithms: location algorithms and tracking algorithms. Between them, the positioning algorithm has higher accuracy and slower inference speed, which is unacceptable for the real-time augmented reality technology. The tracking algorithm satisfies the real-time requirement, but the current instance-level 6D-based pose tracking algorithm basically uses RGB-D data as input, which is obviously extremely high limiting for the equipment used in the system. Therefore, a position-tracking master-slave network for augmented reality applications using RGB images as input is proposed in this paper. Therefore, this article proposes a master-slave network for locating-tracking applications in augmented reality using RGB images as input data, combining the idea of knowledge distillation. By distilling knowledge from the locating

network, a smaller network is obtained for rigid object 6D pose tracking. To train and test the network, this article constructs an RGB image dataset consisting of four objects and tens of scenes.

Based on Unity3D, Visual Studio Code, Xcode and other platforms, the front-end, end-to-end communication and system functions are implemented in this paper. As tracking requires an initial pose and is a process of accumulating errors, this paper combines C++ and socket technology to build a server for placing the locating network model. The initial pose is given and the tracking error is corrected by transferring information from the user to the server.

The experiments carried out demonstrate that the master-slave network proposed in this paper achieves high accuracy in rigid object 6D pose estimation. Furthermore, the tracking network runs at an average inference speed of 129FPS on PC and 25FPS on mobile devices, which fully meets the real-time requirements.

Key words: knowledge distillation; remote guidance system; augmented reality; rigid object 6D pose estimation

目 录

摘 要.....	I
Abstract.....	II
目 录.....	IV
1. 引 言.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 增强现实技术.....	2
1.2.2 AR 在线远程指导系统.....	2
1.2.3 刚体 6D 位姿估计算法.....	4
1.3 论文的主要工作.....	5
1.4 章节安排.....	5
2. 相关理论介绍.....	7
2.1 刚体 6D 位姿估计.....	7
2.1.1 刚体的空间运动.....	7
2.1.2 相机模型.....	8
2.2 知识蒸馏算法.....	9
2.3 PnP 问题.....	10
2.3.1 DLT 算法.....	11
2.3.2 P3P 算法.....	11
2.3.3 EPnP 算法.....	11
2.3.4 BA(Bundle Adjustment)算法.....	12
2.4 点匹配算法.....	12
2.4.1 SIFT 算法.....	12
2.4.2 ORB 算法.....	14
2.4.3 SuperPoint 算法.....	15
2.5 本章小结.....	16
3. 面向增强现实应用的定位—跟踪主从网络.....	17
3.1 网络结构与损失函数.....	17
3.1.1 网络结构.....	17
3.1.2 损失函数与训练策略.....	19

3.2 知识蒸馏.....	19
3.3 实验结果与分析.....	22
3.3.1 数据集.....	22
3.3.2 实验环境与准备.....	23
3.3.3 评价指标.....	24
3.3.4 实验结果与分析.....	25
3.4 本章小结.....	29
4. AR 在线远程指导系统设计与实现.....	30
4.1 需求分析.....	31
4.1.1 系统功能性需求.....	31
4.1.2 系统非功能性需求.....	32
4.2 系统总体设计.....	33
4.2.1 系统架构设计.....	33
4.2.2 系统功能设计.....	34
4.3 系统实现.....	35
4.3.1 系统运行环境与开发平台.....	35
4.3.2 定位功能模块.....	36
4.3.3 跟踪功能模块.....	37
4.3.4 通信功能模块.....	37
4.3.5 AR 功能模块.....	38
4.4 系统展示.....	39
4.4.1 咖啡机指导画面.....	42
4.4.2 雅迪电动车指导画面.....	45
4.5 本章小结.....	47
5. 总结与展望.....	49
5.1 总结.....	49
5.2 展望.....	50
参考文献.....	51
致 谢.....	57
在读期间公开发表论文（著）及科研情况.....	58

1. 引言

1.1 研究背景及意义

随着科技的迅速发展,各种类型的机器、设备的结构与设计都变得较为复杂。一些大型的机器,通常采用多种系统结构组成不同单元并相互合作^[1]。这就给设备的使用与维修带来了一定的困难,此前往往通过预约专业人士进行线下指导的方式解决这些问题^[2]。由于近年来疫情所造成的影响,人们的出行受到了一定程度上的影响,专业人士也无法保证能够到达现场进行指导。而通过语音、视频的方式向专家求助,又需要求助者拥有一定的专业知识,否则可能由于对问题描述模糊不清导致解决方案产生偏差。因此一种合适有效的远程指导办法具有极大的市场需要及发展前景。

增强现实 (Augmented Reality, AR) 技术^[3]是一种将真实世界信息和虚拟世界信息“无缝”集成的新技术,这种技术的目标是在屏幕上把虚拟世界套在现实世界并进行互动^[4]。通过对相关物体进行建模与动画设计,便可以将用户对物体的疑问或对物体的详细操作展示在真实世界中,通过三维世界的演示降低了语言表达的歧义。由于 AR 技术的实时交互的特性^[5],使其在远程指导方面有着广阔的前景。结合 AR 技术的远程指导系统,可以使用户在远程场景中实时获取指导者的指导与反馈,帮助用户准确、快速地完成工作,从而提高工作效率;可以通过远程指导与服务的方式,避免服务者和用户在距离和时间上的限制,达到提高服务质量和效率、减少人力和物力资源的浪费、降低运营成本的目的;可以促进不同地区、行业、企业之间的知识共享与技术交流,推动行业的创新发展;还可以使用户在远程环境中进行虚拟展示和交互,提高用户的体验与参与感。因此 AR 在线远程指导系统的研究有着非常重大的意义,能够给各行各业带来许多积极的影响与经济价值。

在现实生活中,相机可能会移动或旋转,因此为了保证所设计的 3D 模型或动画的位置不出现偏移,增强现实技术的应用还需要结合刚体 6D 位姿估计算法的帮助。目前的刚体 6D 位姿估计算法从任务上可以分为两类:定位与跟踪。所谓定位,是指对物体与相机间的位姿进行估计;所谓跟踪,是指对相机运动前后的位姿进行估计。定位算法往往拥有较高的精度但算法的运行速度较慢。而增强现实技术具有实时性的特征,因此无法容忍低帧率的定位算法。跟踪算法的运行速度能够达到实时的要求,但跟踪需要给予初始位姿并且是一个累计误差的过程。所以跟踪算法的正常使用,还需要定位算法对其进行初始位姿的给予与跟踪误差

的矫正。因此定位算法与跟踪算法的结合对于增强现实技术来说十分重要。

1.2 国内外研究现状

1.2.1 增强现实技术

增强现实（Augmented Reality, AR）技术是一种将真实世界信息和虚拟世界信息“无缝”集成的新技术。它通过计算机图形学、计算机视觉、传感器技术等技术手段，将计算机生成的虚拟信息与真实场景相融合，从而使用户感知到虚拟信息与现实场景的统一。AR技术的核心就是让用户通过某种设备，如手机、平板电脑、头戴式显示器等，通过摄像头或其他传感器，实时获取真实世界的信息，并将计算机生成的虚拟信息与真实世界的信息融合，并以一种自然的方式显示在用户的视野之中。简单来说，AR技术就是将数字内容叠加到真实世界之中，使用户能够看到和感觉到超越现实的内容^{[6][7]}。

增强现实技术在国外已有多年的研究与应用。早期的研究主要集中于AR技术的开发与改进方面，如AR系统的跟踪、画面渲染与通信交互等方面。近年来，随着AR技术的研究趋势逐渐向着多学科交叉的方向发展，如计算机视觉、机器学习、人机交互等方面。国内对于AR技术的研究起步较晚，但近年来发展十分迅速。目前AR技术主要应用于游戏^[8]、教育^[9]、医疗^[10]、艺术^[11]、建筑^[12]等领域。

美国、日本两国的AR技术被广泛应用于游戏开发，如《Pokemon Go》、《Ingress》、《怪物猎人》、《世界名画AR》等制作精良的AR游戏均来自于美、日两国。德国主要将AR技术应用于汽车、航空航天、制造等领域，如奔驰公司在汽车生成过程中，会利用AR技术进行零部件的拼装、虚拟试车等操作。而在我国，增强现实技术被广泛应用于教育领域，如AR教学、AR图书等应用被广泛推广。北京市某小学已利用AR技术对学生开展了AR教学活动，让学生在虚拟环境中更好地学习。

此外，随着AR技术应用场景的不断增多，AR技术所带来的问题也逐渐浮现在公众面前。如何提高AR应用的用户体验、如何保证AR应用的安全性等问题已逐渐成为社会的关注热点^[13]。

1.2.2 AR 在线远程指导系统

远程指导在医疗方面^{[14][15][16]}有着极高的实用价值，因此国内外的远程指导研究最初多是针对远程医疗方面，远程医疗方面多是通过传输图片或视频通信的形式来进行远程指导。随着科技的进步和设备的复杂程度的上升，国内外也有不少生产厂家发明了一些针对设备维修与使用的在线远程指导系统^[17]。

国内相关研究如华为的 HiLeiaAR，该系统基于 AR 眼镜，借助网络进行移动通信传输。该系统的使用需要前端使用者佩戴 AR 眼镜，通过眼镜上的摄像头采集第一视角画面。该系统在音视频通讯基础上，结合物体识别与空间定位等技，通过传输文字、图片、语音和标记来实现 AR 指导。该系统需要配套的 AR 眼镜才能实现相应功能，且成本较高，目前该系统按照可在线数来进行标价，其价格在 8.5 万/年到 40 万/年。华为 HileaiAR 如图 1-1 所示。



图 1-1 华为 HiLeiaAR 效果图

国外相关研究如微软的 Microsoft Dynamics 365 Remote Assist，该系统主要针对一线工作者远程协助场景的量体开发。该系统优势在于可以跨设备协助，对维修人员与相关专家的设备并无特殊规定。双方既可以通过佩戴 HoloLens 眼镜，又可以借助手机、平板等移动设备来进行远程指导。该系统的通信方式是通过结合 Teams 实现。目前该系统具有视频通话、多方讨论、混合现实标记和共享文件等功能。Microsoft Dynamics 365 Remote Assist 如图 1-2 所示。



图 1-2 微软 Microsoft Dynamics 365 Remote Assist 效果图

随着近年来 AR 技术的蓬勃发展与疫情造成的影响, AR 在线远程指导系统已成为当下十分火热的研究方向。但目前大部分 AR 在线远程指导系统都需要搭配特定的设备进行使用, 如 AR 眼镜、AR 头盔等。这些昂贵的设备在经济层面上给用户带来了不小的压力。

1.2.3 刚体 6D 位姿估计算法

刚体 6D 位姿估计算法被广泛应用于机器人抓取^{[18][19]}、无人驾驶^[20]、增强现实等领域。目前主流的刚体 6D 位姿估计算法主要基于以下三种方法: 基于对应点、基于模板和基于投票^[21]。

① 基于对应点的方法是对输入数据与完整 3D 模型之间对应关系查找的方法。当基于 RGB 图像解决问题时, 通过寻找 3D 模型的 2D 像素与 3D 点之间的对应关系并结合 PnP 算法计算刚体的 6D 位姿。当基于 RGB-D 图像解决问题时, 通过所得到的部分视图的 3D 点云与完整 3D 模型之间的 3D 点的对应关系并利用 ICP 算法进行刚体 6D 位姿的计算。其代表算法有: SIFT^[22]、ORB^[23]、Dpod^[24]、3DFeatNet^[25]等。

② 基于模板的方法通过对当前输入和已带有 6D 位姿的模板进行相似性匹配从而得出 6D 位姿。在 2D 情况下, 模板可以从已知的 3D 模型投影成 2D 图像, 并且模板中的对象在相机坐标系下具有相应的 6D 位姿。因此, 刚体 6D 位姿估计问题被转化为图像检索问题。在 3D 情况下, 模板可能是目标对象的 3D 点云, 此时需要找到最佳的 6D 位姿将部分点云与模板保持一致, 此时刚体 6D 位姿估计成为一个从部分到整体的位姿粗略估计问题。其代表算法有: Deep-6DPose^[26]、NOCS^[27]、TEASER^[28]等。

③ 基于投票的方法是对每个像素或 3D 点通过一个或多个选票来构成刚体 6D 位姿估计。投票方式大致分为间接投票和直接投票两种。间接投票的方法是将每个像素或 3D 点对某些特征点进行投票, 这些特征点提供 2D-3D 点对关系或 3D-3D 点对关系。直接投票方法是将每个像素或 3D 点对某个 6D 位姿进行投票。其代表算法有: PVNet^[29]、PVN3D^[30]、6-PACK^[31]、DenseFusion^[32]等。

对于 AR 在线远程指导系统而言, 往往是针对指定的物体进行指导, 因此所使用的刚体 6D 位姿估计算法应是基于实例级别。根据任务的不同, 刚体 6D 位姿估计算法可分为定位与跟踪两个方向。定位是针对刚体与相机间绝对位姿的求解, 跟踪针对刚体运动前后的相对位姿的求解, 需要获取到刚体的初始位姿并且是一个累计误差的过程。其中基于实例级别的刚体 6D 位姿定位算法在为输入数据 RGB 或 RGB-D 图像上均有许多开源算法。而目前基于实例级别的主流刚体 6D 位姿跟踪算法如: DenseFusion^[32]、PVN3D^[30]和 FFB6D^[33]等皆是使用 RGB-D 图像作为输入数据, 而采用 RGB 图像作为输入数据的算法 Deep-6DPose^[26]目前

并未开源。

1.3 论文的主要工作

现有的 AR 在线远程指导系统，大多是以通过对视频进行标记的方式来实现远程指导，相比于传统的文字加视频的方式有着一定优势，但是对复杂的问题依旧不能有效地进行解决。也有部分 AR 在线远程系统采用对三维世界添加 AR 特效的方式进行远程指导，但这些系统往往需要搭配价格高昂的设备才能进行使用。因此，为解决目前 AR 在线远程指导面临的低效率、高成本的问题。本文设计并提出了一种可以部署至智能手机或平板电脑上的 AR 在线远程指导系统。由于 3D 模型的引入，用户可以通过结合虚拟物体的方式对所遇到的问题进行详细的描述。专家也可以通过 AR 动画特效实现手把手教学，提高了远程指导的效率，也降低了用户错误操作的可能性。并且不需要额外购置特定设备，能够为用户减少额外开销。

针对 AR 技术中所使用的刚体 6D 位姿估计算法，考虑到若使用 RGB-D 图像作为输入的算法会极大程度上限制指导系统可应用的设备。因此本文结合知识蒸馏的思想，提出了一种采用 RGB 图像作为输入的面向增强现实应用的定位—跟踪主从网络。通过将位姿估计精准、推断速度较慢的定位网络通过知识蒸馏的方法训练一个位姿估计较为精准且推断速度较快的跟踪网络用于物体的跟踪。并且为进一步提高跟踪网络的推断速度，本文对网络的推断策略进行了适当调整，在后续章节中通过实验证明调整后的推断方式在速度上平均提升了约 85FPS，且精度只有略微下降。由于跟踪需要物体的初始位姿并且是一个累计误差的过程，因此本文结合 C++与 Socket 技术搭建了一个可以与用户端交互的服务器，用于部署定位网络模型，通过定位算法给予跟踪所需的初始位姿并对跟踪误差矫正。为了对本文所提出的主从网络进行训练与测试，还制作了一个由四个物体及数十个场景组成的 RGB 数据集。

在系统方面，本文基于 Unity3D 与 Visual Studio Code 对用户端与专家端的前端界面与业务逻辑进行了设计与实现，还实现了一些 AR 特效。并利用 C#语言与 Scket 技术实现了用户端与专家端之间的视频通信。随后在 Xcode 中，利用 Objective-c 语言对部署至移动端的跟踪网络模型进行调用。

1.4 章节安排

本文内容安排如下：

第一章：本章首先对 AR 在线远程指导系统的研究意义、市场需要与应用前景进行了分析。其次介绍了增强现实技术的发展状况及现有的 AR 在线远程指导

系统的相关情况及它们的不足之处。随后介绍了增强现实技术应用所需要的刚体 6D 位姿估计算法的常用研究方法与发展近况。最后基于现阶段的一些问题，提出了本文的改进措施与研究方法。

第二章：本章主要对后续章节中所涉及的相关理论知识进行介绍。本章首先从刚体的空间运动与相机模型两个角度对刚体 6D 位姿估计的原理进行了介绍。随后介绍了知识蒸馏算法的原理以及目前一些主流方法的思想。然后介绍了解决 PnP 问题的一些主流算法。最后介绍了一些较为主流的点匹配算法。

第三章：本章首先从网络结构、损失函数、正负样本采集方式以及训练与推断策略对本文提出的面向增强现实应用的定位—跟踪主从网络进行了详细介绍。随后介绍了跟踪网络训练时所用到的知识蒸馏方法。随后详细介绍了本文构建的 RGB 数据集与实验所采用的评价指标。最后通过列举对比实验的实验数据来验证本文所采取的策略的有效性、该算法在位姿估计方面在精度与速度方面的优异性以及跟踪算法在前端设备上的实时性。

第四章：本章从系统的需求分析、总体设计两个方面对 AR 在线远程指导系统的设计过程进行了详细介绍。随后对系统功能的设计与实现进行了详细描述。最后结合图片对 AR 在线远程指导系统的界面与功能进行了展示。

第五章：本章对本文所做工作进行了概括性总结，并分析了目前的不足之处，指出了未来的改进方向。

2. 相关理论介绍

本章节对后续章节中所涉及的相关理论进行介绍，主要对刚体 6D 位姿估计的相关原理、知识蒸馏算法、PnP 算法与一些主流的点匹配算法进行介绍。

2.1 刚体 6D 位姿估计

刚体 6D 位姿估计是指估计相机坐标系下物体的 6D 位姿，其中包含了 3D 位置和 3D 姿态。此时物体本身的坐标系被视为世界坐标系，所以刚体 6D 位姿估计即是得到物体所在世界坐标系到相机坐标系的 Rt 转换。由于刚体 6D 位姿估计可以得到刚体的准确位姿，因此它对机器人抓取、无人驾驶和增强现实等领域有着重要作用。本文将从刚体的空间运动及相机模型两个方面来介绍刚体 6D 位姿估计的相关原理。

2.1.1 刚体的空间运动

刚体是指在运动中，形状、大小以及内部各点相对位置不变的物体。刚体的空间运动保证了其坐标系下的向量的模长与夹角在运动前后不会发生改变，这种变化也被称为欧式变换^[34]。欧式变换由旋转和平移构成，其中对于旋转。我们假设某个单位正交基 (e_1, e_2, e_3) 经过一次旋转变成了 (e'_1, e'_2, e'_3) 。那么对于同一个向量 Q （该向量并未随坐标系运动而运动）而言，它在两个坐标系下的坐标分别为 $[q_1, q_2, q_3]^T$ 和 $[q'_1, q'_2, q'_3]^T$ 。由于向量本身并未改变，因此根据坐标定义有：

$$[e_1, e_2, e_3] \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = [e'_1, e'_2, e'_3] \begin{bmatrix} q'_1 \\ q'_2 \\ q'_3 \end{bmatrix} \quad (2-1)$$

为描述两个坐标的关系，我们在上述等式左右同时左乘 $\begin{bmatrix} e_1^T \\ e_2^T \\ e_3^T \end{bmatrix}$ ，则有：

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} e_1^T e'_1 & e_1^T e'_2 & e_1^T e'_3 \\ e_2^T e'_1 & e_2^T e'_2 & e_2^T e'_3 \\ e_3^T e'_1 & e_3^T e'_2 & e_3^T e'_3 \end{bmatrix} \begin{bmatrix} q'_1 \\ q'_2 \\ q'_3 \end{bmatrix} \stackrel{\text{def}}{=} RQ' \quad (2-2)$$

两组基之间的内积所组成的矩阵 R 表现了同一个向量旋转前后的坐标变化，因此我们把矩阵 R 定义为旋转矩阵。并且旋转矩阵是一个行列式为 1 的正交矩阵，因此旋转矩阵 R 拥有三个自由度。对于平移，我们通常用一个 3×1 的平移向量 t 来表示。因此一次欧式变换可以表示为：

$$Q' = RQ + t \quad (2-3)$$

由于这里的式(2-3)中所示欧式变换关系不是一个线性关系，所以该公式会在多次欧式变换后变得较为复杂。因此，我们引入齐次坐标，将式(2-3)改写为：

$$\begin{bmatrix} Q'_1 \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q_1 \\ 1 \end{bmatrix} \stackrel{\text{def}}{=} T \begin{bmatrix} Q_1 \\ 1 \end{bmatrix} \quad (2-4)$$

我们将旋转和平移写在一个四维矩阵之中，将该变换关系变成线性关系，简化了多次变换后的运算公式。在上式(2-4)中，我们将矩阵 T 称为变换矩阵。其中 R 与 t 分别拥有三个自由度，因此刚体在空间中的位姿估计被称为刚体 6D 位姿估计。

2.1.2 相机模型

在普通单目相机中，世界坐标系中的 3D 点通过相机的光心投影到相机的二维成像平面上，所有 2D 投影点构成了照片。习惯上我们让 x 轴向右、 y 轴向下、 z 轴指向相机前方，坐标原点 O 为相机光心构成相机坐标系 $O-x-y-z$ ，如图 2-1 所示(此图观看视角应为 z 轴正方向)^[35]。此时我们可以发现相机的成像模型与类似于小孔成像模型，如图 2-2 所示^[34]。

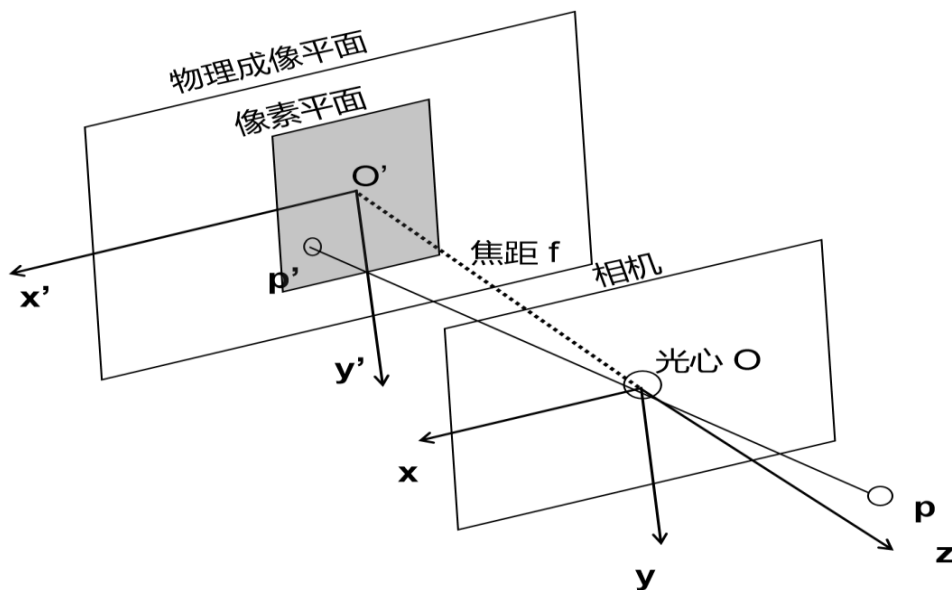


图 2-1 相机成像模型

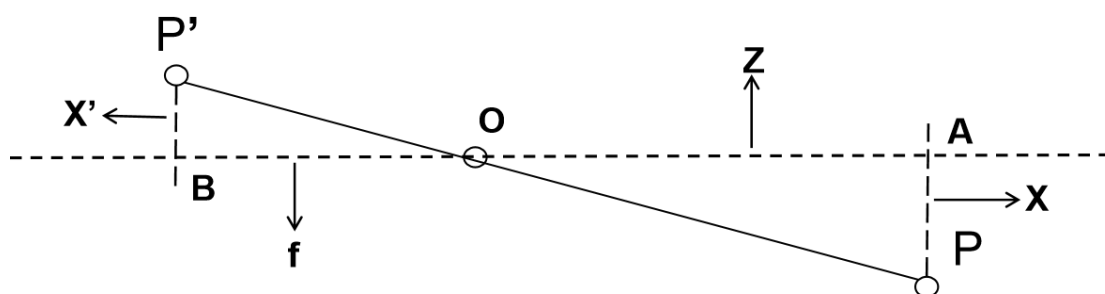


图 2-2 坐标相似三角形关系模型

我们假设现实世界的空间点 P 在相机坐标系下的坐标为 $\{X, Y, Z\}^T$ ，点 P 经过光心 O 投影之后在物理成像平面上的成像点 P' 的坐标为 $\{X', Y', Z'\}^T$ ，且相机焦距为 f 。根据三角形相似关系则有：

$$\frac{Z}{f} = -\frac{X}{X'} = -\frac{Y}{Y'} \quad (2-5)$$

其中负号代表图像是倒立的，而实际上相机图像并非倒像。因此我们等价地将成像平面放在相机前方，将公式中的负号去掉。根据像素坐标系与成像平面之间的关系，我们可以得知像素坐标与成像平面坐标相差了原点平移与缩放可以用如下公式表示：

$$\begin{cases} u = \alpha X' + C_x \\ v = \beta Y' + C_y \end{cases} \quad (2-6)$$

其中， α 和 β 分别表示在 u 轴与 v 轴的缩放倍数， C_x 和 C_y 表示原点的平移量。将代入公式(2-5)并用 f_x 与 f_y 表示 αf 、 βf 则我们可以得到空间点与相机像素平面的转换公式如下：

$$Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \stackrel{def}{=} KP \quad (2-7)$$

这里的矩阵 K 被称为相机内参矩阵，当需要使用到相机内参时，通常可使用张正友标定法^[36]进行内参标定。此时将公式(2-7)中的 P 点坐标相转换成为世界坐标系下的坐标并结合公式(2-4)，则有：

$$ZP_{uv} = Z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = KTP_w \quad (2-8)$$

在公式(2-8)中相机的内参 K 不会随相机运动而改变，因此求解相机的运动轨迹即是求解变换矩阵 T 。

2.2 知识蒸馏算法

知识蒸馏算法由 Hinton 等人于 2015 年提出^[37]，该算法主要用于解决多分类问题的预训练大模型的模型压缩问题。该算法中提出在 Softmax 函数中引入温度 T ，使得原始的硬标签“软化”。原始 Softmax 函数与修改后的 Softmax 函数如公式(2-9)与公式(2-10)所示。

$$Z_i = \frac{\exp(Z_i)}{\sum_{n=1}^i \exp(Z_i)} \quad (2-9)$$

$$Z_i = \frac{\exp\left(\frac{Z_i}{T}\right)}{\sum_{n=1}^i \exp\left(\frac{Z_i}{T}\right)} \quad (2-10)$$

在模型训练方面，引入了教师网络的概念。将训练数据同时输入教师网络与

学生网络，用教师网络生成的软标签与数据本身的硬标签指导学生网络训练。由于软标签相比于硬标签包含了更多信息，软、硬标签示意图如图 2-3 所示，因此学生网络可以从软标签的学到更多的“知识”增加学生模型的泛化能力。知识蒸馏网络结构如图 2-4 所示。

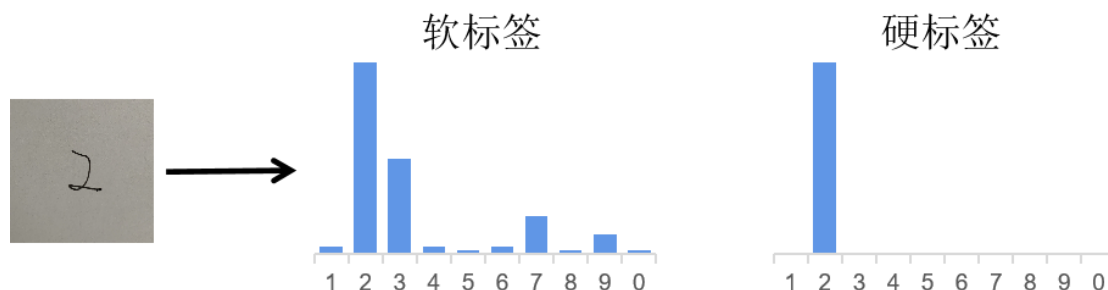


图 2-3 软硬标签示意图

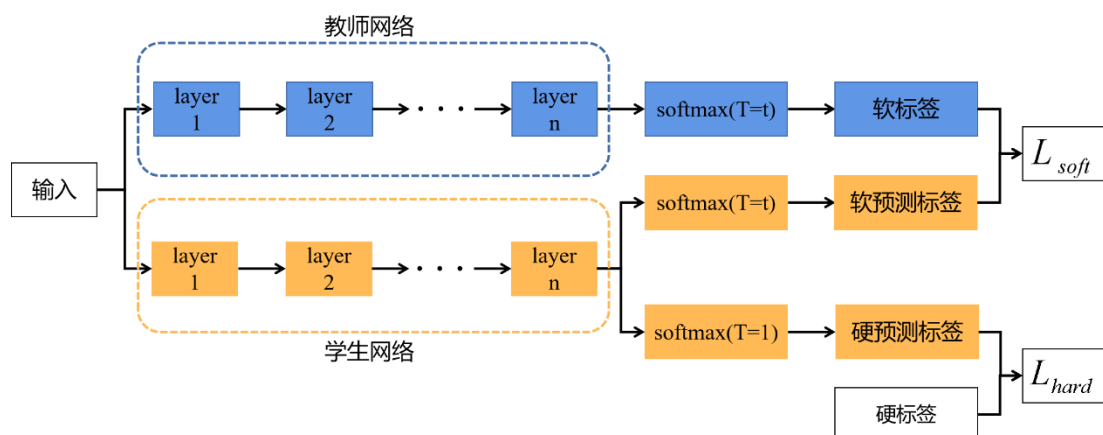


图 2-4 知识蒸馏结构图

由于知识蒸馏算法在多分类问题上的成功，在计算机视觉的其他方向上也纷纷借鉴了知识蒸馏的想法。如：2017 年 Zagorukyo 等人^[38]提出通过提取教师网络生成的注意力图作为“知识”指导学生网络训练。2019 年 Park 等人^[39]提出将多个教师模型的输出作为结构单元，结构单元更能体现出教师网络的结构化特征，使学生网络可以得到更好的指导。2019 年 Wang 等人^[40]提出复杂背景信息包含过多的噪声，导致全特征模仿会使学生模型效果下降，因此定位知识密集（Ground truth）的位置，并让学生模型在这些位置上模仿教师模型。2021 年 Shu 等人^[41]提出传统的知识蒸馏方法将逐个像素严格对齐会导致学生网络过度学习教师网络的输出，可能无法获得最优结果，因此提出对单个通道进行归一化处理并让学生网络学习这个归一化分布的方法。

2.3 PnP 问题

PnP 问题源自于相机的位姿估计，PnP 问题描述的是已知世界坐标系下的 n 个 3D 点坐标及这些 3D 点对应的 2D 像素坐标如何对位姿进行求解，目前主流

的用于解决 PnP 问题的算法有以下几种：

2.3.1 DLT 算法

DLT（直接线性变换）算法利用公式(2-8)所示的相机模型中所建立的线性方程对位姿进行求解。相机内参通常作为已知量，可以得知该方程中含有 12 个未知变量（其中旋转矩阵 R 含有 9 个，平移向量 t 含有 3 个），因此方程需要至少 6 对 2D-3D 点对便可进行求解。在 DLT 算法中，这些未知变量被视为互不相干的未知数，所求得参数可能不符合旋转矩阵的约束。因此还需要对最优旋转矩阵寻找一个符合条件的近似矩阵。

2.3.2 P3P 算法

P3P 算法通常需要 4 对 2D-3D 点对用来解决 PnP 问题，其中 3 对点对用来进行位姿求解，剩下的点对用来对解进行验证。利用 3 对点对可以得到三角形的相似关系如图 2-5 所示，通过构建方程可以得到相机坐标系下这些点的 3D 坐标，从而将问题转换为 3D-3D 的 ICP^[42]问题。但 P3P 算法无法利用更多的点对信息，且 P3P 算法鲁棒性较差，容易受到噪声影响，点对误匹配则会导致算法失效。

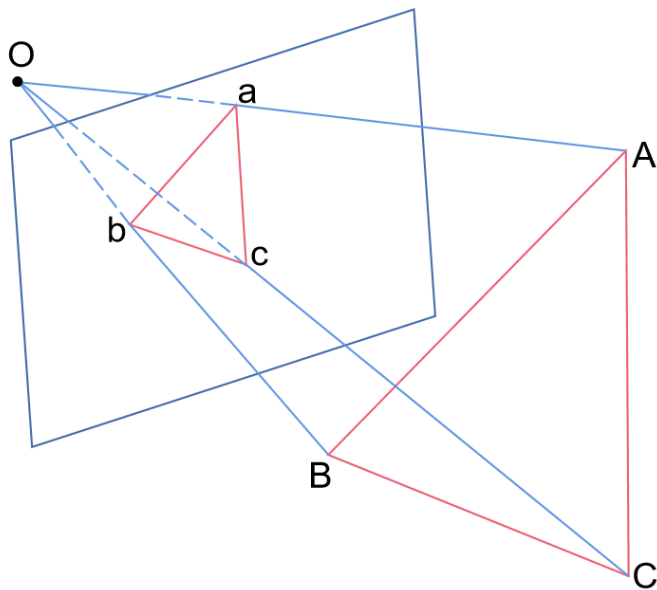


图 2-5 P3P 三角形相似关系图

2.3.3 EPnP 算法

EPnP 算法^[43]是目前使用较多的求解 PnP 问题的方法，其思想与 P3P 算法相似，均是先获得世界坐标系下 3D 点在相机坐标系下的 3D 坐标，从而将 2D-3D 问题转换为 3D-3D 问题并利用 ICP 算法进行求解。但 EPnP 算法利用了更多的点对关系，尽可能地减少噪声带来的影响。

EPnP 算法首先在已知世界坐标系下的 n 个 3D 点中通过 PCA 算法选择 4 个

不共面的控制点，并世界坐标系下的剩余 3D 点用这些控制点进行表示。随后利用相机投影模型与 2D 像素坐标，建立控制点在相机坐标系下 3D 坐标的线性方程组，解出控制点在相机坐标系下的 3D 坐标，从而解出所有 3D 点在相机坐标系下的 3D 坐标，随后利用 ICP 算法对 3D-3D 问题进行求解。

2.3.4 BA(Bundle Adjustment)算法

以上三种算法皆是采用线性方法进行位姿求解，线性方法通常将空间点的坐标视为已知量，通过构建方程组对相机位姿进行求解，而 BA 算法^[44]则将相机位姿与空间点坐标均视为优化变量一起进行优化，将 PnP 问题构造成一个非线性最小二乘问题进行求解。

首先假设空间中有 n 个 3D 点，其世界坐标系下坐标为： $P_i = \{X_i, Y_i, Z_i\}^T (i = 1, 2, \dots, n)$ ，对应的像素坐标系下的齐次坐标为： $P_i = \{u_i, v_i, 1\}^T (i = 1, 2, \dots, n)$ 。若相机位姿已知且观测点没有噪声，则有： $w_i[u_i, v_i, 1]^T = K T [X_i, Y_i, Z_i, 1] (i = 1, 2, \dots, n)$ 其中 w_i 表示深度， K 为相机内参， T 为世界坐标系到相机坐标系的变换矩阵。

由于相机位姿未知且观测点可能存在误差，利用 n 个观测点的重投影误差，并将误差项累加构建最小二乘问题，可以得到如下公式：

$$f(T, P_i) = \sum_{i=1}^n \left\| P_i - \frac{1}{w_i} K T P_i \right\|^2 \quad (2-11)$$

由于变换矩阵 T 中的旋转矩阵 R 受到一定约束，因此通过转换成无约束的李群进行求解，转换后的公式如下：

$$f(\xi, P_i) = \sum_{i=1}^n \left\| P_i - \frac{1}{w_i} K \cdot \exp(\xi^\wedge) \cdot P_i \right\|^2 \quad (2-12)$$

其中 ξ 表示旋转向量， $\xi^\wedge$ 表示旋转向量的矩阵表达， $\exp(\xi^\wedge)$ 表示向量到矩阵的对数映射。然后通过梯度下降的方法确定优化方向，并通过不断迭代求出 n 个点的重投影误差和的最小值对应的变换矩阵。

2.4 点匹配算法

点匹配的思想是本文第三章中所提网络流程中位姿计算的重要基础，因此本小节主要负责介绍目前主流的一些点匹配算法。

2.4.1 SIFT 算法

SIFT 算法^[45]于 1999 年由 David 提出并于 2004 年完善总结，该算法的实质是在不同的尺度空间上查找特征点，并计算出特征点的方向，由于该算法所查找

的特征点是一些十分突出，不因光照、噪声和仿射变换等因素而改变的点。如角点、边缘点等。因此 SIFT 算法在一定程度上可以解决以下因素：

- ① 目标物体的缩放、旋转与平移
- ② 图像仿射变换或投影变换
- ③ 光照
- ④ 遮挡
- ⑤ 噪声

对图像配准、目标识别及跟踪等算法性能的影响。

SIFT 算法中尺度空间的实现通过高斯金字塔表示，高斯金字塔是指通过对原始图像进行多次降采样所得来的塔状结构。为体现尺度的连续性，在高斯金字塔的每层加上了不同参数的高斯模糊。高斯模糊计算公式如下所示：

$$G(r) = \frac{1}{\sqrt{2\pi}\sigma} e^{-r^2/(2\sigma^2)} \quad (2-13)$$

在公式(2-13)中， σ 是正态分布的标准差， σ 的值与图像的模糊程度呈正相关。 r 是模糊半径，指模板元素到模板中心的距离。对于二维模板上的元素 (x, y) ，其对应的计算公式如下所示（ m 与 n 表示模板大小）：

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-m/2)^2 + (y-n/2)^2}{2\sigma^2}} \quad (2-14)$$

将高斯金字塔中相邻两层的进行特征差值计算，得到高斯差分金字塔然后对差分金字塔中的候选点进行极值检测。极值检测是针对某个点的上一层、下一层、同一层的共 26 个点进行比较，选取最大值作为极值点(关键点)。极值点检测图如下：

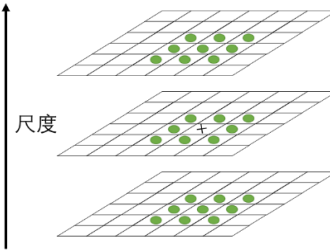


图 2-6 极值点检测图

为了使关键点描述子具有旋转不变性，采集差分金字塔中所检测出的关键点与其邻域窗口内像素的梯度与方向分布特征，梯度与方向公式如下(2-15)、(2-16)所示。

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (2-15)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (2-16)$$

并使用直方图统计邻域内像素的梯度与方向，并将直方图中的最大值代表的

方向作为关键点的主方向，这样便得到了 SIFT 的特征向量。关键点方向直方图如图 2-7 所示。

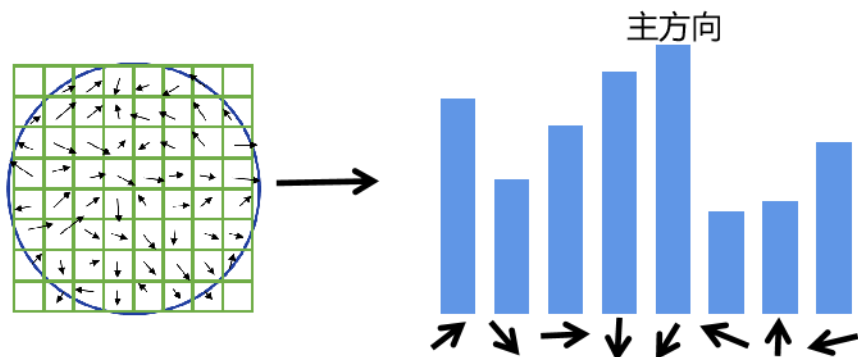


图 2-7 关键点方向直方图

2.4.2 ORB 算法

ORB 算法^[46]由 Rublee 于 2011 年提出，该算法分为特征点提取与特征点描述两部分。ORB 算法使用 FAST 算法进行特征提取，首先以各像素为圆心画一个半径为 3 像素的圆，如图 2-8 所示。其次判断圆周上是否有连续 n 点的灰度值大于或小于圆心像素点的灰度值，则认为该点是特征点（一般情况下将 n 设置为 12）。为了加快特征点的匹配，通常首先对 1、5、9、13 四个位置上的灰度值进行检测，如果没有 3 个或以上的像素点的灰度值大于或小于圆心的灰度值，则排除此点。上述步骤所得到的特征点可能存在特征点相邻的情况，为了解决这种现象，还需要计算并比较相邻特征点与周围 16 个像素点的差分之和。将差分之和较小的点剔除，这种算法也被称为非最大值抑制。

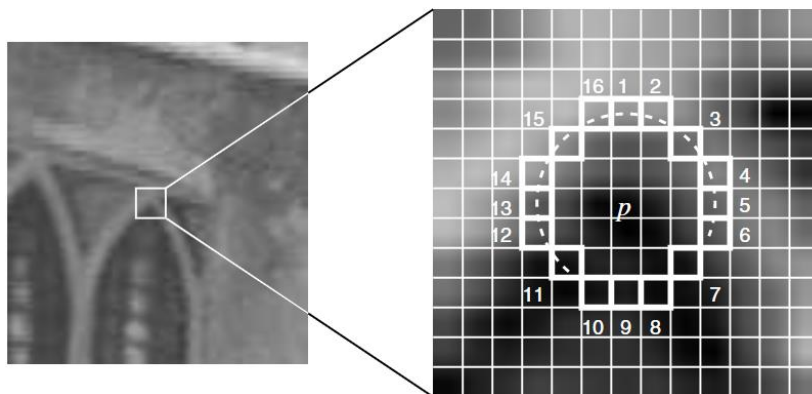


图 2-8 特征点判断示意图

经过上述步骤，便得到了 FAST 算法所提取的特征点，接下来通过 BRIEF 算法对这些特征点生成特征描述子。BRIEF 算法首先使用高斯模糊对图像进行平滑处理，减少噪声的干扰。然后对给定的关键点，在以点 q 为中心，标准差为 σ 的高斯分布中选取一点 q_1 。然后在以点 q_1 为中心，标准差为 $\sigma/2$ 的高斯分布中选取一

点 q_2 。通过比较点 q_1 与点 q_2 的灰度值,若 $q_1 > q_2$ 则描述子中对应位置赋值为 1 反之为 0。通过重复上述操作,最终得到特征点的描述子。

2.4.3 SuperPoint 算法

SuperPionnt 算法^[47]由 DeTone 等人于 2018 年提出,SuperPoint 网络结构图如图 2-9 所示。该算法主要通过三个步骤提取特征点与描述子,第一步是使用虚拟三维物体训练网络用于角点提取,第二步是用第一步中训练的网络模型提取真实场景图片中的角点,第三步将第二步的图片进行几何变换得到已知位姿关系的新图片,并将这两张图片输入网络中,检测关键点和提取描述子。

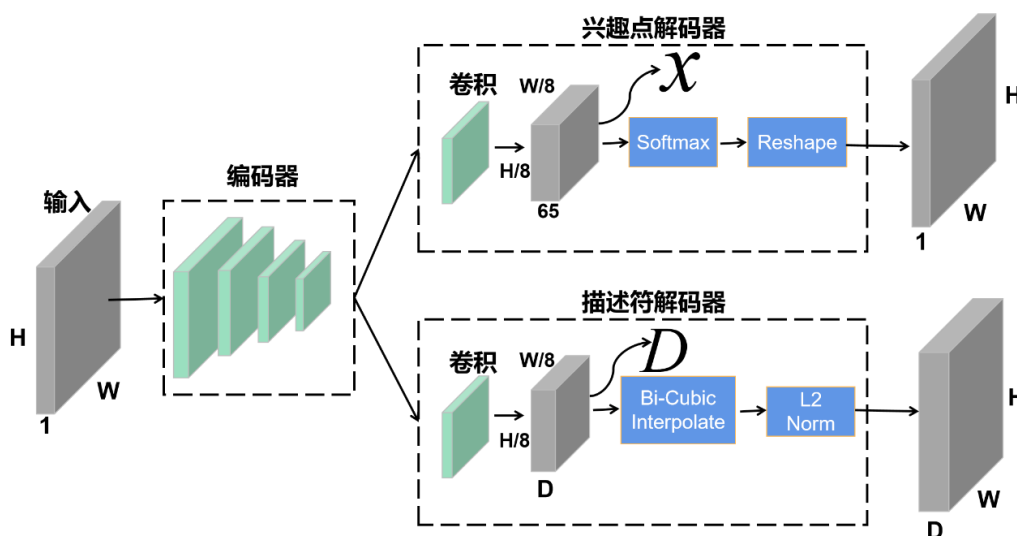


图 2-9 SuperPoint 网络结构示意图

由于初始模型只在基本形状元素的图像上训练,可能对于一般的图像特征提取效果不好。而 MS-COCO 数据集没有特征点真值,因此 SuperPoint 提出了一种真实图像自标注的方法,如图 2-10 所示。该方法将 MS-COCO 中每张图片做 n 种单应变换,得到 n 张变换后的图像。利用初始模型对这些图片进行特征点提取,可以得到 n 个特征点的热力图,将这些热力图累加并使用阈值筛选每个位置的特征点,便得到了原始图像的特征点真值。

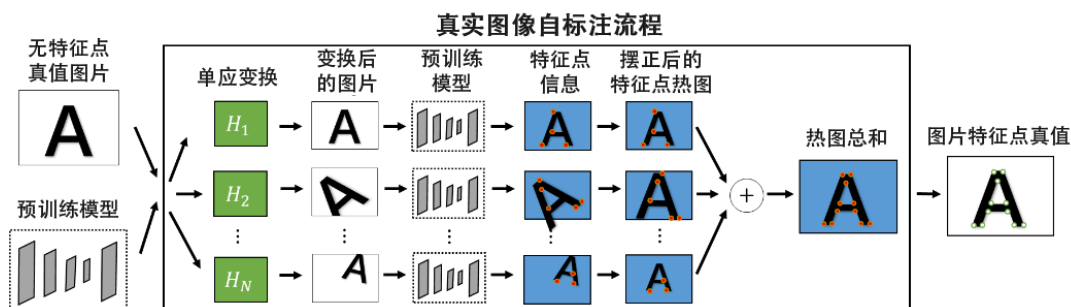


图 2-10 真实图像自标注流程图

2.5 本章小结

本章首先从刚体的空间运动与相机成像模型两个方面对刚体 6D 位姿估计的基本原理进行了详细的介绍。随后介绍了 **PnP** 问题以及 **PnP** 问题常用的几种解决方法。然后, 针对知识蒸馏算法, 本章介绍了知识蒸馏的原理以及在其他方向上的改进策略。最后介绍了一些主流点匹配算法的算法原理。

3.面向增强现实应用的定位—跟踪主从网络

本章主要介绍了本文所提出的面向增强现实应用的定位—跟踪主从网络。首先从网络结构、损失函数、样本采集与训练策略几个方面对该网络进行介绍。其次对跟踪网络训练所采用的知识蒸馏方法与损失函数进行详细介绍。随后介绍了自建数据集与实验中的评价指标。最后罗列了各部分对比实验的详细数据并依照实验数据对实验结果进行了详细分析。

3.1 网络结构与损失函数

3.1.1 网络结构

本文所提出的面向增强现实应用的定位—跟踪主从网络，基于点匹配的思想^[31]，采用将输入图像与模板图像进行点特征匹配的方式得到 2D-3D 点对信息，然后利用 PnP 算法计算相应位姿。其中定位算法由一个图像特征编码器与两个解码器组成，图像特征编码器基于 EncNet^[48]的网络结构，采用 RepVGG^[49]作为编码器的主干网络，为了降低计算复杂度与内存占用，采用了 JPU^[50]模块代替普通的扩张卷积对特征进行上采样。定位网络结构如图 3-1 所示。

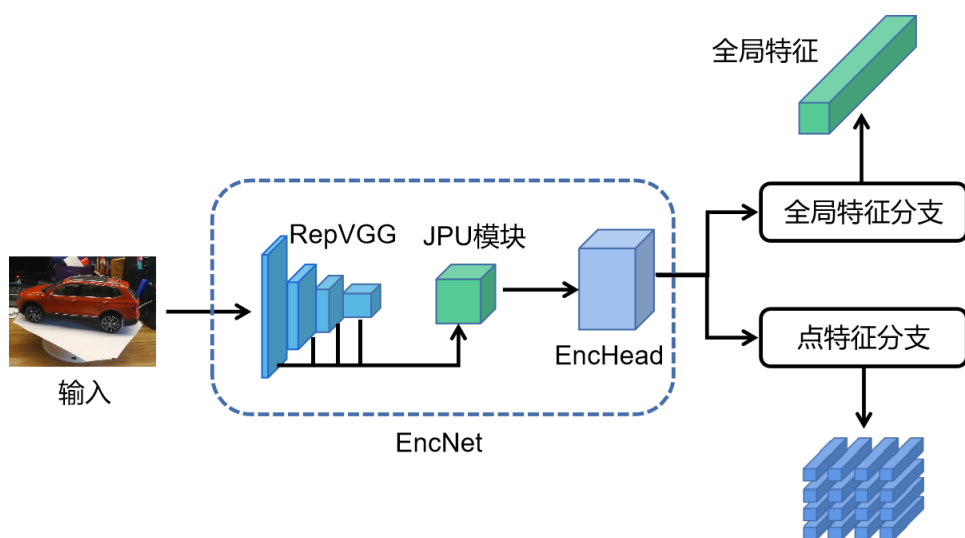


图 3-1 定位网络结构图

在定位网络中，为了达到最好得位姿估计效果，模板图像应与输入图像尽可能相似。因此在定位网络中，除了图像的点特征提取分支之外，还引入了图像的全局特征提取分支。并且为了模型推断，还构建了全局特征数据库，方便查找到最合适的图像作为匹配模板。

将点匹配思想用于物体的位姿跟踪，则模板图像应是前一帧或前几帧图像，因此不需要使用全局特征与数据库对模板图像进行搜索。故本文将对定位网络的网络结构进行解耦，得到点特征分支并将其用于物体的跟踪。跟踪网络结构如图 3-2 所示。

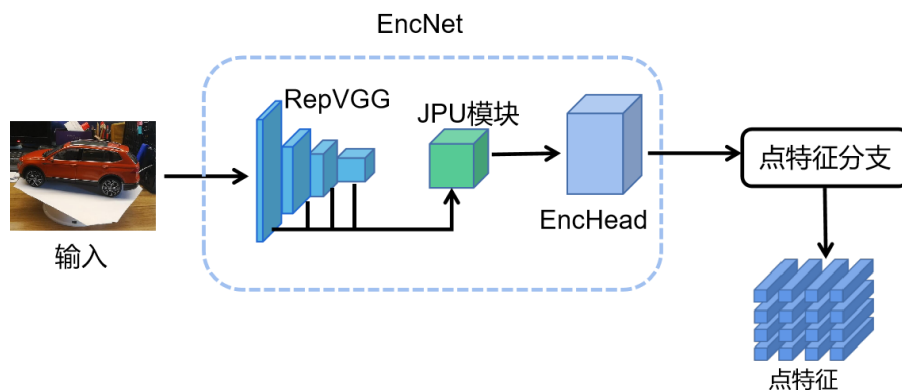


图 3-2 跟踪网络结构图

算法中的全局特征分支结构如图 3-3 所示。该分支将网络输出的特征分别经过不同的池化层与卷积层进行处理，得到粗细粒度不同的四个特征，然后将这些特征进行特征拼接，随后将拼接后的特征经过两个全连接层得到所需的全局特征。

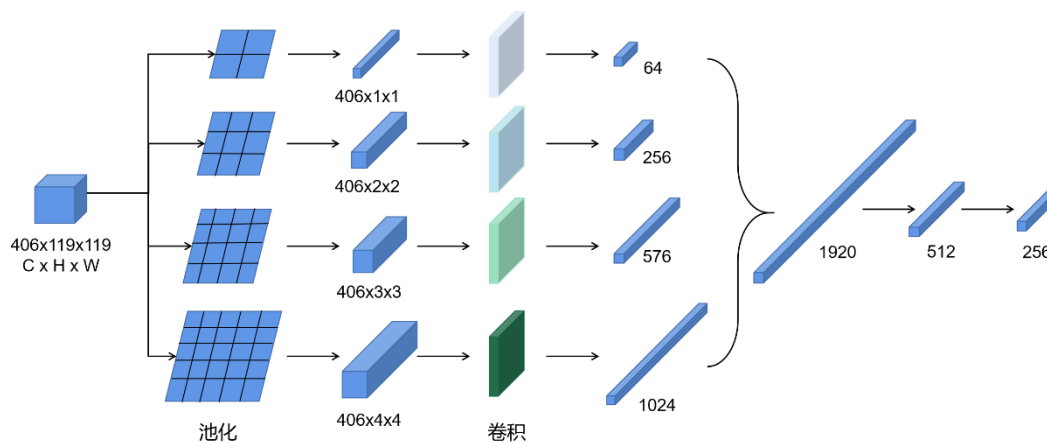


图 3-3 全局特征分支结构图

算法中的点特征分支结构如图 3-4 所示。该分支将网络输出的特征经过两个卷积层随后得到了图像的点特征。

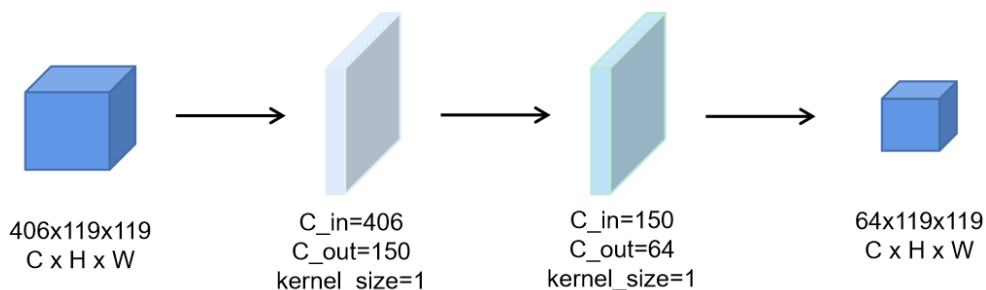


图 3-4 点特征分支结构图

3.1.2 损失函数与训练策略

在损失函数方面，定位算法与跟踪算法均采用了对比损失函数 **Contrastive Loss**^[51]。该函数的作用是为了使网络对同类数据所提取的特征尽可能相似，不同类数据所提取的特征的差异尽可能大。损失函数公式如(3-1)所示：

$$L_c = \frac{1}{2N} \sum_{n=1}^N yd^2 + (1 - y)\max(m - d, 0)^2 \quad (3-1)$$

公式中 y 是样本表示符，1 代表正样本对，负样本为 0。 m 为正样本对和负样本对之间的距离的最小差异度。 d 表示点对中两个点特征之间的欧式距离。

本文中训练数据中正负样本通过如下方法进行筛选，首先计算一组训练数据中任意两帧 F 、 S 之间掩码的交集与并集的比值，记为 $dist_{FS}^{iou}$ ；然后，计算这两帧图片的位姿间的距离 $dist_{FS}^{Rt}$ ；最后，设置正样本阈值 λ_{pos}^{iou} 、 λ_{pos}^{Rt} 并进行判断。当 $dist_{FS}^{iou}$ 大于阈值 λ_{pos}^{iou} 且 $dist_{FS}^{Rt}$ 小于 λ_{pos}^{Rt} 时，则认为 F 与 S 为一对正样本。反之认为 F 与 S 为一对负样本。正负样本判断标准如(3-2)、(3-3)所示：

$$I_{pos} \in (dist_{FS}^{iou} > \lambda_{pos}^{iou}) \cap (dist_{FS}^{Rt} < \lambda_{pos}^{Rt}) \quad (3-2)$$

$$I_{neg} \in (dist_{FS}^{iou} < \lambda_{pos}^{iou}) \cup (dist_{FS}^{Rt} > \lambda_{pos}^{Rt}) \quad (3-3)$$

在点特征模块训练时，为了让跟踪网络在有限的图像上能够学到更多的点特征，本文将物体全部的可见点信息用于跟踪网络的训练。而在模型推断时，为了提高跟踪算法模型的推断速度，则选择少部分可见点信息用于物体的位姿计算。为检验可见点信息的减少与位姿估计、模型推断速度的影响，本文在 3.3.4 节中对用少部分可见点与全部可见点这两种模型推断策略进行了相应的实验评估。

在实际应用时，定位网络与跟踪网络的匹配图像存在明显区别。定位算法检索到的匹配图像往往差异性较大，而跟踪算法则是针对前后帧图像进行匹配。因此在网络训练时，定位网络采用一组随机场景下的物体图像作为输入，跟踪算法则采用一组前后帧图像作为输入。（一组数据包含三张图像）

3.2 知识蒸馏

定位算法是对物体与相机间的绝对位姿进行估计，跟踪算法是对相机运动前后的相对位姿进行估计。本文提出的定位算法与跟踪算法均基于点匹配的思想，通过点特征匹配得到 2D-3D 点对信息，从而计算相应位姿。定位算法与跟踪算法位姿计算流程分别如图 3-5、图 3-6 所示。在定位算法的位姿计算中，数据库中所检索到的模板图像与当前图像往往存在一定的差异，为了保证 2D-3D 点对信息的充足，因此采取全局匹配的方式对各个点特征进行匹配。由于跟踪算法是

对前后帧图像进行匹配，而前后帧图像往往差异性较小。因此，为了进一步提高跟踪算法模型的推断速度，本文在模板图像的可见点位置处选取一个大小为 5×5 的像素空间，将当前图像相应位置的点特征与模板图像的点特征进行匹配（即对图像的点特征进行区域匹配），从而得到 2D-3D 点对信息并结合 PnP 算法计算相应的位姿。

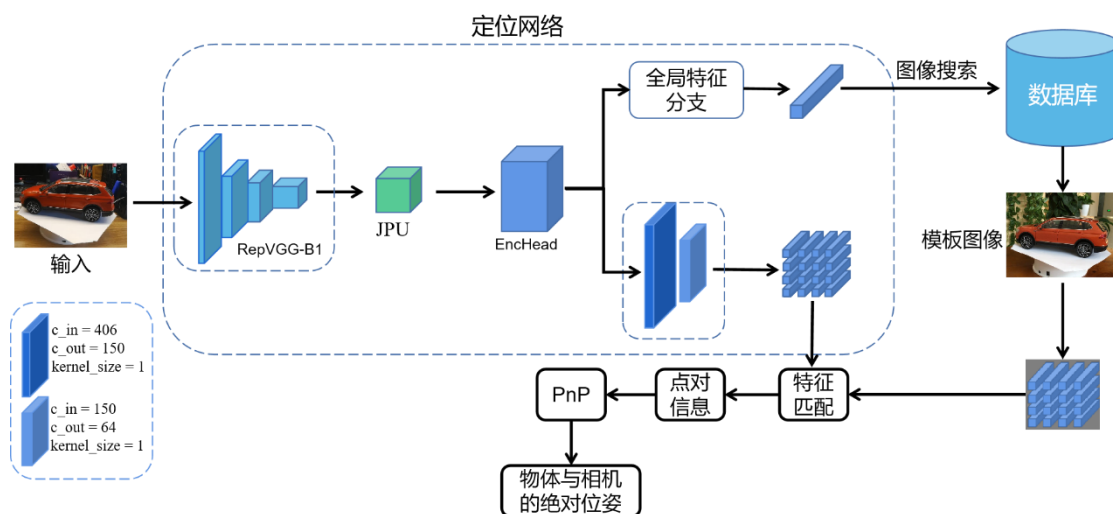


图 3-5 定位算法位姿计算流程图

定位算法需要利用图像的全局特征，从数据库中搜索出与当前图像最近似的模板图像进行点特征匹配，从而获取到当前图像的 2D-3D 点对信息并利用 PnP 算法计算物体与相机间的绝对位姿。而跟踪算法则是对前后帧进行匹配，得到 2D-3D 点对信息从而计算相机运动前后的相对位姿，因此不需要对图像的全局特征进行学习。为了满足跟踪算法实时性的要求，跟踪算法在主干网络上，也选择了更为轻量化的 RepVGG-A0 网络结构。

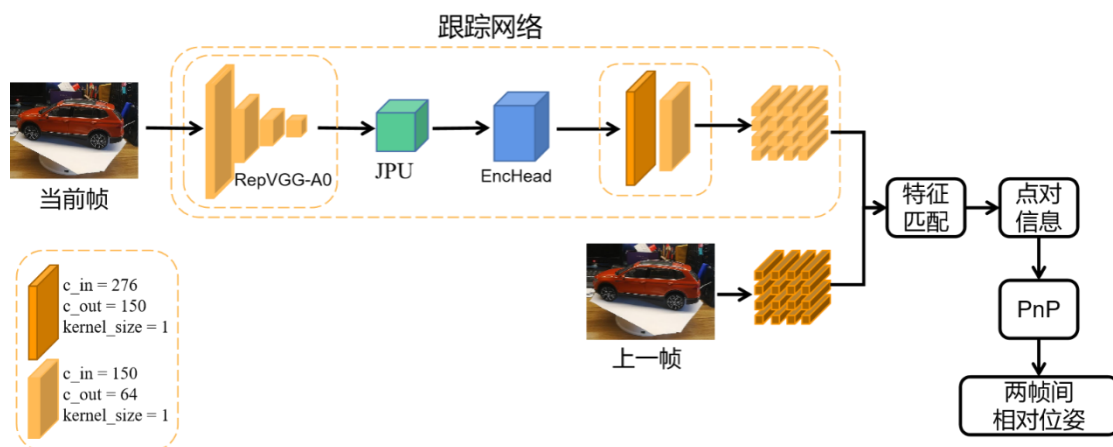


图 3-6 跟踪算法位姿计算流程图

由于跟踪算法的实时性要求，因此在点特征的提取上，定位网络相较于跟踪网络有着更为庞大的参数量，对点特征有着更好的表达能力与鲁棒性，并且定位网络与跟踪网络在结构上有着一定的相似性。因此本文采取知识蒸馏的方法将定

位网络的点特征分支作为教师网络指导跟踪网络的训练，并在后续章节中，通过实验验证了该方法有助于跟踪算法精度的提高。本文中所用到的教师网络与学生网络结构如表 3-1 所示：

表 3-1 教师网络与学生网络结构表

网络模块	教师网络 (定位网络)	学生网络 (跟踪网络)
主干网络	RepVGG-B1	RepVGG-A0
网络层一	[3×3, 128]×4	[3×3, 48]×1
网络层二	[3×3, 256]×6	[3×3, 96]×1
网络层三	[3×3, 512]×16	[3×3, 192]×1
网络层四	[3×3, 2048]×1	[3×3, 1024]×1
EncHead	[256,512,2048]	[256,512,512]
EncModel	512	256
模型大小	311.4M	46M

上表中，网络层对应数据表示该层卷积核大小、输出特征的通道数以及该网络层所对应的卷积层个数。EncHead 与 EncModel 所对应的数据表示该层次结构对应的通道数。

本文采用知识蒸馏的方法对跟踪算法进行训练，在主干网络的特征提取与最后输出的点特征处增加了两个损失函数。首先对于主干网络而言，希望学生网络能够模仿教师网络对特征的表示能力，但又需要防止逐像素对齐会导致学生网络过度学习教师网络从而错过最优解。因此本文将教师网络与学生网络的主干网络所提取的特征进行了标准化，并让学生网络从特征分布上向教师网络学习^[52]。本文采用基于 KL 散度的损失函数作为主干网络部分蒸馏的损失函数。损失函数公式如(3-4)所示：

$$L_f = \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^K y_{ni} \cdot (\log y_{ni} - \log x_{ni}) \quad (3-4)$$

公式中 N 表示特征层总层数， n 表示当前特征层层数， K 表示当前特征分布中元素总数， i 表示当前元素下标， y 表示教师网络特征输出经过标准化与 Softmax 函数处理后的特征分布， x 表示学生网络特征输出经过标准化与 Softmax 函数处理后的特征分布。本文将各层特征分布损失求和并取平均作为损失函数的值。

在点特征的输出方面，本文选择借鉴多分类任务上知识蒸馏的思想，将各像素作为最小分类单元进行独立蒸馏。为了使学生网络尽可能学到教师网络对点特征的描述，本文采取均方误差作为点特征提取部分的损失函数。损失函数公式如(3-5)所示：

$$L_e = \frac{1}{n \cdot m} \sum_{i=1}^n \sum_{j=1}^m (y_{ij} - x_{ij})^2 \quad (3-5)$$

公式中 y_{ij} 表示位置处于教师网络所输出的点特征的第*i*行第*j*列的特征值。
 x_{ij} 表示位置处于学生网络所输出的点特征的第*i*行第*j*列的特征值。

结合 3.1.2 中跟踪网络本身的损失函数与上面所设计的两个损失函数，得到知识蒸馏方法下，跟踪网络训练的损失函数为：

$$L = \alpha \cdot L_c + \beta \cdot L_f + \gamma \cdot L_e \quad (3-6)$$

其中 α 、 β 、 γ 为损失函数权重，本文中一次完整的模型训练将进行八万次迭代。通过实践经验，本文通常将前四万次迭代采用的权重设为[1,1,1]，后四万次迭代采用的权重为[1,1,0.3]。知识蒸馏网络结构图如图 3-7 所示：

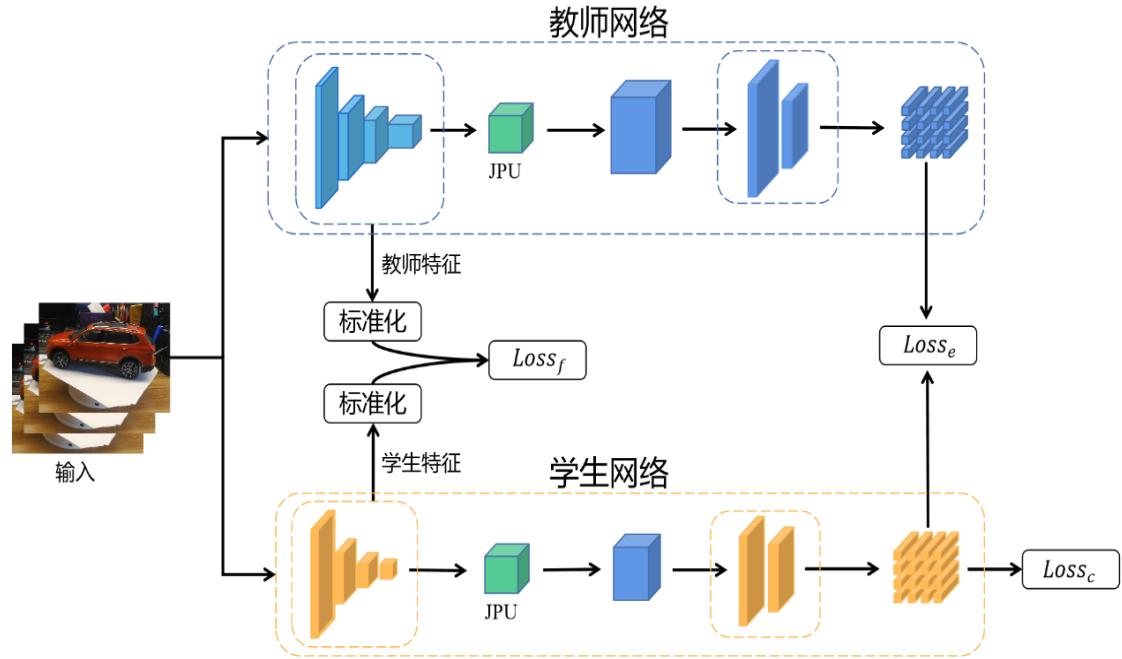


图 3-7 知识蒸馏网络结构图

3.3 实验结果与分析

3.3.1 数据集

本文为网络训练与测试创建了一个由咖啡机、途观车、雅迪电动车与办公大楼四个物体组成的数据集。数据集中包括以下几个部分：RGB 图像、相机内参、物体 3D 模型、物体真实位姿、可见点信息。各物体 3D 模型如图 3-8 所示。

① RGB 图像，本文通过对每个物体在不同光照、不同场景下各拍摄十段时长为一分钟的视频。通过 FFmpeg 工具，对视频进行一秒十帧的方式进行抽帧，构成数据集中的 RGB 图像。

② 相机内参信息，本文通过张正友标定法^[36]利用棋盘格对所有拍摄设备进行内参标定。

③ 物体的 3D 模型，本文利用 RealityCapture 软件与各物体所拍摄的视频生成物体相应的 3D 模型。

④ 物体的真实位姿，本文通过手动标注的方式，标记图片中 2D 关键点。然后利用 PnP 算法计算出各图片中物体所对应的真实位姿。

⑤ 物体的当前可见点信息，本文利用上述步骤获得的 RGB 图像、物体 3D 模型、物体真实位姿和相机内参信息，通过已开源的算法计算得到当前图片对应的可见点信息。

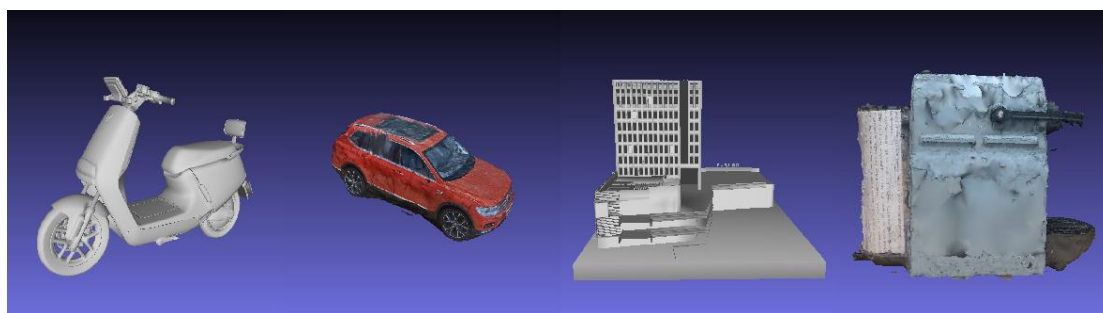


图 3-8 物体 3D 模型示意图

数据集展示图如图 3-9 所示：

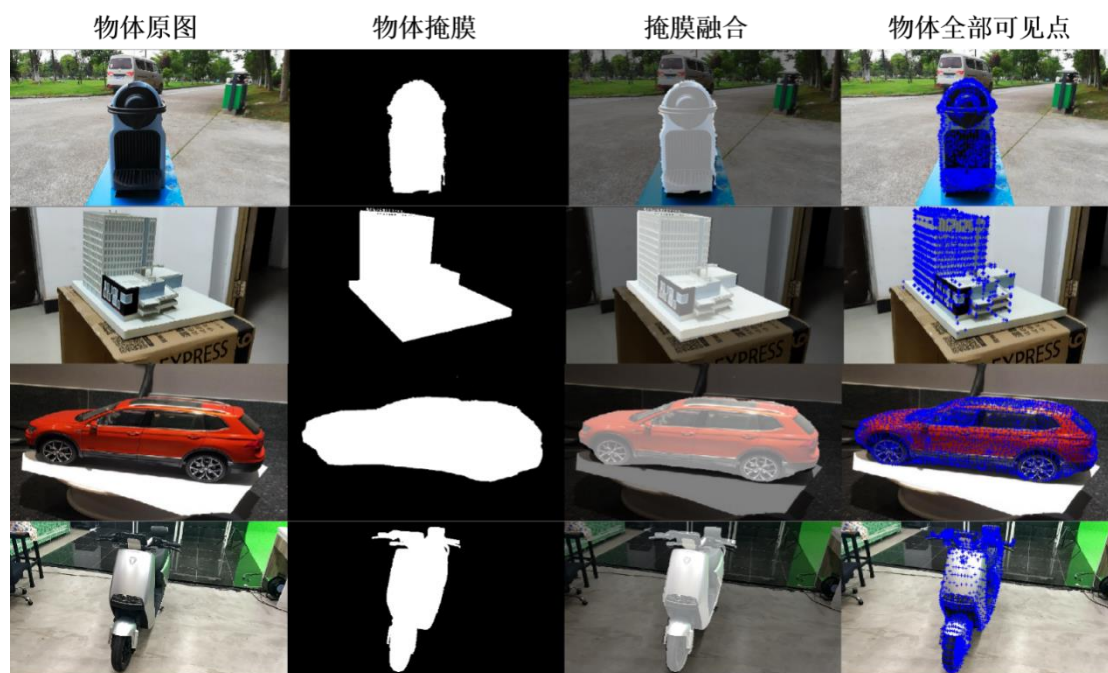


图 3-9 数据集展示图

3.3.2 实验环境与准备

本文中，教师网络与学生网络的训练环境如下表 3-2 所示。

表 3-2 实验环境配置表

网络名称	CPU	GPU	操作系统	运行环境
教师网络	Intel Core i5-6500	NVIDIA RTX 2060 super	Ubuntu 18.04	Python3.7+Pytorch1.8.1+CUDA10.2
学生网络	Intel Core i5-9400F	NVIDIA TITAN V	Ubuntu 22.04	Python3.7+Pytorch1.8.1+CUDA10.2

注：本文中，各项对比实验皆是在教师网络的训练环境中进行。

在训练开始前，将数据裁剪为 473×473 大小并进行数据增强与标准化，以此来加强模型的泛化能力并减少过拟合的可能性。对各物体的数据按场景划分为训练集与测试集（如物体一中场景 1-5 皆为训练集，场景 6-10 皆为测试集），由于各场景中背景、角度与光照等条件不一，给模型测试带来一些噪声以此增加模型的鲁棒性。

3.3.3 评价指标

本文中主要使用了以下评价指标：

① **ADD**^[53]指标：是将物体的真实位姿与预测位姿转换成 3D 模型点，并计算之间的平均距离，通过 3D 模型点间的平均距离与物体真实 3D 模型的直径的 10%或 5%进行比较，若小于则认为位姿估计正确。ADD 计算公式如(3-7)所示：

$$ADD = \frac{1}{m} \sum_{x \in M} \|(R_{gt}x + t_{gt}) - (R_{pre}x + t_{pre})\| \quad (3-7)$$

其中 R_{pre} 和 t_{pre} 分别表示预测的旋转矩阵和平移向量， R_{gt} 和 t_{gt} 分别表示真实的旋转矩阵和平移向量。

② **重投影误差**（下文实验结果中用 **proj** 表示）^[54]：是将物体 3D 模型上的 3D 点，分别使用真实位姿与预测位姿投影至 2D 平面上，并计算两组投影之间的平均距离，若平均距离小于所设定的阈值（由于阈值不同，通常将指标分为 **proj-5** 和 **proj-10**）则认为位姿估计正确。重投影误差计算公式如(3-8)所示：

$$proj = \frac{1}{m} \sum_{x \in M} \|K(R_{gt}x + t_{gt}) - K(R_{pre}x + t_{pre})\| \quad (3-8)$$

其中 K 表示相机内参， R 和 t 的含义与公式(3-7)中一致。

③ 在 **CMD5**^[55]指标中将平移误差与欧式旋转角误差是否小于 5 厘米/5 度作为评价标准。而跟踪算法通过匹配前后帧来计算位姿，往往平移误差与欧式旋转角误差容易满足 5 厘米/5 度的条件，故该指标无法明显的体现出在平移误差与旋转角误差上的差异。因此本文参考 PoseNet^[56]的思想直接将平均旋转角

误差 (R_{error}) 与平均平移误差 (t_{error}) 作为评价指标。误差计算公式如(3-9)、(3-10)所示:

$$R_{error} = \frac{1}{m} \sum_{i=1}^m \frac{180^\circ}{\pi} \arccos \left(\frac{1}{2} [tr(R_{gt}^i{}^{-1} \times R_{pre}^i) - 1] \right) \quad (3-9)$$

$$t_{error} = \frac{1}{m} \sum_{i=1}^m \|t_{gt}^i - t_{pre}^i\|_2 \quad (3-10)$$

其中 R 和 t 的含义与公式(3-7)中一致,上标 i 表示当前所计算的位姿处于数据中的位置。

3.3.4 实验结果与分析

本文的实验内容分为以下四个部分:知识蒸馏效果对比、模型推断效果对比、位姿估计效果对比以及移动端速度测试。在本小节中,将列举以上四个实验的实验数据并依照实验数据进行适当分析。注:下列表格中,加粗的数据表示该数据在此项指标上更为优秀。

① 知识蒸馏效果对比

为验证本文所采取的知识蒸馏方法的有效性,本小节将经过知识蒸馏的跟踪网络模型与未经过知识蒸馏的跟踪网络模型进行对比实验。由于跟踪算法是对相邻帧图像进行匹配,因此位姿估计的精度较高,为了更清晰地展示通过知识蒸馏方法所训练的模型与未通过知识蒸馏方法所训练的模型之间在位姿估计精度的差异,本文使用最为直观的评价指标:平均旋转误差 (R_{error}) 与平均平移误差 (t_{error}) 作为该对比实验的评价指标。相关实验数据如下表 3-3、表 3-4、表 3-5、表 3-6 所示:

表 3-3 咖啡机 R_{error} 与 t_{error} 指标表现

模型	场景一	场景二	场景三	场景四	场景五
类别	R_{error}/t_{error}	R_{error}/t_{error}	R_{error}/t_{error}	R_{error}/t_{error}	R_{error}/t_{error}
蒸馏	1.0653/0.27	1.4226/0.43	2.3556/1.10	1.6971/0.53	1.3304/0.59
非蒸馏	1.2185/0.34	1.8474/0.68	2.4449/ 1.08	1.9098/0.59	1.4520/ 0.59

表 3-4 办公大楼 R_{error} 与 t_{error} 指标表现

模型	场景一	场景二	场景三	场景四	场景五
类别	R_{error}/t_{error}	R_{error}/t_{error}	R_{error}/t_{error}	R_{error}/t_{error}	R_{error}/t_{error}
蒸馏	1.9445/0.63	2.0414/0.67	2.1287/ 0.66	2.1211/ 0.70	2.2993/0.71
非蒸馏	1.9661/ 0.62	2.1062/0.68	2.1097/0.69	2.0917/0.70	2.2776/0.70

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/605342232140011034>