

湖南师范大学

本科毕业设计

题目(中文): 室内定位场景下自适应异质传感器融合的 VR
手柄及定位算法研究

题目(英文): Model on VR Controller
and Pose Estimation of
Adaptive &
Heterogeneous Sensor
Fusion in Indoor Scene

学生姓名: 刘华林

学 号: F201930075021

指导教师: 刘宏

学 院: 信息科学与工程学院

专业班级: 2020 级软件工程辅修班

二〇二二 年 五月

湖南师范大学本科毕业设计诚信声明

本人郑重声明：所呈交的本科毕业论文，是本人在指导老师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议，除文中已经注明引用的内容外，本设计不含任何其他个人或集体已经发表或撰写过的作品成果。对本设计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。

本科毕业设计作者签名：

二〇二 年 月 日

目录

中文摘要	2
英文摘要	3
绪论	5
1 虚拟现实定位	7
1.1 定位传感器描述模型	9
1.1.1 定位传感器一级参数模型	10
1.1.2 定位传感器二级参数模型	12
1.1.3 二级参数的插槽模型推导	14
1.2 VR 追踪器标准及方案	18
1.2.1 VR 追踪器与 PC 通信协议	19
1.2.2 VR 追踪器与传感器通信协议	21
1.3 VR 控制器（手柄）标准及方案	22
1.3.1 VR 控制器（手柄）按键布局及消抖处理方案	24
1.3.2 VR 控制器（手柄）与主机端通讯协议	26
2 定位融合算法及系统	27
2.1 游戏化和数据众包	29
2.1.1 游戏化概念及技术	29
2.1.2 数据众包概念及技术	32
2.2 验证式调试器及活性连接技术	35

2.2.1 验证式调试器流程	36
2.2.2 活性连接技术原理	38
2.2.2.1 活性连接技术与游戏化	40
2.2.2.2 活性连接哲学	43
2.2.2.3 活性连接技术与数据生成	44
2.3 生成器和翻译器设计	45
2.3.1 生成器的理论设计	45
2.3.2 生成器的程序设计	49
2.3.3 生成器效果实验和评价	51
2.3.4 翻译器理论设计	53
2.3.5 翻译器和调试器界面实现	54
2.3.6 面向信息融合的活性连接技术	59
2.4 定位系统总体设计	60
2.4.1 定位系统 Petri 网描述与仿真	61
2.4.1.1 定位系统的 Petri 网描述	61
2.4.1.2 定位系统的 Petri 网仿真	62
3 定位系统部署	64
3.1 FPGA 神经网络加速平台	64
3.2 二值神经网络	67
3.3 二进制神经网络的数值计算结构	70
3.4 二进制神经网络的 Verilog 表示、仿真和评价	74
结论	77

参考文献	8 0
附录	8 7
致谢	8 9

室内定位场景下自适应异质传感器融合的 VR 手柄及定位算法研究

软件工程 2020 级 刘华林

摘要：虚拟现实设备可为用户提供高度沉浸、可交互的内容，而可用的虚拟现实产品必须是高实时性高稳定性的。本文基于室内小场景的虚拟现实 6DoF 定位需求和异质自适应的传感器融合算法目标，面向消费级 VR 电子产品设计的全过程做了软硬件协同的 VR 手柄研究和定位系统的算法研究。产品方面本文研究了 HTC Vive 的 VR 手柄标准和 OpenVR 接口，设计了手柄同 PC 和传感器的串口通信协议及 Verilog 程序实现，还为定位系统的神经网络部署提供了二值化和 FPGA 加速方案。定位系统方面本文从系统级的定位需求出发，设计了传感器的参数描述模型使定位系统的信息融合不必依赖特定传感器特性，并将被动式加速度计型和主动式测距型传感器纳入该模型，在信息融合的算法实现方面采用流行的神经网络架构，并创新出实时产生训练数据的活性连接方法，使定位系统理论上可做到对异质传感器的自适应和即插即用特性，且该种活性连接技术具有通用性。最终得到的 FPGA 定位系统处理时延达 80.64us，提供对传感器的滤波和融合，并且支持 PC 端应用以 12ms/次的速率请求定位和控制数据，满足现代虚拟现实内容对 VR 手柄性能的需求。

关键词：虚拟现实；多模态信息融合；定位系统；在线学习；FPGA

Model on VR Controller and Pose Estimation of Adaptive & Heterogeneous Sensor Fusion in Indoor Scene

Abstract: Virtual Reality devices can provide users with highly immersive and interactive contents, and the usable Virtual Reality product must be highly real-time and stable.

Based on the demand of Virtual Reality 6DoF positioning of indoor and small scenes, and the goal of heterogeneous adaptive sensor-fusion algorithm, this paper studies the VR handle with “Software hardware cooperative” positioning system and its algorithm for the whole process of consumer VR electronics design.

In terms of product design, this paper studies the VR handle standard of HTC Vive and OpenVR interface, proposes a serial communication protocol between the handle and PC or sensor, with implementation of Verilog, and also provides binarization and FPGA accelerator scheme for deployment of positioning system’s neural network.

In terms of positioning system, starting from the description model of the sensor, so that the sensor-fusion of positioning does not have to rely on the specific sensor characteristics, and incorporates the passive accelerometer and active ranging sensor into the model. About the algorithm implementation of sensor-fusion, it adopts the popular neural network architecture, and creates “Active Connection Method”(ACM) to generate training data in real time. Theoretically, the positioning system with ACM can achieve an self-adaptive and “plug and play” performance for heterogeneous sensors, and the ACM is universal.

The processing time-delay of the final FPGA positioning system is up to 80.64us, which provides filtering and sensor-fusion, and support PC applications to request positioning/controlling data at a rate of 12ms/time, so as to meet the requirement of modern Virtual Reality content for VR handle.

Key words: Virtual reality; Sensor fusion; Positioning system; Online learning; FPGA; Pose estimation

1 绪论

1.1 虚拟现实领域介绍

新世纪以来，以 3D 动画、真实物理引擎、人机交互、实时传感器技术为特征的虚拟现实技术（Virtual Reality，以下简称 VR）成为信息技术领域的一大热点技术，虚拟现实技术支持用户在实时渲染的动画环境内切身地体验和与场景内的物体实现贴近现实的交互。虚拟现实产品在产品内容上体现为设计和实现虚拟空间中的可感知对象，如下图 1 所示，区别于传统 IT 产品地而显著具有沉浸式（Immersion）、交互式（Interaction）、幻想式（Imagination）^[1]。凭借在这三种主题上实现既有范式的重建，可以实现具有颠覆力的虚拟现实产品。

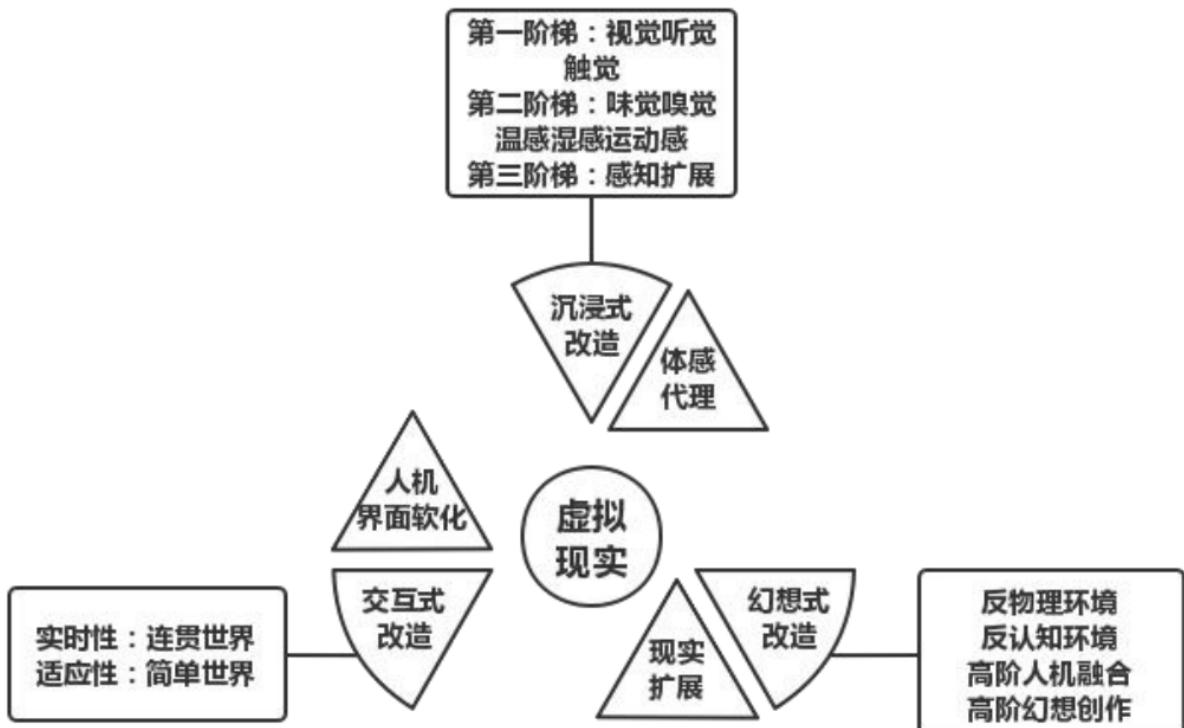


图 1 虚拟现实三个维度上的对传统 IT 产品重构的解析图

1.2 虚拟现实关键技术

虚拟现实产品在体感维度超越了传统 IT 产品的延伸性体感，发展至多维体感并提供感知扩展的可能性。一般的个人电脑仅调用用户的视觉听觉，移动电话的部分软件支持对运动感知的调用（例如 SLAM 技术，为简单的增强现实功能提供支持），但也仅限于此，而虚拟现实产品则将视觉听觉触觉运动感知强调到基本环境的层次，并考虑为更多类别的感知设计解决方案、提升上述感知的精度^[2]，甚至通过脑机接口完成新类别感知的扩展^[3]。

因此，虚拟产品的颠覆性的关键就在于能使用户充分浸入系统所营造的虚拟环境，使用户在虚拟现实环境中的感知与现实相协调、难以察觉区分。这种沉浸要求系统对用户的体感进行代理，这种感知代理有效的关键就在于保证该代理过程的实时性。只有用户的每个动作都能即时得到有意义的反馈，用户才能充分接受虚拟现实系统作为他的感官代理。

1.3 本文的工作

本文的出发点正是从电子产品软硬件工艺的全过程，包括传感器管理、定位算法设计和电子产品与上位机的通讯、VR 硬件驱动的接口标准，去讨论如何设计和实现一款高实时性高稳定性的 VR 追踪器和 VR 控制器。本文的最大亮点在于，尽管只做 VR 手柄的定位系统，但构造的定位系统：传感器描述模型、异质多模态传感器融合技术并不只限于 VR 手柄，而是面向广泛的小范围内（室内场景）的 6DoF 定位需求，使定位系统不必依赖于特定的传感器，其 6DoF 姿势估计具备自适应的特性。

下文包括三大章节和文末总结，第一章讨论虚拟现实定位领域技术的进展，面向定位系统设计了传感器描述模型，使定位系统可以面向抽象的传感

器模型而非具体的传感器设计定位算法，并讨论了 VR 业界的 VR 追踪器和 VR 控制器标准，依此设计了一套 VR 追踪器和 VR 控制器的传感器采样方案和与上位机的串口通讯方案。第二章讨论了游戏化领域和数据众包技术，并介绍了由这两个领域启发的验证式调试器和活性连接训练的多传感器融合算法。第三章讨论了定位系统及其主要成分神经网络如何部署到 FPGA 平台上。文末对定位系统设计和实现的亮点做了总结，并提出了下一步研究的方向。

2 虚拟现实定位

虚拟现实技术是紧贴商业化的，软件端的虚拟现实技术主要是围绕 OpenVR 协议为核心所规定的 VR 定位标准和 VR 渲染标准，硬件端的虚拟现实技术则围绕手势识别、运动姿态估计、实时渲染多个技术热点和难点展开。

OpenVR 协议是 OpenVR 源代码所开放的接口标准体系，Open VR 是一个开源的模块化、轻量级虚拟现实软件平台^[4]，旨在让协调硬件和软件方的开发人员。本设计面向 OpenVR 协议提供的硬件标准开发 VR 追踪器设备，OpenVR 要求追踪器提供所追踪设备的 2 行 3 列 (6DoF) 16 位浮点数姿态估计矩阵：

$$Pose = \begin{pmatrix} x, y, z \\ \varphi, \theta, \psi \end{pmatrix}$$

其中 x, y, z 为坐标轴， φ, θ, ψ 为欧拉角，欧拉角遵循右手坐标系 ($+y$ 为向上， $+x$ 为向右， $-z$ 为向前)。

目前较为成熟的 VR 追踪器设备是 HTC 公司的 Vive Tracker，Caserman et. al 的工作^[5]将 Vive Tracker 视为通用的动作捕捉设备，只使用少量的

Tracker, 通过收集肢体末端的 Tracker 数据并根据人体结构模型进行反向动力学 (Inverse Kinematics) 建模得到人体全体各部位的姿势估计, 但要求足够的算力以保证实时渲染。尽管 Tracker 仅是单点的 6DoF 估计设备, 但 OpenVR 软件端可以采用强大的算法处理这些定位数据展开丰富的应用, 对于本设计的追踪器/定位器, OpenVR 接口的要求就是, 用最快的速度交付最准确且稳定的 6DoF 姿态估计矩阵。

虚拟现实定位的实时性要求极高: 基本与外部现实的动作保持同步, 才能保持虚拟现实环境的沉浸感^[6]。Lee et. al 在 VR 内容流测速实验中^[7]内容流的数据速率提升会提升 FPS 和减小数据抖动, 高速流媒体才可以保证虚拟现实内容的适当帧率以不丢失沉浸感。当这种实时性得到保证时, 虚拟现实技术可以为用户在同一个时空搭建另一个“身体”, 类似 Caserman et. al 的实时反向动力学建模, Ning Zhang et. al. 将 VR 内容流实时导入到机器人^[8], 将人类在虚拟环境内执行的动作序列识别并实时映射到机器人上, 显示出虚拟现实技术应用于实时远程操控和智能机器人领域的巨大空间, 该应用方向的一个重要工作是王盼的使用户通过虚拟现实, 实时指挥目标场景的三维建图机器人进行探索的交互模型及设施^[9]。

本设计采用 FPGA 满足虚拟现实定位的高实时性要求, 这方面成熟的工作参考郑世涛用 ADXL345 倾斜传感器和 HMC5883L 磁阻加速度计作为传感器、Ethnet (以太网协议) 作为数据流传输媒介的 FPGA 虚拟现实定位系统^{[10][11]}, 本设计的最终成品同样采用该传感器组合, 采用 USB 作为与 PC 上位机的传输媒介。

目前主流的虚拟现实定位技术基本可以分为三类: 1. 被动式加速度计定

位；2.主动探测式测距定位；3.主动探测式视觉定位。其中被动式加速度计定位指的是采用倾斜传感器、加速度计等单元测量追踪对象的移动和旋转，传感器直接置放在追踪对象内部；主动探测式测距定位和视觉定位的区别在于测距原理，测距定位基于三角测距、三边测距、ToF 测距等几何原理进行，而视觉定位则基于图像处理技术完成，室内测距定位的实验可以参考刘智伟等人的 UWB（Ultra Wide Band,超宽带无线载波通信技术）测距的精度标定工作^[12]，视觉定位的实验可以参考 Jiachen Yang et. al.的深度学习单目视觉 6 轴姿态估计系统工作^[13]。

虚拟现实技术的定位算法未来的趋势是，不断往手势估计、运动姿势估计的复杂多关节姿势估计发展，这也是本设计的姿势估计算法未来的精进方向，手势估计方面一篇非常有趣的手势识别的姿势估计研究是 Min-Yu Wu et. al.所做的工作^[14]，由深度相机获取用户的手形和手位，根据预定义的手部关节图映射为手部关节的活动序列，进而反映到虚拟现实环境内部使虚拟手发生动作与虚拟物体对象产生交互。

以上讨论了虚拟现实技术的标准和领域现状，本设计由此出发，面向虚拟现实定位技术的核心要求：实时性，以及低成本高精度的性能要求，设计了综合 VR 定位器（VR Tracker）和 VR 控制器（VR Controller）的 VR 手柄，单个手柄的定位系统可提供 6DoF 定位，并且该定位系统允许自由地配置廉价的传感器部件，进行动态的多传感器信息融合实现稳定精确的姿势估计，为虚拟现实设备的广泛应用提供技术条件。

1.4 定位传感器描述模型

本设计的定位系统之核心功能在于对给定的传感器阵列，能融合传感器

信息得到可靠的姿势估计数值。为此必须先定义什么样的传感器阵列是合规的，为传感器表明属性以方便定位系统对带有同类特征的传感器采用相同的处理方法、简化软件设计。

传感器是传递物理量变化的电子设备，将物理过程的模拟量实时转化为数字量，常使用的具备定位功能的传感器包括：测距设备（一对波形发射器/接收器）、加速度计、陀螺仪，但根据传感器的一般性定义可知，只要能利用传感器所感知环境的特性与绝对或相对的空间位置的关系，就可以开发出有用的定位设备，出于各种定位传感器的定位算法实现的复杂性，本文仅关注对 VR 定位系统较重要的一些参数而非传感器的各异的物理定位算法：

A.模拟量分辨的能力，如 $1\text{mm}\sim10\text{mm}$ ，不同传感器分辨的层次不同，如传感器 A 可以确定对象在 5V/m (m 级)，传感器 B 则可以确定对象在 510mV/mm (mm 级)，传感器 B 在更大的数字信号位数下可以输出更精确的数值；

B.采样的速度，不同传感器产生的位置数据频率不一定在同一数量级，例如 GHz 的传感器 A 和 MHz 的传感器 B 是分别以不同频率交付数据的，这就需要融合传感器的信息；

C.采购成本，定位系统需要在性能与成本间做权衡，考虑传感器采购成本的系统设计对该产品实际投产非常有用，低廉的成本也是虚拟现实设备普及的需要；

从 ABC 三点参数出发可以抽象传感器对象的代数结构，得到定位传感器的描述模型。

1.1.1 定位传感器一级参数模型

一级参数(电气级)	参数介绍
帧率 Frame Rate	传感器输出电信号的速度, 表示的是单位时间内采样的数据点个数, 记为 Fr 。
功率 Power	物理信号转为电气信号、A/D 转换等过程需要驱动负载工作, 将单位时间内所负载的总功率记为 Pr (W)。
灵敏度 Sensitivity	在线性工作范围内输出模拟量变动 Δy 对物理信号量变动 Δx 的比率, 记为 Se 。
量程 Range	在线性工作范围内输出量 Δy 的最大值与最小值的代数差, 记为 Ra 。
分辨率 Resolution Ratio	A/D 转换中数字信号输出量 y 对被测量 Δy 最小变化的感知能力, 记为 Rr 。
采购价格 Cost	传感元件生产厂商所提供的参考价格, 记为 Co 。

表 1 定位传感器的一级参数表

一级参数不涉及传感器如何输出定位数据的信号处理层细节, 而只关心传感器作为电气元件的性能, 并且这些电气性能与定位系统所接受的传感器数据紧密相关, 一级参数决定了定位系统以怎样的速度接收是否精确的数据,

并为此耗费多少电能。记一级参数为 $S_1 = (Fr, Pr, Se, Ra, Rr, Pr, Co)$ 。

一级参数作为电气参数不是彼此独立的，帧率 Fr 的提高和分辨率 Rr 的提升一般会加大负载从而提升功率，灵敏度 Se 的提升一般以缩短量程 Ra 为代价，电气参数间的相关关系与传感器的电路结构和传感器的物理原理密切相关。

一级参数间的制约关系需要我们引入控制过程，但处于简单考虑，本文并不进一步讨论定位系统的控制一级参数的算法，只规定列明传感器参数的必要性，必须符合传感器阵列的接口标准，才可确保定位系统可理解插入的传感器的特性。

1.1.2 定位传感器二级参数模型

二级参数（追 踪级）	介绍
探测器位置	安放传感器/探测器的位置，部分探测器的位置是可调节的，记为 Pos_d 。
探测器位移 速率	对于可调节位置的部分探测器，单位时间内相较基准位置所发生的 xyz 三轴向上的移动，记为 Mov_d 。
探测器方位 角	安放传感器/探测器的朝向，部分探测器的朝向是可调节的，记为 Ang_d 。
探测器角速 度	对于可调节朝向的部分探测器，单位时间内相较基准方位角所发生的球坐标 θ, φ (忽略径长) 的移动，记为 Anv_d 。

追踪对象位置	被追踪的对象(信标)的位置, 探测器与追踪对象的位置关系依据传感器的定位机制而定, 为同一位置/存在固定关系的位置/互不相关的位置, 记为 Pos_t 。
追踪对象位移速率	追踪对象(信标)的位置与探测器不相同的, 并且追踪对象的位置是可调节的, 记为 Mov_t 。
追踪对象方位角	被追踪的对象(信标)的方位角, 探测器与追踪对象的方位角关系依据传感器的测角机制而定, 为同一朝向/存在固定关系的朝向/互不相关的朝向, 记为 Ang_t 。
追踪对象角速度	追踪对象(信标)的方位角与探测器不相同的, 并且追踪对象的方位角是可调节的, 记为 Anv_t 。

表 2 定位传感器的二级参数表

二级参数由传感器的定位机制决定, 与传感器之定位算法的数值计算细节紧密相关, 并且考虑了传感元件所安装的空间位置。定位系统不对传感元件自身的运动进行约束, 对传感元件所用算法提出的二级参数变更请求, 允许在定位过程中根据传感器测距算法需要, 调整传感元件的位置或朝向(改变二级参数)以更好地进行定位/测角, 实现“主动定位”的效果。记二级参数为:

$$S_2 = (Pos_d, Mov_d, Ang_d, Anv_d, Pos_t, Mov_t, Ang_t, Anv_t)$$

本设计所用的 VR 手柄在空间结构上集成各种传感器，因此对传感器的二级参数设置一些空间约束关系，对传感元件抽象一个传感元件插槽，不执行具体的定位/测角功能但具有一定位置/方位角参数，接入该插槽的传感器共享其位置/方位角参数，并将传感器位移，传感器朝向旋转，传感器变速位移，传感器变速旋转等各种空间操作封装在插槽上供传感器测距算法调用。

显然，传感元件插槽就是一组传感器标定关系，传感器通过绑定插槽共享一组标定数据，当测距算法需要获取传感器的绝对/相对位置时，可以通过访问插槽查询传感器的当前位置数据。

根据一级电气参数与二级追踪参数，传感器 S 的描述模型可表示为 $S = (S_1, S_2, \xrightarrow{anchor})$ ， S_1 为传感器的电气参数和数字信号输出特性， S_2 为传感器的位置、速度、方位角的数值参数， \xrightarrow{anchor} 为传感器属于哪个插槽，决定访问哪个插槽可获取该传感器位置/方位角数据。

1.1.3 二级参数的插槽模型推导

显然，传感器插槽是确定传感元件定位参数的关键因素，因此这里先对插槽的定位参数模型进行讨论：假设插槽具有两种几何形状：柱形 Cl (Cylinder) 和环形 Cy (Cycle)，分别对应两类传感元件安装方式：固定安装不再移动的、安装以后也支持一定的移动/旋转的。

其中，柱形插槽中的传感器元件是不可活动的(Passive)，而环形插槽假设在机电结构上是一组二轴旋转电机，支持以横轴为法向的旋转(roll)和垂直方向为法向的旋转(yaw)。

假设具有两个 VR 控制器，每个 VR 控制器上装备有一个柱形插槽和一个环形插槽，首先用向量表示柱形插槽的二级参数，0 表示柱形插槽的定位

参数恒定不变，括号内各项表示传感器的定位参数向量的分量：

$$Cl_{1,2} = (0(\dot{x}), 0(\dot{y}), 0(\dot{z}), 0(\dot{\phi}), 0(\dot{\theta}), 0(\dot{\gamma}))$$



图 2 VR 头显设备的三个旋转自由度示意图

参考图 2, x,y,z 表示三个轴向上的速度, ϕ, θ, γ 表示三个方位角 (roll,yaw,pitch) 上的角速度。

绑定于柱形插槽的传感器的描述模型为 $S_{Cl} = (S_1, S_2(Cl_{1,2}), \overrightarrow{Cl_{1,2}}) = (S_1, \vec{0} + (\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\gamma}), \overrightarrow{Cl_{1,2}})$, 由于柱形插槽上的传感器不可活动, 柱形插槽上只安装采取被动式非测距的加速度计型传感器的相对定位方式, 该类传感器不存在探测器/追踪对象之区分, 探测器与追踪对象相统一($d=t$), 描述模型为：

$$S_{Cl} = (S_1, \vec{0} + (\widehat{Mov_t, Anv_t})_{Cl}, \overrightarrow{Cl_{1,2}})$$

这里的 $\vec{0}$ 表示各传感器共享的柱形插槽的定位参数, $(\widehat{Mov_t, Anv_t})_{Cl}$ 表示各传感器自身分别得到的对追踪对象位移速率和追踪对象角速度的估计。

另用向量表示环形插槽的二级参数：

$$Cy_{1,2} = (r, l, \dot{\psi}, \dot{\omega})$$

其中 r 表示环形插槽的半径, l 表示环形插槽的宽度半径, θ 为绕环形法向方向的旋转的角速度, Φ 为绕环形方向的旋转的角速度。

绑定于环形插槽的传感器的描述模型为 $S_{Cy} = (S_1, S_2(Cy_{1,2}), \overset{Cy_{1,2}}{\longrightarrow})$, 环形插槽用于安装探测器与追踪对象相分离的传感器, 在目标追踪过程中追踪对象的速度和角速度一般是不可控制的, 我们可以调整的是探测器的速度和角速度, 故绑定于环形插槽上的传感器均视为主动测距的探测传感器。

对环形插槽中的探测器型传感器, 考虑绕环形法向方向上的旋转, 根据正弦和余弦定理可知:

$$\begin{cases} \dot{x} = r \times \cos \dot{\omega} \\ \dot{y} = r \times \sin \dot{\omega} \end{cases}$$

另外, 绕环形法向方向上的旋转产生一定的 θ 方位角(yaw)旋转:

$$\dot{\theta} = \dot{\omega}$$

考虑绕环形方向上的旋转, 同样根据正弦和余弦定理可知:

$$\begin{cases} \dot{z} = l \times \sin \dot{\psi} \\ \dot{x} = l \times \cos \dot{\psi} \times \cos \dot{\omega} \\ \dot{y} = l \times \cos \dot{\psi} \times \sin \dot{\omega} \end{cases}$$

另外, 绕环形方向上的旋转产生一定的 Φ 方位角(roll)的旋转效果:

$$\dot{\phi} = \dot{\psi}$$

因此, 综合两个方向上的旋转可以得到环形插槽上的探测器描述模型

S_{Cy} :

$$\begin{aligned} & (S_1, (r \cos \dot{\theta} + l \cos \dot{\phi} \cos \dot{\theta}, r \sin \dot{\theta} \\ & + l \cos \dot{\phi} \sin \dot{\theta}, l \sin \dot{\phi}, \dot{\phi}, \dot{\theta}, 0), (\dot{x}, \dot{y}, \widehat{\dot{z}}, \dot{\phi}, \dot{\theta}, \dot{\gamma}), \overset{Cy_{1,2}}{\longrightarrow}) \end{aligned}$$

绑定于环形插槽的探测器整体描述模型:

$$S_{Cy} = \left(S_1, (Mov_d, Anv_d)_{Cy}, (\widehat{Mov_t, Anv_t})_{Cy}, \xrightarrow{Cy_{1,2}} \right)$$

这里的 $(Mov_d, Anv_d)_{Cy}$ 表示传感器 S_{Cy} 所属的环形插槽的定位参数， $(\widehat{Mov_d, Anv_d})_{Cy}$ 表示传感器 S_{Cy} 对目标的位移速率和探测器角速度的估计。

这里我们可以得到一个对 VR 手柄的传感器分类描述模型，柱形插槽可以安装 N 个传感器，不可活动，而环形插槽只可安装 1 个传感器，但可以在两个轴向上自由旋转：

$$\left\{ \begin{array}{l} cl(+\vec{0}) \left\{ \begin{array}{l} S_2 = (\widehat{Mov_t, Anv_t}) \\ S^1 \sim S^n, \text{sensor array} \\ \text{count}(S^1 \sim S^n) \geq 1 \end{array} \right. \\ Cy(+ (Mov_d, Anv_d)_{Cy}) \left\{ \begin{array}{l} S_2 = (\widehat{Mov_t, Anv_t}) \\ S^1 \sim S^n, \text{sensor array} \\ \text{count}(S^1 \sim S^n) = 1 \end{array} \right. \end{array} \right.$$

基于该传感器分类描述模型可以为 VR 手柄或 VR 追踪器规范输入的定位数据，本设计处于工作量考虑，不使用测距式传感器而仅使用加速度计型的被动式定位传感器，因此也不设置环形插槽，仅设置柱形插槽表示传感器的位置和朝向关系。

本设计最终敲定采用 ADXL345 以及 HMC-5883L 作为 VR 手柄所安装的传感器，以下是这两个传感器的一级电气参数和二级定位参数表：

一级参数	帧率	功率	灵敏度	量程	最高分辨率	价格
ADXL345	0.1~3200Hz	0.216~0.36mW	230~282 LSB/g	8g ~16g	3.9mg/LSB	\$2.77/\$8.40
HMC-5883L	15Hz	85mW~225mW	230~1370L SB/g	-8g ~+8g	4.35mg/LSB	\$4.99

表 3 ADXL345 和 HMC-5883L 一级参数（电气参数）表（注：两款传感器的实际表现非常复杂，该参数表选取的是传感器主要工作模式下的平均参数或参数区间。）

二级参数	位置	方位角
ADXL345	(-2cm,+1.3cm,+1.2cm)	(0°, -180°, 0°)
HMC-5883L	(+1.8cm,+1.7cm,+1.5cm)	(0°, +5°, 0°)

表 4 ADXL345 和 HMC-5883L 二级参数（定位参数）表（注：实际上取决于手柄结构的设计如何安排传感器放置的位置，这里的位置基准是安置传感器的 PCB 板中心位置，方位角基准是 PCB 板中心点的右手坐标系，X, Y 轴在板内，Z 轴垂直 PCB 板。）

这里 PCB 板就是柱形插槽的物理实现，获取 ADXL345 与 HMC-5883L 的绝对坐标需要先获取 PCB 板的位置，基于这两个传感器的参数描述模型，下文将展开规范的对融合传感器的定位系统的建模。

1.2 VR 追踪器标准及方案

本节讨论 VR 手柄的 VR 追踪器部分的软硬件结构，参考 HTC 官方提供的 Vive Tracker 开发手册^[15]，Vive 追踪器可以采用 USB 数据线的方式向上位机提供追踪数据，本文也采用同样的数据传输渠道：

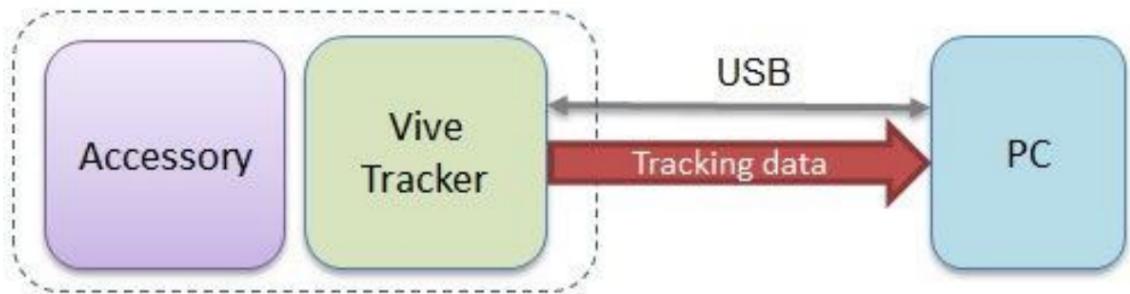


图 3 Vive 追踪器纯 USB 数据传输方式

如图 3 所示, Vive 追踪器依附于所追踪的设备 (Accessory), PC 从 USB 数据线建立与追踪器的连接并安装相关驱动, Vive Tracker 则根据驱动 (基于 OpenVR 协议) 定时向 PC 提交姿势数据包。

根据 Vive Tracker 开发手册, 从外设到 Vive 追踪器以 USB 高速模式进行, 至少提供 USB2.0 级别的总线协议, 传输的数据格式是通过 USB-HID 设备类进行特征报告传输的。类似于 Vive Controller, Tracker 两次数据传输的间隔至少大于 10ms, 这里将 Tracker 交付数据的频率设定为 12ms, 由于定位数据字节量大交付频次高, 应保证 Tracker 与 PC 通信有足够高的波特率, 本文处于简单考虑, 以下采用最高的波特率实现 Tracker 与 PC 的通信, 保证能足够快足够多地交付定位数据。

1.2.1 VR 追踪器与 PC 通信协议

本设计采用基于 UART(RS232) 传输协议的 USB 高速串口通信模式, 由上位机(PC)驱动程序发送信号, 追踪器返回定位数据的过程构成。具体参数为, 主机端驱动向 COM7 串口 (笔者电脑为其初始化的名称, 可重新配置其设备描述符修改串口名) 以 115200 的波特率发送消息, 发送的字节需满足以下的格式 (ADXL 和 HMC 为本定位系统所用的传感器):

X0 编 码位 (ADXL)	X1 编 码位 (AD XL)	Y0 编 码位 (ADXL)	Y1 编 码位 (AD XL)	Z0 编码位 (ADXL)	Z1 编码位 (A DXL)
0x78 x	0x58 X	0x79 y	0x59 Y	0x7 a z	0x5 a Z
X0 编 码位 (HMC)	X1 编 码位 (HM C)	Y0 编 码位 (HMC)	Y1 编 码位 (HM C)	Z0 编码位 (HMC)	Z1 编码位 (H MC)
0x75 u	0x55 U	0x76 v	0x56 V	0x7 7 w	0x5 7 W

表 5 Tracker 的串口通讯主机端控制协议

上述每个编码位表示一个可供传输的数据字节，X/Y/Z 表示不同的输入特征，0/1 表示同一输入特征的低位/高位，ADXL/HMC 表示该数据字节采样于哪个传感器，串口驱动根据编码位选择对应字节（例如要获取 Z0 编码位(HMC)可发送 0x77，ASCII 编码为 w）通过 COM7 串口以 115200 的波特率发送，并即时读取追踪器返回的相应数据字节。

实际上 Vive Tracker 作为 HID-USB 设备类发送 HID-report（报表）向 PC 发送数据，由于本设计采取串口通信模式故忽略相关的 Vive Tracker HID 设备描述符和报表协议，本设计所面向的软件驱动方接口仅有 OpenVR 协议，该协议源代码的 DRIVER_API 头文件中的枚举类 ETrackedDeviceProperty 定

义了适用于所有 VR 设备类的设备描述符标准, 本设计亦重编程 UART-USB 设备的设备描述符, 以遵循 OpenVR 提出的设备描述符标准。

1.2.2 VR 追踪器与传感器通信协议

同时, 追踪器定时对传感器采样, 追踪器由 FPGA 的 Verilog 模块实现, 定时并行访问传感器阵列内各单元, 本设计的传感器阵列包括两个传感器: ADXL345 与 HMC5883L, FPGA 以更高的主频 (FPGA 时钟 $>100\text{MHz}$, 传感器采用标准 IIC 时钟 $=100\text{KHz}$) 支持对采样数据的更细致处理, 如下图 4 所示:

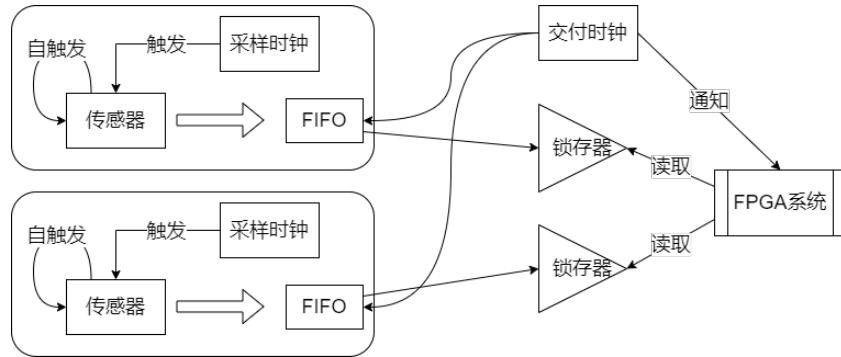


图 4 FPGA 实现的追踪器从传感器阵列采样数据过程示意图

FPGA 以传感器的 IIC 通信所能支持的最高时钟速率与之通信, 并将之采样数据缓存到 FIFO 中, FIFO 限定深度, 后至数据挤入 FIFO 并移开最先放入的数据。具体上, 两个传感器模块六个数据各设置一个 FIFO 模块。

FPGA 高速时钟必须足够快, 以保证高速翻转的采样时钟。在集合 $\{t_{s1}, t_{s2}, t_{spl}\}$ 中, t_{s1}, t_{s2} 分别是两个传感器给出一个采样值的估计时间, t_{spl} 是定位系统向各传感器索取采样值的估计时间, t_{spl} 小于 t_{s1}, t_{s2} , 也可能介于这两个传感器采样周期之间 (本设计所预设的传感器异质性的异频率性), 也可能较 t_{s1}, t_{s2} 更长。为能辨识到这三个采样频率, 设置 t_{res} 用于作为参考时钟的周期, 每个 t_{res} 使参考计数器数值加一, 并且 t_{res} 对 t_{s1}, t_{s2}, t_{spl} 有大小关系:

$$t_{res} \ll \min\{t_{s1}, t_{s2}, t_{spl}\}$$

设 t_{res} 的计数器为 cnt_{res} , t_{s1}, t_{s2} 的计数器分别为 cnt_{s1}, cnt_{s2} 。 t_{s1}, t_{s2} 的传感器计数器数值表示这两个已经发生的采样次数。由于 cnt_{res} 比 cnt_{s1}, cnt_{s2} 增长更快, 可以驱动滤波器综合一段时期内的传感器数据得到更稳定的采样数值, 以最简单的均值滤波为例, $\frac{\partial cnt_{res}}{\partial (cnt_{s1} + cnt_{s2})}$ 决定了均值滤波的滤波窗口能有多大, 进而决定采样滤波的稳定性。

本文在此提出一种面向高速采样滤波场景的简单两期滤波算法。该滤波要求 FIFO 记忆两期传感器数据, 滤波输出值的根据 FIFO 所记忆的前两期的采样点连线与采样时间垂线的交叉点计算。例如, FIFO 中的数值为 $\{Val_{nt}, Val_{(n+1)t}\}$, 此时已经进行的采样时间为 t_{spl} , 那么该传感器向系统交付的值为 $Val_{(n+1)t} + (t_{spl} - (n + 1)t) \cdot \frac{Val_{(n+1)t} - Val_{nt}}{t}$, 这是一种快速的线性拟合方法。在采样时间彼此相差不大 ($\{t_{s1}, t_{s2}, t_{s3}, t_{spl}\}$ 都差不多) 时, 这种方法适合存储资源紧缺、处理速度快、累积误差不大的传感器滤波需求。

1.3 VR 控制器 (手柄) 标准及方案

参考 Vive Tracker 开发手册^[15]提出的 VR 外设与 Vive 追踪器的命令流, Vive 追踪器应接收来自 VR 外设的命令流:

字节编号	数据内容	备注
0	标签编号	表示送出数据版本。当前数据格式版本默认值是 0
1	按键	扳手键 0x01
		缓冲键 0x02

		菜单键 0x04
		系统键 0x08
		平板键 0x10
		平板键按下 0x20
		预留 0x40
		预留 0x80
2	平板 X 值	int16 值范围： -32768~32767
3		
4	平板 Y 值	int16 值范围： -32768~32767
5		
6	扳机键原 始值	uint16 值范围： 0~65535
7		
8	电量水平	预留
9		

表 6 Vive 跟踪器外设的数据格式^[15]

依据该命令流标准，本设计的 VR 控制器（VR 手柄）应支持交付 9 字节的字节流，9 字节即 72 比特，HIDEMI SHIMIZU^[16]的研究表明，越高的时钟频率平均可以带来越高的键盘响应时间。由于 USB2.0 全速传输协议提供了理论最高 12Mbps 的速率，平均到毫秒级为 12Kbit，显然可以支持足够快的键盘响应速度。研究^[16]显示主流键盘扫描时间在 2.96~10.30 毫秒内，同时 Vive Tracker 开发手册^[15]提出的 Vive Controller 传输速率标准，两次向 USB 交付数据的间隔不小于 10ms，出于使 VR 控制器与 VR 软件端主流需求适配

的目的，本设计将 VR 控制器键钮扫描时间设定为 12 毫秒。

1.3.1 VR 控制器（手柄）按键布局及消抖处理方案

对 VR 控制器的按键具体布局，参考 Vive Controller 按键布局图：

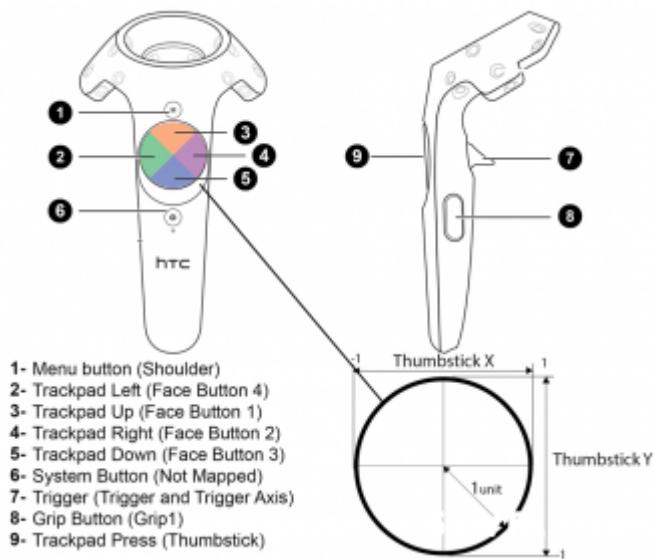


图 5 Vive 控制器（VR 手柄）的按键布局^[15]

借鉴 Vive 控制器的按键布局，本设计得到如下按键布局样式及功能说明表：

按键序号	按键功能	按键样式	实现元件
1	菜单键，发送呼出菜单的信号	光滑的圆形按钮	微触开关 四个引脚分别接线到电路板
2~5	圆形触控板的四个方向键，通常用于选择方向、迅速旋转	粗糙质感的圆面，边缘硬朗	JoyStick 模块 X,Y 轴向模拟量输出引脚

	转		接线到电路板
6	系统按钮, 功能尚未定义	光滑的圆形按钮	微触开关 四个引脚分别接线到电路板
7	扳机键, 通常用于使用物品的行为	光滑的、带弹簧的扳机形状按钮	HM0810 电镐扳手型开关引脚接线到电路板
8~9	侧键, 通常用于配合扳机键, 选择物品的行为	光滑的、椭圆形平面的柱状的按钮	微触开关 四个引脚分别接线到电路板
10	圆形触控板的整体按下, 配合扳机键的物品行为	粗糙质感的圆面, 边缘硬朗	JoyStick 模块 Z 轴数字信号输出引脚接线到电路板
11	调试键, 按下调试键迅速通知终端激活任务	光滑的圆形按钮	微触开关 四个引脚分别接线到电路板

表 7 VR 手柄按键布局、样式和实现

本设计由 **FPGA** 实现控制器处理程序, 包括按键消抖和与主机端的通讯协议, 按键消抖采用边沿检测法, 对按键 **button[7:0]** 的值打四拍寄存器 **key_r** 并对第二拍和第三拍求与, 以检测按键值波形的正负边沿 **key_pos** 和 **key_neg**,

用类似的办法打两拍寄存器缓存 button[7:0]并每隔 12ms 就进行一次采样，这样用很少的寄存器资源和与门就可以满足稳定扫描按键键值的需求：

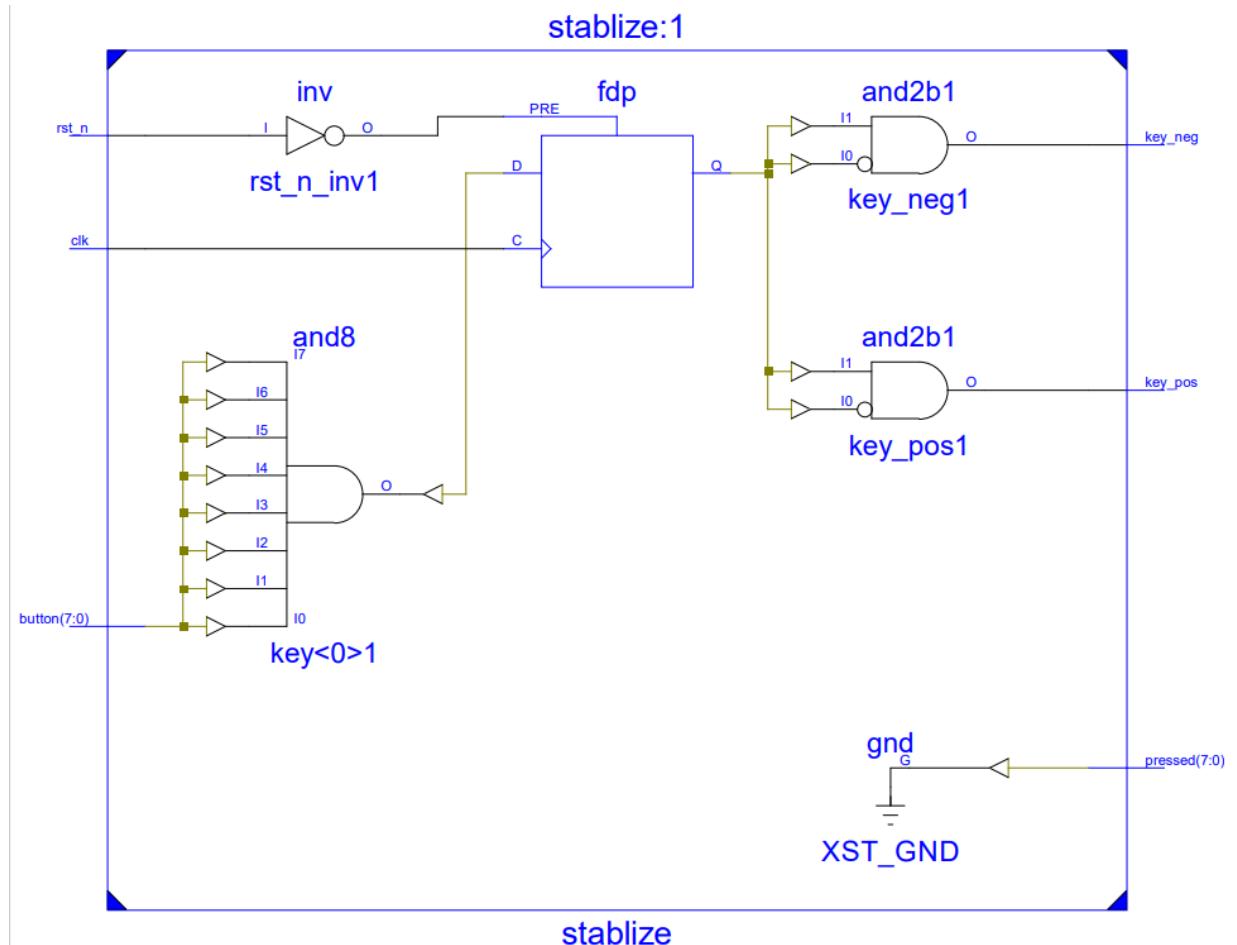


图 6 按键消抖 Verilog 原理图 (模块名: Stablize)

1.3.2 VR 控制器 (手柄) 与主机端通讯协议

根据该按键布局，仿照表[1]的 Tracker 的串口通讯主机端控制协议得到 Controller 串口通讯主机端控制协议：

X 方 向编码位	键盘 扫描编码 位 1	键盘 扫描编码 位 2	Y 方 向编码位
0x7B	0x5B	0x5D	0x7D

{	[]	}
---	---	---	---

表 8 Controller 串口通讯主机端控制协议

用两个字节(int8 数据类型)可以表示 Controller 的圆形触控板 X,Y 两个方向的运动的模拟量, 用一个比特可以表示圆形触控板 Z 轴方向上的按压, 将按键{1,6,7,8,9,10,11,Z}用一个字节: 键盘扫描编码位进行表示, 键盘扫描编码位 2 保留不用, 即本设计对 Controller 如何与主机端交流信息的协议设计。

第二章、定位融合算法及系统

给定 N 个传感器数据流, 略过各传感器数据流各自的数据过滤步骤(滤波处理), 本文的目的在于设计一种特征级和决策级融合数据的深度学习方案, 使这种信息融合满足以下两个目标:

- (a).即插即用, 对新插入的未知分布的传感器快速将其进行特征级融合;
- (b).异质适应, 不要求传感器的数据在单位或者格式上进行标准化的处理;

假设这 N 个传感器的任意一个都可以表示为, 按照某种采样频率生成的观测数据序列, 有:

$$S = \{\vec{s}\}_i$$

其中 \vec{s} 是每次采样得到的观测数据向量, 这一向量的宽度根据传感器具体类型有所不同; $\{\cdot\}_i$ 表示根据指标集 i 进行的采样, 指标集 $i = \{jT | j \in N\}$, 其中 T 为采样频率。

设每次进行信息融合的数据个数为 M 个(批的大小), Bubeck 和 Sellke 的工作显示^[17], 要平滑地拟合一个 D 维空间中的 N 个数据点, 需要 $N \times D$ 个待定参数, 因此本文设计一个过参数化的神经网络, 以保证神经网络的容量

足够大能拟合复杂的传感器组合情形。

作为示例，信息融合方案的“异质适应”特性要求传感器的输出可以出现如下情况：

$$\left\{ \begin{array}{l} \text{传感器①: } \vec{s} = (z, x, y) \\ \text{传感器②: } \vec{s} = (\varphi, \omega, \theta) \\ \text{传感器③: } \vec{s} = (10 * x, 10 * y) \\ \text{传感器④: } \vec{s} = (?, ?) \\ \text{目标输出: } \vec{t} = (x, y, z, \varphi, \theta, \omega) \end{array} \right.$$

其中 xyz 表示轴向速度， $\varphi\omega\theta$ 表示欧拉角速度，? 表示传感器输出的物理意义不明。

特征级融合的目的是，寻找一个合适的非线性函数将传感器①~④变换为姿势矩阵 $\begin{pmatrix} x, y, z \\ \varphi, \theta, \psi \end{pmatrix}$ 的格式，采用深度学习作为模型，则目标输出特征数为 6，而输入特征数则为各个传感器的特征数之和。决策级融合的目的是，根据特征级融合后输出的估计向量与真实向量进行比对，提供高精度和低时延的输出。

将特征级融合的方案视为一个融合编码器，由一些神经网络层构成。对 \vec{s} 记其特征个数为 $|\arg(\vec{s})|$ ，记存在 I 个传感器输入分别为 \vec{s}_i ，则融合编码器的输入层特征个数为 $\sum_{i=1}^I |\arg(\vec{s}_i)|$ ，输出层特征个数为 6，融合编码器可表示为：

$$\begin{aligned} \text{Encoder: Layer} \left(\text{input_size} = \sum_{i=1}^I |\arg(\vec{s}_i)| \right) \\ \rightarrow \text{Layer}(\text{output_size} = 6) \end{aligned}$$

根据上文提出的过参数化训练策略，设计一个过参数化层序列，使参数总数为 $\sum_{i=1}^I |\arg(\vec{s}_i)| \times N$ 个，使得 Encoder 可表示为一个 $N+2$ 个全连接层的

深度学习模型：

$$\left\{ \begin{array}{l} InputLayer \left(input_size = \sum_{i=1}^I |arg(\vec{s}_i)|, output_size = \sum_{i=1}^I |arg(\vec{s}_i)| \right) \\ N \times HiddenLayer \left(input_size = \sum_{i=1}^I |arg(\vec{s}_i)|, output_size = \sum_{i=1}^I |arg(\vec{s}_i)| \right) \\ OutputLayer \left(input_size = \sum_{i=1}^I |arg(\vec{s}_i)|, output_size = 6 \right) \end{array} \right.$$

该深度学习模型的训练过程是执行 MiniBatch-Training 的批训练，每一批训练 N 个数据计算平均 Loss，再根据 Loss 反向传播更新各层参数。

给定以上的训练方法，如何确定结束该训练过程的条件即停止参数更新的收敛条件？由于该模型应满足即插即用的要求，预设一个测试数据集判断神经网络是否收敛的思路并不合适，应根据实际插入的传感器阵列调整收敛条件，尽管不能保证动态变化的传感器阵列能存在收敛的可能，但可以从每批训练数据进行重采样并添加一定噪声作为测试集，用以可能的神经网络准确度测试需求。

本文的信息融合方法重在即插即用和异质适应而非准确度一个指标，测试数据并非刚需，下文的活性连接技术是本文的信息融合技术的核心，该方法支持实时地训练神经网络，由用户决定神经网络的准确度是否满足需要、是否停止训练。

2.1 游戏化和数据众包

本设计所用的定位融合算法之核心就在于引入活性连接技术，从多模态信息融合的语境理解，就是采用游戏化和众包技术完成多个传感器的特征级融合，这也是本设计的创新所在。以下分别讨论游戏化技术、数据众包技术以及结合两者应用到神经网络数据生成的活性连接技术。

2.1.1 游戏化概念及技术

游戏化通常被认为是在非游戏情景下对游戏元素的使用，但这一定义忽视了游戏化的非技术因素；Kronisch 提出^[18]，游戏化应理解为任意使用者出于好玩而进行的互动性过程。游戏元素可以分为形式元素和戏剧元素，前者是交互方式，后者是沉浸感的来源，单一用形式元素（各种游戏机制的设计）指导游戏化过程是市场化的庸俗理解，游戏化应当为用户的游戏体验创造机会、为非娱乐性目的提供新动机。

用户的体验千人千面，因此，有效的游戏化设计依赖于对用户的有效建模，Claudia et.al. 发现^[19]并不存在用户类型和游戏化设计的完备且一致的关联模式，通过用户类型定义和游戏化设计分类技术建立两个分类集合，并发现两者的特定组合（例如“游戏者+个性”和“游戏者+生态”）存在统计学关联。

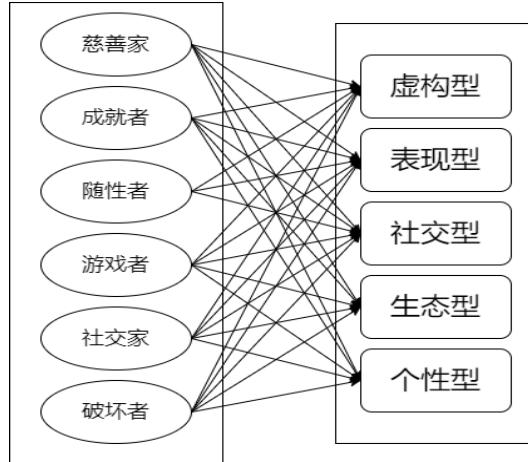


图 7 用户类型和游戏化设计维度的映射^[19]

在游戏化的游戏元素设计方面，Marlena et. al. 提出一套分析现存的游戏元素以及派生的新游戏元素的迭代流程并运用到了对一个在线学习应用程序的界面设计^[20]。Blohm et. Al.发现适当的游戏元素设计可以改变用户行为，实现 IT 应用的服务增强的效果^[21]。Strmečki et.al 的游戏化改造对在线学习

App 服务效果的提升^[22]可作为此观点的一个例证。尽管游戏化是更高效的学习流手段,但为此适配的游戏元素设计量和设计者工作量是较大的^[23]。因此游戏化领域发展出许多自动化方法简化游戏化过程的工作量问题。

王雅倩提出一般的软件游戏化分析方法^[24],对需求模型分解为活动,定义用户行为的度量指标(例如执行了某个动作的次数)和评价方法(采用游戏元素如积分勋章排名激励用户):

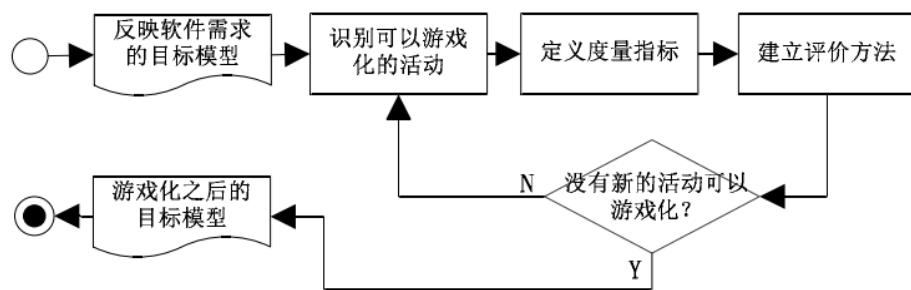


图 8 基于目标的软件游戏化分析过程^[24]

在游戏化的机制设计方面,Knutas et. al. 提出一套基于关联规则的 CN2 规则归纳器对用户个性化的设计流程^[25]: a.选取游戏化的个性化策略; b.启发式地提出若干个设计目标; c.构造“游戏化”任务的形式与分类(算法的训练数据的结构); d.个性化策略的算法实现。

在游戏化的评价方面, Villegas et. al. 提出基于游戏化机制和玩家动机的相关性^[26]估测一个游戏化过程是否足够有效的方法。刘俊提出了量化估计游戏化教学中游戏元素与行为动机关联性的方法^[27]。

在游戏化的众包方面, Haworth et. al. 认为^[28], 架构设计是 NP 复杂问题,但参与的众包人员可以作为动态地解析问题的资源,参与目标问题的解空间的搜索,这种做法存在的缺点是作为一种启发式算法会丢失很多有用的解,众包过程的游戏化可以缓解这个问题。作为例证, Bayas et. al. 设计了游戏化

的众包 App 收集地理数据的算法^[29]，尽管用户提供的数据质量并不高，但只要稍加拍摄技巧学习就能为数据集获取大量有用的细节样本。

显然，众包与游戏化的结合就好像发生在现实环境的多主体遗传算法，强大的游戏化机制可以积极调用用户的认知资源，产生满足实时性和多样性要求的数据。并且必须使用户的主动性通过丰富的交互机制充分发挥出来。

单调地使用用户资源只会使复杂的用户退化为行为简单的“粒子”，就会成为一个简单的启发式算法，退化后的用户众包数据还不如用模拟采样的算法生成一堆训练数据，游戏化众包的意义就在于提供足够丰富、难以预测的行为数据。

在下文的活性连接技术中，用户的认知资源甚至是一个基本假设。活性连接技术中的翻译器（Translator）充分考虑了游戏化过程应具备的标准和游戏化机制的设计，提供了实时渲染窗口和游戏任务窗口，并同时提供了命令行输出和 3D 图形界面输出，这可以为用户充分发挥自身的能动性提供良好的游戏化环境。

2.1.2 数据众包概念及技术

数据众包（Crowd-sourcing）是互联网和社交媒体时代兴起的，获取人工智能算法所需的训练数据的策略。利用这些新兴数据源的搜索算法相继被开发出来^[30]。典型的利用众包数据作为软件系统数据源的包括：生物医药领域的 BioSharing 维护有关各种生物代谢物的记录^[31]；地理信息领域的匹配众包数据和权威机构地理数据的算法、以及众包地理建模和灾害预测的技术^{[32][33]}；出行领域的采用众包数据训练旅行向导系统的服务^[34]；文化领域的非结构化数字地方志资源的众包收集和数据规范化^[35]。

在物联网层次上，数据众包可分为三大层：感知层、网络层和应用层：

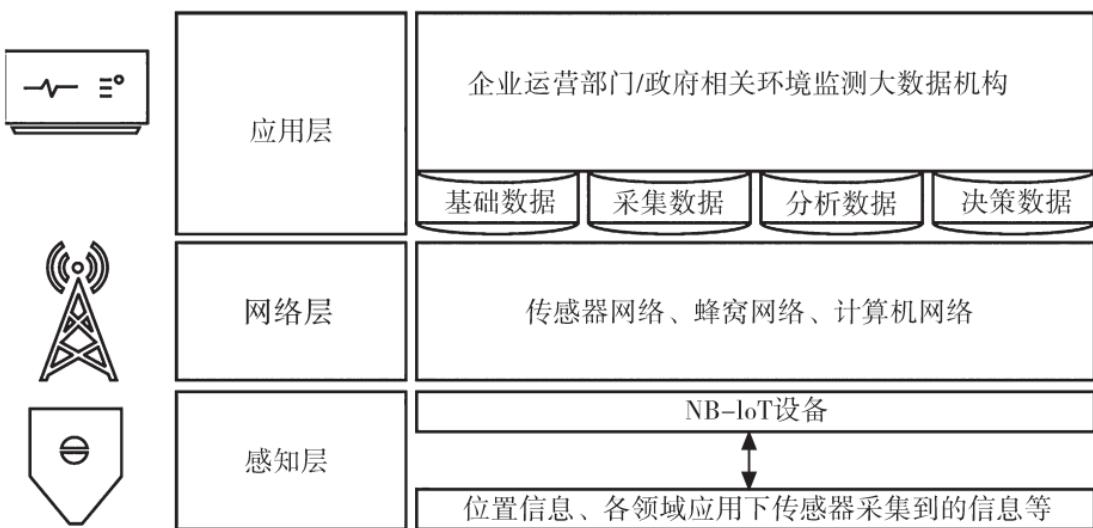


图 9 数据众包流程的物联网层次划分示意图^[36]

感知层是终端的物联网设备，在差异化的场景中由传感器采集数据；网络层是感知层和应用层的交互渠道，将终端的数据进行编码、加密、传输；应用层是用于部署与开发各种物联网服务的大数据机构，负责接收、分类和评估来自终端的众包数据^[36]。

众包数据的涌现要求对众包数据的有效利用和评估，基于该目标研究人们在使众包数据与人工智能算法适配、众包数据的清洗和预处理、众包数据与专家数据的比较等多个方面进行了研究。

李绍园,姜远考虑了多标记数据的众包学习^[37]，认为多标记众包学习的目标是从多个有偏标注中估计样本的真实标注，多标记众包相较于单标记众包在于多标记存在相关性，足够多的标注者会使这种相关性涌现出低秩结构，可用低秩标注结构近似原标注结构。

叶晨等人的研究^[38]基于真实众包平台认为数据清洗技术应具备三个条件：通用性（与标签分类器的领域无关）、批量性（流处理）、容错性（标注

存在噪声甚至恶意偏差), 设置出一个众包处理机制, 机器学习模型评估最有众包价值的数据交付人工平台进行标注, 工人对算法给出的结果进行反馈, 众包优化算法综合这些工人的结果得到最终反馈的标记数据, 因此完成一个数据清洗的流程。

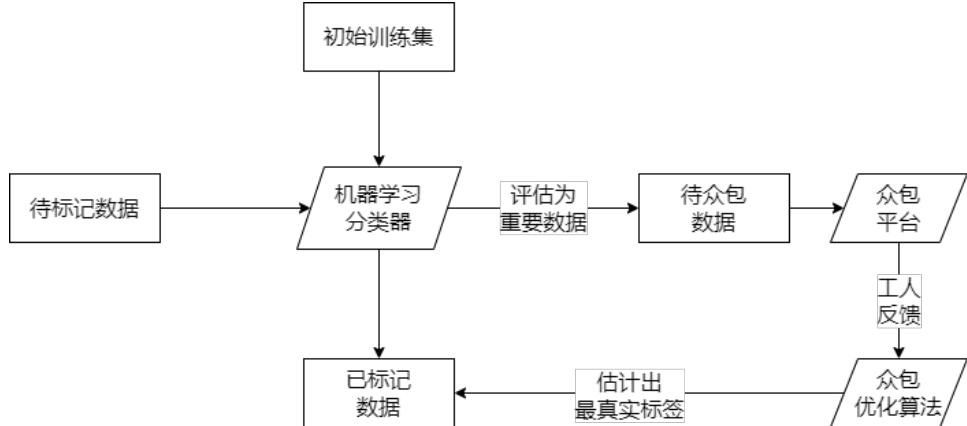


图 10 众包清洗数据的机器学习算法及众包反馈机制^[38]

对众包数据标注过程中存在的数据偏差问题更为系统的研究参考张静^[39], 讨论了对非真实标注 (二分类标注为主) 的真值推理和估计问题。

由众包数据与专家数据的比较可以引出众包数据的质量控制和质量评价问题, 面向数据质量的众包系统 (评价质量、分配众包任务、控制质量) 是一种有效的众包系统设计方法。研究者采取特殊的任务分配策略评价众包工人的质量, 区分出恶意众包工人与优秀众包工人, 同时构建工人的专业度向量, 以便为工人分配最合适的众包任务^[40]。在质量控制问题上, 研究者为众包样本数据动态分配差异化的学习权重, 以抗衡数据点疏密的不平衡, 面对极端的众包任务场景更具稳定性^[41]。

综上考虑, 一个有效的众包系统应当满足以下标准:

1. 可扩展的, 支持众包任务, 但对某些专业领域保留一定的领域分区;

2. 高度可交互的，充分动员用户的认知资源以标注数据；
3. 任务分配最优的，能评价不同众包者的时间、效率、费用和准确度，构建众包工人画像，为其分配最合适的工作；
4. 质量管理合格的，对众包者的带偏差标注进行综合，去除极端和恶意标注，消除冗余标注使数据疏密程度恰当。

全子飞开发的通用开放性众包平台 Linne 就是符合这些标准的一个软件实现^[42]，可作为众包平台的参考。本设计的众包同样是一个通用开放的众包框架，但该框架并不对众包者做很多量化工作，也不为这些人设计评价机制和众包者数据画像。该框架所重视的并非众包者的经济价值规划：众包者领多少钱、做多少事以及做哪些任务。实际上，本设计的传感器系统收集数据完全可以由个人完成而不需一群众包者，我们关注的是众包的关键特征：将训练数据的获取划分为离散的任务包，本文提出的算法正是基于训练数据标注的任务化思想所发展的，但具体执行又有所差异，传统众包技术一般要求众包者的标注要尽量靠近真实估计，并以此为基础展开优化过程，要么是让数据向众包者靠齐，要么是让众包者向数据靠齐，本文的活性连接技术所用的众包理念却不包括这点假设，具体的活性连接见下一节。尽管如此，上文提到的诸多数据众包理论和技术，尤其是综合的众包系统评价标准，是下文的活性连接技术的生成器设计的出发点。

2.2 验证式调试器及活性连接技术

本文所用的信息融合算法主体是神经网络模型，神经网络模型应用的关键是为神经网络提供足够多的高质量训练数据，而训练数据集一般是事先得到用于训练神经网络的，基于本设计所用的神经网络并非离线场景，故不能

采用预设的数据集而应采用预设的数据生成的策略。

回顾“即插即用”和“异质适应”的要求，两者一般是相互矛盾的，“即插即用”要求对各传感器组件预先配置参数以满足传感器管理系统的需要，同时各个组件必须在很多数据结构层面同构，“异质适应”则反之，不要求各个组件的数据结构同构，也不需要复杂的初始配置/参数，但为得到每一套组件的方案如何配置各个组件的作用，都需要一个长周期的动态计算和更大的空间复杂度（保存生成的参数）。

2.2.1 验证式调试器流程

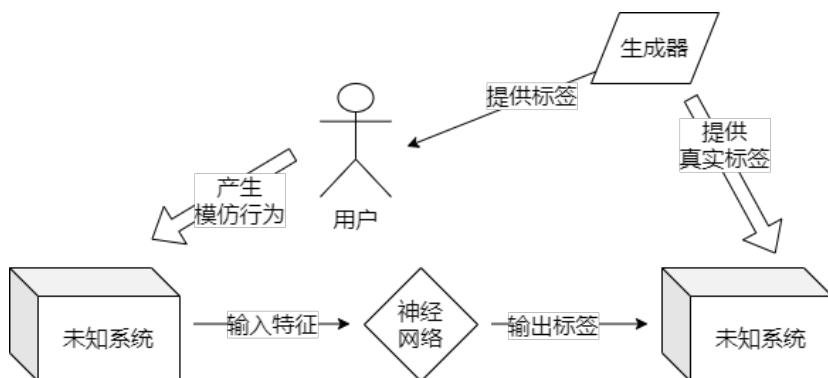


图 11 验证式调试器工作流示意图

对此问题本设计提出“验证式调试器”的方案，“验证式调试器”的理念是，调试器根据某个预定策略，生成一组目标标签，用户参考生成的目标标签发生行为产生实时数据，实时数据流入神经网络，输出的估计标签与目标标签的差值作为 Target Loss。

“验证”的意思是用户为提高手柄的定位准确性，需要不断模仿系统所提示的行为，“调试”则是系统会考虑用户已经产生的训练数据，选定一些欲优化的方向，生成一些新的目标标签供用户生成。

根据对验证式调试过程的定义，以下可以得到“验证式调试器”实现的若干问题：

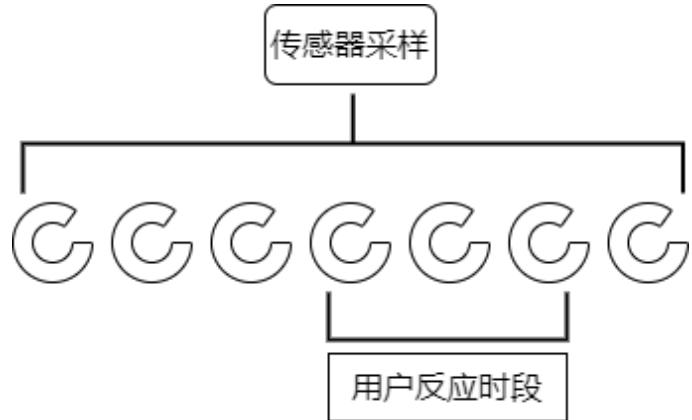


图 12 用户时间与数据时间对齐问题示意图

1. 对齐用户时间与数据时间。例如传感器可能数毫秒内返回一个采样点，而用户的反应时间和操作时间可能长达数百毫秒至数秒，需要识别用户产生目标标签的时间区段。
2. 生成区间关键点识别。用户动作的开始，关键帧（动作的转折点）和结束都在数据上是可识别的，我们可以将用户动作和标签数值建立稳定的一一对应。
3. 目标标签正则化。不允许复杂的目标标签，以方便用户理解标签并容易生成行为数据，例如，每次给出的目标标签仅涉及一部分特征的变动，手柄只做几个轴向的旋转或移动。
4. 可视化。总是为用户可以方便地做出动作而考虑，根据目标标签生成一副时长控制在数秒内的动画，并且附加一些辅助图形帮助用户理解在哪些方向上旋转和移动物体。
5. 保收敛。若干次调试训练后神经网络在测试数据集上的 Target Loss 趋向于稳定，假设测试数据集与生成器提供的批训练数据采样自同一概率分布，那么测试数据即为批训练数据的带噪重采样，故若干次调试训练后的神经网络应在批训练数据集的带噪重采样上估计的 Target Loss 逐步缩小。

引入验证式调试器的核心待定问题是如何生成一组新的目标标签。分别假设传感器产生的输入、参数、神经网络输出和目标输出为 x, w, y, \tilde{y} ，由于用户总是在模仿调试器所给出的动作，故 y 在一轮轮的 BP 优化下存在向 \tilde{y} 靠近的趋势，而 \tilde{y} 是可控的，因此神经网络会存在一定的可变性。

传统的神经网络通常假设训练数据集 x, \tilde{y} 符合一个未知的待学习的概率分布 $P(\tilde{y}|x)$ ，这个概率分布是几何上平滑连续的。但 \tilde{y} 是由调试器生成的，我们并未规定 \tilde{y} 需要按照某个概率分布来生成，为此我回到验证式调试器的语境以解决该问题。

2.2.2 活性连接技术原理

验证式调试器的数据流来源是使用户模仿计算机生成的标签而产生数据，将计算机生成的 \tilde{y} 与用户模仿而产生的 $y(w \cdot x)$ 合为一个二元组 (y, \tilde{y}) ，称为一个活性连接（也作 Livelink）。

活性连接是可分解的，假设这个活性连接建立在三层神经网络上，也即 $y = w_3(w_2(w_1 \cdot x))$ ，那么对每层都可以构造一个活性连接，活性连接的分解等价于，构造多个细粒度的活性连接且并行运作。

本文不讨论一个层或多个层构造 Livelink 的区别，仅将多层神经网络视为一个整体的黑盒，使用总的 Livelink 对整个神经网络产生训练数据，以减小问题复杂度。

通过对活性连接的定义可知，在神经网络体系中，活性连接与神经网络的优化器无关，而有关训练数据的获取过程。下面我抽象一个问题情景以便更好地阐述活性连接技术的应用场景。

假设存在两个结构未知的现实中的黑盒系统，要么因为系统的高复杂性，

要么因为系统的高随机性，甚至单纯假设为不可知，对这两个系统无法进行很好的建模。

然而，由于该两个系统处于现实中，可以主动地彼此建立关联，例如本文中从多个传感器到 6 轴定位，可采用神经网络连接两个系统（传感器阵列和姿势系统）的输入和输出端口，同时神经网络的数据来源于主动建立关联的过程所产生的数据，这就是“活性连接”技术的背景。

“活性连接”技术的关键在于如何确定合适的标签生成器，单一重复的标签显然只能让神经网络处理生成器所重复生成的简单样例。应设置合理的标签生成策略，使神经网络可应对充分多的情景。

确定标签生成器要求先定义用户的性质，标签生成器主要是一个引导设备，实现数据生成过程的是用户，用户的非机械性使得用以拟合 Loss 的实际标签值 Real Label 与用户实际模仿的目标标签值 Target Label 存在一定差异。

引入一种流形的现代技术作例子来考虑这个标签值差异问题：机器人仿真技术。机器人仿真所用的数据源于图形学的物理仿真技术：先构建一个虚拟环境，接着改造这个环境以定制一个充分多样的虚拟世界，部分或全部地代替实地模拟。

以数据生成的角度作为切入点，从活性连接技术的视角看，机器人仿真使用了一个“虚拟用户”：仿真环境，仿真环境发挥了“主体能动性”为机器人提供训练数据，生成器则是视工程师的理解而定，工程师根据自己的需求确定仿真环境的布景和道具。

然而该种“虚拟用户”的解释并不符合活性连接技术的观点，活性连接

技术认为提供训练数据的“用户”必须具备主体能动性：能充分理解生成器给出的标签并做出行为模仿，这种模仿的相似度是与目标一致的（无恶意曲解），并且实时地迅速地做出响应。

为此需要建模“仿真环境”本身，将该对象黑盒化，视作一个未知的复杂系统，并设置一种仿真环境的表示方案，将仿真环境的一个表示进行黑盒化，也视作一个未知的复杂系统，创建一个神经网络使其输入是仿真环境的表示，是黑盒，输出是仿真环境，同样是黑盒。

2.2.2.1 活性连接技术与游戏化

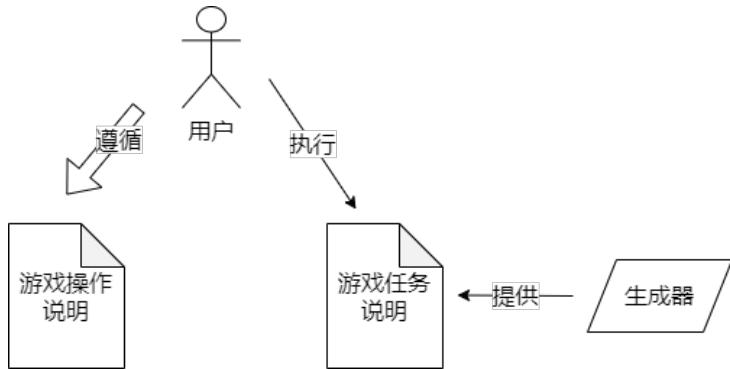


图 13 活性连接技术的游戏化解读

从游戏化的角度理解活性连接技术，生成器是一个任务系统，激发用户遵循其给出的任务目标，使用户产生一定的行为，用户操作一些对用户来说并不需要清楚其内部构造或功能的游戏道具，以完成一系列的游戏任务，这些任务整体上会覆盖一个概率分布 $P(\tilde{y}|x)$ ，也即活性连接技术应提供的待学习概率分布。

对于机器人仿真的仿真环境而言，要做的就是发明一个“游戏”，使得用户使用一些小工具和构成元素（比如砖块、障碍物或者一些高级物体）来搭建一些各种情形的仿真环境，这些搭建仿真环境的任务足够充裕和多样，进而为神经网络提供充分的训练数据。

假设用户一定具备“主体能动性”，也即可以充分理解生成器给出的标签并产生模仿性行为，用户作为程序资源就被标准化同质化了，而系统的复杂度因此就集中于生成器上。

如果不允许生成器给出的“游戏任务”过于复杂，那么就只能将一个游戏分解为多个游戏实现，多个游戏分别运行一个更简单的生成器，或者也可以支持生成器达到一定的复杂度，缩并游戏系列规模。不同的生成器复杂度设计应以一个目的为出发：能更好地实现理想用户假设。

具体上生成器给出的任务应能使用户模仿行为的丰富性和实时性达到一个权衡，使用户方面可以较好地满足“主体能动性”的假设，一个生成器首先不能过于复杂，这会让用户难以理解疲于应付，其次也不应太简单，这会无法充分发挥用户的能动性。

用户的能动性在活性连接技术中的地位是一种一般化资源：一定通用性的复杂认知能力。

本文的定位系统假设用户资源较为紧张并且本设计所欲实现的定位系统的输出特征空间也仅有 6 维、不拓宽特征空间，因此游戏任务的复杂度可以得到控制，可仅采用一个游戏过程来生成神经网络的训练数据。

从游戏过程的用户和神经网络协同角度考虑，用于训练定位系统的神经网络的数据必须是完整的，也即传感器的输入有效当且仅当相应的标签数据同时在发挥作用。

这就是训练过程的关键点识别问题，包括确定训练（每一次任务）开始的时间点和结束的时间点，也可以是中间某个关键的特殊时间点，必须确定每一批用于训练的数据点具体是如何，在物理上分离的神经网络训练环境和

用户模仿环境中统一的。

关键点的识别工作可以放到软件或硬件中完成，例如，可以在软件中进行一些初级的动作响应识别机制，当输入特征的某些维度产生超过阈值的变化即认为已抵达动作关键点，这种做法类似于消抖处理算法：超过某个阈值的信号被视为产生了需要的数据。但笔者认为应该在硬件中完成关键点识别工作，将关键点识别本身作为一个用户自主决定按按钮开启的游戏任务。

这是出于保持游戏化过程：训练数据的生成应完全来自于未知系统，即使是数据生成的预处理步骤也应该尽量表达为游戏元素，如无必要勿增实体。

因此，对关键点识别问题专设关键点开启按钮，用户自按下至松开交付一段过程的训练数据（传感器阵列交付最近的传感数据、生成器交付最近的标签数据）。这种做法实际上是将关键点识别问题开放化游戏化，避免过早设置的算法使程序向用户封闭重要的游戏细节，这会违背游戏化的交互性原则。如要提高关键点识别的效率，游戏化的做法就是重新将关键点识别过程提取出来进行游戏化改造。

规范的训练过程还必须确定生成器产生标签的具体机制，生成器一方面需要向用户提供可理解的操作，另一方面需要向神经网络提供方便数值运算的标签。

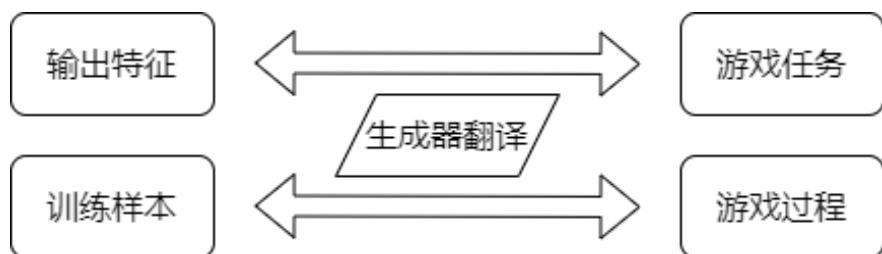


图 14 生成器的翻译功能解读

基于该角度，生成器是一种翻译器，将方便神经网络计算的特征向量 \vec{y} 转

化到用户可理解的“游戏任务”，将神经网络一轮轮训练过程的样本点 (\vec{x}, \vec{y}) 转化为一批批“游戏过程”。

2.2.2.2 活性连接哲学

这里引出的问题是如何让生成器有效地执行这个“翻译”功能？在此作者首先引出一些复杂系统领域的工程哲学观，为翻译功能的实现获取哲学意见。

回顾前文提及的两个“未知系统”/“复杂系统”，分别作为神经网络的输入和输出，在定位系统的设计中，“输入未知”是指传感器从现实物理空间中的振动，到定位数据进入算法的逻辑模块，这一个未知的数据流过程，“输出未知”是指定位数据流出逻辑模块，重新表示在现实物理空间（屏幕、机器人电机等）的未知数据流过程。

这两个未知是逻辑性未知，即永远无法将这两个过程精确表示为逻辑模型。这实际上是一种精简手段，将模型的专注点放在“活性连接”的构造上，而使其他部分尽可能地自由不受限。

与自由相对的是不自由或者受限，也即各种琐碎的标准、协议、语法和算法环节，这些都是工程师在开发其所预期的系统时面对 BUG 和例外情况所做的各种妥协和修补。

所有为解决预期中系统存在的问题的行为都是交互，工程师所写的复杂算法就是不断地调试和交互中的痕迹与成型，是软件进化完毕的完成式而非软件进化的进行时，但这样的最终成型未必能解决实践中不断涌现的新问题，甚至不能验证该软件真的完善地解决了预期中存在的 BUG（参考图灵机停机定理），为此就需要验证。

有效的验证不在于绝对的正确性，因为根据图灵机停机定理，对庞大系统的行为做有效且快捷的判定是不可能的。

现代软件发展的路径明显是追求快捷性而权衡有效性，各种轻量级软件的开发，以及内部存在复杂依赖和大量复用的现代软件生态，两者可以印证这个观点。包括深度学习的神经网络技术目前的应用困难主要在于可解释性不佳和复杂的训练流程、漫长的训练周期，未充分动员其用户群体的能动性。

因此，在工程哲学上，活性连接技术的目的是在现代神经网络技术的基础上将神经网络模型轻量化、可交互化。活性连接技术必会对已有的神经网络工作流有所革新，但又基本由神经网络模型组成。

2.2.2.3 活性连接技术与数据生成

以人脸识别的情绪分类（Emotional Recognition）任务为例，在活性连接技术的语境下，输入 x 是未知的摄像头和通信算法得到的影像信息，导入到终端的神经网络；神经网络的输出 y 是情绪标签，生成器将数值化的情绪标签翻译为用户可理解的情绪表现，用户根据情绪表现做出相应的表情或者展示相应的图片，以牵动摄像头传感器截取信息 x ，得到一批完整的训练样本 (x, y) 。

作为对比，常规深度学习训练项目中也基本保持该流程，标注人员为图片 x 打上标签 y ，每一个 (x, y) 组都可以作为一个训练样本，另有一部分作为测试样本，评估神经网络的表现，以帮助工程师对超参数一定的调整。这些超参数调整，在训练数据调整方面，就包括数据增强和扩充训练数据。

显然活性连接技术与深度学习常规训练方案并无本质区别，那么如何证明活性连接技术的差异性？实际上，这两个工作流在目的上存在一个显著区

别：活性连接技术是生成器引导用户产生训练数据，由 y 引导 x ，常规训练则是根据先验的常识产生训练数据，由 x 引导 y 。

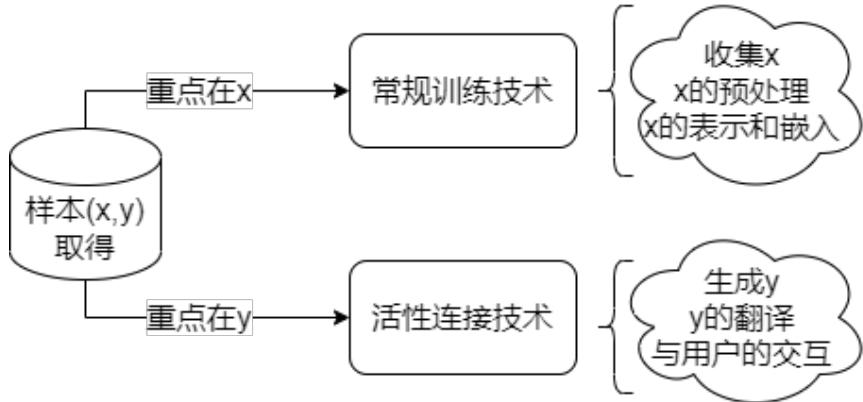


图 15 活性连接与常规训练的差异

这种差异奠定了活性连接技术的基调，相比于大量挖掘“优质”数据 x 的常规训练，活性连接技术的核心是另一个：大量开发“优质”的数据 y 。常规训练花费大量精力在数据 x 的收集、预处理和表示上，活性连接技术则花费大量精力在数据 y 的生成、翻译和用户交互上。

2.3 生成器和翻译器设计

本节包括生成器的理论设计、程序设计和实验展示，以及翻译器的理论设计和界面实现程序。

2.3.1 生成器的理论设计

本文的定位系统有 6 个自由度，也即输出特征为 6 个，那么如果一个标签仅选取一个特征生成，那么有 6 种选取可能，选取两个特征生成则有 C_6^2 个，选取 N 个特征就有 C_6^N ($N \leq 6$) 个。

同时，假设一个动作由一组连续无转折变化的标签组成，那么对标签的连续无转折变化模式也有多种方案可以采取，如一直采用同一个常数的常值模式 $f = C$ ，线性变化模式 $f = kx$ ，变速变化模式 $f = e^x$ 或 $f = \ln x$ 等。

选哪些特征、特征在数轴上怎么采样，这两者都是标签生成的待定问题，该采样策略必须保证生成器满足数据众包的标准，生成器提供游戏任务的机制应满足自动化、离散化、高质量的要求，成为一个本地化的数据众包平台为神经网络训练提供有效的数据。下文通过更精细的定义来设计这个采样过程的细节。

在本定位系统中， y 可以表示为 $\vec{y} = (x, y, z, \varphi, \theta, \psi)$ ，每个维度标准化到 $[-1, 1]$ 的范围，例如 $\vec{y} = (0, 0, 1, 0.51, -0.51, 0)$ 就可以表示为一个动作的一个暂态，从暂态到暂态可以确定一个游戏过程，这个游戏过程会连续地产生 x 数据。假设通过均值的办法采样 M 个数据点，则对 M 个 x 数据应产生 M 个 y 数据与之对应，这里同样选择将暂态 \vec{y}_1 到暂态 \vec{y}_2 的切分为 $M - 1$ 个区间段，分别对应到这些 x 数据。

例如，在 $2s$ 内产生了 2000 个 x 数据点（ $1ms$ 一个），并提供一个起始 y 数据和终点 y 数据，分别为 $\vec{y}_1 = (0, 0, 0, 0, 0, 0)$ 和 $\vec{y}_2 = (0, 0, 1, 0, 0, 0)$ ，即向前移动了 1 个单位。我们取每 200 个数据点（即 $M = 10$ ）做均值 $x' = \frac{x_i + x_{i+200}}{200}$ ，得到 10 个 x 数据点；另一方面，将向量空间 \vec{y}_1 到 \vec{y}_2 切分为 9 段，从 $(0, 0, 0, 0, 0, 0)$ 到 $(0, 0, 0.1, 0, 0, 0)$ 到 $(0, 0, 0.2, 0, 0, 0)$ 最终到 $(0, 0, 1, 0, 0, 0)$ ，因此就得到了 M 个 (x, y) 样本点用于训练数据。

上面这个采样方法是线性模式 $f = kx$ ，如要采取变速模式，以 $f = e^x$ 为例，有：

$$|y_i - y_{i-1}| = k|y_{i+1} - y_i| (k < 1)$$

拉宽两个暂态的距离即可呈现变速模式的 y 数据，而常值模式使 $y_i = y_{i+1}$ 即可。

常规训练的数据追求 x 数据面的多样性，活性连接技术则关注 y 数据面的多样性。考虑到每一个生成器给出的动作转化到玩家产生的动作都会带有一定的噪声，我们所生成特征的动作，等效于在六维特征空间 $(x, y, z, \varphi, \theta, \psi) = \Sigma Y$ 进行一系列的随机采样，一个有效的 ΣY 采样策略应当使得：

$$\forall \vec{y}_i, \vec{y}_j (i \neq j) \ argmax\{d(\vec{x}_i, \vec{x}_j)\}, F(\vec{x}_{i,j}) = \vec{y}_{i,j}$$

对于任意两个不同的输出特征， ΣY 采样策略最大化两个输出特征所对应的输入特征在输入空间中的距离，该策略可以使得每个采样出的标签都尽量“有用”：对不同的 \vec{x}_i 有区分力。

进一步设定一个假设，在已出现的 x 数据尚未遍历特征空间 ΣX 前，足够大的 y 距离使得 x 距离足够大，也就是在 ΣY 中足够分散的 y_i 可以最大化 x 采样点 $\{x_i\}$ 的相互距离，那么只需 $argmax\{d(\vec{y}_i, \vec{y}_j)\}$ 即可 $argmax\{d(\vec{x}_i, \vec{x}_j)\}$ ：

$$L < \delta, d(\vec{y}_i, \vec{y}_j) \propto d(\vec{x}_i, \vec{x}_j), \vec{y}_i, \vec{y}_j \in \{y\}_L$$

具体来说就是，连续进行 ΣY 的 L 次随机采样得到 $\{y_1, \dots, y_L\}$ ，使得当 ΣY 的随机采样值 \tilde{y} 与上述策略给出的 y ，其与已知采样值的距离 $d(\{y_1, \dots, y_L\}, \tilde{y})$ 比上述策略给出的采样值的距离 $d(\{y_1, \dots, y_L\}, y)$ 大于 σ ，当这个距离小于 σ 时，可以认为采样策略已经基本遍历了 ΣY ，若要进一步采样，可以随机生成采样点了， δ 为此时已经采样的数据点个数。

例如，先从 ΣY 采样一个点，得到 y_1 ，继而搜索 y_2 使得 $argmax\{d(y_1, y_2)\}$ ，再搜索 y_3 使得 $argmax\{d(y_1, y_3) + d(y_2, y_3)\}$ ，搜索 y_i 使得 $argmax\{\sum_{j=1}^{i-1} d(y_j, y_i)\}$ ，同时每一次采样生成一个 ΣY 的随机点 \tilde{y} ，若 $|\sum_{j=1}^L d(y_j, y_i) - \sum_{j=1}^L d(y_j, \tilde{y})| < \sigma, \sigma = 0.01$ 则认为该种采样策略效果与随机采样无可观测差异，则采取随机采样的 \tilde{y} 加入到 $\{y_1, \dots, y_L\}$ 。

实际上， $\text{argmax}\{d(\vec{y}_i, \vec{y}_j)\}$ 即推出 $\text{argmax}\{d(\vec{x}_i, \vec{x}_j)\}$ 这个假设过强，如果算力充裕，也可以采用在 ΣX 计算采样点的距离而不在 ΣY 计算相应距离（ ΣX 一般比 ΣY 大得多），采取类似 ΣY 采样策略的方法计算满足要求的 x ，并输入到神经网络中求出 y ，以输出的 y 作为新的 ΣY 采样点。在采样未充分地密集时，这两种采样策略， ΣX 和 ΣY ，可能最终效果上不会相差很大，但由于未作数学论证，这两种策略的采样效果仍是一个开放性问题。

为了用户可实操，还应实现对 y 采样点一定的正则化以降低随机采样带来的复杂度，随机采样的复杂性包括数值不规整的复杂以及数值组合长度的复杂，因此正则化操作分两类：

- a) 数值正则化，将随机生成的复杂浮点数改为简单的小数；
- b) 组合正则化，将复杂的多维向量拆解为依次组合的多个简单向量。

对于(a)，规定一个定点数 D ，例如，对 0.98234345 的 8 位浮点小数，规定定点数 $D = 2$ ，那么就会将其改为 0.98000000 的 8 位浮点小数，也即 0.98 的 2 位定点小数；

对于(b)，规定一个组合复杂度 C ，表示将向量 y 分拆为多少个子向量，借鉴机器学习中的独热编码 (one-hot representation)，若 \vec{y} 含有 n ($n \leq 6$) 个非零维度，那么， \vec{y} 可以分解为至多 n 个非零向量， $C \in (1, n]$ ，例如，假设随机生成的 \vec{y} 在数值正则化以后有 4 个维度是非零的，可以表示为 (0,1,2,0.5,0,1)，那么可以构造 $C = 1, C = 2, C = 3, C = 4$ 的四种分拆。例如对 $C = 2$ 的分拆，相当于从 1,2,0.5,1 四个数中选取一组在 2 个维度上非零的向量使这些向量的总和为原向量，例如选取 (0,1,2,0,0,0) 和 (0,0,0,0.5,0,1)，或者 (0,1,0,0,0,1) 和

$(0,0,2,0.5,0,0)$ 。

本文不设置具体选取哪一个作为分拆向量 \vec{y} 的策略，仅仅记下非零维度的索引，每次需要分拆时作混洗，将混洗结果按 C 正序分组为子向量，随机选择。

由此可将数值和组合的正则化策略表述为 $Re(D, C)$ ，两个参数即可将目标特征 \vec{y} 简化，变为一个更规整更易操作的 \vec{y} ，一个目标标签的生成过程可总体描述为一个四元组：

$$Gen(Y) = \left(\sum Y, M, \sigma, Re(D, C) \right)$$

上述四元组中， $\sum Y$ 为目标标签的特征空间， M 为每两个标签点间的采样次数， σ 表示 $\sum Y$ 的距离最大化采样策略的判定点， $Re(D, C)$ 为对采样出的 \vec{y} 再予以正则化的策略。

于是总的目标标签生成流程是，先进行 $\sum Y$ 的随机采样，计算距离是否小于 σ ，若不小于则继续流程否则终止，接着对所得数据进行正则化，然后以初始零向量为起点，正则化所得的向量序列为后继，两两标签向量间产生一个采样过程，采样 M 个数据点。

2.3.2 生成器的程序设计

根据目标标签的生成过程的四元组定义，生成器程序提供生成策略的接口、采样次数和正则化策略，其数据处理流程如下图所示：

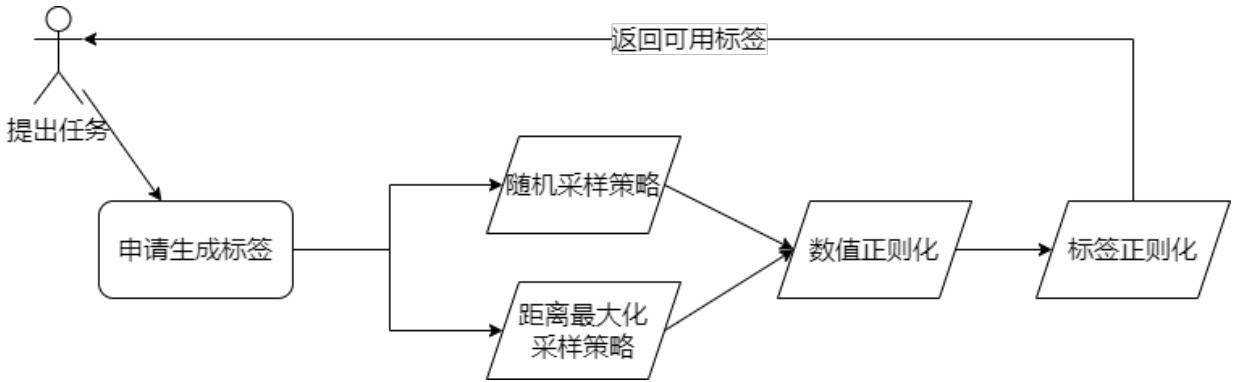


图 16 生成器事件处理流程

数值正则化与标签正则化的实现较为简单，这里关注距离最大化采样策略如何实现，根据连续函数的最值定理可知，凸性连续函数必然在定义域的边界取到最大值，故只需关心如何采样边界。本文假定随机标签值从 $[-1, +1]$ 单位区间采样，则对一个维度只有两个边界点，对 N 个维度有 2^N 个边界点，因此可以随机采样一组 N 维的二进制向量，将其映射为端点值的标签，仅在这样的二进制向量中进行比较即可求出距离最大化的标签值采样点，此即距离最大化采样的算法原理。

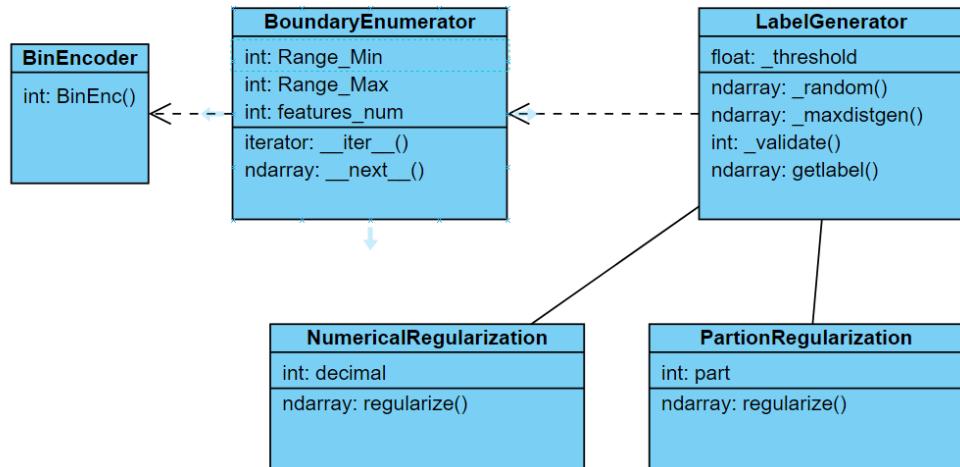


图 17 生成器程序类图

如图 2 所示，以 LabelGenerator 类为核心，维护两种采样策略 `_random()` 和 `_maxdistgen()` 以及采样点质量的评价方法 `_validate()`，向外提供 `getlabel()` 接口以供用户访问；其中 `_threshold` 是 `_validate` 用以评价最大化距离采样策

略得到的标签是否比随机采样更好的阈值；BoundaryEnumerator 是一个迭代器，根据特征空间的边界值和特征数迭代遍历空间边界点组成的 ndarray 类型标签（numpy 数组），用户同时自行决定调用 NumericalRegularization 和 PartitionRegularization 决定是否对从 LabelGenerator 得到的标签进行进一步正则化以减少复杂度，因为最大化距离采样策略获取的是标签点、不需要数值正则化策略，并且标签的复杂度也是对具体用户而言个体理解不同的，所以用户可以自由设置。

2.3.3 生成器效果实验和评价

使用生成器采用最大化距离差值采样策略生成 100 个标签，该标签组与随机采样的差值、每个标签（最大距离）的数值、“与随机采样差”对同期最大距离的比值实验如下：

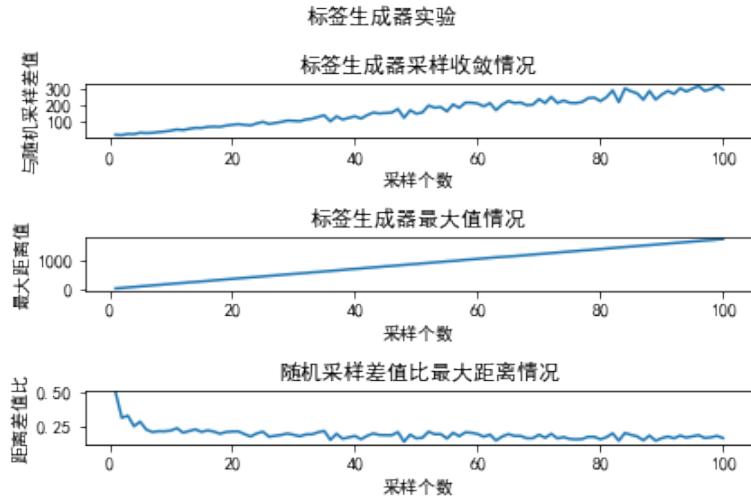


图 18 标签生成器的最大化距离采样的标签质量实验

如图 1 所示，横轴是最大化距离采样每次迭代新扩充的标签，纵轴分别是①：与随机采样的差值、②：该采样的距离总和值；③：①/②。可以观察到与随机采样的数值差距随采样个数线性波动增长，而采样个数与最大值的关系是稳定的线性增长，随机采样差值的波动越发巨大，这是数据点密集化

的结果。有趣的是， $\frac{MaxDiff - RandomDiff}{MaxDiff} \rightarrow 0.2$ ，也即 $RandomDiff \rightarrow 0.8MaxDiff$ ，这可能也是大量边界数据点造成的高距离差结果。

这说明，特征空间 ΣY 各标签距离最大化的策略是存在问题的，很多边界值被重复采样了许多次，这样目标标签只会生成极端标签：出于最大化彼此差距的需要。那么假设特征空间是可动态变动的，对于我们的实际需要来说，这个动态变动是指迅速缩小（收敛），该策略的表现如何呢？

作为一个探索性数据分析（EDA, Exploring Data Analysis）的尝试，每次完成若干个标签的迭代生成，就对特征空间缩小一个固定比例的边界，例如下面的实验就直接尝试一个激进的策略：每生成 5 个标签就缩小一倍的特征空间，前 5 个标签为一开始设定的 BoundaryEnumerator 的边界最小值和最大值，5~10 个标签缩小到一半，15~20 个标签缩小到 1/4。

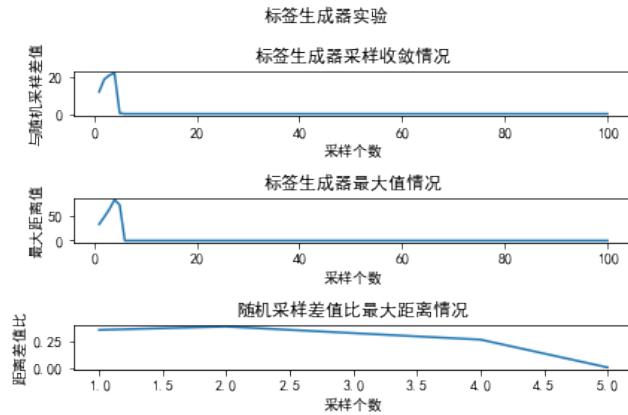


图 19 边界空间迭代缩小下的最大化距离采样实验图形

如图 2 所示，该种激进采样使最大化距离采样策略所取得的标签值距离总和在达到一个高峰（采样 2~3 个）后就迅速收敛到零，可知边界采样的最大化距离策略对特征空间边界的变动是敏感的，在特征空间不变时线性增长，在特征空间的边界指数组小时同样迅速缩小。鉴于标签生成器的表现，本文推荐的标签生成策略是一开始的 2~3 组标签采用最大化距离采样策略，选择

缩小边界并选择 2~3 组边界点标签，随后再采用在原空间（未经缩小的特征空间）进行随机采样。

当然，这只是不讨论组合正则化的结果，组合正则化允许设置某些特征维度为零（不在这些维度上做出模仿行为，类似 Dropout 技术）得到一些不那么极端的标签，并且可以有效降低复杂度，在实际情况中应该更多地进行使用。

2.3.4 翻译器理论设计

翻译器的需求是将输出特征翻译为游戏任务，将训练样本翻译为游戏过程，且这种翻译遵循游戏化的设计原则。

翻译问题的关键在于，将标签数据表达为用户可理解的多种媒介形式，因为游戏需要发生在一定的媒介上。翻译的结果必须支持用户进行灵活的操作，以支持理想用户假设：用户具备一定通用性的复杂认知能力。

这一过程就是将一个标签打包为一个游戏任务，例如 $(0,1,0,0,0,0)$ 既可表示“向上移动一个单位”——文字性描述，也可以表示为三维空间中一个手柄物体的向上平移——图像性描述。翻译器产生用户可理解的内容，以便用户基于对任务的理解产生有效的模仿行为，就可将游戏过程反映为训练过程。

在这里，由于定位系统的 6DoF 姿势估计任务较为简单，并且 $\sum Y$ 可直接理解为现实行为，故可直接构建两个并行的游戏。

第一个是文字性游戏，分别为（左右移动 X 个单位，上下移动 X 个单位，前后移动 X 个单位，上下翻转 Y 个单位，左右翻转 Y 个单位，前后翻转 Y 个单位），其中 X 个单位根据 \vec{y} 值的平移维度进行判定，在同一组标签中的最小平移值作为衡量单位； Y 个单位根据 \vec{y} 值的旋转维度进行判定，在

同一组标签中的最小旋转值作为衡量单位。

对于每个 \vec{y} 值（正则化后可能是数个 \vec{y} 的分量），文字性游戏将其翻译为一组任务，若 \vec{y} 的格式不变，每个任务具有上面的相同表述格式；文字性游戏同时会实时地根据玩家模仿行为所产生的 x 数据经过神经网络后得到的 y 数据翻译为类似文字任务的文字表现，这样就实现了标签和任务的隔离，也是神经网络界面和用户界面的隔离。

另一个游戏是图像性游戏，类似于文字性游戏其会将这些操作转化为一个手柄在三维空间中的 6DoF 运动，每个标签会使手柄图形从原点出发产生运动。因此每一组标签就可以表达为一段带多个关键点的动画。动画可以反复播放，因此向玩家提供直观、易理解的界面，同样实现了神经网络和用户界面的隔离。

2.3.5 翻译器和调试器界面实现

翻译器和调试器界面采用 PyQt+VTK+QtGraph 设计，主界面分别为三大选项卡：实时翻译器、实时生成器、活性连接训练。

实时翻译器包含左右区域：任务目标展示区域和实时行为展示区域，两个区域各包含一组 VTK 可交互可视化图形窗口和 QTextBrowser 文字输出窗口，分别实现图像性游戏和文字性游戏，其运行界面如下图 1 所示：



图 20 实时翻译器界面运行示例图

如下图 2 所示，实时翻译器的实时行为展示区域接收串口数据并结合文本表达和图形表达各自的特色实现对应窗口内用户可理解可交互的输出；任务目标展示区域同步任务流得到的目标标签，图形窗口播放动画，文本窗口输出目标标签的说明。

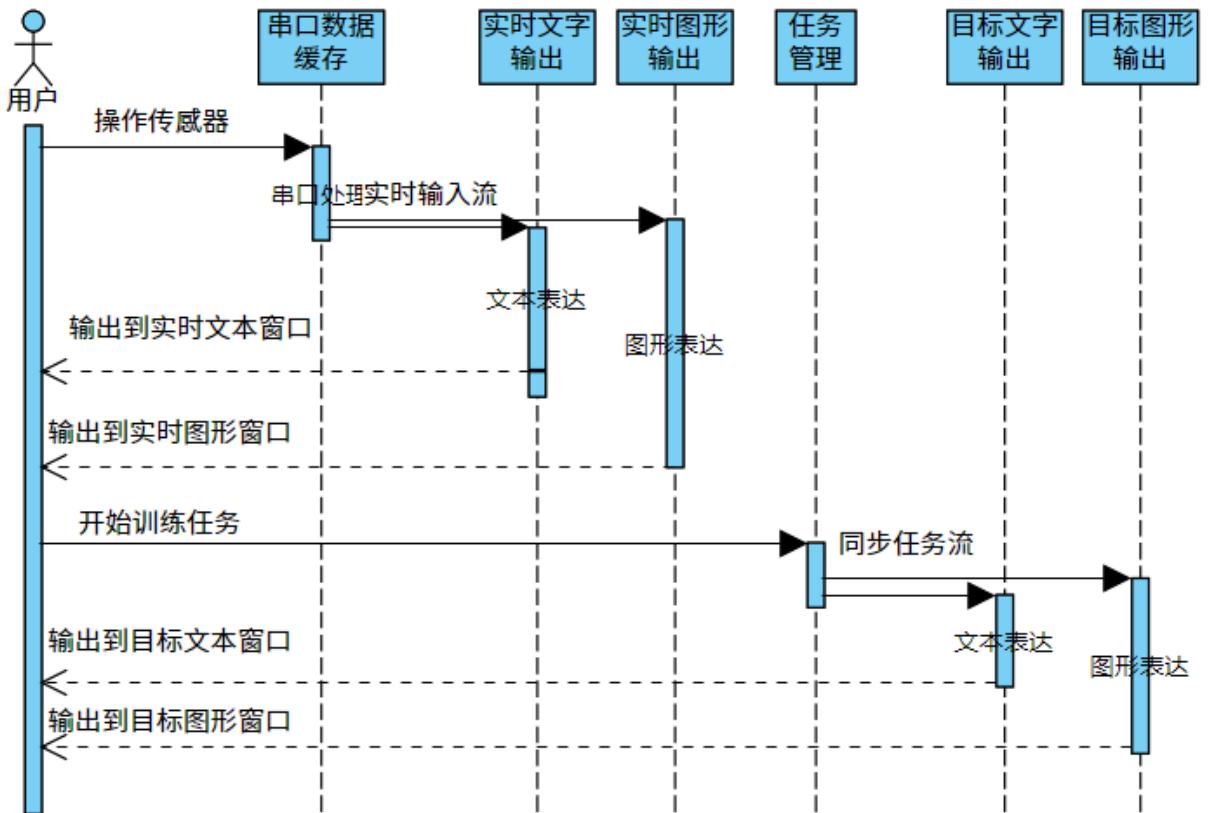


图 21 实时翻译器各组件与用户消息流的序列图

实时生成器包括中上位置的任务流区域和中下位置的生成器区域，任务流区域允许用户选择任务、开始/重启/跳过任务以及结束任务，并通过波形图实时输出从串口端得到的传感器定位数据，以及所选定的任务数据提供的目标标签波形图，用户在结束任务的时候可以截取该段任务时期得到的(X,Y)数据对，并在“截取数据对”的窗口展示；生成器区域允许用户选择标签正则化策略和标签生成策略得到建议的标签，并在右侧窗口展示，由用户自行选择是否根据该标签创建任务。实时生成器的运行界面如下图 3 所示：

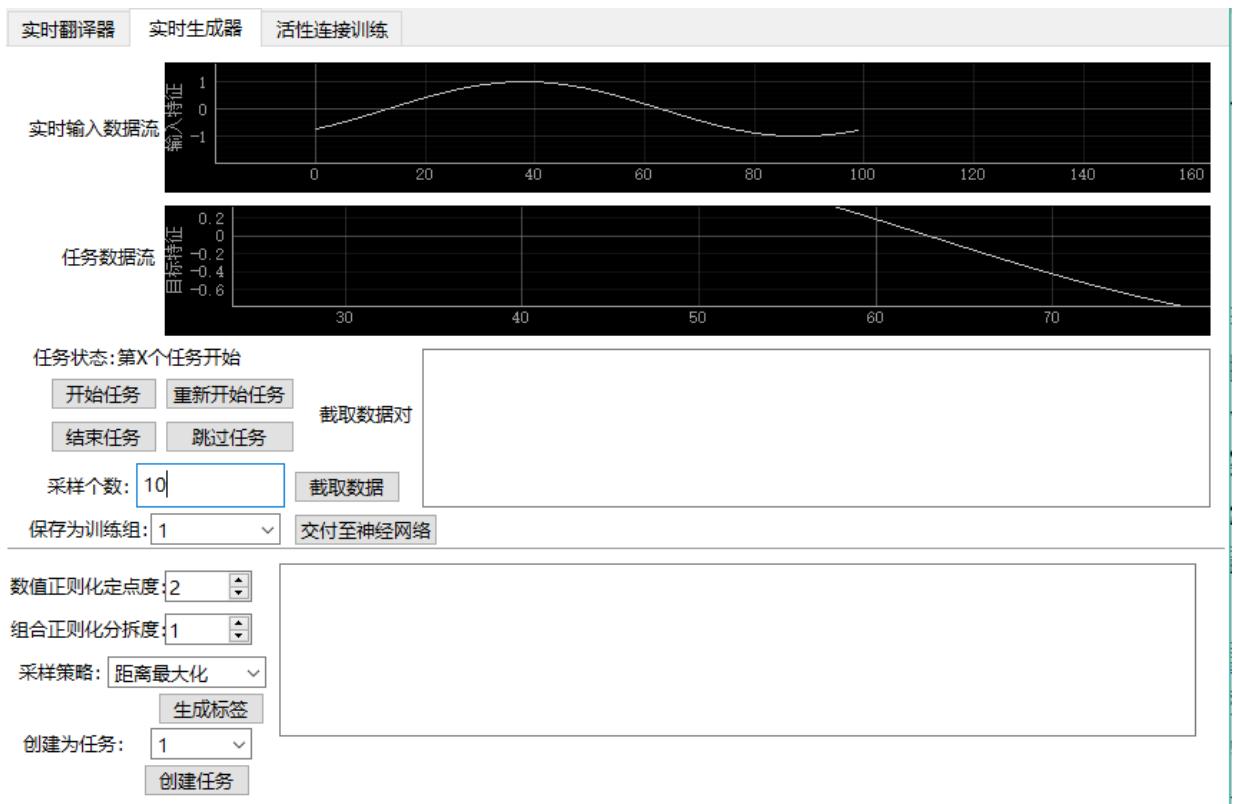


图 22 实时生成器界面运行示例图

由图3可直观感受到生成器是一个提供很多操作的输入性界面而非翻译器这样的输出性界面，用户自主操作的功能板块分三大类：生成标签、控制任务和数据交付，每一板块包括细分的子类用例行为；另有系统定时器实时更新输入数据的波形图和任务数据的波形图，如下图4所示。

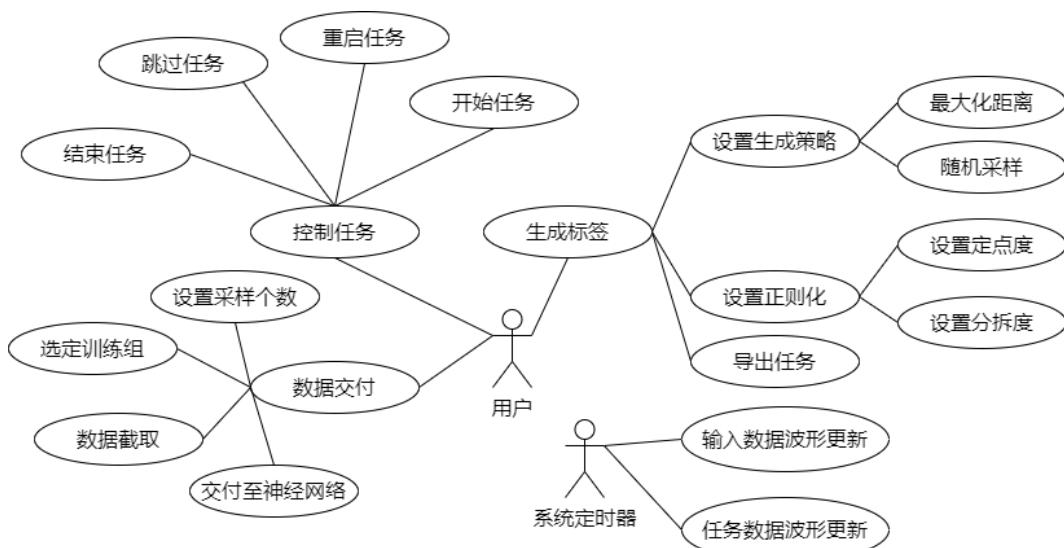


图 23 实时生成器用户行为的用例图展示

活性连接训练界面目前最为简单，允许用户自定义隐层个数，但输入特征和输出特征个数都是 6 个不可修改，隐层也只能是全连接层，暂不支持改变损失函数仅能是 **MSELoss** 均方误差损失，优化器仅为 **Adam**，学习率仅设置为 0.005，据此可以启动/重启神经网络模型；用户可指定批训练大小（神经网络每次从“截取数据对”里取多少组用来训练），开启神经网络实时训练流；实时训练流过程中用户可在右上角观察目前已进行的训练次数和训练数据对个数，并在区域中下位置察看神经网络的 Loss 波形图，以及预测精度波形图，其中预测精度是从历史训练数据中取 10 个估计并计算偏差得到的。活性连接训练界面的运行实例如下图 5 所示：

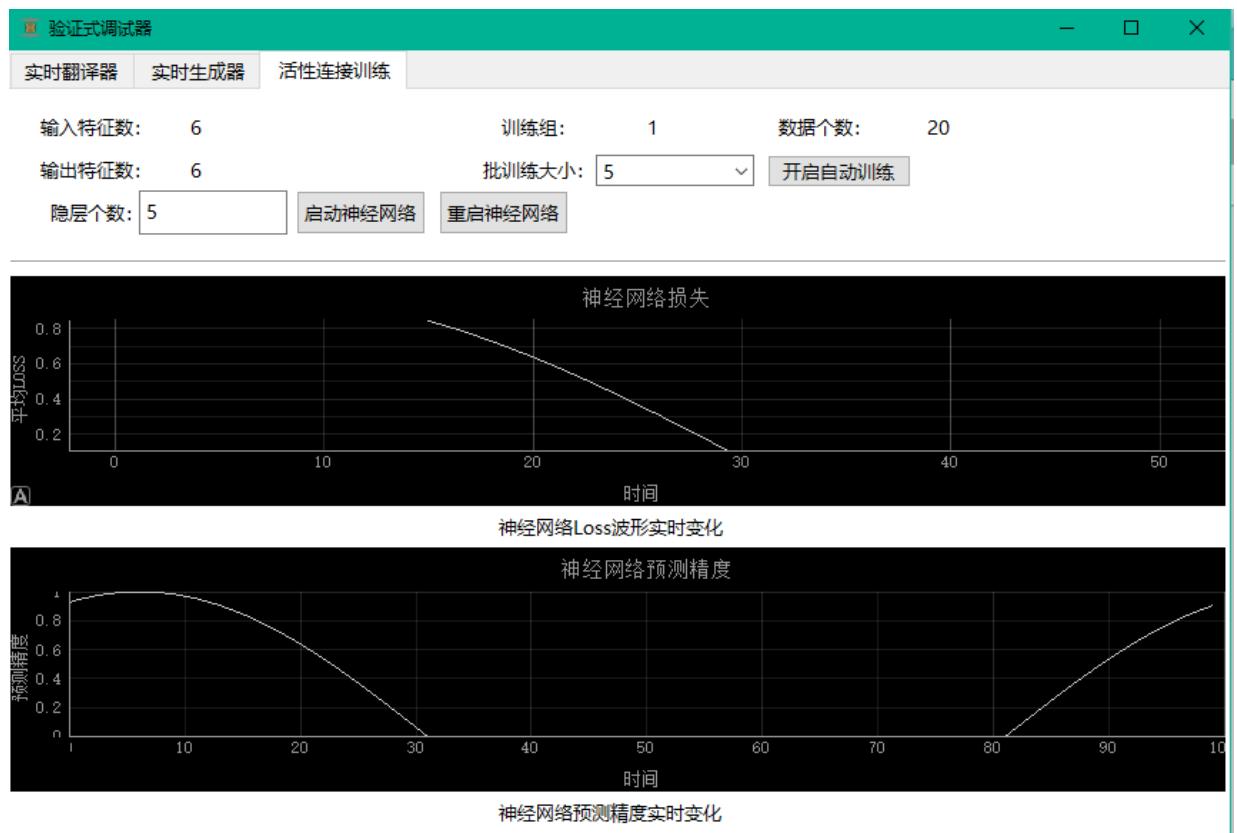


图 24 活性连接训练界面运行示例图

两个波形图模块中，神经网络的预测精度波形图是具备实时性的，因为

精度预测模块所需的测试数据可以从历史训练数据中采样取得，但 Loss 波形图实际上并非实时的，由于本设计并不允许神经网络重复使用训练数据进行训练，故只有用户截取了数据对交付到神经网络以后，神经网络的自动训练模块才可从中取出批训练数据用于训练，以实现 Loss 波形的更新。

2.3.6 面向信息融合的活性连接技术

信息融合的目的在于借助多个传感器获得更鲁棒或快速的定位数值，并且对单个传感器失效或异常具有稳定性。本文将提出的验证式调试器将特征级融合和决策级融合在一个神经网络内完成，可以将许多传感器融合问题转化到神经网络内部，转化为训练数据中的特殊模式，选择合适的神经网络结构使该神经网络可以学习到这类模式出现时的正常输出。

例如，若需要定位系统对突发的传感器的失效具有鲁棒性，可以模拟失效场景（一个或多个传感器的输入值波动巨大或无变化），对每一类失效场景进行训练，使神经网络记住该类“失效输入”的特征，再次应对类似场景时即可更具适应性。只需不断验证各类失效场景，就可维持定位系统的对传感器表现异常的鲁棒性。生成器负责生成各类失效场景（提供变化模式奇异的目标标签），使用户应对多样的传感器失效情况，并进而为神经网络提供丰富的数据。

类似表示学习（Representation Learning）对输入 x 的改造，活性连接技术也支持对输出 y 的改造，对目标标签的特征空间 ΣY ，当 ΣY 非常大或复杂时，如果用户的认知能力非常强大，可以分别理解 ΣY 各维度的标签数值变化并做出相应举动，认知资源足够强大能支撑 ΣY 的复杂性，就可以扩充输出特征空间 ΣY 以实现更复杂的功能。

如果用户的认知能力不足以覆盖 ΣY 的各维度标签变化，就需要多个用户合作以覆盖 ΣY 的各维度标签变化，也就是构造多个游戏过程相衔接。例如，将本文的 6DoF 定位系统扩充为 600 维特征空间， $\Sigma Y = 600$ ，相当于原先的每一个维度扩充为 100 个维度，用这一百个维度表示在某个自由度上运动的更丰富的信息。

为应对这个高复杂输出空间，这里本文提出一种简单的用户资源分配策略：调动 $S = \log_n \Sigma Y$ 个用户，其中每个用户处理上游用户提供的 n 个特征， S 个用户的顺序合作即可完成 ΣY 大空间的覆盖，将空间复杂度对数化降低，不过这需要非常复杂的游戏化设计工作。

当然，活性连接技术能做到多少性能是待实践检验的，这里主要是为了展示活性连接技术面向实际问题中更复杂需求的可扩展性，活性连接技术在理论上具备作为一种一般的信息融合方法的潜力。

2.4 定位系统总体设计

综上可知本设计的定位系统是典型的软硬件协同定位系统，软硬件协同的基本要求是软件方面具备完善的逻辑结构以确保硬件落地的顺利进行，以下从系统整体层次描述本文的定位系统模型，并基于该模型给出一种 Petri 网的软件形式化规格说明，同时进行一些对系统整体的 Petri 网模型基本的仿真测试，以明确该定位系统的软件总体特性。

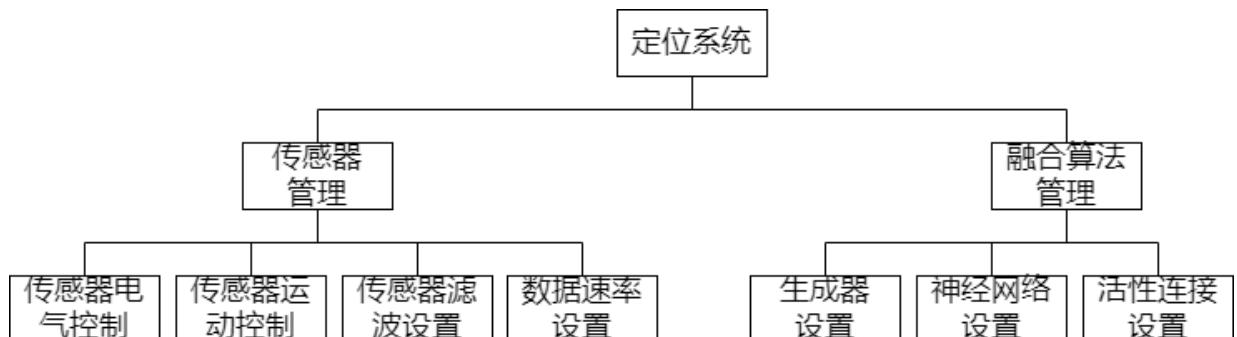


图 25 定位系统总体设计的层次图

如图 1 所示，定位系统主要可分为两大功能模块：传感器管理和融合算法管理。传感器管理包括传感器描述模型的一级二级参数控制（电气控制、运动控制）和软件层的滤波算法以及时序方面的数据速率设置；融合算法管理包括生成器设置（训练数据的标签如何生成）、神经网络结构及超参数设置、活性连接设置（神经网络训练数据的数据源特征）。

2.4.1 定位系统 Petri 网描述与仿真

定位系统是一种对实时性和稳定性要求特别高的并发系统，因此可以采用非常适宜描述并发活动的 Petri 网^[43]对本文的定位系统进行建模和说明，实际上，部署定位系统所用的 FPGA 硬件的 Verilog 程序设计也是高度并发的，参考陈珑等人的工作^[44]将 FPGA 组合逻辑（高并发性）程序映射为 Petri 网模型的编译算法。因此，Petri 网无论从软件需要而是实际硬件环境来说都是本文形式化描述整体定位系统的逻辑模型的不二之选。

2.4.1.1 定位系统的 Petri 网描述

用 Petri 网描述定位系统并非事无巨细地表示整个定位系统，而是发挥 Petri 网强大的时序和并发描述能力，提取定位系统中重要的时间变化节点，并罗列这些时间变化结点间的状态转移关系。

对本文的定位系统，最长的状态变化链条是：对传感器采样→对传感器数据滤波→交付定位数据（→用户截取为训练数据→训练神经网络）→经由神经网络融合信息→输出综合定位信息，这也是定位系统的主要功能所必需的数据的位置及迁移链条，可表示为如下图 1Petri 网图形：

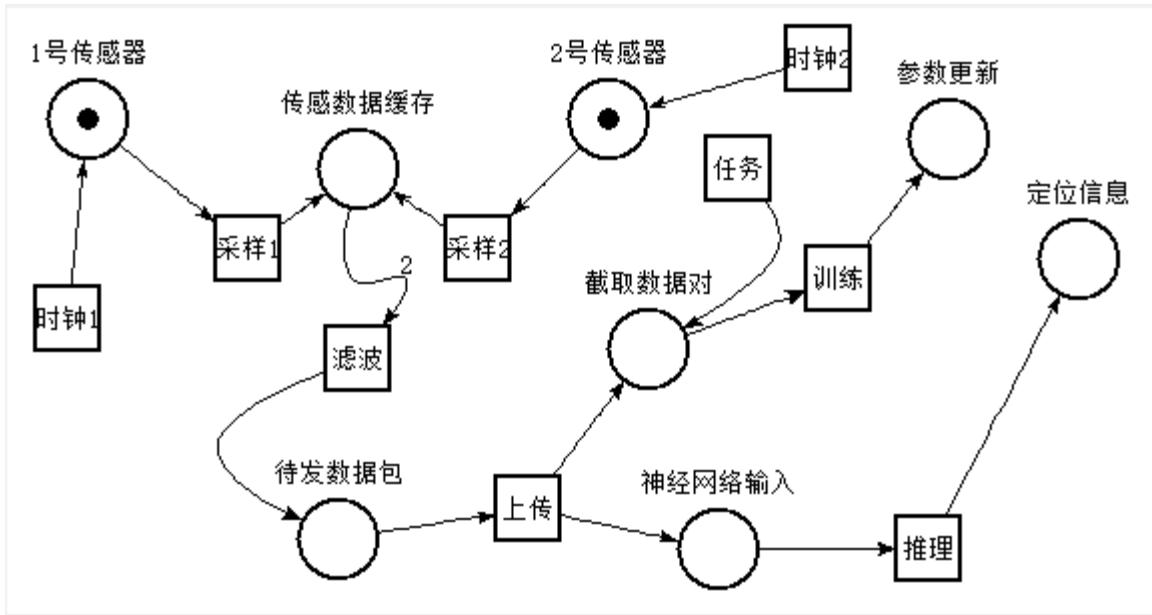


图 26 定位系统主干功能的数据“位置-迁移”Petri 图形

如该图所示，“时钟 1”和“时钟 2”是源迁移，无条件送出令牌到 1 号 /2 号传感器，表示对这些传感器采样的定时触发机制，“采样”表示将临时采样数据保存到用于缓存传感器数据的寄存器的过程，实际上是通过 IIC 通信协议取出传感器的数据；随后滤波迁移要求缓存位置已具备 2 个令牌，并发送 1 个令牌到待发数据包位置；待发数据包就是等待通过 UART 向 PC 通信发送定位数据字节时刻以前，这些定位数据的状态。上传迁移引向两大分支：训练分支和推理主干，推理主干就是定位系统的主要功能，实现融合传感器信息为定位数据，训练分支是定位系统质量上可用的必要功能，由用户发起任务截取数据对和上传的数据包共同引发训练任务迁移，发送一个令牌到“参数更新”位置表示发生了一次神经网络的参数更新。

2.4.1.2 定位系统的 Petri 网仿真

本文采用 LAAS/CNRS 团队的开源软件 Tina 绘制 Petri 网和进行 Petri 网仿真。仿真程序采用步进模拟器进行，初始仿真条件为各位置均无令牌，网络全局步数延迟基本在 0.0~1000 内波动但基本控制在 100 以内，处于程

序兼容原因将所有位置和迁移的名字改为了英文，接下来分别展示约 10,000、30,000、100,000、175,000 时间点的模拟情况：

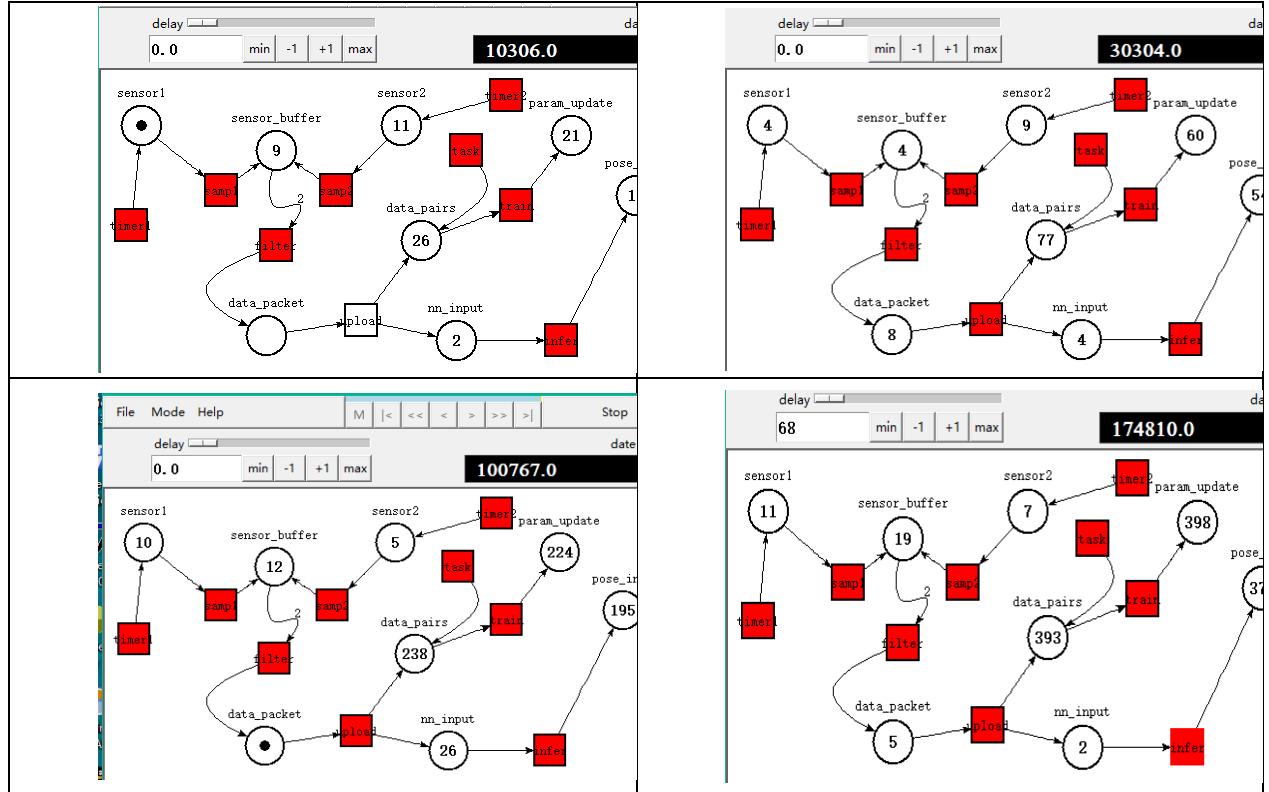


表 9 定位系统在约 10,000、30,000、100,000、175,000 时间点的步进模拟器仿真情况

由四个时间点的仿真情况可推断，两个传感器以及传感器数据的缓存这个硬件数据传输过程，数据包到训练和推理分支的软件数据传输过程，这两个数据传送过程较不稳定，容易产生较大的数据漂移窗口使各模块间的数据收发不同步。

因此对这些数据传输易堵塞位置应设置充裕的 FIFO 模块，例如对传感器采样共用一个同步的定时器，数据缓存共用一个 FIFO，数据上传采用同步式传输协议，以控制定位系统各模块的时序尽量衔接而尽量少地错接。例如本定位系统虽然采用 UART 异步传输协议，但实际上数据包级的传输是同步的，PC 向 FPGA 传感器管理系统发送的每一个字节都被编码为请求相应

数据的协议，因此在数据包级将异步协议实现为了同步协议。

第三章、定位系统部署

设立本章的必要性在于定位系统的部署问题与定位系统的逻辑结构同样重要，能很好地适配特定硬件环境的定位系统部署方案很大程度上可以保证定位系统的可用性，如何在 FPGA 环境上设置神经网络模块，并保证这种部署尽量节约资源、降低延迟和高速运行，就是定位系统部署要解决的核心问题。

3.1 FPGA 神经网络加速平台

神经网络应用场景的拓宽和融入学科的增加使其结构和计算图的复杂度迅速提升，传统的 CPU 串行计算架构难以满足神经网络扩张的计算量，主流的神经网络加速主要在 ASIC,FPGA, GPU 三类芯片架构方面进行扩展。其中有关 FPGA 的深度神经网络加速主要分为：在复杂指令集计算机上管理数据流，输入数据流到 FPGA 上进行计算；在 FPGA 上设置 SoC 的精简指令集系统同时负责数据流和计算流^[45]。

衡量 FPGA 加速器的效果可采用以下指标：加速比（程序的串行版本运行时间对并行版本运行时间的比值）、效率（加速比对处理单元个数的比值）、资源利用率（处理单元空闲时间对忙碌时间的比值），据此 FPGA 应在 4 个方面提升加速器性能：加速算法的计算核、归纳出共性特征、算法并行化改造、数据传输高速化^[46]：

1. 机器学习算法反复执行、耗时最高的部分即为计算核(Kernel)，对计算核的加速可以显著提升加速器性能；
2. 机器学习算法具有共性：可以分解为大量的线性代数运算，可以

把张量、矩阵之积表示为乘累加(MAC)，以及多数算法所复用的激励函数；

3. 机器学习算法的执行顺序、同步和异步步骤允许为其设计管道流水线等架构提高数据吞吐率；
4. 机器学习算法频繁存取数据带来的高 I/O 开销，可以用多级缓存的方式降低通信耗时。

例如，一种复杂指令集计算机+FPGA 加速器范式的架构为^[47]：

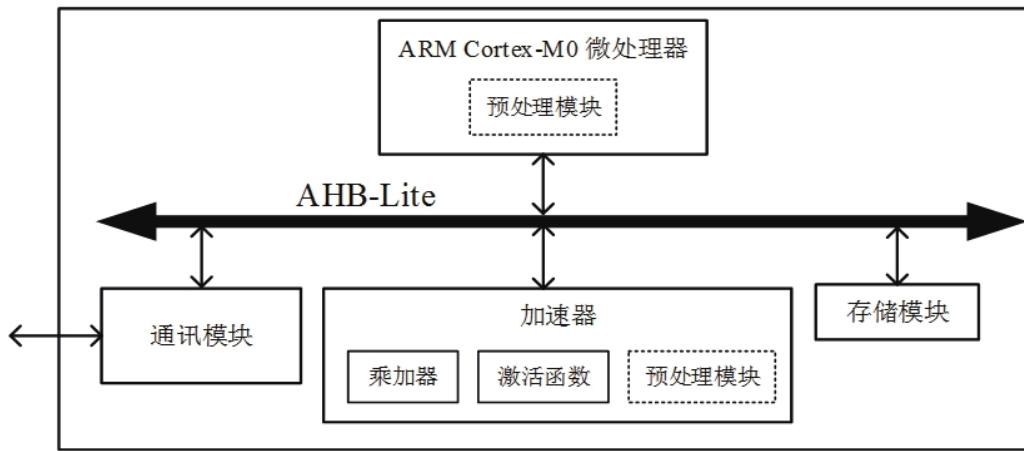


图 27 复杂指令集计算机+FPGA 加速器范式的一种架构^[47]

由微处理器负责控制流和时序同步，在 AHB-Lite 总线上传输数据流和控制流，加速器只负责运算，数据传输和缓存交由通讯模块和存储模块处理。加速器可细分为乘加器，激活函数模块和数据预处理模块，FPGA 加速器分别例化这些模块并由 ARM Cortex-M0 产生的控制时序进行并行计算处理。其中，乘加器负责矩阵乘代数运算，激活函数模块用一阶多项式 $bx + c$ 和二阶多项式 $ax^2 + bx + c$ 拟合 Sigmoid 函数 $f(x) = \frac{1}{1+e^{-x}}$ ，采用乘加器实现多项式计算，数据预处理模块负责将浮点数表示为 32 位定点数，可节省大量浮点乘法器资源，适用 FPGA 硬件结构。

传统的 FPGA 神经网络加速器围绕量化神经网络（也即定点数神经网络）

设计各种子计算模块并连接，这方面的工作可参考黄航对多层感知机、多个激活函数、反向传播单元以及卷积模块的具体实现^[48]，和参考石帅连接计算模块间的脉动阵列流水线的具体实现^[49]。

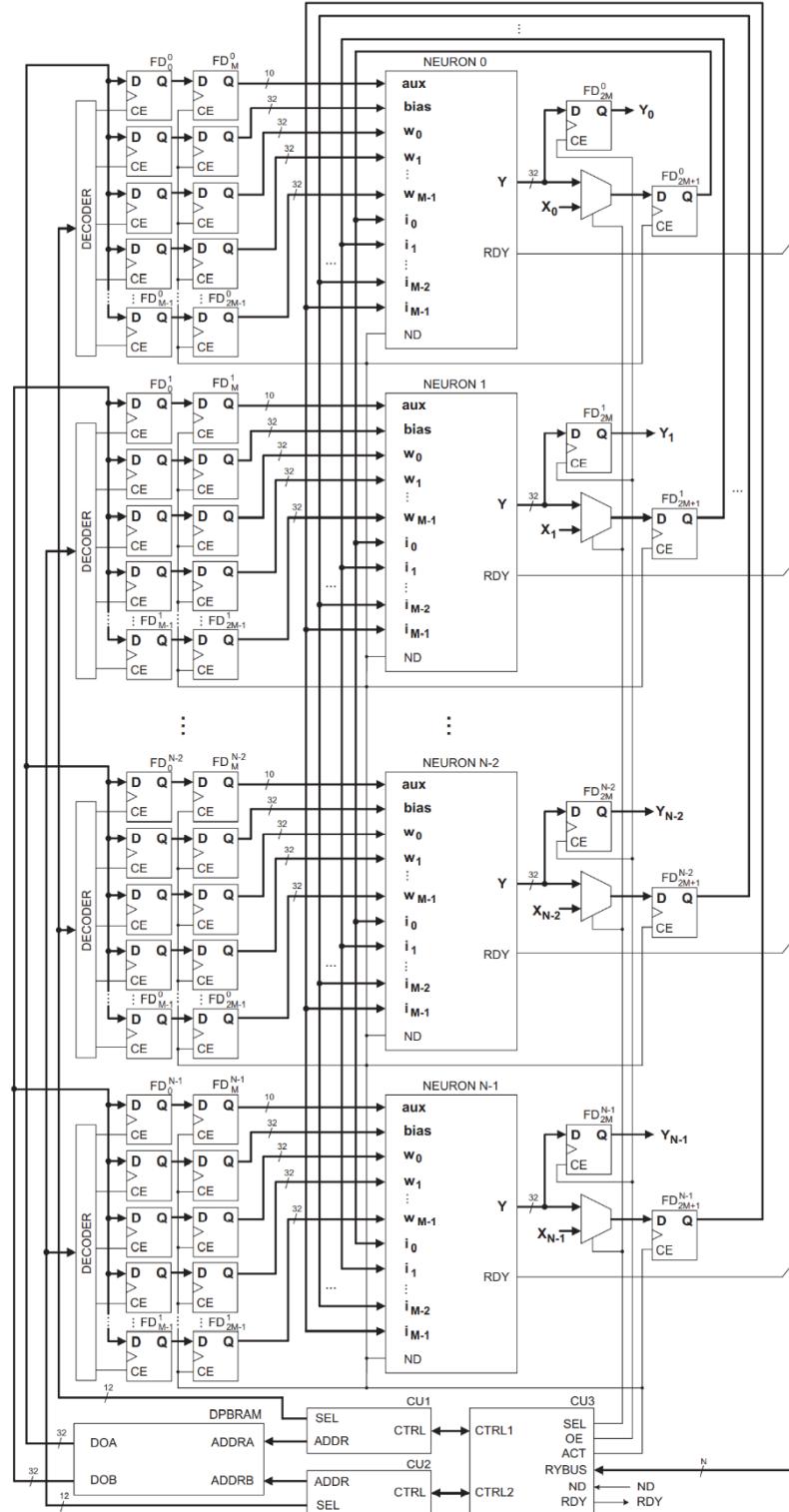


图 28 硬件级的神经网络 FPGA 加速器的原理图^[50]

一个硬件级的神经网络 FPGA 加速器的原理图实现可参考 Zbigniew Hajduk 的工作^[50]，由上图可见，若干个 D 触发器阵列寄存器构成一个解码器的下游缓存，由控制单元根据地址从块内存（BRAM）读出数据以供解码器转化为定点数处理，从缓存中接线至神经元模块，每个神经元仅计算输出 \hat{y} 的每一位 y_i ， $\hat{y} = \sum_{i=1}^N 2^{i-1} * y_i$ ，将 \hat{y} 的各位输出通过移位寄存器链组织起来就得到 y_i 的整体定点数输出。

因此，FPGA 完全可以成为神经网络的部署平台，神经网络的层级结构、代数算子都可以在 FPGA 平台上找到对应的硬件结构，因此 FPGA 可以提供对抽象神经网络模型的硬件加速器。

3.2 二值神经网络

本文欲采用 BNN（二进制神经网络）而非 QNN（量化神经网络/定点数神经网络）实现 FPGA 上的神经网络，这方面的奠基研究可参考 Bengio 和 Rastegari 的工作^{[51][52]}，通过将输入 x 及权重 w 表示为二值的 $\{0,1\}_N$ 或 $\{-1,1\}_N$ 向量，将矩阵乘 $w * x$ 、激活函数 $f(\cdot)$ 、卷积算子 $Conv(\cdot)$ 、池化算子 $Pooling(\cdot)$ 、归一化算子 $BatchNorm(\cdot)$ 表示为向量操作，辅以将浮点数或定点数编码为二值向量及将二值向量解码为定点数的编解码器，就可得到仅采用逻辑门完成主要运算（主要是 $Xnor$ 和 $Bitcount$ 操作）的神经网络。

纯逻辑门架构非常适合 FPGA 的实现，FPGA 在硬件架构上主要是由 LUT 组合的 SLICE 阵列，逻辑资源按 SLICE 分配，而 LUT 则是 SLICE 完成逻辑运算的主要模块，LUT 也即 Lookup-Table 查找表，根据固定的地址返回一个结果，FPGA 厂商大多采用 6LUT（6 扇入 1 扇出）的 LUT 结构，因此可实现 6 入 1 出的真值表，在逻辑门的意义上就是可以实现 5 个逻辑门

[53]。而 BNN 主要是同或门 *Xnor* 和 *Bitcount* 操作，前者可以直接由 LUT 实现，后者可以采用 LUT 的思路用更大的查找表结构简单实现。

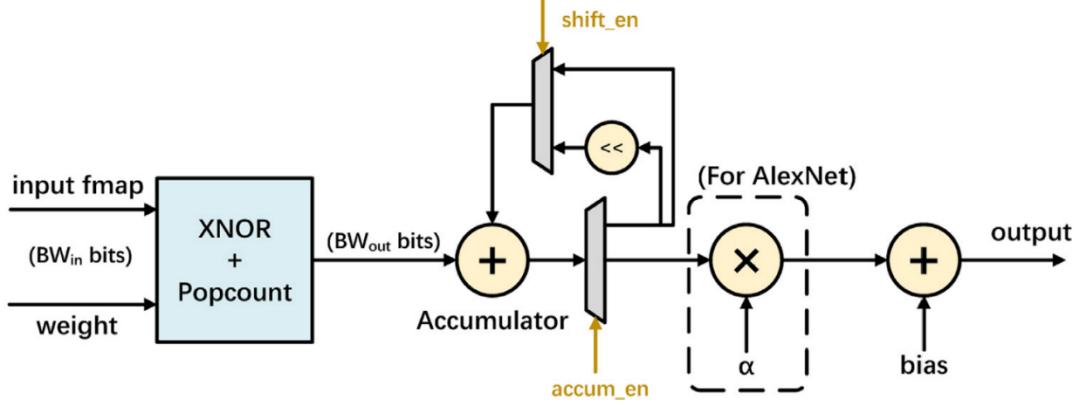


图 29 单层二值化神经网络数据变换流^[54]

在更高层的 FPGA 硬件结构上，输入和权重的二值向量经过 *Xnor* 和 *Bitcount* 操作可得到第 N 位上的输出，通过累加器和移位寄存器，将各位输出乘修正系数并累加可得到最终的定点数 \hat{y} ，再加以偏置可得 $y' = \hat{y} - bias$ 即为最终输出^[54]。

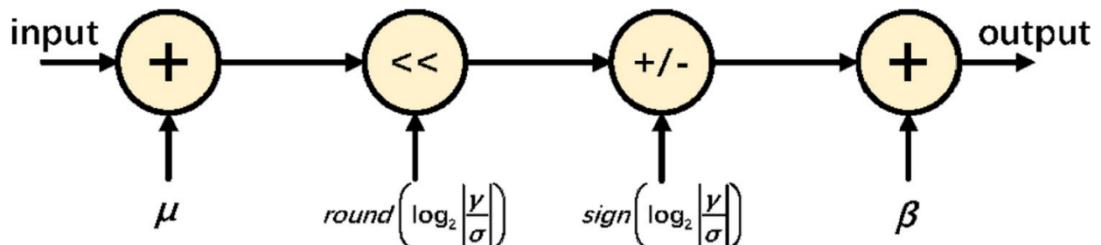


图 30 二值化神经网络的 BatchNorm 层实现^[54]

Shuang Liang et. al. 的工作中还包括 BatchNorm 层的硬件级数据流设计^[54]，BatchNorm 也即归一化层， $y = \frac{x-\mu}{\sigma} \cdot \gamma + \beta$ 为归一化层的代数表示，假设均值 μ 和标准差 σ 已被计算，那么 BatchNorm 层可表示为：加 μ + 左移 $round(\log_2 |\frac{y}{\sigma}|)$ + 乘正负符号 $sign(\log_2 |\frac{y}{\sigma}|)$ + 加偏置 β 的流水式操作。这样做的原因是未归一化的输出 y 在流经二值化激活函数 $f^b(\cdot)$ 后可能会损失很多信

息，但采用 $f^b(BatchNorm(\cdot))$ ，对已经归一到 [-1,1] 区间的输出 y 采用激活函数就可以减少激活函数带来的信息流失，这种策略在该文献中也被证明是非常有效的。

这里推荐 Umuroglu et. al. 的工作 “FINN” [55]，FINN 是另一个完善且开源的端到端 DNN-FPGA 加速框架，从神经网络 Theano 代码到 FPGA 硬件级实现的 FPGA-BNN 框架，支持卷积神经网络为主的深度学习算子，快速部署到 FPGA 目标平台上。

由于本文不涉及计算机视觉相关的卷积运算，除却 BatchNorm 层和 Linear 层，剩余神经层仅有 BNN 的输入层和输出层：如何将浮点实数表示为二进制向量和如何将二进制向量解码为浮点实数的具体实现。

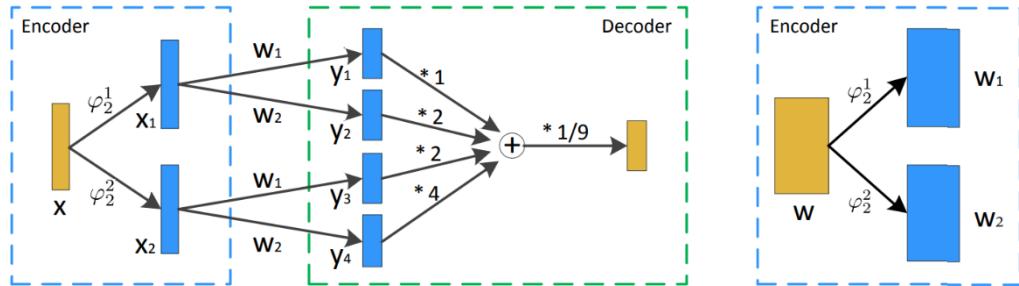


图 31 定点数神经网络量化分解为多分支二进制神经网络的计算结构^[56]

这里参考 Qigong Sun et. al. 的编解码器设计^[56]，如上图所示，定点数 x 被编码器 φ_N^i 分解到 N 个 x_i ，例如对于 32 位定点数，编码器为 φ_{32}^i ，将定点数 x 分解为 32 个 x_i ， $x = \sum_{i=1}^{32} i * x_i$ ，权重 w 采取类似的做法。再将每个 x_i 采用 32BitEncoder 处理，Qigong Sun^[56] 提出用正弦函数结构来实现这个编码器：

$$MBitEncoder = \begin{cases} \varphi_M^m(x): \text{sign} \left(\sin \left(\frac{2^M - 1}{2^m} \pi * x \right) \right), m = M \\ \varphi_M^m \left(-\sin \left(\frac{2^M - 1}{2^m} \pi * x \right) \right), m \in \{1, 2, \dots, M-1\} \end{cases}$$

该函数可对 $[-1, 1]$ 内的实数值编码到二进制值 $\{-1, +1\}$ ，超出该范围的实数值可采用整流函数 $\text{Sigmoid}(\cdot)$ 或 $\text{ReLU}(\cdot)$ 实现。

该编解码器的解码器部分较为简单，使与 x_i 对应的 y_i 乘以对应的权重系数 i 再求和，累加和乘以一个修正系数 α 可返回定点数输出 y 。

通过将普通的浮点数神经网络改造为二进制神经网络，我们能轻易用 FPGA 逻辑资源完成神经网络的前向推理甚至是反向传播过程，这就为 FPGA 上部署足够大足够复杂的神经网络奠定了基础；并且浮点数并行处理并非 FPGA 的强项，FPGA 的强项在于大量的查找表逻辑资源（一个 6 扇入查找表可实现 5 个逻辑门的任意组合）和高速串行数据传输，仅用逻辑门完成运算就可以充分发挥 FPGA 的高实时性优势，符合高性能定位系统的低时延高响应需求。

3.3 二进制神经网络的数值计算结构

本文的定位系统只需实现全连接层神经网络即可，由于 PC 端 Pytorch 训练采用 32 位精度进行，故本文要做的是将 32 位定点数转化为二进制的量化神经网络得到适配 FPGA 芯片的方案，本文的神经网络训练阶段在 PC 端进行，并且 FPGA 只需从 PC 接收训练完毕的参数权重数据，本文通过重新烧录 FPGA 系统整体配置文件来实现参数权重数据的传送。

因此假设已得到一个训练好的神经网络，剩余的工作仅为前向推理。输入 6 个特征，每个特征为 32bit Double，神经网络为七层神经网络（批训练

数据大小为 5 个时): $f: x \rightarrow h, g: h \rightarrow y$, 其中 h 为隐层序列, 由于隐层目前均为 6 个特征的神经网络, 故只需考虑输入层和输出层即可清楚隐层的 Verilog 表示, 进而得到神经网络整体的表达方案。从数据流的观点看, 首先将权重和特征拆分为二进制数, 每一层求 $y = \text{activate}(w \cdot x)$, 再将 y 二值化为 y^b 作为新一轮运算的 x 。

采用上文提到的 Qigong Sun et. al.^[56] 用 -1 和 +1 量化神经网络的方法, $x^i = \sum_{m=1}^M 2^{m-1} \cdot c_m^i, c_m^i \in \{-1, +1\}$, 该种表示方法可以表达 $\{-(2^M - 1) \sim (2^M - 1)\}$ 的奇数值, 将 x^i 表示为放缩系数向量 $(2^{m-1}, \dots, 1)$ 和二值向量 $(c_m^i, c_{m-1}^i, \dots, c_1^i)$:

$$x^i = (2^{m-1}, \dots, 1) \times (c_m^i, c_{m-1}^i, \dots, c_1^i)$$

同样分解权重 $w^i = \sum_{k=1}^K 2^{k-1} \cdot c_k^i, c_k^i \in \{-1, +1\}$, 将 w^i 表示为放缩向量和二值向量之点积:

$$w^i = (2^{k-1}, \dots, 1) \times (d_k^i, d_{k-1}^i, \dots, d_1^i)$$

由向量点积运算的交换性, 可以得到:

$$\begin{aligned} x^i \cdot w^i &= \left((2^{m-1}, \dots, 1) \times (c_m^i, c_{m-1}^i, \dots, c_1^i) \right) \\ &\quad \times \left((2^{k-1}, \dots, 1) \times (d_k^i, d_{k-1}^i, \dots, d_1^i) \right) \\ \therefore x^i \cdot w^i &= (2^{m-1}, \dots, 1) \times (2^{k-1}, \dots, 1) \\ &\quad \times \left((c_m^i, c_{m-1}^i, \dots, c_1^i) \times (d_k^i, d_{k-1}^i, \dots, d_1^i) \right) \\ \therefore x^i \cdot w^i &= (2^{m-1}, \dots, 1) \times (2^{k-1}, \dots, 1) \times \text{Bitcount} \left(\text{Xnor}(\vec{c^i} \cdot \vec{d^i}) \right) \end{aligned}$$

其中 Xnor 为同或操作: 输入的两个比特位 (+1/-1) 相同则输出 +1(1) 不同则输出 -1(0), Bitcount 为比特计数操作: 对比特向量 $\{-1/+1\}_i$ 求和; 根据此公式可在逻辑门级 (Gate-Level) 迅速计算二值向量相乘结果, 具体对本文

的定位系统神经网络，对 $6 \times 32\text{bit}$ 的输入 x ，设置 32bit 编码器为：

$$\begin{cases} x^{(1)} = (x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, x_4^{(1)}, x_5^{(1)}, x_6^{(1)}) \\ \dots \dots \dots \\ x^{(32)} = (x_1^{(32)}, x_2^{(32)}, x_3^{(32)}, x_4^{(32)}, x_5^{(32)}, x_6^{(32)}) \end{cases}$$

输入 x 共 32 条总线，每条总线位宽为 6，分别储存 x 每一个数位上各输入特征的值。例如，对特征 x_5 ，有 $x_5 = 4,294,967,296 * x_5^{(31)} + \dots + 4 * x_5^{(3)} + 2 * x_5^{(2)} + 1 * x_5^{(1)}$ ，且 $x_5^{(i)} \in \{-1, 1\}$ 。

以同样的方式将权重 w 分解为 32bit ，在全连接层中，对任意一个输出 y ，计算出该输出的 6 个输入 x 分别要乘 6 个权重，可以得到 w 的 $6 \times 32\text{bit}$ 编码器：

$$\begin{cases} w^{(1)} = (w_1^{(1)}, w_2^{(1)}, w_3^{(1)}, w_4^{(1)}, w_5^{(1)}, w_6^{(1)}) \\ \dots \dots \dots \\ w^{(32)} = (w_1^{(32)}, w_2^{(32)}, w_3^{(32)}, w_4^{(32)}, w_5^{(32)}, w_6^{(32)}) \end{cases}$$

权重在神经网络前向推理过程中是不变的，以 32 宽度 6 深度的存储为一个基本单位，每一个层包含 $6^2=36$ 个这样的权重基本单位，用 ROM 存储一层权重耗费 6912 个 bit，FPGA 强大的 LUT-6 支持作为 64bit 的分布式 ROM 存储使用，因此可用 108 个 LUT-6 分布式 ROM 实现权重的储存。

由输入 x 和权重 w 的 $6 \times 32\text{bit}$ 编码器可以得到输出 y 的相应表达：

$$y_{11} = w^{(1)} * x^{(1)}, \dots, y_{32_32} = w^{(32)} * x^{(32)}, y_{ij} = w^{(i)} * x^{(j)} = \text{popcount}(\text{Xnor}(w^{(i)}, x^{(j)}))$$

最终输出 $y = \alpha \cdot \sum_{i=1}^{32} \sum_{j=1}^{32} (2^{i-1} * 2^{j-1}) * y_{ij}$ ，其中 α 为调整输出的缩放因子，最常用的表示为 $\alpha = \frac{1}{(2^M-1)(2^K-1)}$ ，这里 $M = K = 32$ ，则 $\alpha = \frac{1}{8589934592^2}$ ，这样得到的输出 $y \in [-1, +1]$ ，处于实数范围内。输出 y 的激活函数采用 $\text{Htanh}(x)$ 作为二分函数：

$$Htanh(x): \begin{cases} -1, & x \leq 0 \\ +1, & x > 0 \end{cases}$$

该神经网络的输入值 x 被限定到 $[-1, +1]$ 的实数值内, 对于 8bit 编码器, 给出下列编码程序:

$$\begin{cases} \varphi_{32}^{32}(x): sign\left(\sin\left(\frac{8589934591\pi}{8589934592} \cdot x\right)\right), & \text{for } x^{(32)} \\ \dots\dots\dots, & \text{for } x^{(31)} \sim x^{(3)} \\ \varphi_{32}^2(x): sign\left(-\sin\left(\frac{8589934591\pi}{4} \cdot x\right)\right), & \text{for } x^{(2)} \\ \varphi_{32}^1(x): sign\left(-\sin\left(\frac{8589934591\pi}{2} \cdot x\right)\right), & \text{for } x^{(1)} \end{cases}$$

该函数组对给定的实数 x 返回 32bit 编码下各个数位的 +1 或 -1 编码 (sign 符号位), 因此对输入的 $[-1, +1]$ 内的实数值 x 就可将其编码为实数, 若 x 不在该范围内, 可用整流函数如 Sigmoid 将 x 投影到 $[-1, +1]$ 区间, Sigmoid 的指数函数 e^{-x} 在 FPGA 中可用二阶多项式 [3] 拟合实现, 这样就回到了用 FPGA 乘法器和加法器资源构造整流函数的步骤:

$$Sigmoid(x) = \frac{1}{e^{-x} + 1}: R \rightarrow [-1, +1]$$

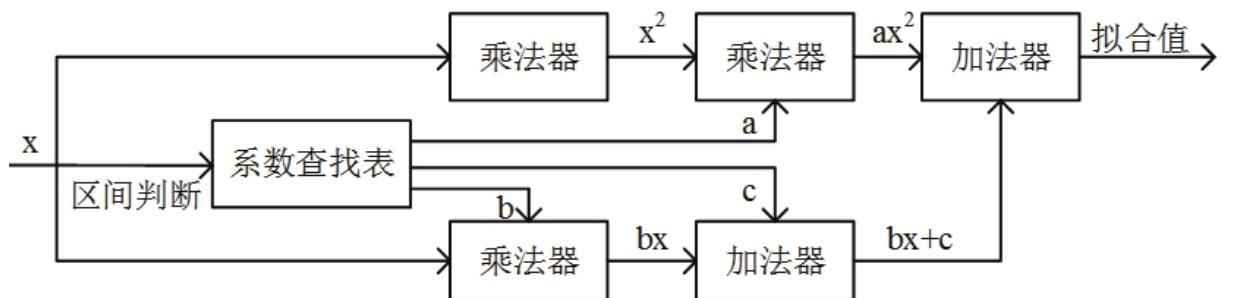


图 32 Sigmoid 整流函数的二阶多项式拟合, 用乘累加器实现 (图源^[47])

神经网络二值化的另一部分是权重 w 的二值化, 需要将权重收缩到 $[-1, +1]$ 内才可进行二值编码, 在训练后为权重进行归一化操作可得到这样

的小数权重。每一层全连接层神经网络计算完毕后，若该层不是最后一层，则将输出 y 重新应用二值化编码得到下一层所用的输入 x ，若该层为最后一层则不做处理转交 UART 发送器模块等待上位 PC 请求数据。

3.4 二进制神经网络的 Verilog 表示、仿真和评价

二进制神经网络推理时所用的最多模块就是 $w*x$ 的二进制向量点积运算模块、累加 $32^2=1024$ 个点积结果乘以一个系数的乘累加器 (MAC) 运算模块，以及正弦函数编码器模块，本文主要讨论该两大模块的具体实现，该部分因被频繁使用故采用并行结构实现：

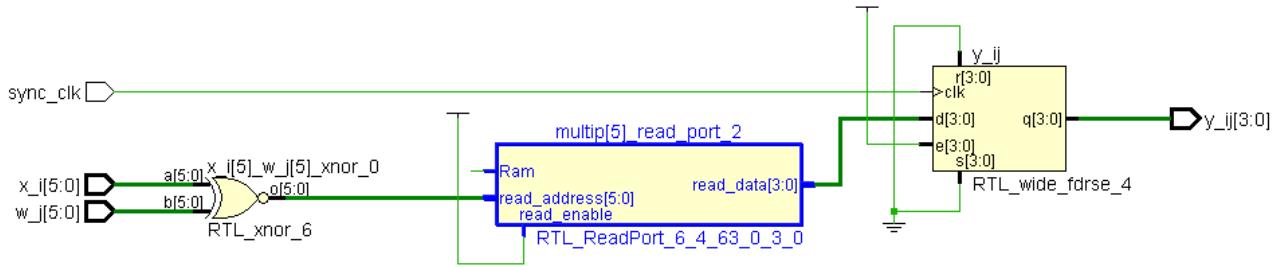


图 33 Verilog 点积运算模块的原理图

如图 1 所示， $x_i[5:0]$ 与 $w_j[5:0]$ 表示任意数位上的六维向量，经过同或门得到 $xnor[5:0]$ ，Bitcount 操作采用查找表的方式实现：设置一个 $2^6=64$ 深度的 ROM 保存任意 $xnor[5:0]$ 的 Bitcount 结果，6 维向量的 Bitcount 操作结果数用 4bit 宽度即可表示，在实操中，最后一位始终为 0，故只需使用宽 3 深 64 的查找表实现 Bitcount 操作，由同步时钟 $sync_clk$ 同步该模块的时序，每一个同步时钟上升沿由上层模块输入各数位上的六维向量，并输出相应的点积运算结果。

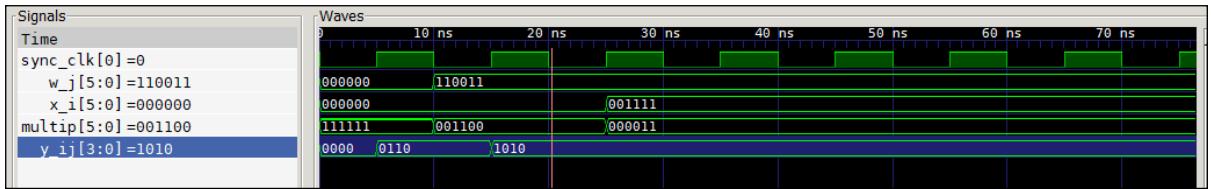


图 34 点积运算模块的 Icarus Verilog 仿真实例波形图

如图 2 所示进行该模块的仿真，仿真时间单位为 1ns，误差 1ps，同步时钟信号设置为 5ns， y_{ij} 输出在每个上升沿输出点积运算结果，本设计所用的 FPGA 主频时钟为 100MHz（时钟周期 10ns），本设计采用串行的方式累加点积运算结果，则约 320ns（0.32us）可以计算一对(x, w)的输出 y ，每一个神经网络的层输出 \vec{y} 耗费 $36*0.32=11.52\mu s$ 时间，7 层神经网络若采取流水线式操作前后输入输出衔接、无时延，则主要计算过程总耗费 $80.64\mu s$ 时长，这是将数据从传感器送到串行通信端口所必需的处理时延。对本设计的毫秒级通信速率而言，这是可以接受的。

另一大模块是乘累加器，采用串行总线的方式从 8 位总线分批次读取 32x6bit 的数据 x 和 w ，切分为 6x32bit 的格式并保存到本地寄存器，通过同步时钟 $sync_clk$ 按步放入(x, w)数据对到点积运算模块，返回 y 值，与 y 值相乘的系数均是 2 的幂次，直接使用移位实现 y 值与系数相乘，并用除法器使 y 值乘以调整因子，得到最终的输出 y 值，由于程序原理图过于庞大，故不在此处展示。

逻辑资源类别	点积与乘累加模块的消耗	Spartan-6 XC6SLX9-TQG144 可提供
片寄存器	95	11440
片 LUT	127	5720

“LUT-FF”寄存器对	86	136
IOB I/O 块资源	20	102
BUFG/BUFG CTRL 缓存块资源	2	16

表 10 点积运算模块与乘累加运算模块所耗费逻辑资源

如表 1 所示, 这两个模块耗费了大量的 LUT 查找表和 LUT-FF 寄存器对, 其中 LUT-FF 寄存器对在 FPGA 平台上并非很充裕, 这也是这两个程序未来的改进方向: 优化寄存器资源的利用; 但 LUT 的利用是完全合理、充分利用的, 片寄存器则尚未被充分使用, 由于该程序在神经网络层间是串行结构, 故表 1 的资源消耗就是 7 层神经网络总体的消耗。

第三个大模块是正弦函数实数编码器, 正弦函数的计算非常频繁且复杂, 对 FPGA 的时间资源带来较大负担, 故采用上文提到的点积运算模块的查找表法:

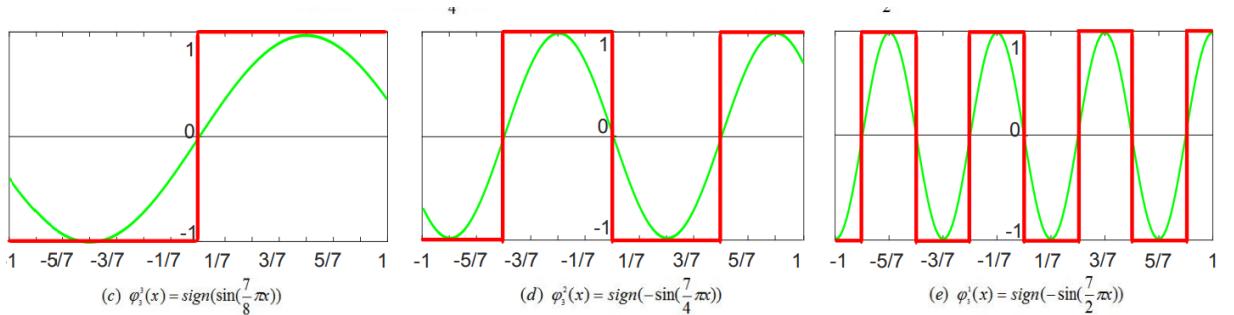


图 35 $\varphi_3^1(x) \sim \varphi_3^3(x)$ 编码器的编码映射^[56]

如图 3 所示, 编码映射存在清晰的边界点, 在离线环境下预先计算 $\varphi_{32}^1(x) \sim \varphi_{32}^{32}(x)$ 的边界点表并储存到 FPGA 片上 ROM, 在神经网络进行编码

操作时直接按地址查找、与边界点比较即可。

结论

本文从设计 VR 手柄和异质传感器融合的定位算法出发展开了全方位的研究，包括适用于定位系统的传感器描述模型、VR 追踪器和 VR 控制器数据通讯协议、以神经网络融合多模态传感器数据以及通过活性连接技术为神经网络提供训练数据、定位系统数据流的规划和仿真、神经网络在 FPGA 目标平台上的部署，使用先模型再工程的思路，基于异质适应、即插即用和高实时性的任务目标，设计出多个抽象模型组成定位系统模型，再指导定位系统的软件工程实践。

虚拟现实是一个高度学科交叉的领域，设计 VR 硬件设备不仅是传统意义上的电气层级、门级、寄存器级的电子产品设计，为满足日益增长的高实时性高稳定性需求，VR 手柄（追踪器+定位器）需要从系统级考虑数据传输的时序约束以及信息融合算法的性能，市场需要的不仅是一个娱乐设备或工具，而是一套完善的定位系统方案，为此就需要从 VR 手柄数据流的各个环节：从逻辑门到 PC 串口收到的数据结构，从传感器的接口到 OpenVR 协议的接口，全程去规划一款高性能的 VR 手柄。

从定位系统的层次来看，本文尚有诸多细节未完善，如在第一章中的传感器描述模型，一级参数（电气参数）间的互作机制未能进行量化，市面上的多数传感器都并非电气参数固定不变的，而是可以动态调整的，尤其是分辨率、量程经常是可以由微控单元动态调整的，定位系统应能设置算法根据环境需要自动调整电气参数，而相对地二级参数（定位参数）的动态调整已在本文进行了一定研究。在未来的相关定位系统的传感器管理算法研究中应关注对电气级参数的控制，例如功率的控制对高功耗定位设备的耗能，帧率的控制对多传感器定位设备的信息融合，影响是不可忽视的。

第一章的 VR 手柄串口通讯协议践行了软件工程的面向对象思想：不是让数据生产者设定发送数据的频率，而是取决于数据消费者的消费需求，由 PC 端应用根据通讯协议向 VR 手柄发送需求命令，本章末提出的简单二期滤波算法是对异质传感器的一种情形：异速情形的算法，在传感器与系统采样速度的异速情况不严重时，该种方法作为一种时间序列平滑法有一定滤波作用。

第二章是本文定位系统的核心所在，基于游戏化理念和数据众包技术设计的验证式调试器界面和活性连接数据生成方法，从数据标签的角度而非数据输入的角度设计将多模态数据对齐的办法，避免了变换输入数据的复杂的联合学习/协同学习编码器，该方法从抽象的神经网络训练过程出发而不仅可应用于定位系统，具备应用到更多领域中多模态融合方法的潜力。

验证式调试器主要包括生成器、翻译器和神经网络结构设计，本文出于训练数据的多样性提出了距离最大化的设计原则，介绍了两种典型的翻译器媒介：图形和文字。对于未来在其他问题中应用可能出现的超大标签特征空

间，提出用多个游戏化过程拼接以对数化分解大空间的方法。

第二章的最后总结了定位系统的数据流，用 Petri 网描述定位系统时序模型并进行仿真分析，发现传感器位置和通讯端口位置容易发生数据包不同步，对传感器的堵塞/不同步可以用 FIFO 缓存和同步的时钟统一操作，对通讯端口的不同步本文的通讯协议在逻辑上可以忽视。

第三章讨论了神经网络的二值化和 FPGA 部署，虽然神经网络是大量操作浮点数的，但仍然允许 FPGA 用大量逻辑门资源部署神经网络并在可接受的处理时延 80.64us 完成前向推理过程。但该部署仍有诸多可改进之处：能否进一步减小处理时延、提高实时性？能否做更多的并行转串行环节以减小高昂的布线（面积）代价？

综上本文最终设计出一款初步可交付的 VR 手柄和定位系统服务，该 VR 手柄相对一般的 VR 手柄具备特殊的立意：异质适应和即插即用特性。这是一种定位系统本位而非定位产品本位的价值观，该设备软件性的成分更强，传感器只是插入系统的异构可换的单元，这种设计最终导向的将是通用的定位服务，即定位功能不依赖于特定的硬件元件实现，这也是使虚拟设备能走向大众、广泛应用的软件工程解法。

参考文献

- [1] Burdea G. Virtual Reality System and Application[C]. Electro 93, 1993.
- [2] 陈劲舟. 虚拟现实及触觉交互技术的趋势与瓶颈 [J]. 轻工科技.Vol.37,2021(09):59-65.
- [3] 张翔. 基于增强现实脑机接口的仿人机器人控制系统[D]. 天津:天津工业大学, 2021.
- [4] ValveSoftware. Openvr[EB/OL]<https://github.com/ValveSoftware/openvr>.
- [5] Caserman, P., Garcia-Agundez, A., Konrad, R. et al. Real-time body tracking in virtual reality using a Vive tracker. *Virtual Reality* 23, 155–168 (2019).
- [6] Misek, Richard. (2021). ‘Real-time’ virtual reality and the limits of immersion. *Screen*. 61. 615-624.
- [7] Lee G, Yun WJ, Ha YJ, Jung S, Kim J, Hong S, Kim J, Lee YK. Measurement Study of Real-Time Virtual Reality Contents Streaming over IEEE 802.11ac Wireless Links. *Electronics*. 2021; 10(16):1967.
- [8] Zhang N, Qi T, Zhao Y. Real-Time Learning and Recognition of Assembly

Activities Based on Virtual Reality Demonstration. Sensors (Basel). 2021;21(18):6201.

[9]王盼. 基于三维建图与虚拟现实的人-机器人交互技术研究[D]国防科技大学,2017.

[10]郑世涛. 基于 FPGA 的虚拟现实定位技术的研究[D]华中科技大学,2012.

[11]史典阳,高俊雄,王耘波,郑世涛. 基于 FPGA 的虚拟现实定位系统设计[J]. 计算机与数字工程,2013,41(1):127-130.

[12]刘智伟,李建胜,王安成,贾骏超,王俊亚.基于运动捕捉系统的 UWB 室内定位精度标定方法[J].测绘科学技术学报,2017,34(02):147-151.

[13] Yang, J, Lei, Y, Tian, Y, Xi, M. Deep learning based six-dimensional pose estimation in virtual reality. Computational Intelligence. 2022; 38: 187– 204.

[14] Wu, Min-Yu & Ting, Pai-Wen & Tang, Ya-Hui & Chou, En-Te & Fu, Li-Chen. (2020). Hand Pose Estimation in Object-Interaction based on Deep Learning for Virtual Reality Applications. Journal of Visual Communication and Image Representation. 70. 102802. 10.1016/j.jvcir.2020.102802.

[15]HTC. Vive Tracker Developer Guidelines[EB/OL]https://dl.vive.com/Tracker/CN/Guideline/HTC_Vive_Tracker_Developer_Guidelines_v1.3-V2.pdf.

[16]Shimizu, H. Measuring keyboard response delays by comparing keyboard and joystick inputs. Behavior Research Methods, Instruments, & Computers 34, 250–256 (2002).

[17] Bubeck, S., & Sellke, M. (2021). A Universal Law of Robustness via Isoperimetry. NeurIPS.

[18]Kronisch. Gamification – Concepts and Theories. Minor Candidacy Exam

Reading Course(2016).

- [19] Santos, A.C.G., Oliveira, W., Hamari, J. et al. The relationship between user types and gamification designs. *User Model User-Adap Inter* 31, 907–940 (2021).
- [20] Sofia Marlena Schöbel, Andreas Janson & Matthias Söllner (2020) Capturing the complexity of gamification elements: a holistic approach for analysing existing and deriving novel gamification designs, *European Journal of Information Systems*, 29:6, 641-668.
- [21] Blohm, Ivo & Leimeister, Jan Marco (2013) Gamification: Design of IT-Based Enhancing Services for Motivational Support and Behavioral Change. *Business & Information Systems Engineering (BISE)*, 5 (4). 275-278.
- [22] Strmečki, Daniel & Bernik, Andrija & Radosevic, Danijel. (2015). Gamification in E-Learning: Introducing Gamified Design Elements into E-Learning Systems. *Journal of Computer Science*. 11. 1108-1117.
- [23] Dichev, Christo & Dicheva, Darina & Angelova, Galia & Agre, Gennady. (2014). From Gamification to Gameful Design and Gameful Experience in Learning. *CYBERNETICS AND INFORMATION TECHNOLOGIES*. 14. 80-100.
- [24] 王雅倩. 软件游戏化的方法与现状分析研究[D]河南大学,2017.
- [25] Knutas, A., van Roy, R., Hynninen, T. et al. A process for designing algorithm-based personalized gamification. *Multimed Tools Appl* 78, 13593–13612 (2019).

- [26] Villegas E, Fonseca D, Peña E, Bonet P, Fernández-Guinea S. Qualitative Assessment of Effective Gamification Design Processes Using Motivators to Identify Game Mechanics. *Sensors* (Basel, Switzerland). 2021 Apr;21(7).
- [27] 刘俊. 游戏化学习活动建模及其应用实践验证[D]华东师范大学,2017.
- [28] Haworth B, Usman M, Schaumann D, Chakraborty N, Berseth G, Faloutsos P, Kapadia M. Gamification of Crowd-Driven Environment Design. *IEEE Comput Graph Appl*. 2021 Jul-Aug;41(4):107-117.
- [29] Laso Bayas, Juan & See, Linda & Fritz, Steffen & Sturn, Tobias & Perger, Christoph & Dürauer, Martina & Karner, Mathias & Moorthy, Inian & Schepaschenko, Dmitry & Domian, Dahlia & McCallum, Ian. (2016). Crowdsourcing In-Situ Data on Land Cover and Land Use Using Gamification and Mobile Technology. *Remote Sensing*. 8. 905.
- [30] Brambilla, Marco & Ceri, Stefano & Halevy, Alon. (2013). Special issue on structured and crowd-sourced data on the Web. *The VLDB Journal*. 22. 10.1007/s00778-013-0327-9.
- [31] McQuilton, Peter & González-Beltrán, Alejandra & Rocca-Serra, Philippe & Thurston, Milo & Lister, Allyson & Maguire, Eamonn & Sansone, Susanna-Assunta. (2016). BioSharing: curated and crowd-sourced metadata standards, databases and data policies in the life sciences. *Database*. 2016. baw075. 10.1093/database/baw075.
- [32] Du, H., Alechina, N., Jackson, M. and Hart, G. (2017), A Method for Matching Crowd-sourced and Authoritative Geospatial Data. *Trans. in GIS*, 21:

406-427.

[33] Ratner J, Sury J, James M, Mather T, Pyle D. Crowd-sourcing structure-from-motion data for terrain modelling in a real-world disaster scenario: A proof of concept. *Progress in Physical Geography: Earth and Environment*. 2019;43(2):236-259.

[34] Basiri, A., Amirian, P., Winstanley, A. et al. Making tourist guidance systems more intelligent, adaptive and personalised using crowd sourced movement data. *J Ambient Intell Human Comput* 9, 413–427 (2018).

[35] 李欣,于亚秀,程静.基于众包的数据优化——以数字方志特藏资源建设为例[J].图书馆论坛,2019,39(02):73-79+118.

[36] 冉卓衡.基于 NB-IoT 网络的实时环境监测系统在众包模式下的应用[J].通信电源技术,2020,37(12):88-90.

[37] 李绍园,姜远.多标记众包学习[J].软件学报,2020,31(05):1497-1510.

[38] 叶晨,王宏志,高宏,李建中.面向众包数据清洗的主动学习技术[J].软件学报,2020,31(04):1162-1172.

[39] 张静. 众包系统偏置标注问题研究[D]合肥工业大学,2015.

[40] 杨璐. 面向 RDF 数据质量评价的众包任务分配机制研究[D]武汉科技大学,2019.

[41] 史珩. 基于自步学习的众包分类数据质量控制模型[D].大连理工大学,2017.

[42] 全子飞. 通用众包标注系统的设计与实现[D].浙江大学,2015.

[43] 张海藩. 软件工程导论 (第五版) [M].北京: 清华大学出版社,2008:82-85.

- [44] 陈珑, 黄颖坤, 罗继亮. FPGA 组合逻辑程序的 Petri 网建模方法[J]. 华侨大学学报(自然科学报), 2015, 36(01): 29-33.
- [45] 张祖扬. 深度神经网络硬件加速研究[D]. 电子科技大学, 2019.
- [46] 王超, 王腾, 马翔, 周学海. 基于 FPGA 的机器学习硬件加速研究进展[J]. 计算机学报, 2020, 43(06): 1161-1182.
- [47] 欧阳魁. 基于 FPGA 的低功耗低资源的 BP 神经网络推理加速器设计[J]. 集成电路应用, 2020, 37(04): 10-12.
- [48] 黄航. 基于 FPGA 的深度神经网络设计与实现[D]. 电子科技大学, 2020.
- [49] 石帅. 神经网络算法的 FPGA 实现及其性能优化[D]. 中北大学, 2020.
- [50] Zbigniew Hajduk. Reconfigurable FPGA implementation of neural networks[J] Neurocomputing. Volume 308(2018) 227-234.
- [51] Courbariaux, Matthieu & Hubara, Itay & Soudry, Daniel & El-Yaniv, Ran & Bengio, Y.. (2016). Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1.
- [52] Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A. (2016). XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science(), vol 9908. Springer, Cham.
- [53] (日) 天野英晴 (主编) .FPGA 原理和结构[M]. 北京: 人民邮电出版社, 2019: 47-48+61-63.
- [54] Shuang Liang, Shouyi Yin, Leibo Liu, Wayne Luk, Shaojun Wei. FP-BNN: Binarized neural network on FPGA. Neurocomputing. Volume

275(2018)1072-1086.

- [55] Yaman Umuroglu, Nicholas J. Fraser, Giulio Gambardella, Michaela Blott, Philip Leong, Magnus Jahre, and Kees Vissers. 2017. FINN: A Framework for Fast, Scalable Binarized Neural Network Inference. In Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '17). Association for Computing Machinery, New York, NY, USA, 65–74.
- [56] Sun, Qigong & Shang, Fanhua & Yang, Kang & Li, Xiufang & Ren, Yan & Jiao, Licheng. (2019). Multi-Precision Quantized Neural Networks via Encoding Decomposition of $\{-1, +1\}$.

附录

- 图 1 虚拟现实三个维度上的对传统 IT 产品重构的解析图
- 图 2 VR 头显设备的三个旋转自由度示意图 (图源网址：<http://www.woshipm.com/pd/3040331.html>)
- 图 3 Vive 追踪器纯 USB 数据传输方式
- 图 4 FPGA 实现的追踪器从传感器阵列采样数据过程示意图
- 图 5 Vive 控制器 (VR 手柄) 的按键布局
- 图 6 按键消抖 Verilog 原理图 (模块名: Stabilize)
- 图 7 用户类型和游戏化设计维度的映射
- 图 8 基于目标的软件游戏化分析过程
- 图 9 数据众包流程的物联网层次划分示意图
- 图 10 众包清洗数据的机器学习算法及众包反馈机制
- 图 11 验证式调试器工作流示意图
- 图 12 用户时间与数据时间对齐问题示意图
- 图 13 活性连接技术的游戏化解读
- 图 14 生成器的翻译功能解读
- 图 15 活性连接与常规训练的差异
- 图 16 生成器事件处理流程
- 图 17 生成器程序类图
- 图 18 标签生成器的最大化距离采样的标签质量实验
- 图 19 边界空间迭代缩小下的最大化距离采样实验图形
- 图 20 实时翻译器界面运行示例图
- 图 21 实时翻译器各组件与用户消息流的序列图
- 图 22 实时生成器界面运行示例图
- 图 23 实时生成器用户行为的用例图展示
- 图 24 活性连接训练界面运行示例图
- 图 25 定位系统总体设计的层次图

- 图 26 定位系统主干功能的数据“位置-迁移”Petri 图形
图 27 复杂指令集计算机+FPGA 加速器范式的一种架构
图 28 硬件级的神经网络 FPGA 加速器的原理图
图 29 单层二值化神经网络数据变换流
图 30 二值化神经网络的 BatchNorm 层实现
图 31 定点数神经网络量化分解为多分支二进制神经网络的计算结构
图 32 Sigmoid 整流函数的二阶多项式拟合, 用乘累加器实现
图 33 Verilog 点积运算模块的原理图
图 34 点积运算模块的 Icarus Verilog 仿真实例波形图
图 35 $\varphi_3^1(x) \sim \varphi_3^3(x)$ 编码器的编码映射

- 表 1 定位传感器的一级参数表
表 2 定位传感器的二级参数表
表 3 ADXL345 和 HMC-5883L 一级参数 (电气参数) 表
表 4 ADXL345 和 HMC-5883L 二级参数 (定位参数) 表
表 5 Tracker 的串口通讯主机端控制协议
表 6 Vive 跟踪器外设的数据格式
表 7 VR 手柄按键布局、样式和实现
表 8 Controller 串口通讯主机端控制协议
表 9 定位系统在约 10,000、30,000、100,000、175,000 时间点的步进模拟器仿真情况
表 10 点积运算模块与乘累加运算模块所耗费逻辑资源

致谢

感谢我这一路上收到的来自各个家人、朋友、同学、网友对我提供的帮助！尤其感谢我的小佩宝对我的任务难处的理解和无微不至的陪伴！感谢妈妈不问缘由的财政支持和鼓励！

感谢室友容忍我熬夜晚睡的打搅！感谢互联网上无私分享知识和代码的同志们！尤其是 GitHub、Academia、Researcher Gate、CSDN 让我不知道省了多少麻烦！感谢世界上最优秀的软件们！尤其是 Microsoft Edge、Anaconda Jupyter Lab、Python、ISE、Notepad++、Visual Studio Code、GitHub Copilot，你们使人延年益寿！还要感谢蔡美玲老师和刘宏老师对这篇毕业设计的细致指导和关心，感谢师大为我提供的数字图书馆支持！

时光荏苒，辅修软件工程学位是约莫在大一下期结束的时候，如今过了两年，世界变化真大，我也变了很多，但唯一不变的就是自己那热爱真理、想钻研一切理解一切的“超人”心。

我要感谢自己和永恒的知识之神，赞美探险家！赞美勇气！赞美所有把生命的多彩和主体的能动性推向历史最远彼岸的勇者们！

外文翻译译文

基于深度学习的虚拟现实六轴姿势估计

JiaChen Yang,Yutian Lei,Ying Tian,Meng Xi

摘要

随着虚拟现实技术的不断发展，虚拟现实正逐步应用于医疗、教育、商业等领域。在该技术的应用中，位置和姿势估计是关键的空间定位技术。传统的姿势估计存在对环境依赖性强、复杂度高等问题，但卷积神经网络(CNN)和其他的智能计算技术为该困难带来了转机。本文基于深度学习的 CNN 理论，依托单目视觉系统和标注好的的目标样本集，提出了一种估计目标位置和姿态的新方法，同时详细描述了在模拟环境得到标注数据集的一般方法。在本文中通过不同网络结构的对比，我们发现该估计方法可以避免人工提取复杂图像特征的大工作量，实现快速、任意且精确的估计。此外，我们还基于四元数证明了世界坐标系与图像坐标系间视觉上的一致性。

关键词

计算智能 卷积神经网络 数据集 姿势估计 虚拟现实

1.引言

随着信息技术的不断发展，虚拟现实技术将在人类生活中发挥越来越重要的作用^{[1][2]}。利用多媒体，三维(3D)建模，智能化交互、计算智能等技术，虚拟现实技术能将虚拟现实信息融入现实世界，这主要取决于虚拟现实技术发展中的许多关键技术，如计算智能、3D实时图形显示、定位和跟踪、触觉和嗅觉传感技术、人工智能技术、并行计算技术和人类行为学研究。虚拟现实技术在各领域中得到了广泛应用，包括军事、医学^[3]、工业^[4]、航空航天、教育、生产、娱乐等行业。随着虚拟现实技术的发展，虚拟现实的真正实现将使人类生活和发展发生巨大改变。

姿势估计是虚拟现实技术应用的必要技术条件^[5]。该技术不仅可用于空间定位，也用于物体姿势估计。在现有的空间定位技术中，图像识别技术是

低成本便携设备的必备工具^[6]，尽管该技术准确度尚低。

给定包含物体的 RGB 彩色图像或深度图像数据集，要求完成目标姿势估计任务（估计该物体在环境中的位置），对该任务 RGB 彩色摄像头具有一系列优势，包括广视角、高分辨率、适合户外工作等^[7]。除此外，姿势估计算法若支持不带深度信息的 RGB 彩色图像，必然可得到广泛应用。因此，RGB 彩色信息与图像姿态估计技术的结合将在未来广泛应用，这种方法将提高虚拟现实领域的用户体验以及人机交互效率^[8]。

与此同时，近年来深度学习的代表性算法之一的卷积神经网络（CNN）^[9]发展非常迅速。其广泛应用于计算机视觉领域的目标检测和分类，各类 CNN 架构相继涌现：LeNet^[10]，AlexNet^[11]，VGG-Net^[12]，Inception^[13]，R-CNN^[14]，ResNet^[15]，DenseNet^[16]……CNN 提供了可以广泛应用于多个领域的深度学习方案^[17]，可解决复杂的分类或回归问题^[18]。该方法避免了推导复杂的函数式，直接用神经网络表达函数关系，这体现了计算智能技术的威力。然而，神经网络的训练必须提供足够的数据作为支撑^[19]。因此，一种通用且具备数据模式的训练集生成方法也是待解决的问题之一^[20]。有效的训练集生成方法不仅可以使整个姿势估计过程更加智能，还可以节省大量的手工劳动。

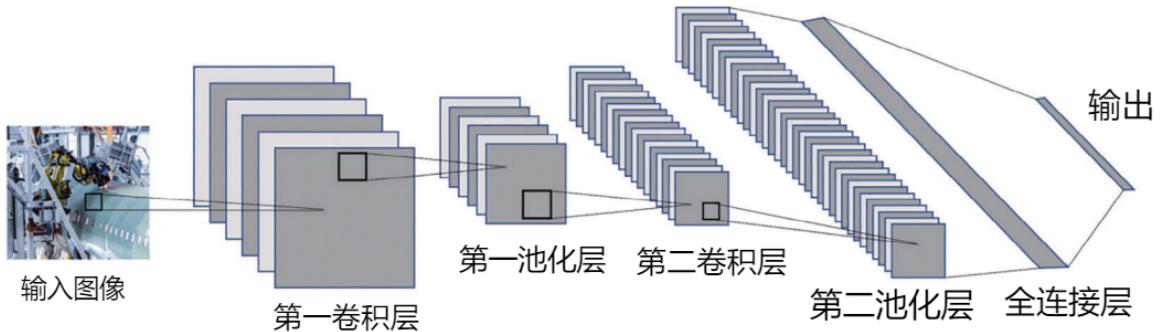


图 1：本文所用的卷积神经网络的主要结构，该神经网络由卷积层、池化层和全连接层顺序连接组合而成。

基于以上考虑，本文提出了一种在虚拟现实场景中用 RGB 摄像机估计目标姿势的方法。首先，采用卷积神经网络的理念，我们提出基于深度学习的网络结构，如图 1 所示。也就是说，该网络以带标记的目标样本图片作为 CNN 的输入，并输出有关目标位置和姿势的六维参数。我们就可以回避摄

像机畸变校正的复杂工作，快速地直接测量目标的位置和姿势。其次，针对以上提出的算法，本文给出了所需的具体数据集，并详细阐述了该数据集的生成方法。这是一种通用方法，可应用于各种目标位置和姿势估计算法的训练。第三，本文证明了该算法流程的通用性。这种基于深度学习、单一用途的估计技术可广泛应用于目标的位置和姿势估计算法。

2. 相关工作

目标位置和姿势的估计领域有大量的工作，估计方法包括使用陀螺仪^[21]、加速计^[22]、摄像头^[23]、各类传感器^[24]等。这些方法中，最方便的就是摄像头，摄像头估计可分为单目视觉估计和双目视觉估计^[25]。与单目视觉相比^[26]，双目视觉估计还提供距离信息，但这会涉及到必要的摄像机校准工作。如果相机损坏、内部参数不准确，估计误差就将失控。单眼的视觉估计系统^[27]包括对协同和非协同目标的测量^[28]。简而言之，协同目标是指在测量过程中与之发生接触的目标，而非协同目标是指在测量过程中不产生直接接触的目标^[29]。根据图像进行目标姿势估计的方法既包括传统的数学算法，又有流行的深度学习方法。

外文翻译原文

Deep learning based six-dimensional pose estimation in virtual reality

Jiachen Yang¹ Yutian Lei¹ Ying Tian² Meng Xi¹

Abstract

Virtual reality technology, with its continuous development, is gradually applied to healthcare, education, business, and other fields. In the application of the technology, position and attitude estimation, as a space positioning technology, is indispensable. Traditional pose estimation has the problems of high dependence on environment and great complexity. But convolutional neural network (CNN) and other technologies with computational intelligence provide a strong guarantee for the progress of pose estimation. This article, based on the theory of CNN in deep learning, as well as monocular vision system and target sample set with markers, proposes a method for estimation of target position and attitude, and at the same time, describes in detail a general way of making dataset with markers based on simulation environment. In this article, the comparative experiments of different network structures show that this measurement method can avoid manual extraction of complex image features, and realize fast, arbitrary and accurate measurement, which plays a key role in pose and attitude measurement. Moreover, the visual correspondence between the world coordinate system and the pixel coordinate system is proved effectively by quaternion.

KEYWORDS

computational intelligence, convolutional neural network, dataset, position and attitude measurement, virtual reality

1 INTRODUCTION

With the continuous development of the information technology, virtual reality (VR) technology will play an increasingly important role in human life.^{1,2} VR technology can bring virtual information into the real world, using multimedia,

three-dimensional (3D) modeling, intelligent interaction, computational intelligence, and so forth. It mainly depends on the development of many key technologies, such as computational intelligence, 3D real-time graphics display, positioning and tracking, tactile and olfactory sensing technology, artificial intelligence technology, parallel computing technology, and human behavior research. VR technology is widely used in many fields, such as the military, medicine,³ industry,⁴ aerospace, education, production, entertainment, and so forth. With the development of VR technology, the real realization of VR will cause great changes in human life and development. Pose estimation is indispensable in the application of VR technology.⁵ It can be used not only for space positioning, but also for object pose and attitude estimation. Among the existing spatial positioning technologies, image recognition is indispensable for the low cost and good mobility,⁶ though its accuracy is low. There is a target pose estimation task, provided with the RGB(D) image sample set containing the object, requiring accurate estimation of the object's position in the environment. Completing this task, RGB camera has a series of advantages, such as wide viewing angle, high resolution, suitability for outdoor work, and so forth.⁷ In addition, RGB image without depth information can expand the application range of pose estimation. Therefore, the combination of RGB information and image pose estimation technology will be widely used in the future. And this method will improve the user experience and efficiency of human-computer interaction in VR technology.⁸ At the same time, in recent years, convolutional neural network (CNN),⁹ one of the representative algorithms of deep learning, develops very rapidly. It is widely used in target detection and classification in computer vision, and a variety of CNNs have come out one after another—LeNet,¹⁰ AlexNet,¹¹ VGG-Net,¹² Inception,¹³ R-CNN,¹⁴ ResNet,¹⁵ DenseNet,¹⁶ and so forth. CNN has provided a deep learning method, which is used in many fields,¹⁷ to solve complex problems of classification or regression.¹⁸ This method can avoid the derivation of complex functional relationship, and directly use neural network

to express the functional relationship. This is an embodiment of computational intelligence. However, the training of neural network inevitably needs enough datasets as support.¹⁹ So a general and patterned training set generation method is also one of the problems to be solved.²⁰ This cannot only make the whole pose estimation process more intelligent, but also save a lot of manual work. With the above considerations, this article proposes a method of using RGB camera to estimate the target's pose and attitude in the VR scene. First of all, using the idea of CNN, we propose a network structure based on deep learning, as shown in Figure 1. That is to use the target sample picture with a marker as the input of CNN, which directly outputs the six-dimensional (6D) parameters of its position and attitude. The complex work of camera distortion correction can thus be avoided, and the position and attitude of the target can be measured quickly and directly. Second, to the algorithm proposed above, this article proposes the specific dataset required, and elaborates on the method of the dataset making. This method is a general method that can be applied in all kinds of algorithms to measure the position and attitude deviation of the target by using markers. Third, this article proves the generality of the algorithm process. This kind of deep-learning, single-purpose measurement method can be widely applied to the position and attitude measurement of the target.

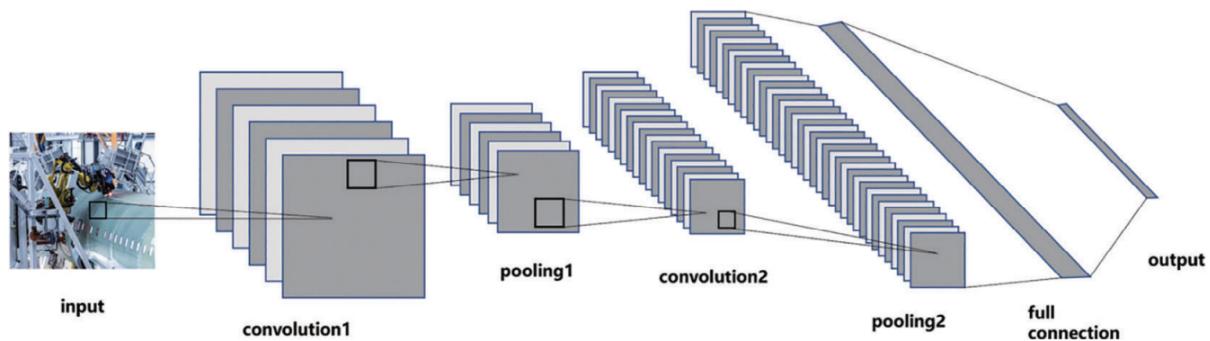


FIGURE 1 The main structure of convolutional neural network used in this article. This Neural network is combined with convolution layer, pooling layer and fully connection layer.

2 RELATED WORK

A lot of work focuses on the estimation of target position and attitude. The measurement methods include using gyroscope,²¹ accelerometer,²² camera,²³ various sensors,²⁴ and so forth. Of these methods, the most convenient one is the use of camera, which can be divided into monocular vision and binocular vision measurement.²⁵ Compared with monocular vision,²⁶ binocular vision has one more distance information, but it inevitably involves camera calibration. If the camera's internal parameters are not accurate, the measurement error will be out of control. Monocular vision measurement system²⁷ includes the measurement of the cooperative and noncooperative targets.²⁸ In short, cooperative targets are ones with communication during measurement, while noncooperative targets are ones not directly contacted in the measurement process.²⁹ In the method of obtaining target pose from images, there are the traditional mathematical solution method and the popular deep-learning method.