|  |  |
| --- | --- |
| **ONI结构化数据传输协议设计** | |
| [备注] | |
|  |  |
| 密级 | 机密 |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **修订历史** | | | |
| 版本 | 修订者 | 审核者 | 日期 | 备注 |
| v0.1 | 常胤（林卫华） | 大鱼（余学稳） | 2020-11-09 | 1. 初版 |
| v0.2 | 常胤（林卫华） | 大鱼（余学稳） | 2021-02-23 | 1. 新增AI流 |
| v0.5 | 常胤（林卫华） | 武松（谢利军） | 2021-07-19 | 1. 新增相位流 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

[ONI结构化数据传输协议设计 3](#_Toc84609355)

[1 概述 3](#_Toc84609356)

[1.1 背景 3](#_Toc84609357)

[1.2 目的 3](#_Toc84609358)

[1.3 对象 3](#_Toc84609359)

[2 数据结构 3](#_Toc84609360)

[2.1 基本单元 3](#_Toc84609361)

[2.2 数据总览 4](#_Toc84609362)

[3 数据节点 5](#_Toc84609363)

[3.1 骨架 5](#_Toc84609364)

[3.1.1 节点定义 5](#_Toc84609365)

[3.1.2 数据负载 6](#_Toc84609366)

[3.2 抠图 7](#_Toc84609367)

[3.2.1 数据负载 7](#_Toc84609368)

[3.3 人体测量 7](#_Toc84609369)

[3.3.1 类型定义 7](#_Toc84609370)

[3.3.2 数据负载 9](#_Toc84609371)

[3.4 平面检测 9](#_Toc84609372)

[3.4.1 数据负载 9](#_Toc84609373)

[3.5 相位流 10](#_Toc84609374)

[3.5.1 数据格式 10](#_Toc84609375)

[3.5.2 数据负载 11](#_Toc84609376)

[4 交互协议 11](#_Toc84609377)

[4.1 AI端点 11](#_Toc84609378)

[4.2 AI流控制 12](#_Toc84609379)

[5 风险 15](#_Toc84609380)

[6 附录 15](#_Toc84609381)

ONI结构化数据传输协议设计

# 概述

## 背景

昆仑山项目是基于RK1126平台，打造TV端的RGBD软硬件标品方案，为了更好地向客户展示，推广RGBD相机在电视行业中的应用，基于产品端提出的演示功能需求，实现AR魔镜演示Demo，主要包括以下功能：

* 3D骨架
* 实时抠图
* 人体测量
* 平面检测

当前OpenNI协议仅支持传输常规的视频流，例如：IR、Depth、Color，为适配设备端的AI结构化数据流，需对OpenNI协议进行扩展。

## 目的

设计先行，为后续方案讨论、编码实现、迭代维护，提供参考依据。

## 对象

本文档适用人员（Orbbec STAR内部人员）：

* 技术
* 产品
* 项目

# 数据结构

## 基本单元

数据节点以TLV格式为基本结构单元，TLV即Tag/Type-Length-Value，是一种可变数据格式，TLV协议简单高效，广泛适用于各种通信场景，具有良好的可扩展性。TLV协议主要包括三个域，分别为：

* Tag/Type：标识域，表示编码数据类型
* Length：长度域，表示数据域实际长度
* Value：数据域，实际数据内容

考虑到客户实际应用场景，特别是在TV端，普遍采用网络协议，而非OpenNI协议传输，故在TLV协议基础上，增加时间戳域，即TTLV格式，并在数据域增加状态码字段，如下图所示：



图 2-1

## 数据总览

在昆仑山项目中，RV1126平台集成的数据节点主要有：

* 骨架
* 抠图
* 人体测量
* 平面检测
* 相位流（新增节点，由常规视频流模式调整为AI流模式）

数据节点负载链路，如下所示：



图 2-2

以上数据节点排列顺序并无强制要求，数据节点可传输单个、或多个，可根据实际应用场景排列组合，上层根据标识域进行节点数据解析。

# 数据节点

## 骨架

### 节点定义

骨架算法支持2D/3D两种模式，骨架点统一定义如下：



图 3-1

骨架点类型定义如下：



图 3-2

### 数据负载

骨架数据负载定义如下：

图 3-3

字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| X/Y/Z | double | 3D模式下为空间坐标；2D模式下为图像坐标，Z轴为0 |
| Score | double | 当前骨架点置信度（<= 0为无效骨架点） |
| Width | uint16 | 计算骨架时输入的RGB图像宽度 |
| Height | uint16 | 计算骨架时输入的RGB图像高度 |
| Group Num | uint32 | 骨架组数量（一人表示一组） |
| User ID | uint32 | 骨架ID，每组骨架对应一个ID |
| Node Num | uint16 | 每组骨架对应的节点数量（0 < num <= 15） |
| Node | struct | 骨架节点：{ X, Y, Z, Score } |

## 抠图

### 数据负载

受RV1126平台算力限制，抠图模块输出的结果为人体掩码数据，非扣好的完整图像，上位机接收到掩码数据后，需对彩色图和人体掩码做进一步的融合渲染处理。

人体掩码数据负载定义如下：



图 3-4

字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| Width | uint16 | 人体掩码图像宽度（像素单位：640） |
| Height | uint16 | 人体掩码图像高度（像素单位：384） |
| Data | uint8[] | 人体掩码数据（bytes = width \* height） |

## 人体测量

### 类型定义

人体测量模块，主要功能是测量人体身高、身材比例等，在AR魔镜中，该功能定义为单人测量。

体型类型定义如下：



图 3-5

类型说明：

* ONI\_FIGURE\_THIN：体型偏瘦
* ONI\_FIGURE\_OVERWEIGHT：体型偏胖
* ONI\_FIGURE\_MEDIUM\_BUILT：体型适中

体型数据定义如下：



图 3-6

字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| ID | uint16 | 被测量的人体ID（保留字段） |
| Ratio | float | 身材比例（保留字段） |
| Height | float | 身高 |
| Waist | float | 腰宽（保留字段） |
| Waistline | float | 腰围（保留字段） |
| Bust | float | 胸围（保留字段） |
| Hips | float | 臀围（保留字段） |
| Shoulder | float | 肩宽（保留字段） |
| Figure | uint32 | 体型（保留字段） |
| Points | unt32[] | 三个人体点坐标（用于辅助调试，判断是否检测到人体） |

### 数据负载

体型数据负载定义如下：



图 3-7

字段说明：

* 当前算法仅支持单人测量，为方便后续扩展，协议设计为支持单人、或多人测量
* Length为TLV协议头数据长度字段

## 平面检测

### 数据负载

平面检测模块输出结果为平面向量和掩码数据，空间坐标定义如下：



图 3-8

平面数据负载定义如下：

图 3-9

字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| Center | OniPoint | 平面中心点 |
| Normal | OniPoint | 平面法向量 |
| Width | uint16 | 平面掩码图像宽度（像素单位：640） |
| Height | uint16 | 平面掩码图像高度（像素单位：480） |
| Data | uint8[] | 平面掩码数据（bytes = width \* height） |

## 相位流

### 数据格式

相位流的数据量和TOF Sensor的工作模式密切相关，不同工作模式，其对应的数据量是不一样的，以33D和Pleco为例，各模式下的相位流数据格式如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sensor/Mode | 单频-NoShuffle | 双频-NoShuffle | 单频-Shuffle | 双频-Shuffle |
| 33D | 1280 x 960 | 1280 x 960 x 2 | 1280 x 960 x 2 | 1280 x 960 x 4 |
| Pleco | 1920 x 480 | 1920 x 480 x 2 | 1920 x 480 x 3 | 1920 x 480 x 6 |
| 备注 | 相位数据解包后每像素占2个字节，每张相位图对应一行扩展数据，数据量计算方式：  Total Bytes = width x (height + 1) x bit-wide / 8 x Num(图片张数) | | | |

在昆仑山项目开发前期，优先适配了33D，33D相位流按逐帧方式输出（非帧组方式），故在OpenNI初版设计中，把相位流当成常规的视频流来处理，而Pleco是以帧组的方式输出，为兼容33D和Pleco，需额外做以下适配工作：

* 平台端需对Pleco帧组进行拆分，按逐帧方式打包发送
* 上位机收到数据后，需把拆分后的帧数据，重新进行组帧

这种方式不仅增加了代码复杂性，在相位流帧率较高的情况下，还会导致上位机应用经常出现组帧失败的情况，为解决以上弊端，统一按帧组的方式，通过TLV格式传输，将相位流当成特殊的AI流节点来处理，这种方式的优势体现在：

* 省去了拆帧、组帧操作，降低了代码复杂性
* 统一了33D和Pleco的处理方式，代码更通用
* 避免帧率过高，导致上位机应用出现组帧失败的情况
* TOF Sensor的工作模式将变得透明，SDK根据节点标识域即可解析出不同模式下的数据
* 相位流不再和深度流复用一个端点，在平台端支持的情况下，可同时输出相位和深度流

### 数据负载

相位流节点数据负载定义如下：



字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| Width | uint32 | 图像宽度（像素单位） |
| Height | uint32 | 图像高度（像素单位） |
| Stride | uint32 | 行字节数 |
| Image Node | uint8[] | 图像数据节点，首行为扩展数据  bytes = stride \* (height + 1) |

# 交互协议

## AI端点

指定EP-0x83，作为AI数据流统一传输端点，定义如下：

|  |  |  |
| --- | --- | --- |
| 端点（EP） | 流类型 | 描述 |
| 0x0 | 控制端点 | 用于控制指令等小型数据传输 |
| 0x81 | Depth | 用于深度流传输 |
| 0x82 | IR | Color | 用于IR或彩色流传输（端点复用） |
| 0x83 | AI | 用于AI流传输（相位、骨架、抠图、人体、平面） |

新增USB数据包类型，以区分AI数据流：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Packet Type | IR | Color | Depth | AI | 描述 |
| SOF | 0x8100 | 0x7100 | 0x6100 | 起始包 |
| Normal | 0x8200 | 0x7200 | 0x6200 | 中间包 |
| EOF | 0x8500 | 0x7500 | 0x6500 | 结束包 |

注意：AI流不是常规视频流，是单个、或多个结构化数据的组合，当数据量较小时，需考虑只有一个数据包的情况。

## AI流控制

新增AI Video Mode类型，用于AI开流指令交互：



图 3-10

新增AI流常规属性，用于流属性获取、设置交互：



图 3-11

新增AI Formats类型，用于AI流模式切换：



图 3-12

字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| JOINT\_2D | enum | 2D骨架 |
| JOINT\_3D | enum | 3D骨架 |
| BODY\_MASK | enum | 抠图掩码 |
| FLOOR\_INFO | enum | 平面信息 |
| BODY\_SHAPE | enum | 体型数据 |
| PHASE | enum | 相位流 |
| FACE | enum | 人脸信息（保留字段） |
| GESTURE | enum | 手势信息（保留字段） |

AI模式切换，交互指令如下：



图 3-13

字段说明：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| Magic | uint16 | 魔数（Host -> Device: 0x4D47） |
| Len | uint16 | 指令数据长度（DWORD类型，即：Data Size / 2） |
| Opcode | uint16 | 操作码 |
| Tag | uint16 | 指令标签 |
| Param Type | uint16 | 参数类型（此处为AI FORMAT属性） |
| Param Value | uint16 | 参数值（此处为XnIOAIFormats） |

注意：AI模式可以是单个、或多个类型的组合（除2D/3D骨架不能同时输出），例如期望同时获取骨架和体型数据，Param Value可设置为：XN\_IO\_AI\_FORMAT\_JOINT\_3D | XN\_IO\_AI\_FORMAT\_BODY\_SHAPE。

由于当前TOF Sensor不支持同时输出相位和深度流，因此依赖深度的算法节点和相位节点是互斥的，依赖深度的算法节点有：

* 3D骨架
* 平面检测
* 人体测量

以上算法不能和相位流同时使能。

# 风险

* 在AR魔镜功能定义中，AI功能模块（骨架、抠图、人体测量、平面检测）单独运行，无需同时运行所有AI模块，考虑实际应用场景，如游戏开发，可能需要用到多维度数据，如骨架、手势、平面、视频流等，此场景下，平台算力、传输带宽可能成为瓶颈。

# 附录

略。