AVS\_RPP接口使用手册V1.0.0

任亦立

浙江省北大信息技术高等研究院

AVS中心AIOT实验室

2020年12月16日

**目 录**

[1. 引言 1](#_Toc59022252)

[1.1 设计目标 1](#_Toc59022253)

[1.2 适用范围 1](#_Toc59022254)

[1.3 相关术语 1](#_Toc59022255)

[1.4 参考资料 1](#_Toc59022256)

[2. 宏 2](#_Toc59022257)

[3. 枚举 3](#_Toc59022258)

[4. 结构体 3](#_Toc59022259)

[4.1.1 AVS\_BUF 3](#_Toc59022260)

[4.1.2 AVS\_POINT\_I 3](#_Toc59022261)

[4.1.3 AVS\_POINT\_F 4](#_Toc59022262)

[4.1.4 AVS\_MEM\_TAB 4](#_Toc59022263)

[4.1.5 AVS\_MEM 4](#_Toc59022264)

[4.1.6 AVS\_FRAME\_INFO 4](#_Toc59022265)

[4.1.7 AVS\_RADAR\_POINT\_INFO 4](#_Toc59022266)

[4.1.8 AVS\_RADAR\_INFO 5](#_Toc59022267)

[4.1.9 AVS\_CAMERA\_INFO 5](#_Toc59022268)

[4.1.10 AVS\_RADAR\_CALIB 5](#_Toc59022269)

[4.1.11 AVS\_RADAR\_PROCESS\_IN 6](#_Toc59022270)

[4.1.12 AVS\_RADAR\_PROCESS\_OUT 6](#_Toc59022271)

[4.1.13 AVS\_RADAR\_POSTPROCESS\_IN(待完善) 6](#_Toc59022272)

[4.1.14 AVS\_RADAR\_POSTPROCESS\_OUT(待完善) 6](#_Toc59022273)

[5. 分析设计 7](#_Toc59022274)

[5.1 设计思想 7](#_Toc59022275)

[5.2 外部接口 7](#_Toc59022276)

[5.2.1 AVS\_GetMemSize 7](#_Toc59022277)

[5.2.2 AVS\_CreatMemSize 7](#_Toc59022278)

[5.2.3 AVS\_PreProcess 8](#_Toc59022279)

[5.2.4 AVS\_PostProcess 8](#_Toc59022280)

[5.3 数据流分析 9](#_Toc59022281)

[6. 错误码 10](#_Toc59022282)

[7. 修订记录 10](#_Toc59022283)

[附录1 XXXX 11](#_Toc59022284)

1. 引言
   1. 设计目标

为了构建实时的端到端的雷达-图像融合算法，需要对雷达数据和图像数据进行预处理后送入神经网络进行推理，然后拿到推理数据进行后处理得到目标框，速度等多传感器融合信息，为此设计融合数据预处理和后处理的接口供嵌入式平台系统进行数据处理。

* 1. 适用范围

本文档目标范围是AVS中心嵌入式软件实验室和AIOT实验室相关算法人员和软件开发人员。

* 1. 相关术语

|  |  |  |
| --- | --- | --- |
| 术语 | 说明 | 备注 |
| RPP | Radar Info PrePost Processing | 雷达信号预处理后处理库 |
|  |  |  |
|  |  |  |

* 1. 参考资料

无。

1. 宏

|  |  |  |
| --- | --- | --- |
| 宏定义 | 值 | 说明 |
| Avs\_Base.h | | |
| AVS\_MAX\_MEM\_NUM | 4 | 算法库支持的最大内存表数量 |
| AVS\_MEM\_ALIGN\_128BYTE | 128 | 算法库内存的对齐字节数 |
| QWORD | unsigned long long | 算法库内存地址数据类型 |
| AVS\_PI | 3.141592653 | π值 |
| Avs\_Radar.h | | |
| AVS\_MAX\_RADAR\_NUM | 125 | 雷达单词扫描支持的最大雷达点数 |
| AVS\_RADAR\_MEM\_TAB | 1 | 算法库用到的内存表数量 |
| AVS\_MAX\_RADAR\_CACHE | 20 | 算法库所需的最近的雷达扫描次数 |
| AVS\_POST\_OUT\_LENGTH | 46035 | model输出向量维度 |
| AVS\_POST\_OUT\_MXA\_OBJ | 128 | 输出最多目标数 |
| AVS\_INFRARED\_CHANNEL | 1 | 红外图像（白热）channel |
| AVS\_RADAR\_INPUT\_CHANNEL | 3 | 预处理输入向量channel |
| AVS\_RADAR\_OUTPUT\_CHANNEL | 5 | 预处理输出向量channel |
| AVS\_MAX\_BUF\_LEN | 1024 | 临时缓存大小 |

1. 枚举

|  |  |  |
| --- | --- | --- |
| 宏定义 | 值 | 说明 |
| Avs\_Base.h | | |
|  |  |  |
| Avs\_Common.h | | |
| AVS\_RADAR\_INVALID\_STATES | AVS\_RADAR\_INVALID\_STATE\_VALID | 0x00 |
| AVS\_RADAR\_INVALID\_STATE\_OTHER\_VALID | 0x04,0x08,0x09,0x0a,0x0b,  0x0c,0x0f,0x10,0x11 |
| AVS\_RADAR\_INVALID\_STATE\_OTHER\_INVALID | 0x01,0x02,0x03,0x05,0x06,  0x07,0x0d,0x0e |
| AVS\_RADAR\_DYNPROP\_STATES | AVS\_RADAR\_DYNPROP\_STATE\_MOVING | 0 |
| AVS\_RADAR\_DYNPROP\_STATE\_STATIONARY | 1 |
| AVS\_RADAR\_DYNPROP\_STATE\_ONCOMING | 2 |
| AVS\_RADAR\_DYNPROP\_STATE\_STATIONARY\_CANDIDATE | 3 |
| AVS\_RADAR\_DYNPROP\_STATE\_UNKNOWN | 4 |
| AVS\_RADAR\_DYNPROP\_STATE\_CROSSING\_STATIONARY | 5 |
| AVS\_RADAR\_DYNPROP\_STATE\_CROSSING\_MOVING | 6 |
| AVS\_RADAR\_DYNPROP\_STATE\_STOPPED | 7 |
| AVS\_RADAR\_AMBIG\_STATES | AVS\_RADAR\_AMBIG\_STATE\_INVALID | 0 |
| AVS\_RADAR\_AMBIG\_STATE\_AMBIGUOUS | 1 |
| AVS\_RADAR\_AMBIG\_STATE\_STAGGERED\_RAMP | 2 |
| AVS\_RADAR\_AMBIG\_STATE\_UNAMBIGUOUS | 3 |
| AVS\_RADAR\_AMBIG\_STATE\_STATIONARY\_CANDIDATE | 4 |
| AVS\_RADAR\_OBJ\_CLASS | AVS\_RADAR\_OBJ\_BACKGROUND | 0 |
| AVS\_RADAR\_OBJ\_CAR | 1 |
| AVS\_RADAR\_OBJ\_BUS | 2 |
| AVS\_RADAR\_OBJ\_MOTORCYCLE | 3 |
| AVS\_RADAR\_OBJ\_TRAILER | 4 |
| AVS\_RADAR\_OBJ\_TRUCK | 5 |
| AVS\_RADAR\_OBJ\_BIBYCLE | 6 |
| AVS\_RADAR\_OBJ\_HUMAN | 7 |
| AVS\_RADAR\_OBJ\_END | 8 |
| Avs\_Radar.h | | |
|  |  |  |

1. 结构体
   * 1. AVS\_POINT\_I

|  |  |  |
| --- | --- | --- |
| Avs\_Base.h | | |
| 定义 | 成员 | 说明 |
| AVS\_POINT\_I | int x; | 三维点x坐标，整型 |
| int y; | 三维点y坐标，整型 |
| int z; | 三维点z坐标，整型，二维坐标置为0 |

* + 1. AVS\_POINT\_F

|  |  |  |
| --- | --- | --- |
| Avs\_Base.h | | |
| 定义 | 成员 | 说明 |
| AVS\_POINT\_F | float x; | 三维点x坐标，浮点型 |
| float y; | 三维点y坐标，浮点型 |
| float z; | 三维点z坐标，浮点型，二维坐标置为0 |

* + 1. AVS\_MEM\_TAB

|  |  |  |
| --- | --- | --- |
| Avs\_Base.h | | |
| 定义 | 成员 | 说明 |
| AVS\_MEM\_TAB | unsigned int size; | 内存段大小 |
| unsigned int alignment; | 内存段对齐位数 |
| void \*base; | 内存段起始地址，未分配时为NULL |

* + 1. AVS\_MEM

|  |  |  |
| --- | --- | --- |
| Avs\_Base.h | | |
| 定义 | 成员 | 说明 |
| AVS\_MEM | int mem\_tab\_num; | 内存表数量 |
| AVS\_MEM\_TAB mem\_tab[AVS\_MAX\_MEM\_NUM]; | 内存表定义 |

* + 1. AVS\_FRAME\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_FRAME\_INFO | int frame\_num; | 帧号 |
| int width; | 图像宽 |
| int height; | 图像高 |
| int channel; | 数据通道数 |
| float \*data; | 图像数据，行优先，RGB平铺 |

备注：RGB平铺为RRRGGGBBB格式，RGB混合为RGBRGBRGB格式。

* + 1. AVS\_RADAR\_POINT\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POINT\_INFO | unsigned int id; | 雷达点id |
| unsigned int dyn\_prop; | 雷达点运动信息 |
| float rcs; | 雷达目标有效散射截面积 |
| int is\_quality\_valid; | 雷达点质量有效性状态 |
| int ambig\_state; | 多普勒模糊性状态 |
| int invalid\_state; | 雷达点有效性状态 |
| int phd0; | 雷达点的虚警概率 |
| AVS\_POINT\_F radar\_point; | 雷达点坐标 |
| AVS\_POINT\_F radar\_v; | 雷达点速度 |
| AVS\_POINT\_F radar\_v\_comp; | 雷达点相对速度 |
| AVS\_POINT\_F radar\_rms; | 雷达点均方根误差 |
| AVS\_POINT\_F radar\_v\_rms; | 雷达点速度均方根误差 |

* + 1. AVS\_RADAR\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_INFO | int radar\_point\_num; | 当前时刻的雷达点数量 |
| AVS\_RADAR\_POINT\_INFO\* radar\_point\_info; | 雷达点信息 |

* + 1. AVS\_CAMERA\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_CAMERA\_INFO | float fu; | x方向归一化焦距 |
| float fv; | y方向归一化焦距 |
| float cu; | x光轴中心 |
| float cv; | y光轴中心 |
| float pitch; | 俯仰角 |
| float yaw; | 水平角 |
| float roll; | 横摆角 |
| float height; | 相机高度mm |
| float k1; | 畸变系数k1 |
| float k2; | 畸变系数k2 |
| float k3; | 畸变系数k3 |
| float p1; | 畸变系数p1 |
| float p2; | 畸变系数p2 |

* + 1. AVS\_RADAR\_CALIB

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_CALIB | float radar\_radian; | 雷达对于相机的横摆角 |
| float radarInCamera\_X; | 雷达对于相机的x平移 |
| float radarInCamera\_Y; | 雷达对于相机的y平移 |
| float radar\_height; | 雷达假设高度 |
| float radar\_top; | 先验值，物体真实高度，3000mm |

* + 1. AVS\_RADAR\_DEBUG\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_DEBUG\_INFO | FILE \*fp[AVS\_MAX\_RADAR\_CACHE]; | 雷达信息调试信息文件句柄 |
| FILE \*fp\_point; | 雷达点调试信息 |
| FILE \*fp\_point\_xyz; | 雷达点调试信息 |
| FILE \*out\_map; | 输出的雷达特征图 |

* + 1. AVS\_IMAGE\_CONFIG

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_IMAGE\_CONFIG | int undistort\_avalible; | 是否进行畸变校正 |
| int infrared\_align; | 是否进行红外对齐 |

* + 1. AVS\_INFRARED\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_INFRARED\_INFO | double m11; | 红外相机的仿射变换矩阵 |
| double m12; |
| double m13; |
| double m21; |
| double m22; |
| double m23; |
| double m31; |
| double m32; |
| double m33; |

* + 1. AVS\_RADAR\_PREPROCESS\_IN

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_PREPROCESS\_IN | int radar\_info\_cache\_num; | 雷达信息缓存数量 |
| AVS\_RADAR\_INFO radar\_info[AVS\_MAX\_RADAR\_CACHE]; | 不同时刻雷达信息 |
| AVS\_FRAME\_INFO frame\_info; | 当前视频帧 |
| AVS\_FRAME\_INFO infrared\_frame\_info | 当前红外帧 |
| AVS\_CAMERA\_INFO cam\_info; | 相机参数 |
| AVS\_RADAR\_CALIB radar\_calib; | 雷达参数 |
| AVS\_IMAGE\_CONFIG image\_config; | 配置信息 |
| AVS\_RADAR\_DEBUG\_INFO debug\_info; | 输入调试信息 |

* + 1. AVS\_RADAR\_PREPROCESS\_OUT

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_PREPROCESS\_OUT | int width; | 输出特征图宽 |
| int height; | 输出特征图高 |
| int channels; | 输出特征图通道 |
| float \*data; | 输出特征图数据 |

* + 1. AVS\_RADAR\_INTERPOLATIONPROCESS\_OUT

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_INTERPOLATIONPROCESS\_OUT | AVS\_RADAR\_PREPROCESS\_OUT rgb\_info; | 降采样rgb信息 |
| AVS\_RADAR\_PREPROCESS\_OUT infrared\_info; | 降采样红外信息 |

* + 1. AVS\_RADAR\_POSTPROCESS\_CLASSIFICATION

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POSTPROCESS\_CLASSIFICATION | float conf[AVS\_RADAR\_OBJ\_END]; | 置信度信息 |

* + 1. AVS\_RADAR\_POSTPROCESS\_REGRESSION

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POSTPROCESS\_REGRESSION | float x1; | x1回归信息 |
| float y1; | y1回归信息 |
| float x2; | x2回归信息 |
| float y2; | y2回归信息 |

* + 1. AVS\_RADAR\_POST\_DEBUG\_INFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POST\_DEBUG\_INFO | FILE \*classification; | 分类的调试信息句柄 |
| FILE \*regression; | 回归的调试信息句柄 |

* + 1. AVS\_RADAR\_POSTPROCESS\_IN

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POSTPROCESS\_IN | AVS\_RADAR\_POST\_DEBUG\_INFO post\_debug\_info; | 后处理输入调试信息 |
| AVS\_RADAR\_POSTPROCESS\_CLASSIFICATION \*classification[AVS\_POST\_OUT\_LENGTH]; | 后处理输入分类信息 |
| AVS\_RADAR\_POSTPROCESS\_REGRESSION \*regression[AVS\_POST\_OUT\_LENGTH]; | 后处理输入回归信息 |

* + 1. AVS\_RADAR\_POSTINFO

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POSTINFO | int valid; | 有效标识位 |
| int x1; | 左上角x坐标 |
| int y1; | 左上角y坐标 |
| int x2; | 右下角x坐标 |
| int y2; | 右下角y坐标 |
| int cls; | 类别 |
| float confidence; | 置信度 |
| float velocity; | 速度 |
| float distance; | 距离 |
| float temp; | 根据红外图像像素值映射的温度 |
| float temp\_pixel; | 该目标红外图像像素值 |

* + 1. AVS\_RADAR\_POSTPROCESS\_OUT

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_POSTPROCESS\_OUT | int obj\_num; | 输出目标数量 |
| AVS\_RADAR\_POSTINFO \*obj\_info[AVS\_POST\_OUT\_MXA\_OBJ]; | 目标信息 |

* + 1. AVS\_RADAR\_PROCESS\_IN

|  |  |  |
| --- | --- | --- |
| Avs\_Radar.h | | |
| 定义 | 成员 | 说明 |
| AVS\_RADAR\_PROCESS\_IN | AVS\_RADAR\_PREPROCESS\_IN pre\_input; | 预处理结构体 |
| AVS\_RADAR\_POSTPROCESS\_IN post\_input; | 后处理结构体 |

1. 分析设计
   1. 设计思想

该算法库设计分为六个外部接口，解释说明如下：

|  |  |
| --- | --- |
| 术语 | 说明 |
| AVS\_GetMemSize | 计算算法库所需内存 |
| AVS\_CreatMemSize | 分配算法库所需内存 |
| AVS\_UndistortProcess | 畸变矫正 |
| AVS\_InterpolationProcess | 降采样 |
| AVS\_PreProcess | 雷达数据预处理 |
| AVS\_PostProcess | 雷达数据后处理 |

* 1. 外部接口
     1. AVS\_GetMemSize

配置AVS\_RADAR\_PROCESS\_IN结构体基本参数，初始化AVS\_MEM\_TAB内存表，调用AVS\_GetMemSize接口进行内存表信息更新，即计算得到算法库所需内存。

// 初始化输入句柄 包含预处理句柄和后处理句柄

AVS\_RADAR\_PROCESS\_IN input;

memset(&input, 0, sizeof(AVS\_RADAR\_PROCESS\_IN)); // 句柄初始化

AVS\_RADAR\_PREPROCESS\_IN \*pre\_input = &input.pre\_input;

pre\_input->frame\_info.width = image\_resize.cols;

pre\_input->frame\_info.height = image\_resize.rows;

pre\_input->frame\_info.channel = 3;

pre\_input->infrared\_frame\_info.width = image\_infrared\_resize.cols;

pre\_input->infrared\_frame\_info.height = image\_infrared\_resize.rows;

pre\_input->infrared\_frame\_info.channel = 1;

pre\_input->infrared\_calib\_info.m11 = 0.989852860539301;

pre\_input->infrared\_calib\_info.m12 = -0.1461011482095614;

pre\_input->infrared\_calib\_info.m13 = 355.6865876136517;

pre\_input->infrared\_calib\_info.m21 = 0.03590646316358077;

pre\_input->infrared\_calib\_info.m22 = 0.8554565306626785;

pre\_input->infrared\_calib\_info.m23 = 89.58020698233526;

pre\_input->infrared\_calib\_info.m31 = 2.360413366804856e-05;

pre\_input->infrared\_calib\_info.m32 = -8.960393897429334e-05;

pre\_input->infrared\_calib\_info.m33 = 1;

// 内存计算

AVS\_MEM\_TAB mem\_tab[AVS\_RADAR\_MEM\_TAB];

AVS\_GetMemSize(&input, mem\_tab);

* + 1. AVS\_CreatMemSize

因为算法库内部不涉及任何内存申请操作，计算算法库所需内存后，一定要在算法库外部首先使用AVS\_alloc\_mem\_tab进行内存申请，注意按要求对齐。

计算并申请到内存之后调用AVS\_CreatMemSize，该接口会对AVS\_RADAR\_PROCESS\_IN以及void \*filter中的指针进行内存分配。

// 内存申请

AVS\_alloc\_mem\_tab(mem\_tab, AVS\_RADAR\_MEM\_TAB);

// 内存分配

void \*filter = nullptr;

AVS\_CreatMemSize(&radar\_param, mem\_tab, &filter);

* + 1. AVS\_UndistortProcess

上述两个接口在算法库初始化的时候调用一次即可，接下来逐帧进行畸变矫正，降采样，预处理和后处理。

// 图像数据输入 rgb数据

// get\_image\_info(pre\_input, image\_resize);

// 相机参数设置

pre\_input->cam\_info.fu = 1007.84;

pre\_input->cam\_info.fv = 1014.16;

pre\_input->cam\_info.cu = 649.10;

pre\_input->cam\_info.cv = 360.32;

pre\_input->cam\_info.height = 1750; // mm 相机离地面高度

pre\_input->cam\_info.pitch = 1.4 \* (AVS\_PI \* 1.0 / 180);

pre\_input->cam\_info.yaw = 0 \* (AVS\_PI \* 1.0 / 180);

pre\_input->cam\_info.k1 = -0.3923900430945697;

pre\_input->cam\_info.k2 = 0.1413877908438226;

pre\_input->cam\_info.k3 = 0.002443674780940085;

pre\_input->cam\_info.p1 = -0.003599841241478862;

pre\_input->cam\_info.p2 = -0.01032105435135168;

// 雷达数据输入 x, y, rcs, invalid\_state, dyn\_prop, ambig\_state, radar\_v\_comp一定需要

//get\_radar\_point(pre\_input, txts);

// 雷达参数设置

pre\_input->radar\_calib.radar\_height = 500; // 雷达架设高度

pre\_input->radar\_calib.radar\_top = 3000; // 目标估计高度，固定为3000

pre\_input->radar\_calib.radar\_radian = 0 \* (AVS\_PI \* 1.0 / 180);

pre\_input->radar\_calib.radarInCamera\_X = 0; // 假设没有水平距离

pre\_input->radar\_calib.radarInCamera\_Y = 0; // 假设没有前后距离

// 畸变校正参数设置

pre\_input->image\_config.undistort\_avalible = 0; // 开启畸变校正

// 图像畸变矫正接口

AVS\_RADAR\_PREPROCESS\_OUT undistort\_output;

t = (double)cv::getTickCount();

AVS\_UndistortProcess(filter, pre\_input, sizeof(AVS\_RADAR\_PREPROCESS\_IN), &undistort\_output, sizeof(AVS\_RADAR\_PREPROCESS\_OUT));

t = ((double)cv::getTickCount() - t) / cv::getTickFrequency();

cout << "undistortprocess cost time " << t << "s" << endl;

* + 1. AVS\_InterpolationProcess

畸变矫正后进行降采样。

// 降采样参数配置

pre\_input->image\_config.infrared\_align = 1; // 开启红外对齐，warpPerspective功能

// 图像降采样接口 目前固定到640\*384

AVS\_RADAR\_PREPROCESS\_OUT resize\_output;

t = (double)cv::getTickCount();

AVS\_InterpolationProcess(filter, pre\_input, sizeof(AVS\_RADAR\_PREPROCESS\_IN), &resize\_output, sizeof(AVS\_RADAR\_PREPROCESS\_OUT));

t = ((double)cv::getTickCount() - t) / cv::getTickFrequency();

cout << "interpolationprocess cost time " << t << "s" << endl;

* + 1. AVS\_PreProcess

对每一帧调用接口AVS\_PreProcess对雷达和图像数据进行预处理，得到的预处理数据AVS\_RADAR\_PROCESS\_OUT直接用于神经网络的前向推理。

// 雷达数据预处理

AVS\_RADAR\_PROCESS\_OUT output;

AVS\_PreProcess(filter, input, sizeof(AVS\_RADAR\_PROCESS\_IN), &output, sizeof(AVS\_RADAR\_PROCESS\_OUT));

* + 1. AVS\_PostProcess

之后将预处理得到的数据送入神经网络进行前向推理，得到feature map，用得到的feature map进行后处理。

// 雷达数据后处理

AVS\_RADAR\_POSTPROCESS\_OUT post\_output;

t = (double)cv::getTickCount();

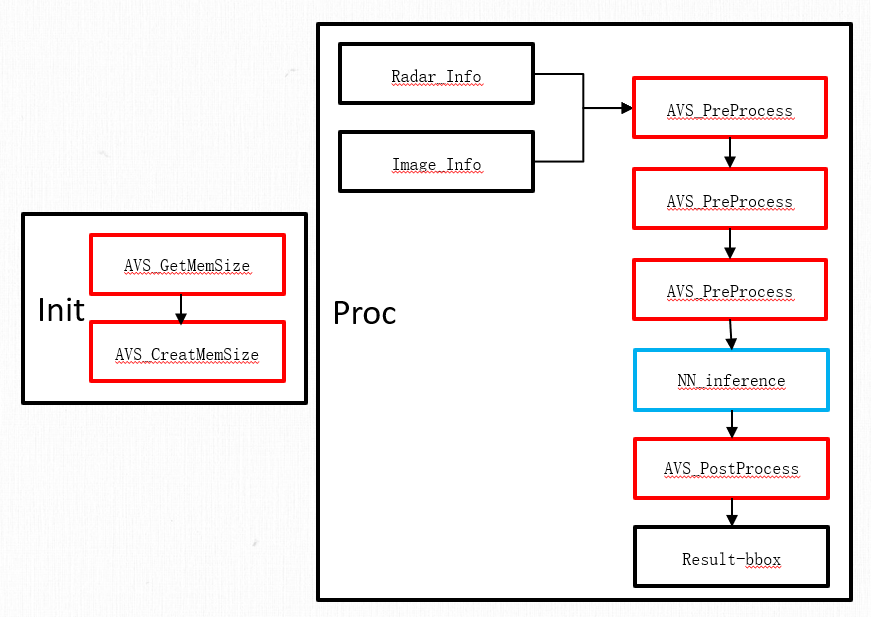
AVS\_PostProcess(filter, post\_input, sizeof(AVS\_RADAR\_POSTPROCESS\_IN), &post\_output, sizeof(AVS\_RADAR\_POSTPROCESS\_OUT));

t = ((double)cv::getTickCount() - t) / cv::getTickFrequency();

cout << "postprocess cost time " << t << "s" << endl;

* 1. 数据流分析

如下图所示，红色框图为算法库接口，蓝色是神经网络前向推理。Init在取流前运行一次即可，Proc逐帧进行算法处理。



1. 错误码

|  |  |  |
| --- | --- | --- |
| 宏定义 | 值 | 说明 |
| Avs\_ErrorCode.h | | |
| AVS\_LIB\_OK | 0x00000000 | 成功 |
| AVS\_LIB\_MEM\_OUT | 0x60000000 | 内存不够 |
| AVS\_LIB\_RESOLUTION\_UNSUPPORT | 0x60000001 | 分辨率不支持 |
| AVS\_LIB\_PTR\_NULL | 0x60000002 | 传入指针为空 |
| AVS\_LIB\_KEY\_PARAM\_ERR | 0x60000003 | 高级参数设置错误 |

1. 修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改日期 | 修改人 | 修改内容 | 备注 |
| 1.0.0 | 2020-12-16 | 任亦立 | 新建 |  |
| 1.1.0 | 2020-12-24 | 任亦立 | 增加畸变，降采样，后处理接口 |  |
| 1.2.0 | 2021-01-11 | 任亦立 | 增加红外信息输入输出 |  |

附录1 XXXX