

朝闻道

TAObotics

HandsFree 开源软硬件系统 机器人及行业应用解决方案

深圳朝闻道智能信息科技有限公司



目录



TAObotics

1.团队简介

Company Profile

2.HandsFree开源项目

Handsfree Open Source Project

3.项目生态和应用案例

Project Ecology And Application Cases

关于我们

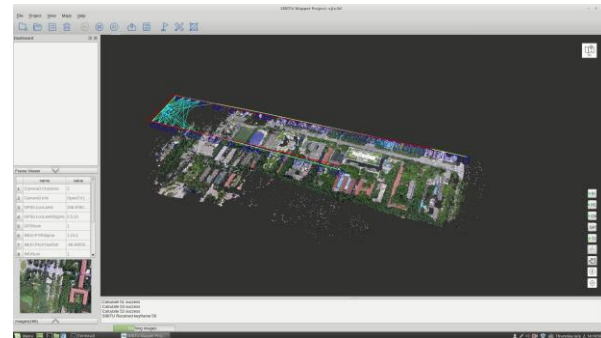
ABOUT US



飞行器智能感知与控制实验室 (PI-Lab) 实验室主要从事机器人智能感知与控制方面的研究与系统开发，重点研究视觉SLAM、地图重建、图像处理、机器学习等方面的研究。研究过程中提出了一系列原创的方法，包括通用SLAM平台GSLAM，地图融合算法Map2DFusion、TerrainFusion、DenseFusion等方法。并有多个开源软硬件项目，涵盖视觉SLAM、机器人、无人机等领域，受到国内外广泛关注。



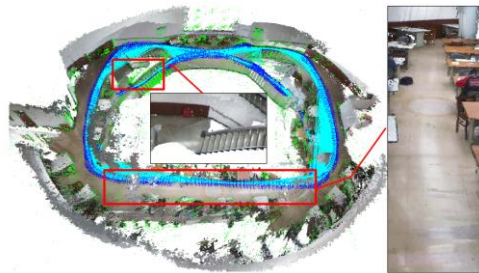
实时二维重建



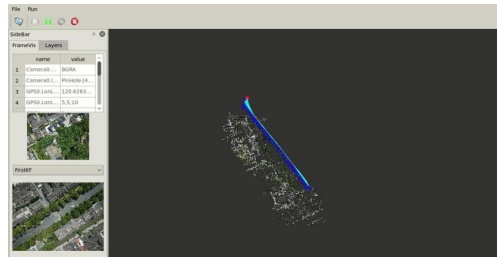
实时三维重建



差异检测



大规模场景三维重建



视觉SLAM



路径优化与控制



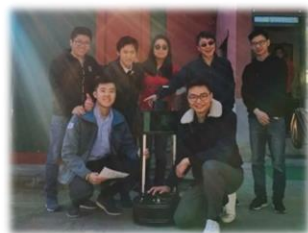
成长历程

GROWTH PROCESS

TAObotics



公司在深圳市南山区成立，核心技术为机器人室内大场景“司南”定位导航系统及机器人软硬件控制系统。同时专注机器人产品和解决方案



与顶级食品商玛氏集团、华润集团旗下多个子品牌达成合作、推出智慧零售机器人并完成交付使用

2014年11月



创始人于西北工业大学智能系统实验室，发起**开源机器人项目—HandsFree**
超200所高校科研机构合作
超2000名活跃机器人开发者社区

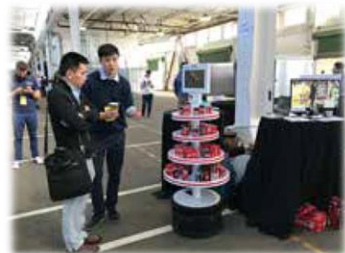


2018年5月

2019年4月



获得来自YC中国、百度Apollo 无人驾驶项目发起人，前微软全球副总裁**陆奇博士**天使轮融资；
导师团为拼多多公司创始人**黄峥先生**
宝宝树集团创始人**王怀南先生**



2019年6月

持续深耕零售数字化场景，拓展墨西哥零售市场，与南非，墨西哥，阿根廷商超等代理商达成合作；

2020年4月



全向系列®

四驱系列®

履带系列®

差速系列®

服务系列®

Octopus



Thunder



Soldier



Turtle



Stone



Octopus Pro



Thunder Pro



Soldier Pro



Turtle Pro



Stone Pro



行业机器人解决方案

Industrial Robotic Solution

TAObotics

智慧零售

MallBot系列®

1st Generation



2nd Generation



智慧安防巡检

Guardian系列®

工厂巡检员



抗疫卫士

Anti-Covid系列®

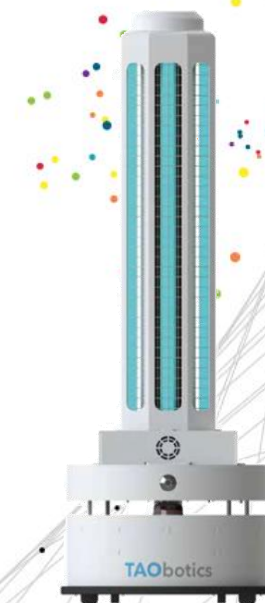
喷雾消杀机器人



抗疫卫士

Anti-Covid系列®

紫外线灭菌机器人Pro



2. 开源项目介绍

Open Source Project Introduction



HandsFree 开源项目主要包含以下几个部分：

实时控制&通讯
OpenRE-
Robolink-
RobolinkGUI



应用操作系统
&模块组织
ROS-
PICMake-Svar



感知&规划&决策
行业级司南
导航系统-
GSLAM

2.1 机器人实时控制系统 OpenRE



Robot Real-time Control System OpenRE

OpenRE全称Open Source Robot Embedded Library，是一个专业的机器人嵌入式开源库。

完备鲁棒的体系：涵盖底层CPU解耦层，设备和传感器驱动，算法，机器人模型，通信件与操作系统组件等。经过不断实验和优化，开源库变得鲁棒和丰富。

泛化性能强：全C，C++编写，可移植性能强，支持多种处理器，多种机器人模型。

易用和开放：跨平台，支持Linux和Windows开发环境，能方便的结合ROS，PIXHAWK，源码开放，具备很好的研习价值。

OpenRE Hands Free Middleware

Main Components

OS Components

RTOS

GUI

FATFS

SHELL
Etc..

LWIP

USB

Communication :
HFLINK(Robolink), Mavlink

Robot Abstract: chassis ,arm(s)
gripper(s),head , hand,...

Device
Drivers

Algorith
m
Libs

Mathlibs,
Matrix,...

Hardware Abstract :// different board and standard operation interface

Hardware :// IO Interface

PWM

UART

CAN

SPI

ENCODER

USB

IIC

ADC

TIMER

IO

SDIO

etc

Hardware :// Sensors and IO Equipments/

RC

IMU/GPS
3D ACC / Gyro /
MAG / Baro

LCD

FLOW

Memory

RF

SEVORS

MOTORS

etc

Various MCUs : STM3,M4,M7; AVR etc..

应用展示和产业化案例

Application Demonstration And Industrialization Case

全向系列®

Octopus



Octopus Pro



四驱系列®

Thunder



Thunder Pro



履带系列®

Soldier



Soldier Pro



差速系列®

Turtle



Turtle Pro



服务系列®

Stone



Stone Pro



智慧零售
MallBot系列®
1st Generation



2nd Generation



智慧安防巡检
Guardian系列®
工厂巡检员



抗疫卫士
Anti-Covid系列®
喷雾消杀机器人



抗疫卫士
Anti-Covid系列®
紫外线灭菌机器人Pro



TAObotics

开源通用机器人控制器



OpenRE的主要功能模块：

运动控制：伺服电机控制、运动学/动力学解算、里程计融合等。

电源管理：外设供电&电池管理、安全保护&低功耗。

执行机构管理：底盘、云台、升降、机械臂等机构的控制管理。

传感器管理：IMU，GPS，超声波等。

通讯管理：参数管理、Robolink和机器人抽象。

交互模块：本地可视化监控、云端监控和控制。

开源地址：<https://github.com/HANDS-FREE/OpenRE>



不同运动模型的机器人

Open Source Education Solutions

TAObotics

全向系列®

四驱系列®

履带系列®

差速系列®

服务系列®

Octopus



Thunder



Soldier



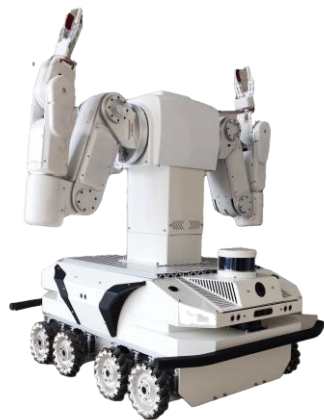
Turtle



Stone



Octopus Pro



Thunder Pro



Soldier Pro



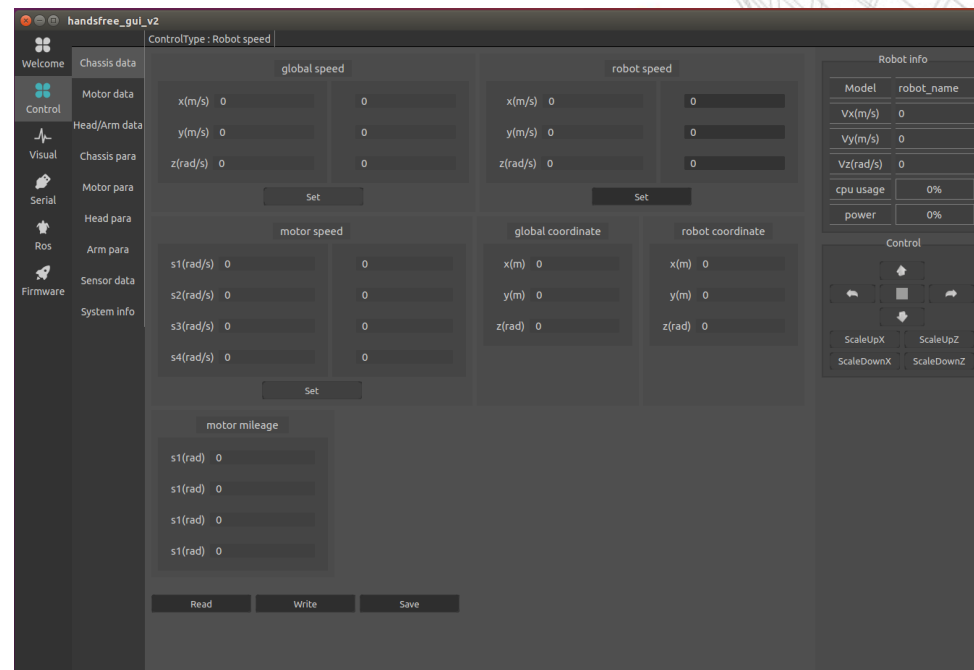
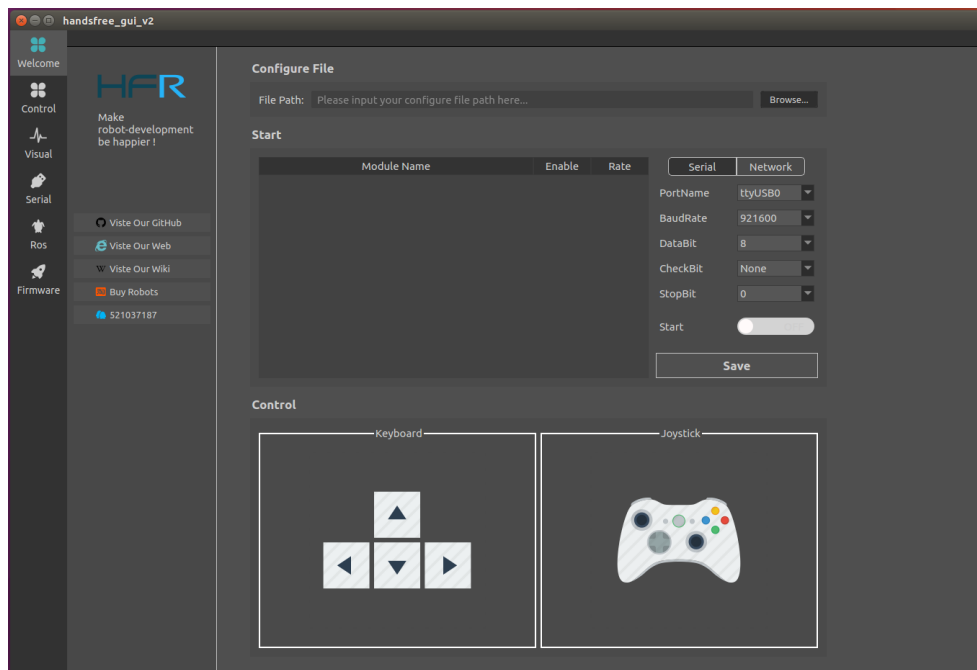
Turtle Pro



Stone Pro



Robolink： 轻量、高效、稳定的移动机器人通讯控制协议。
支持C++、Python。支持ROS。
RobolinkGUI： 可视化开发调试工具，参数管理。

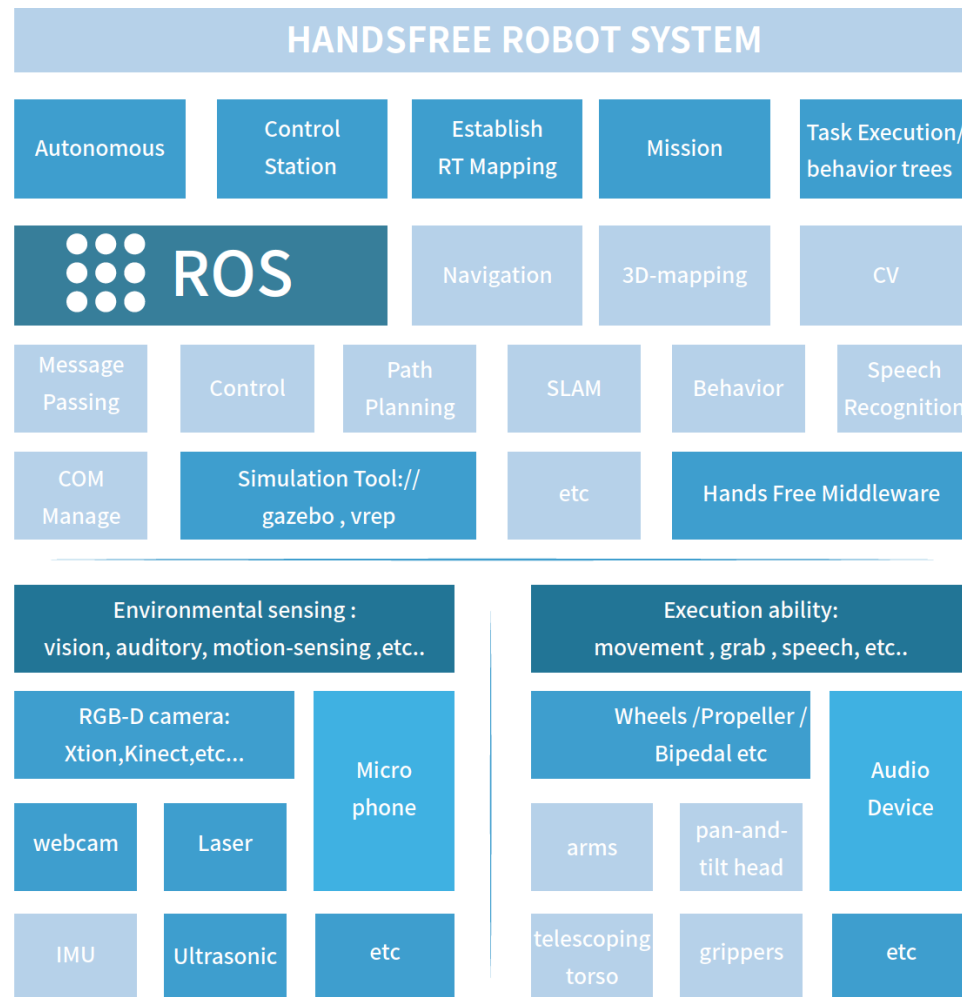


2.2 应用操作系统&模块组织

Application Operating System & Module Organization

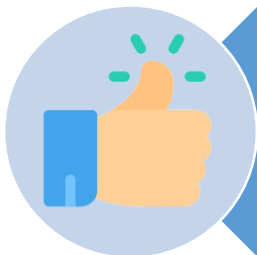


- 多传感器支持：立体视觉，Laser，RGB-D等
- 分布式架构设计，支持机器人ROS系统
- 完备的开源机器人开发研究平台产品线
- 软硬件开源，完善的开发手册、教程
- 丰富的开发调试可视化工具，手机、WEB工具
- HandsFree开源社区支持，数十个开源应用案例，150+机构和2000+开发者用户
- 配合OpenRE实现机、电、控、软产业化系统方案
- 可脱离ROS实现产业化开发，提供产业化技术支持





是否曾消耗大量时间解决（开源）代码编译依赖问题？是否曾为C++的开发测试效率，模块解耦头疼？



如何不依赖ROS的同时和ROS高效通信？ROS如何连接广域网实现云端管理？



ROS只能在几个固定Ubuntu、Linux发行版使用？
ROS可以如何被移植/替代？

知乎 首页 会员 发现 等你来答

跨平台 编译 CMake

CMakeLists.txt文件都是手写的吗?

很多跨平台的开源项目使用CMake作为编译管理工具，每个源代码目录下都有一个CMakeLists.txt文件，长度动辄100-200行。这些文件都是手写的吗？感觉好麻烦，即便是有模板可用，管理和定制一个200行的文件也不轻松。

有一次看见一个文件夹下只有2个源文件，总计400行，相应的CMakeLists.txt竟有200行。。

[关注问题](#) [写回答](#) [邀请回答](#) [好问题](#) [添加评论](#) [分享](#) [收起](#)

13 个回答 默认排序

 **ruki**
xmake&tbox作者, github.com/waruqi

8 人赞同了该回答

可以试试 xmake。。比起直接写makefile 以及 cmake 简单多了。。而且跨平台。。基于Lua, 灵活方便, 很轻量, 并且提供各种插件。。

```
target("console")
  set_kind("binary")
  add_files("src/*.c")
```

编辑于 2020-07-30

[赞同 8](#) [12 条评论](#) [分享](#) [收藏](#) [喜欢](#)

13 个回答 默认排序

 **ruki**
xmake&tbox作者, github.com/waruqi

8 人赞同了该回答

可以试试 xmake。。比起直接写makefile 以及 cmake 简单多了。。而且跨平台。。基于Lua, 灵活方便, 很轻量, 并且提供各种插件。。

```
target("console")
  set_kind("binary")
  add_files("src/*.c")
```

编辑于 2020-07-30

[赞同 8](#) [12 条评论](#) [分享](#) [收藏](#) [喜欢](#)

 **Xi Yang**
每天吃三斤GV971, 在大是大非面前谈科学和真相

7 人赞同了该回答

那是因为大型项目的复杂度就已经到了那个程度。

你要是嫌写CMake麻烦, 可以试试自己弄个CMakeMaker, 然后还是麻烦, 再弄个CMakeMakerMaker。。。。。。

=====

cmake的优点是有内置的自动功能, 会比较方便。缺点是你需要了解那些自动功能到底偷偷干了什

```
1 # + testCMakeApp
2 # -- main.cpp
3 # -- CMakeLists.txt
4
5 cmake_minimum_required(VERSION 2.8)
6 project(testCMakeApp)
7
8 set(CMAKE_AUTOMOC ON)
9
10 find_package(Qt4 REQUIRED) # find Qt4
11 find_package(OpenCV REQUIRED) # find OpenCV
12
13 include_directories(${QT_INCLUDES} ${OPENCV_INCLUDE_DIRS})
14 add_executable(${PROJECT_NAME} src/main.cpp src/another.cpp)
15 target_link_libraries(${PROJECT_NAME} Qt4::QtGui Qt4::QtXml ${OpenCV_LIBS})
```

灵魂三问：

1. 为什么每个库的依赖写法不一样？你是否记得？
2. 为什么cmake要一个一个添加源文件？
3. 如何知道你找到的依赖库是否正确？

```
1 cmake_minimum_required(VERSION 2.8)
2 project(testCMakeApp)
3
4 include(PICMake)
5
6 pi add target(${PROJECT_NAME} BIN src REQUIRED Qt4 OpenCV)
```

极简风：一行解决所有！

目标名 目标类型 源文件/目录列表 依赖库

一行命令自动生成CMakeLists.txt，解放双手(HandsFree)：picmake -src src -require Qt4,OpenCV

```
41 pi_add_target(qviz_display_map SHARED GSLAM/plugins/qviz/displays/mapviz REQUIRED System)
42 pi_add_target(qviz_display_traj SHARED GSLAM/plugins/qviz/displays/trajviz REQUIRED System)
43
44 #pi_add_target(gslam_optimizer SHARED GSLAM/plugins/optimizer_ceres REQUIRED System Eigen3 Ceres GLOG)
45 #pi_add_target(gslam_estimator SHARED GSLAM/plugins/estimator REQUIRED Eigen3)
46
47 pi_add_target(gslamDB_euroc SHARED GSLAM/plugins/datasets/DatasetEuroc.cpp REQUIRED OpenCV)
48 pi_add_target(gslamDB_kitti SHARED GSLAM/plugins/datasets/DatasetKITTI.cpp REQUIRED OpenCV)
49 pi_add_target(gslamDB_npudronemap SHARED GSLAM/plugins/datasets/DatasetNPUDroneMap.cpp)
50 pi_add_target(gslamDB_npurgbd SHARED GSLAM/plugins/datasets/DatasetNPURGBD.cpp REQUIRED OpenCV)
51 pi_add_target(gslamDB_cvmono SHARED GSLAM/plugins/datasets/DatasetOpenCVMono.cpp REQUIRED OpenCV)
52 pi_add_target(gslamDB_rtm SHARED GSLAM/plugins/datasets/DatasetRTMapper.cpp REQUIRED Qt)
53 pi_add_target(gslamDB_tummono SHARED GSLAM/plugins/datasets/DatasetTUMMono.cpp REQUIRED OpenCV)
54 pi_add_target(gslamDB_tumrgbd SHARED GSLAM/plugins/datasets/DatasetTUMRGBD.cpp REQUIRED OpenCV)
55
56 pi_add_target(gslam_evaluate BIN GSLAM/evaluation/evaluate_memory REQUIRED System)
57 pi_add_target(gslam_metric_cpu SHARED GSLAM/evaluation/metric_cpu REQUIRED System)
58 pi_add_target(gslam_metric_time SHARED GSLAM/evaluation/metric_time REQUIRED System)
59 pi_add_target(gslam_metric_traj SHARED GSLAM/evaluation/metric_trajectory REQUIRED System)
60
61 pi_add_target(svar_gslam SHARED GSLAM/svar_gslam REQUIRED System Python LT0)
62 pi_report_target()
63 #####
64 # Now do make install
65 get_property(TARGETS2COMPILE GLOBAL PROPERTY TARGETS2COMPILE)
66 pi_install(HEADERS GSLAM/core HEADER_DESTINATION include/GSLAM
67           TARGETS ${TARGETS2COMPILE} CMAKE "${PROJECT_SOURCE_DIR}/cmake/FindGSLAM.cmake.in")
68
69 #####
70 # Add Tab Completion Support
71 if(UNIX)
72     install(SCRIPT ${CMAKE_CURRENT_LIST_DIR}/cmake/tab_completion.cmake)
73 endif()
```

```
--OpenGL:  
OPENGL_INCLUDES: /usr/include  
OPENGL_LIBRARIES: /usr/lib/x86_64-linux-gnu/libGL.so;/usr/lib/x86_64-linux-gnu/libGLU.so  
OPENGL_DEFINITIONS: -DHAS_OPENGL
```

```
--GLEW: VERSION 1.13.0  
GLEW_INCLUDES: /usr/include  
GLEW_LIBRARIES: /usr/lib/x86_64-linux-gnu/libGLEW.so  
GLEW_DEFINITIONS: -DHAS_GLEW
```

依赖库信息

```
--Python:  
PYTHON_INCLUDES: /usr/include/python3.5m  
PYTHON_LIBRARIES: /usr/lib/x86_64-linux-gnu/libpython3.5m.so  
PYTHON_DEFINITIONS: -DHAS_PYTHON
```

目标统计

```
-- The following targets will to be build.  
-- LIBS(): libgslam_qviz.so; libgslam_tests.so; libgslam_resource.so; libgslam_doc.so; libgslam_play.so; libgslam_gmap.so; libgslam_topic.so; libqviz_panel  
win3d.so; libqviz_panel_displays.so; libqviz_panel_frameinfo.so; libqviz_panel_pubsubs.so; libqviz_panel_logger.so; libqviz_display_map.so; libqviz_display_t  
raj.so; libgslamDB_euroc.so; libgslamDB_kitti.so; libgslamDB_npudronemap.so; libgslamDB_npurgbd.so; libgslamDB_cvmono.so; libgslamDB_rtm.so; libgslamDB_tummo  
no.so; libgslamDB_tumrgbd.so; libgslam_metric_cpu.so; libgslam_metric_time.so; libgslam_metric_traj.so; libsvar_gslam.so
```

```
APPS(): gslam; gslam_evaluate  
C++ flags (Release): -O3 -DNDEBUG
```

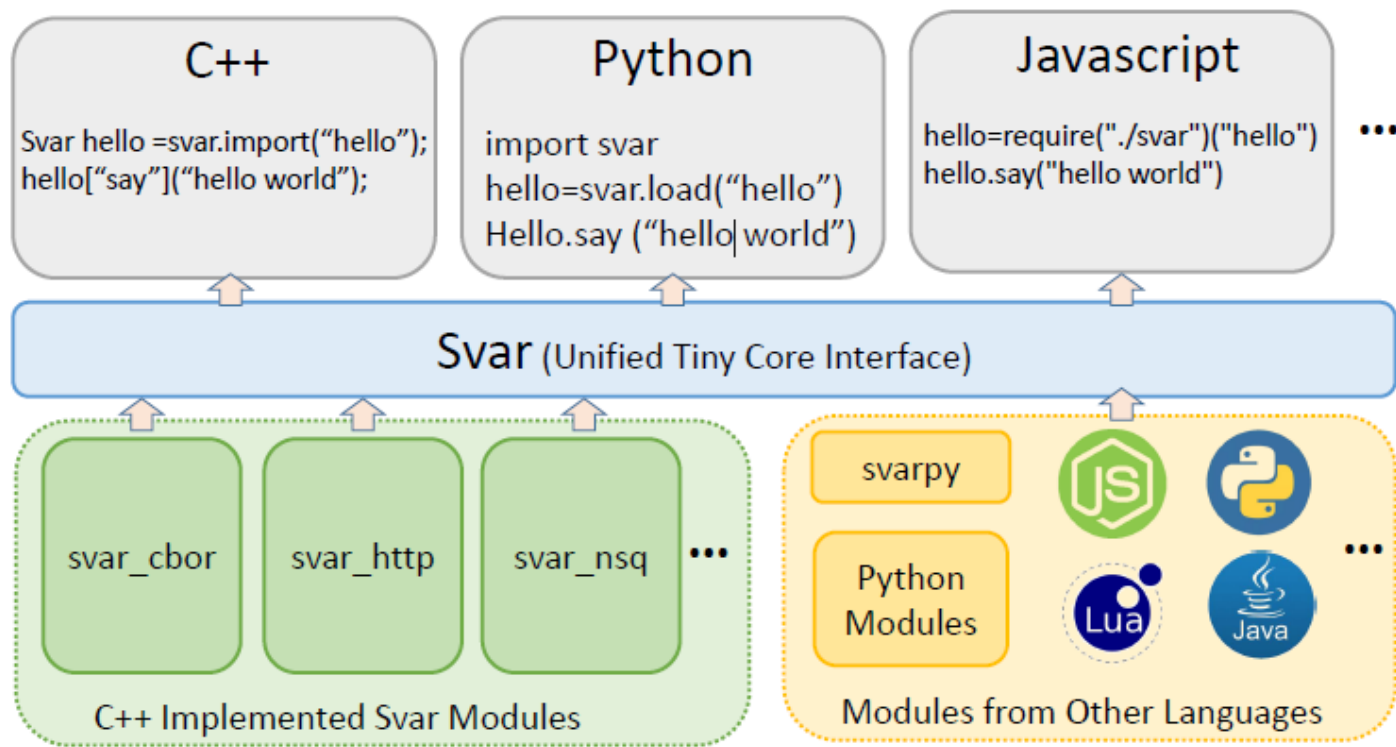
编译选项总结

```
-- configuring done  
-- Generating done  
-- Build files have been written to: /data/zhaoyong/Program/Apps/GSLAM/build  
build>
```

开源地址：<https://github.com/zdzaoyong/PICMake>

Svar开源项目

C++封装神器，多语言交互工具



设计目标

- 提供一个C++中的动态类型数据结构，支持变量，函数，类和JSON等的操作；
- 实现C++中的import机制，实现动态库加载，解决C++的编译及依赖问题；
- 基于以上数据结构实现库的导出，且同时支持多种语言调用；
- 基于以上数据结构实现其他语言库的导入和调用；
- 基于Svar与多种语言之间的接口，实现多语言之间的库调用；

轻量：单头文件无依赖

好用：使用简洁，易上手

强大：高效微内核，扩展能力强

TAObotics

Svar开源项目

C++封装神器，多语言交互工具

TAObotics

1. Bind C++ to hello.so	2. Import by C++	3. Import by Python
<pre>#include <Svar.h> void say(std::string v){ std::cerr<<v<<std::endl; } REGISTER_SVAR_MODULE(hello) { svar["say"] = say; } EXPORT_SVAR_INSTANCE</pre>	<pre>#include <Svar.h> auto m=svar.import("hello"); int main(){ m["say"]("hello world"); return 0; }</pre>	<pre>import svar hello = svar.load("hello") hello.say("hello world")</pre>
4. Import by Javascript	5. C++ import Python	6. Javascript import Python
<pre>svar = require('./svar') hello = svar('hello') hello.say("hello world")</pre>	<pre>#include <Svar.h> auto py=svar.import("svarpy"); int main(){ auto os = py["import]("os"); std::cout<<"Pid is:" <<os["getpid"](); return 0; }</pre>	<pre>svar=require("./svar") python=svar("svarpy") os=python.import("os") print("Pid is", os.getpid())</pre>

让你的库被
各种语言
import!

以下情况考虑使用ROSBridge：

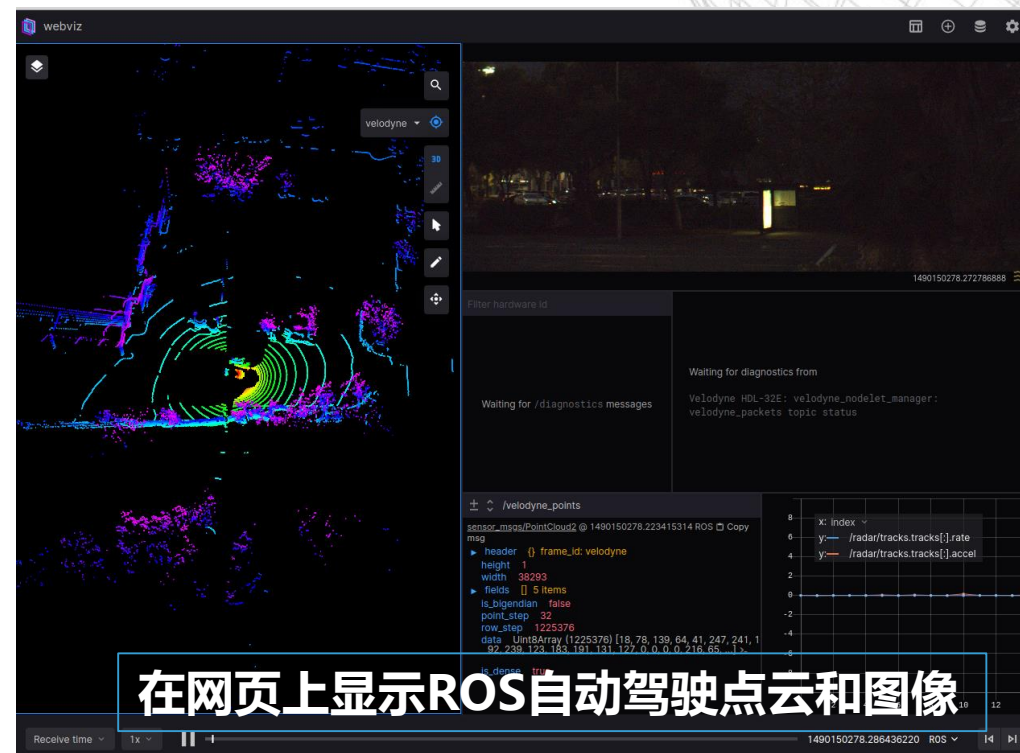
- 在机器人的安卓|Web前端中与ROS系统进行通信，从而实现状态显示&控制；
- 通过广域网消息队列实现云端机器人管理；
- ROS太重了，需要与ROS通信，但不想引入ROS依赖

ROSBridge特点：

- 可通过TCP、UDP、WebSocket实现ROS的通信转发；
- 使用JSON（包含二进制CBOR）实现消息的表达；

广域网消息中间件：

- 物联网（轻量消息）：MQTT；
- 大吞吐：Kafka，NSQ，ZeroMQ；
- 阿里云：RocketMQ



svar_rosbridge

不依赖ROS的ROS通信中间件

TAObotics

svar_rosbridge特点：

- 依托Svar实现多语言支持，实现C++中消息的表达；
- 可基于CBOR支持高效图像等二进制数据转发；

```
1 import svar
2 import time
3
4 ros=svar.load('svar_rosbridge')
5
6 def cbk(msg):
7     print(msg)
8
9 pub = ros.ROSPublisher('/test_int',{'type':"std_msgs/Int32"})
10 sub = ros.ROSSubscriber('/test_int',cbk,{'type':"std_msgs/Int32"})
11
12 while True:
13     time.sleep(1.0)
14     pub.publish({"data":20})
15
```

Python发布订阅示例

```
1 #include <Svar.h>
2
3 auto ros=svar.import('svar_rosbridge')
4
5 void cbk(sv::Svar msg)
6 {
7     std::cout<<msg<<std::endl;
8 }
9
10 int main(int argc,char** argv){
11     auto pub = ros["ROSPublisher"]("/test_int",{{"type","std_msgs/Int32"}})
12     auto sub = ros["ROSSubscriber"]("/test_int",cbk,{{"type","std_msgs/Int32"}})
13
14     while(True){
15         usleep(1000000);
16         pub.call("publish",{{"data":20}})
17     }
18     return 0;
19 }
20
```

C++发布订阅示例

开源地址：https://github.com/zdzaoyong/svar_rosbridge

Messenger

更易用，更轻量的通信中间件

TAObotics

```
1 #include <GSLAM/core/Messenger.h>
2
3 void cbk(int msg){
4     std::cout<<"received:"<<msg<<std::endl;
5 }
6
7 int main(){
8     auto pub_int = messenger.advertice("/int",0);
9     auto sub_int = messenger.subscribe("/int",0,cbk);
10
11     messenger.publish("/int",10);// direct publish
12     pub_int.publish(100);
13 |
14     return 0;
15 }
```

	Messenger	ROS
消息类型	进程内：任意类型 进程间：JSON+Buffer	固定消息类型
序列化	JSON/CBOR	ROS/Protobuf
平台	任何支持C++11的设备	Ubuntu、Linux
延迟	支持阻塞调用/ns级	ms级
多语言	支持	支持
进程间通信	兼容ROS NSQ MQTT等	内置网络协议

无需提前定义消息类型，进程内消息甚至可以是函数和类；

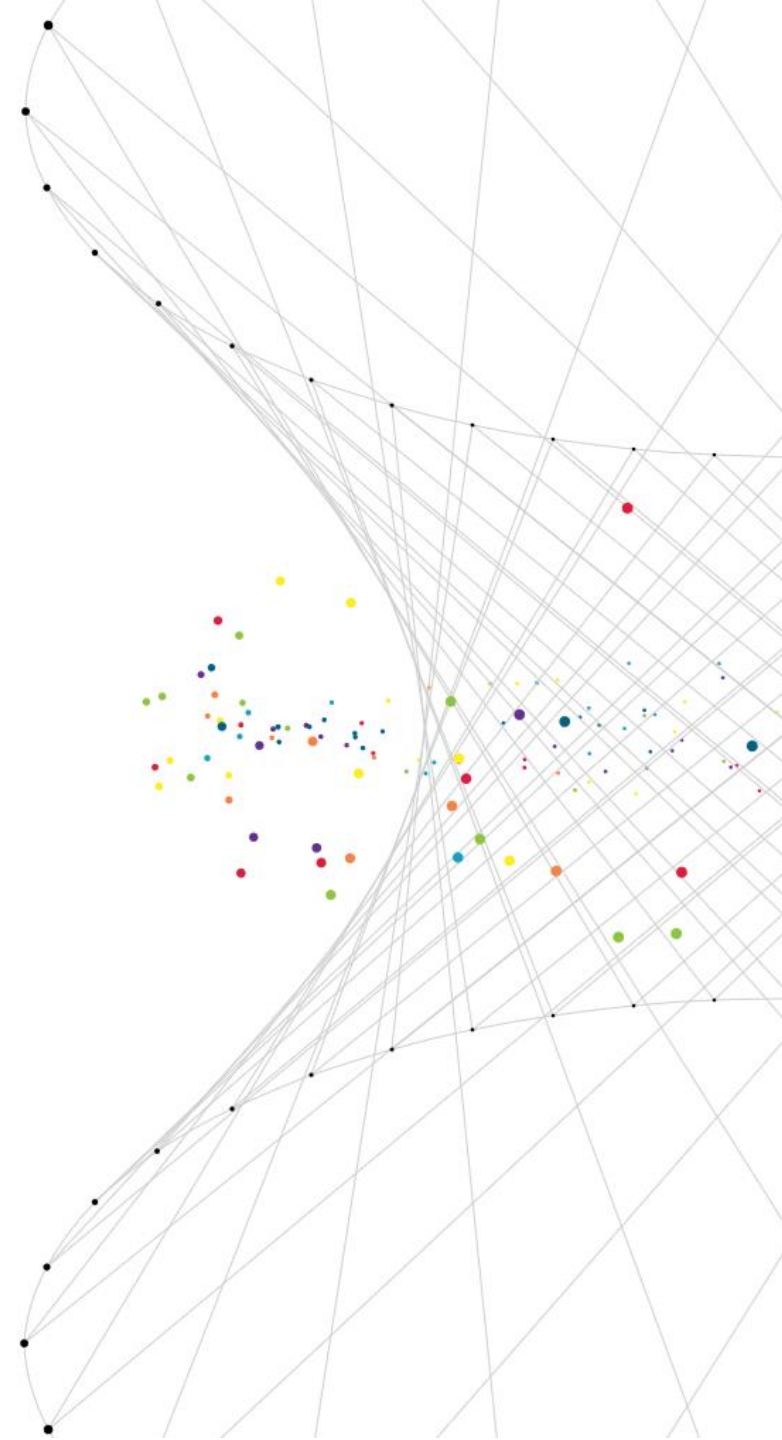
支持阻塞调用，零延迟；

单头文件，把微内核坚持到底；

开源地址：https://github.com/zdzaoyong/svar_messenger

2.3 感知&规划&决策

Perception & Planning & Decision



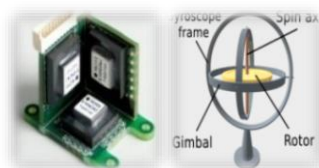
司南定位导航系统

Sinan Positioning and Navigation System

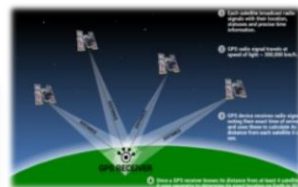
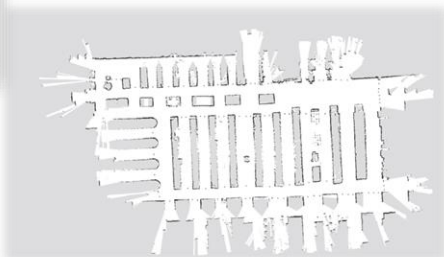
TAObotics

司南定位导航系统是我司自主研发的室内大场景多传感器融合定位导航系统，它整合了传统的定位导航方案和前沿的视觉SLAM技术，为移动机器人行业提供低成本、高精度，可靠稳定的定位导航模块。

千元级硬件成本实现5万平方米以上定位导航功能



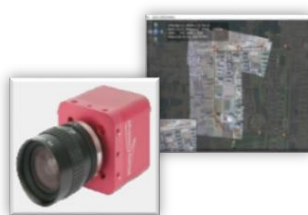
惯导系统



卫星定位



激光雷达



视觉定位

自主返回充电

三维路径规划、避障系统

300连续工作小时稳定性

低于1万元的成本优势，降低60%以上成本

5万平米（室内）、100万平米（室外）的大场景建图

实现5cm高精度定位能力

20ms行业优秀避障反应速度

技术亮点

司南定位导航系统

Sinan Positioning and Navigation System

TAObotics

基础算法：SLAM系统、路径规划、自动充电等。

功能组件：视觉检测、语音识别、梯控系统。

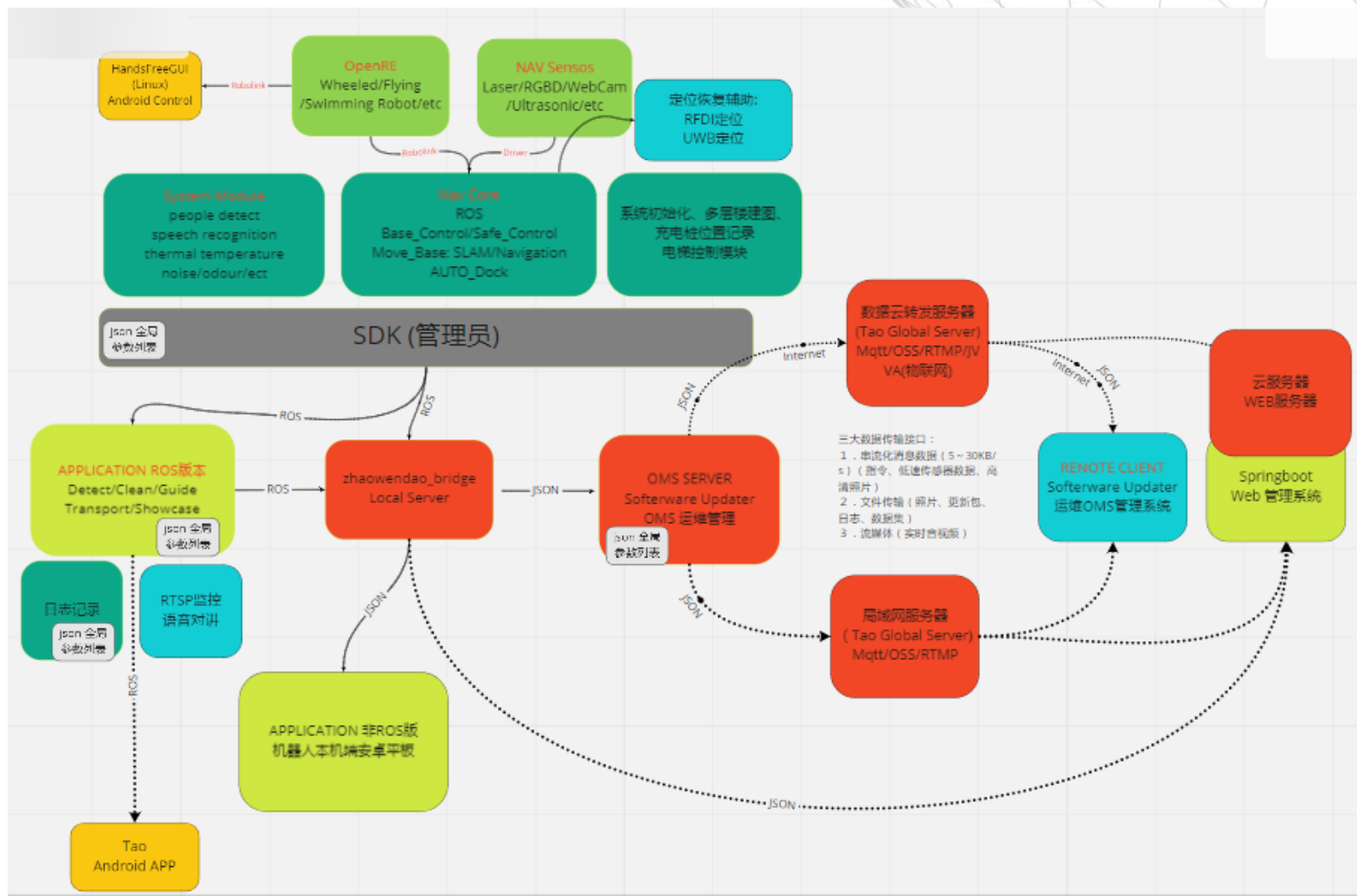
地图管理：多地图管理、特殊点和特殊区域的位置信息，比如充电桩、UWB/RFID、任务点的位置，或者运行区域、禁行区域等。

设备管理：机器人本机的外设和传感器、UWB基站、梯控模块、呼叫器等设备检测和管理。

云端管理：局域网和广域网的可视化监测管理，消息服务、文件服务、流服务。数据库搭建和用户管理。

交互框架：本机安卓交互、移动手机端、局域网WEB监控。

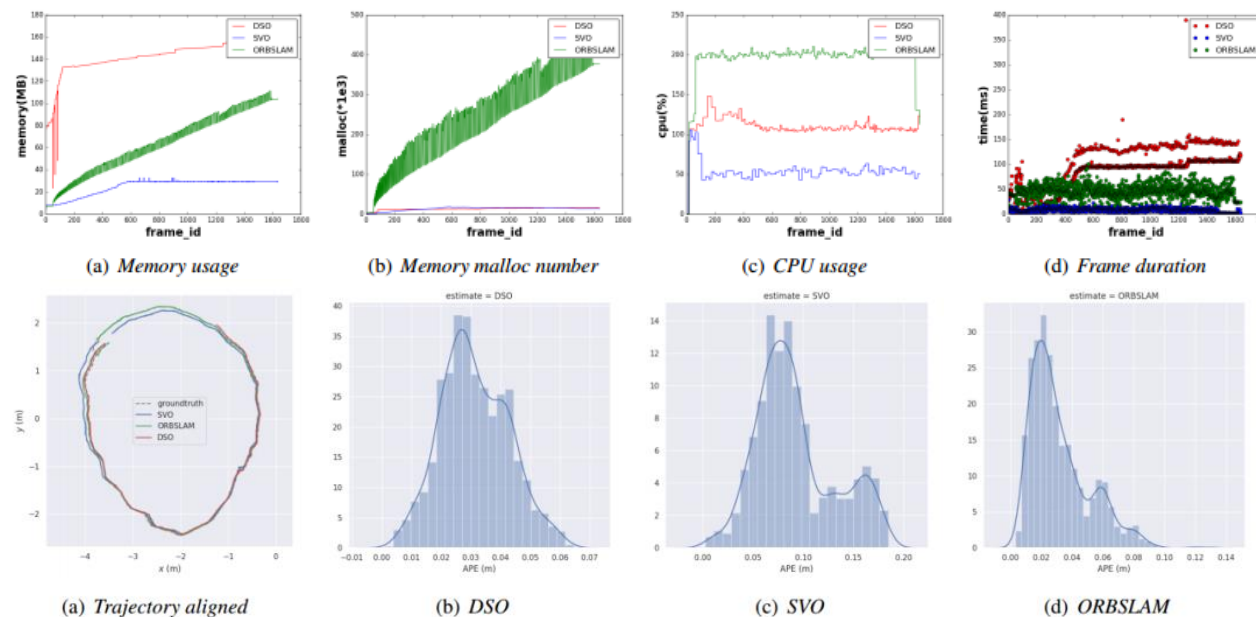
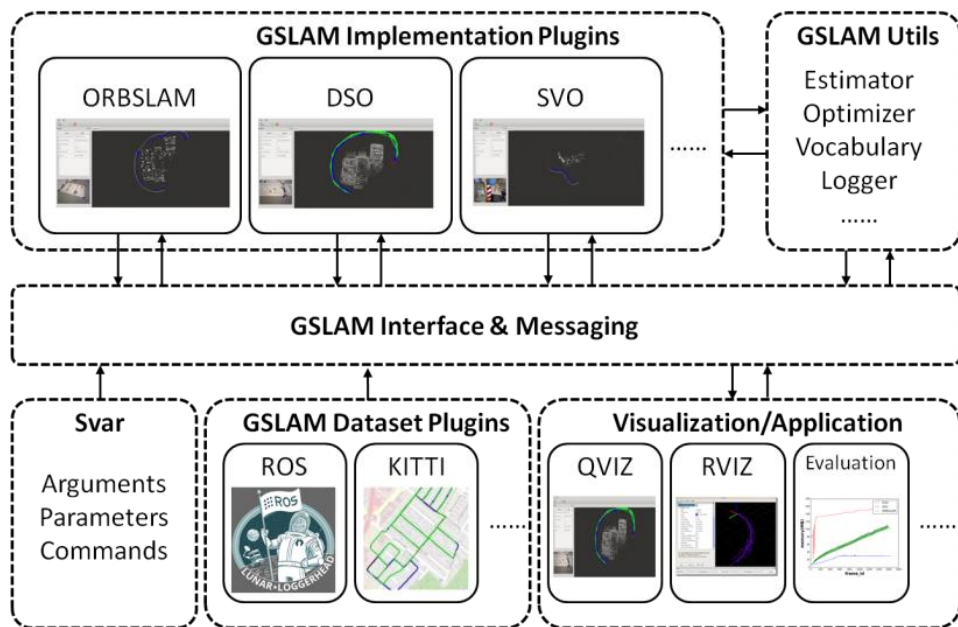
任务管理：应用任务的规划、日志记录等。



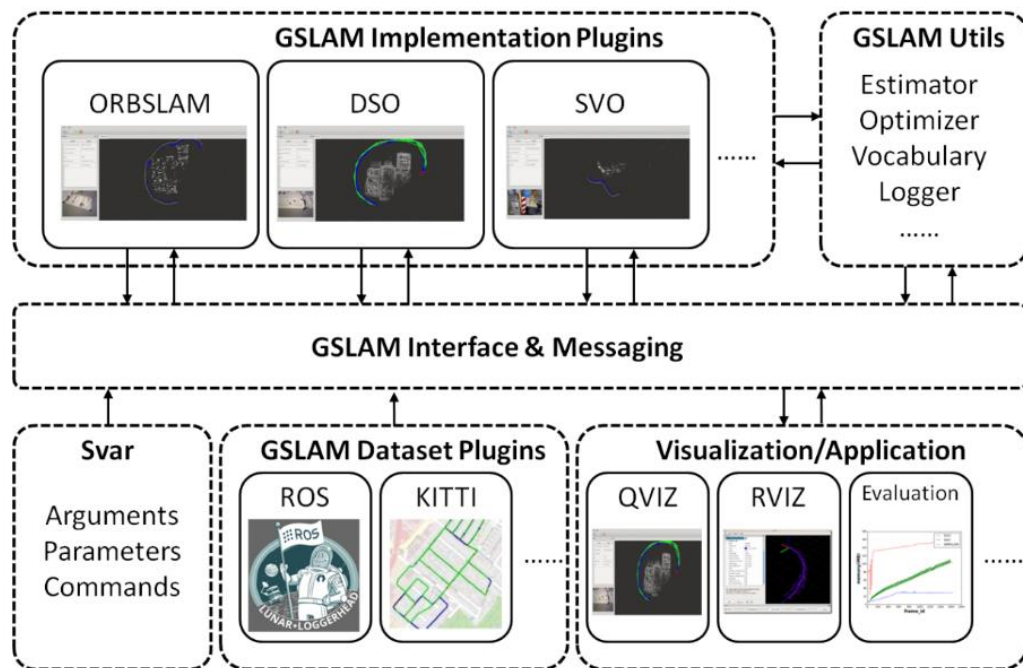
GSLAM: A General SLAM Framework and Benchmark

Yong Zhao, Shibiao Xu, Shuhui Bu, Pengcheng Han, Hongkai Jiang

ICCV 2019



- Support C++, Python, Node-JS languages
- Source Code(750+ Star): <https://github.com/zdzaoyong/GSLAM/>
- SLAM领域热门开源项目, 源代码单日访问量峰值2000+
- 知乎整理的《[100 项开源视觉 SLAM 方案够你用了吗?](#)》, 仅个人上榜两个开源项目



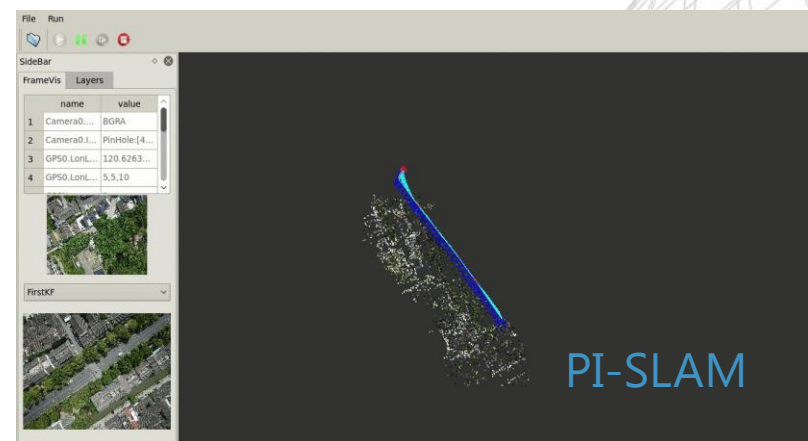
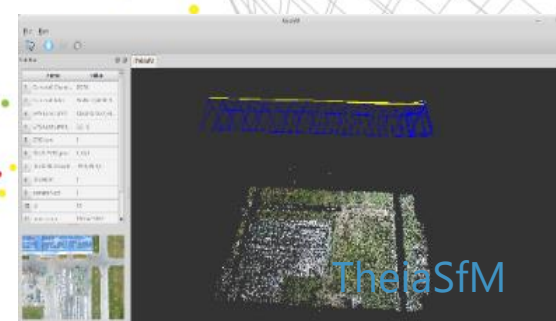
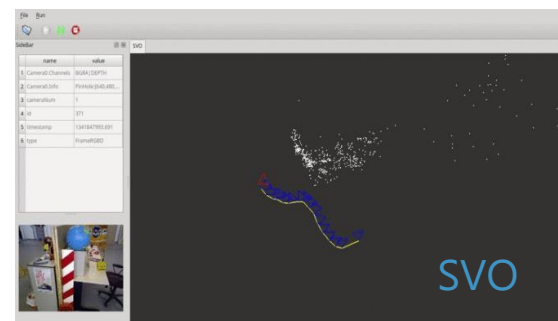
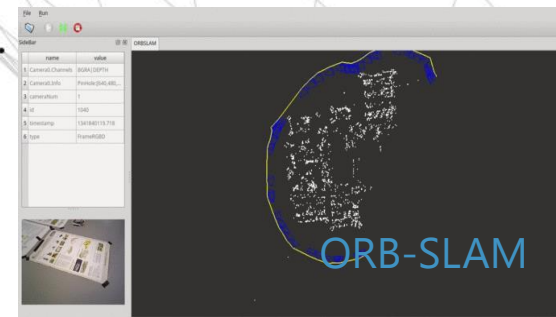
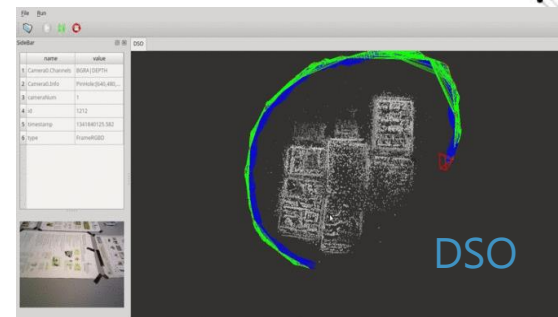
通用的SLAM（同时定位于地图构建）开发框架和平台

插件式架构设计

高性能组件和函数库

支持C++ 11

提供Python/Javascript 开发接口



3. 项目生态和应用

Project Ecology And Application



项目生态

Project Ecology

TAObotics



开源软硬件
平台



开源算法和
开发工具



学习教程和
交流社区



实验/行业
级案例



开源社区

Open source community

我们与超过150所大学与科研机构展开了合作,创立了自己的社区,吸引超过2,000位开源社区的成员。我们拥有丰富的教程、开发工具以及众多机器人平台和开源硬件,同时也提供各大开源社区的支持。真正做到为开发者提供软硬件、学习教程、开发调试工具、应用案例及社区交流。



教学解决方案 TEACHING SOLUTIONS

结合应用实践,设计实验课程和教学教案,为高校提供从设备、课程和师资培训的一整套解决方案。

Arrange lab courses and tutorials combined with practical applications and provide a full set of solutions from equipment, courses and training for colleges and universities.



线上线下培训 ONLINE AND OFFLINE TRAINING

构建机器人线上线下教育生态系统,线上的开源社区,线下的培训机构为求职就业,技能增值提供定向化的短期培训。

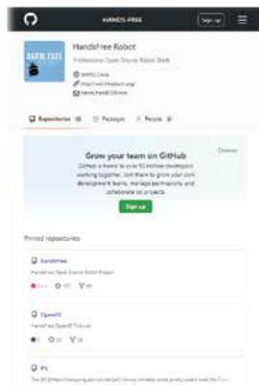
Build a robot educational ecosystem online and offline with open source community, short-term training of skills and job seeking.



实验室定制 LABORATORY CUSTOMIZATION

提供机器人知识体系教学和课外机器人竞赛的解决方案,包括了实验室的空间设计、教学实验设备的组合搭配、实验指导书和课件等配套资源,以及相关的培训和交流研讨活动。

Provide comprehensive solutions of robot related knowledge and competitions, including customizing robot labs, lab facilities, educational materials and PPTs, training and discussions.



在线社区
Online communities



在线教程
Online courses

2000+
Developer user

150+
Research institutions

100+
Application cases

主要开源社区的支持

Support of major open source communities



我们的开源社区
Our open source communities



一步一步学编程



90分钟学编程

基础知识与编程课程



Learn SLAM Step by Step
一步一步学SLAM

一步一步学SLAM



机器学习
Machine Learning
布列辉



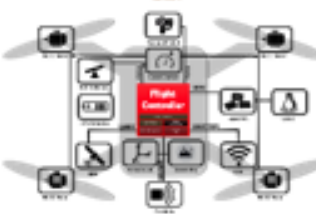

机器学习



一步一步学ROS

一步一步学ROS

视觉SLAM、机器学习、ROS课程



研究课题与项目



课程的网站

由深入浅，一步一指导，降低学习过程的痛苦指数，重视基础知识与技能的培养

全向系列®

四驱系列®

履带系列®

差速系列®

服务系列®

Octopus



Thunder



Soldier



Turtle



Stone



Octopus Pro



Thunder Pro



Soldier Pro



Turtle Pro



Stone Pro



实验级开源案例展示

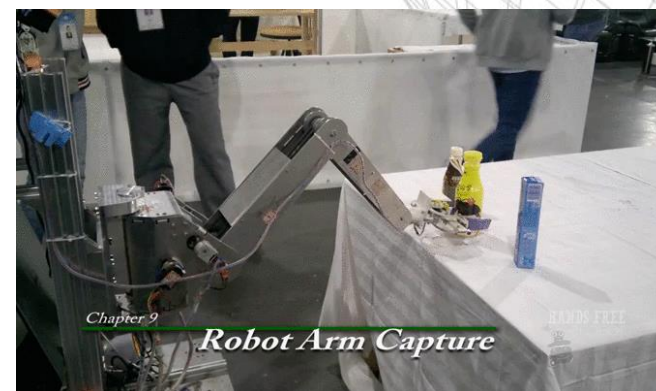
- HandsFree开源了大量的应用案例，通过开放源码来促进社区交流，同时引导开发者进行科学研究和产品应用开发。包含自主定位导航，多机器人协同，机械臂运动/规划，视觉抓取，智慧农业机采摘，安防巡检，视觉跟踪等。所有应用案例基于统一的HandsFree软件系统框架。



➤ 安防巡检



➤ 定位导航系统



➤ 机械臂运动规划



➤ 双臂机器人



➤ 无人视觉跟踪



➤ 智慧农业机器人

智慧零售

MallBot系列®

1st Gen

2nd Gen



工作范围：200-4000m²

最大续航：5-7hr

自主回充：支持

视觉抓取：深眸系统

行为分析：识别率98%

云端管理：HandsFree

Cloud 管理

智慧安防巡检

Guardian系列®

工厂巡检员



工作范围：3500-10000m²

最大续航：8-10hr

自主回充：支持

视觉抓取：深眸系统

安全分析：支持

云端管理：HandsFree

Cloud 管理

抗疫卫士

Anti-Covid系列®

喷雾消杀机器人



工作范围：200-5000m²

最大续航：5-7hr

自主回充：支持

消毒喷雾：5L (4-8hr)

杀菌范围：2400-4800m³

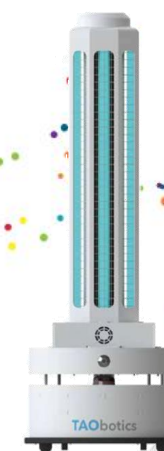
云端管理：HandsFree

Cloud 管理

抗疫卫士

Anti-Covid系列®

紫外线灭菌机器人Pro



工作范围：200-5000m²

最大续航：5-7hr

自主回充：支持

紫外杀菌：支持

杀菌范围：3600m³+

云端管理：HandsFree

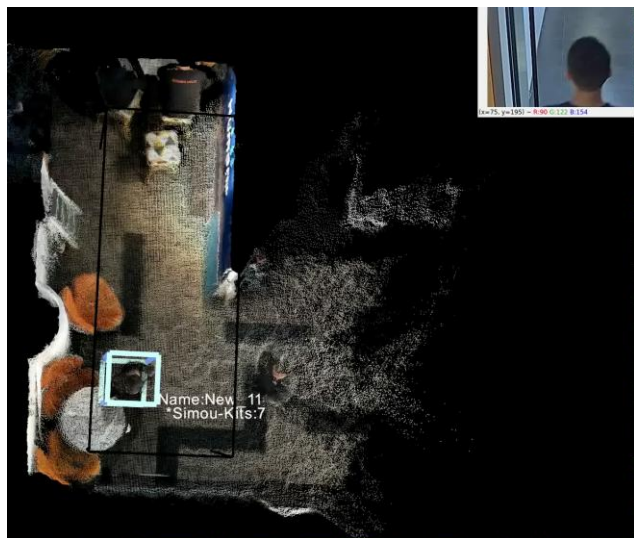
Cloud 管理

智慧零售MallBot系列-深度视觉感知

Smart Retail MallBot Series-Computer Vision Structure

机器人搭载深度感知与 AI 计算融为一体的智能视觉传感器，支持人体特征识别、人体跟踪、人体行为识别、商品识别等 AI 算法，边缘加速计算，无需额外服务器。

监测能力	>1000 人	每分钟监测人流量
测量距离	<6m	支持安装高度
识别精度	>98%	客流量计数准确度
计算速度	<3s	系统计算反应时间
人群覆盖	>99.9%	性别、年龄识别



联系我们

Contact US



微信公众号



QQ群二维码

朝闻道

TAObotics

公司主页：

<https://www.taobotics.com>

开源项目主页：

<http://wiki.hfreetech.org>

项目代码：

<http://www.github.com/hands-free>



The logo for TAObotics, featuring the word "TAO" in a bold blue font and "botics" in a grey font. The logo is positioned in the top right corner of the slide. The background of the slide features a complex network of grey lines and dots, with a cluster of colorful dots (red, yellow, green, blue, purple) in the center-right area. There are also solid blue rectangular shapes in the top-left and bottom-right corners.

TAObotics

Thank you!