Nvidia Jetson TX2 开发板测试报告

Zhang Yifei(yidadaa@qq.com)

UESTC — July 8, 2019

1 环境设置

Nvidia Jetson TX2 系列开发板提供了大约 1.5T Flops 的运算能力，可以运行大多数轻量级神经网络。本次测试使用开发板配套的 Jetpack 套件，测试常见的 SLAM 系统以及视觉算法在开发板上的表现。

1.1 解决 opencv 无法使用lopencv_nonfree 的问题

Jetpack 自带的 Opencv 不支持 nonfree 库，因此就不能使用 SIFT/SURF 这种专利算法。要使用 nonfree 库有两种解决解决方法：

Note about SIFT/SURF in the nonfree module: OpenCV4Tegra doesn't include the opencv_nonfree package (containing SIFT & SURF feature detectors) since those algorithms are patented by other companies and therefore anyone using opencv_nonfree is at risk of liability.

If you need something from the nonfree module, you have 2 options:

- Analyze the public OpenCV source code then copy/paste the parts of the nonfree module that you want (eg: SURF feature detector) from OpenCV into your own project. You will have the CPU optimizations of OpenCV4Tegra for most of your code and will have the GPU module and will have the non-optimized patented code that you need from the nonfree package such as SURF. So this option gives full performance (for everything except the nonfree code) but is tedious.
- Ignore OpenCV4Tegra, and instead, download & build public OpenCV (by following the instructions below for natively compiling the OpenCV library from source). You will still have the GPU module but not any CPU optimizations, but you won't need to spend time ripping out parts of the OpenCV non-free module code. So this option is easiest but produces slower code if you are running most of your code on CPU.

并且 libopencv4tegra 是 2.4 版的 opencv，要使用 3.0+ 版本的还是得自己编译 opencv 才行。(缺点是自己编译的 opencv 没有 CPU 优化)。

1.2 开启被屏蔽的 2 块 CPU 并设置为最大频率

使用位于 home 目录的 tegrastats 命令可以查看 TX-2 的使用情况：

```
$ ~/tegrastats
RAM 1282/7854MB (lfb 1x256kB)
cpu [21%@2035,off,off,15%@2035,17%@2035,15%@2035]
```

开启被屏蔽的两颗 CPU：

```
$ sudo su
$ echo 1 > /sys/devices/system/cpu/cpu1/online
$ echo 1 > /sys/devices/system/cpu/cpu2/online
```

使用以下命令开启最大频率：

```
~/jetson_clocks.sh
```

再次查看 CPU 情况：

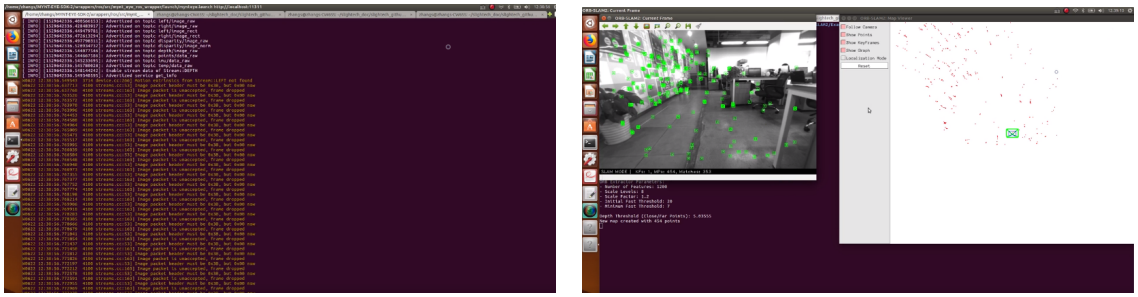
```
$ ~/tegrastats
RAM 979/7854MB (lfb 1545x4MB)
cpu [0%@2035,0%@2419,0%@2419,0%@2035,0%@2035,0%@2035]
```

同时由于频率提升，可以观察到开发板风扇开始转动。

2 测试 SLAM 系统

2.1 ORB-SLAM

ORB-SLAM 是西班牙 Zaragoza 大学的 Raúl Mur-Arta 编写的视觉 SLAM 系统。它是一个完整的 SLAM 系统，包括视觉里程计、跟踪、回环检测，是一种完全基于稀疏特征点的单目 SLAM 系统，其核心是使用 ORB (Orinted FAST and BRIEF) 作为整个视觉 SLAM 中的核心特征。



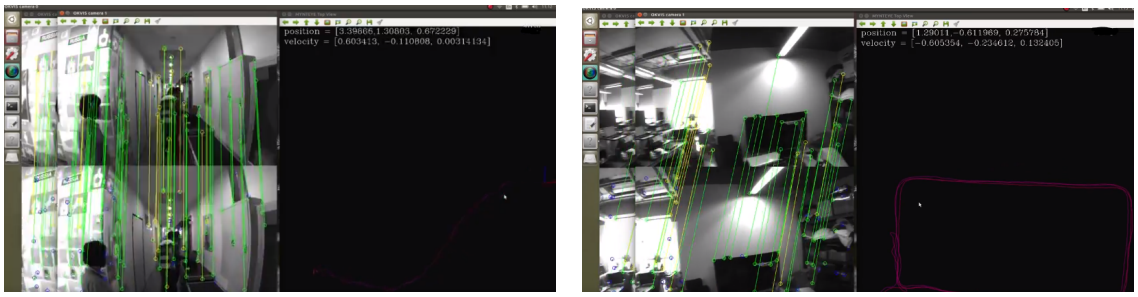
(a) (b)

Figure 1: ORB-SLAM 测试截图

在测试过程中，使用 720P 分辨率的图像作为输入，处理速度可以稳定在 30FPS，基本可以达到可用级别。

2.2 OKVIS

OKVIS 是由 Stefan Leutenegge 等人提出的基于双目 + 惯导的视觉里程计，属于 VIO (Visual Inertial Odometry) 。OKVIS 系统由于需要将双目图像与 IMU 数据同步，需要大量运算，在 720P 视频下，可以



(a) (b)

Figure 2: ORB-SLAM 测试截图

达到 15FPS 的速度，在不影响使用的情况下，将分辨率降至 480P 可以使帧率提升到 27FPS 左右，基本达到可用水平。

3 总结

本次测试使用两个机器人导航中常用的 SLAM 算法，Jetson TX2 基本可以胜任此类算法的运行任务，可以在此平台上进一步开发用于机器人导航或环境建图的边缘端计算平台。