技术学习详细计划

注：因为技术知识很多，分清轻重缓急，切勿心大，需循序渐进，主次分明。

* 时间安排

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 时间 | 6:30 | 7:00-12:30 | 12:30-  14:00 | 14:30-18:30 | 18:30-19:00 | 19:00-20:00 | 20:00-21:00 | 21：00-23:00 |
| 事件 | 起床 | 先看1小时英语  《私房菜》  树莓派  （2遍，后期根据实际需要再看第2遍，可间隔久一点 ） | 吃饭、午休 | 《数字图像处理》（3遍）  《SLAM十四讲》（3遍）  《opencv3》（2遍）  《统计学习方法》（2遍） | 吃饭 | C++ primer（2遍） | 锻炼，洗澡 | C++ primer、1小时《高效能人士的七个习惯》  （1遍） |

* 招聘韩哥

重点：先把理论搞透彻，代码能力练起来，包括：物理原理，数学应用和原理，解算过程，优化过程，代价。系统：orbslam2（基础，完整，看的人多），okvis。

* 新计划方案（没考上最多到7月10日，新生入职之前）：

1. 出成绩前：先把私房菜、slam十四讲、数字图像处理、opencv编程入门、C++ primer（闲时敲敲代码）、数字图像处理、统计学习方法看完，大致理解；英语练习听力和口语，直到出成绩（1小时）；仿照光哥程序玩stm32（每天少于1小时），完成小车避障和姿态计算，重点练代码能力，由C过渡到C++，尽量不百度，自己想
2. （1-2个月，考虑加入新书）看第二遍书，回顾1中几本书和之前的机器学习，解决未解决问题，做十四讲习题、看相关英文文献，搞透彻相关理论，搞清楚书上代码，代码结构和思想等。
3. 看第三遍书，学习ROS、Git上找经典、主流的orb\_slam2程序，在ros上调试，用树莓派平台，仿照orb\_slam2代码自己敲
4. 习惯

学习过程中需注意养成好习惯提高效率。习惯总结如下：

1. 注意使用快捷键，尽量少用鼠标；
2. 注意敲键盘姿势，提高盲打能力和速度；
3. 勤于反思和总结，在电脑和简书做笔记，每隔一段时间将简书上内容合并到电脑和上传到云盘，并在简书上做好标记“//日期//以合并内容//日期//”，之后建立博客做笔记；
4. 注意细节，优化做事步骤；
5. 注意借助工具；
6. 编程或做事前先思考一下，理清思路；
7. 切记重复确认，对于稍重要的事一般确认1次，最多确认2次；
8. 不要固化思想，多参考，查看不同方法和思想，持续尝试新方法及优化；
9. 用谷歌搜索查找问题解决方法，尝试看英文解释，手机翻墙软件：蓝灯专业版，电脑翻墙软件：xx-net（GitHub）；
10. 使用GitHub帐号管理代码，多看GitHub上优秀开源程序；
11. 有空看看沟通、心理等非技术方面的书。
12. 编程时养成使用git的习惯
13. 做每个实验都在git上写上总结和问题
14. 具体计划（旧的，先完成一中新计划）

推进进度：吴恩达机器学习课程还剩连续强化学习部分

1. 机器学习：（-12.23）
2. 主要学习机器学习，看1遍《机器学习》，然后浏览一遍吴恩达视频；
3. 以吴恩达视频重点学习视频上面所讲算法为主线，看廖雪峰网站教程和《python基础编程》，根据《python机器学习实战》熟悉python和机器学习主要算法，先在window上使用pycharm编辑器，看《机器学习》、视频及讲义，做笔记，争取看懂七八成，有需要会用到MATLAB；
4. 结合实际运用（网上找项目、牛客网等更贴近工作的实战）回顾巩固，可放在后面做

注：

* 1. 深度神经网络：CNN、DBN、sparsecoding、RNN、DNN、LSTM、GRU、softmax的集成学习等；
  2. 深度学习开源框架：TensorFlow、pytorch、net、torch、theano等；
  3. 语言：C、C++（slam十四讲要求）、MATLAB、python、java（网站、app、Android）、JavaScript（前端，如网站页面）等；
  4. 系统：Linux（Ubuntu、CentOS、Fedora）、ROS、Windows、freertos/ros等；
  5. Python编辑器（推荐 10 款最好的 Python IDE - CSDN博客 https://blog.csdn.net/liang19890820/article/details/51069097）：vim（slam十四讲要求花1小时）、Eclipse with PyDev、Sublime Text、Emacs、Komodo Edit、PyCharm、Wing、PyScripter、The Eric Python IDE、Interactive Editor for Python等；
  6. Python机器学习编辑器（五款实用免费的Python机器学习集成开发环境（5 free Python IDE for Machine Learning）（图文详解） - 大数据和AI躺过的坑 - 博客园 https://www.cnblogs.com/zlslch/p/7977039.html）：JuPyter/IPython Notebook、PyCharm、Spyder、Rodeo、Geany等；
  7. 机器人算法： SLAM算法（谷歌开源cartographer），stereo算法，frertos/ros机器人操作系统，AGV机器人调度算法等
  8. 数据库（有需要时参考《SQL必知必会》）：MySQL、sqlite（python嵌入式数据库） 、PostgreSQL 、Oracle等。
  9. 互联网比较重要知识点：系统涉及的知识点，像进程，线程，文件这些；感觉做这个互联网最重要的还是tcp/ip和操作系统原理；
  10. 机械方面书籍：《机械设计基础》、《机械制造基础》、《工程力学》、《轻功？量》
  11. 网站框架:flask，django，tomado，mvc模型，
  12. 人工智能竞赛平台：kaggle待发掘

1. 图像处理：VS、C++、OpenCV、主要参考《OpenCV3入门》、《学习OpenCV》、《数字图像处理》等；(三维图构建)
2. 以《编程入门》为编程学习主，同时将《数字图像处理》先看一遍；
3. 动手实践，开始准备设计和搭建小车，不断巩固；
4. 按《Visual studio和opencv总结》第1条“基础概念、基本思想、算法研究”步骤不断巩固。

3、Linux（Ubuntu）：以《鸟叔私房菜》为主，C、C++等书籍为辅熟悉Ubuntu操作，熟悉python及其编辑器pycharm在Ubuntu上使用，可尝试搭博客（先装虚拟机，后面考虑双系统）；

4、SLAM算法：首先，通过《机器人学、机器人视觉与控制》加深对机器人整体了解；然后，通过《视觉SLAM十四讲》在Ubuntu上使用C++熟悉SLAM算法、ros系统，期间会简单使用vim；

5、学习TensorFlow深度学习框架及MATLAB算法：通过《TensorFlow实战Google深度学习框架》学习TensorFlow框架；以《机器人学、机器人视觉与控制》为主，《MATLAB一本通》为辅，练习机器学习在MATLAB中的使用。根据实际需要确定先后顺序。

三、实战构想

具体见文件小车构想

* 树莓派及相机：前端视觉里程计、后端优化、回环检测、建图、路径规划（导航）、避障