**NACIT 2016视觉系统介绍**

# 前言：

1. 源码中很多类函数都有int num = x之类的输入参数，这代表着随着x值的增大，该函数的更新；x一般取值4,5,6,7，代表更新的月份；
2. 本文 = 90% 正经解说 + 10% 装逼扯犊子吐槽。各位师弟不喜请绕路～

# 总体思路：

由于我们从一开始就把视觉系统视为一个传感器使用，所以笔者用模块化的思路构建了整个视觉系统，即把这个模块卸下来之后，就算直接插在其他提供视觉系统接口的平台上也可以直接使用。这样的话，负责图像的同学只需要负责更新维护自己的视觉系统，并将更新后的视觉系统源码拷贝给策略控制负责人，然后让控制负责人编译即可完成对接，这样分工更加明确，效率也会更高（虽然最后图像和控制都落笔者头上……）

# 视觉系统大概介绍：

主要分为两部分，一是自定义函数部分，二是视觉类部分；另外还有依赖的第三方库，存储数据的XML文档集，以及其他的一些辅助小程序。

1. 自定义函数CustomizeFunctions：

该部分主要收容了笔者在构建各项工程时，为了编程方便而自定义的函数。主要下分为四种分类：

1. 自定义结构体CustomizeStructs；
2. 通用图像处理函数GeneralImageProcess；
3. 特殊图像处理函数SpecialImageProcess；
4. 其他支持函数SupportFunctions；
5. 视觉类VisionClass：

该部分直接构建了视觉系统。结构如图所示：



视觉系统模块为NCVision，主要构成如下：

1. 摄像机类NCCamera；
2. 摄像机模型类NCModel，并向下派生俯视/前视摄像机模型NCDownModel、NCFwdModel；
3. 图像处理类NCImgPro；
4. 其他类函数；
5. 视觉分析信息VisionInfo及其他宏定义数据；
6. 第三方库ThirdParty：

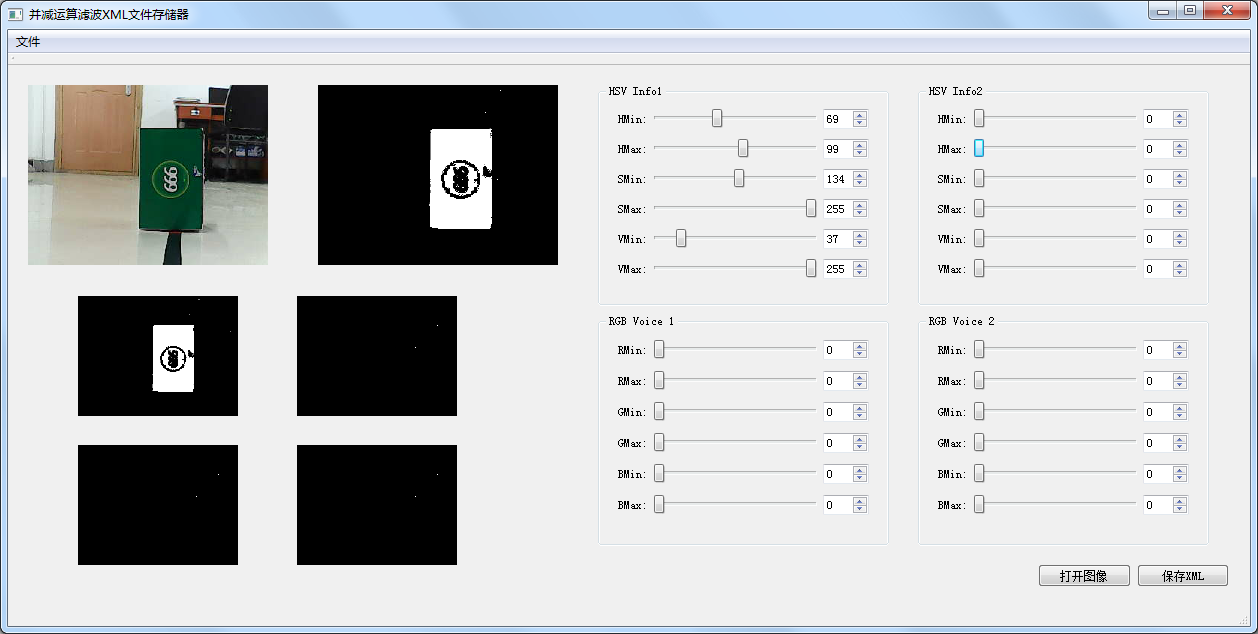
主要包含了opencv与Devices两个文件夹。opencv文件夹中包含了opencv开源库的lib文件与dll文件，Devices文件夹中包含了Guppy Pro相机的API接口函数。

1. XML文档集XML Save Path：

主要存储了图像阈值、摄像机参数等视觉系统需要的参数，包含了不同时间、不同任务目标中的对应数据。

1. 辅助程序：

笔者为存储某些数据，编写了一些小程序，用于某些数据的获取与存储，如图像HSV/RGB空间中感兴趣目标的阈值范围；通过小程序获得了这些数据之后，再用XML文档以一定格式存储；运行程序时，调用自定义的xmls\_fun.cpp中的函数提取出这些数据为程序所用。



# 自定义函数CustomizeFunctions：

1. 自定义结构体CustomizeStructs：

在其他自定义函数中，某些参数经常集体出现，笔者便将其组合成结构体形式，并将其整理在该文件夹中。

目前只有图像处理自定义结构体ImageProcessHeader.h。

1. 通用图像处理函数GeneralImageProcess：

存放了笔者自定义的图像处理函数，这些图像处理函数是通用的，除了这个比赛之外如果其他工程需要，也可直接使用。笔者认为这部分的程序最为有价值。

1. 轮廓处理相关函数：

存放了所有笔者对于轮廓的操作函数，包括对轮廓筛选、轮廓排序、轮廓删除、轮廓最小外包矩形拟合等函数。主要用于图像的分析部分，使用频率很高。

具体操作直接见源码。

相关源码：contours\_fun.h, contours\_fun.cpp

1. 绘制图像函数：

存放了所有笔者用于绘制图像的操作函数，包括绘制直线、矩形等函数，可以在一帧图像上绘制出对应的形状。主要用于航行器调试时图像的显示，使用频率很高。

相关源码：draw\_fun.h, draw\_fun.cpp

1. 拟合直线函数：

存放了所有笔者拟合曲线使用的操作函数。主要用于寻线任务。

相关源码：findline\_fun.h, findline\_fun.cpp

注：A、标注初版本、二代版本都是以前的遗本，但后来由于程序改变之后弃置不用的程序；

B、笔者在写这部分程序的时候还是一个Rookie、Newbie（现在依然是），所以这部分程序虽然能完成寻线任务，但我始终认为我这种拟合方法消耗系统资源太多，所以相关的函数以后尽量考虑重写。

1. 图像预处理函数：

存放了所有笔者用于图像预处理使用的操作函数，包括阈值处理、畸变图像还原、图像裁剪、图像拉伸、二值化图像并/减运算等函数。主要用于图像采集后的预处理工作。使用频率很高。

相关源码：imageprocessing.h, imageprocessing.cpp

1. 特殊图像处理函数SpecialImageProcess；

之前想与“通用图像处理函数”相对应来一个特殊图像处理函数，然而后来把这部分完全移植到视觉类中了，故该部分完全弃置不用，使用频率零。权当当时笔者蛋疼写了该部分程序吧……

1. 其他支持函数SupportFunctions；

存放了笔者自定义的图像之外其他函数，这些函数是通用的，除了这个比赛之外如果其他工程需要，也可直接使用。在图像系统中，主要有用的部分是XML相关函数。

1. 串口相关函数：

用于串口数据格式的转换。主要用于策略主控，图像部分没有该部分的涉及。

相关源码：ports\_fun.h, ports\_fun.cpp

1. 字符串相关函数：

用于字符串格式的转换。适用面很广，但主控与图像部分都没有使用…… 所以鸡肋了。

相关源码：string\_fun.h, string\_fun.cpp

1. XML文档处理函数：

用于XML文档的读写。由于很多视觉系统的参数都以XML文档形式存储，所以该部分对视觉系统也是很重要的。XML文档存储的信息有：感兴趣目标HSV、RGB的阈值、摄像头标定参数等。

相关源码：xmls\_fun.h, xmls\_fun.cpp

# 视觉系统

1. 摄像机类NCCamera：

该类集成了摄像头的相关成员及函数，可实现摄像机初始化、图像采集、将GuppyPro相机存储于内存中的图像数据与OpenCV相接。

其中图像捕获有两种模式，一种模式是从实际摄像机中捕获图像（一般用于工控机），一种模式是从视频文件中捕获图像（一般用于调试用的PC机）。

相关源码：nccamera.h, nccamera.cpp, ncinit.cpp

注：

1. 不管师弟们是否要更换摄像机，但摄像机模块的开发，一定要详读摄像机厂商提供的资料，了解摄像机的工作机制，这样才能驾驭摄像机的使用。如果这方面偷懒，极其容易出问题，然后笋干包扎……
2. 该段程序中的GUID，是Guppy Pro相机的编号。从厂家产出的相机设备，每一个相机都有自己的编号，即GUID。当时我们买了三个相机，笔者查到了它们的GUID，便可以通过GUID赋值的方法，按照相机的安装位置（前/俯视）确定对应位置相机（即成员函数setCameraGUID的作用）。
3. 摄像机模型类NCModel，向下派生俯视/前视摄像机模型NCDownModel、NCFwdModel：

该类及其派生类集成了真实世界摄像头3D坐标系与图像平面2D坐标系的对应数学模型。该类具有理论模型的意义，但实际工程中并没有使用。不过了解摄像机模型，对于调试的顺利也是很重要的。

相关源码：ncmodel.h, ncmodel.cpp, ncdownmodel.h, ncdownmodel.cpp, ncfwdmodel.h, ncfwdmodel.cpp

1. 图像预处理类NCImgPro：

该类集成了图像预处理的数据成员与类函数。类函数包括摄像头原图像的标定重绘、图像裁剪、图像拉伸、各任务对应图像的阈值化预处理函数、以及提供给外界的各set/get函数；数据成员包括各类函数中缓存的图像数据、以及各感兴趣目标的阈值范围信息。

相关源码：ncimgpro.h, ncimgpro.cpp, ncinit.cpp

1. 其他类函数：

NCVision类的其他类函数，主要包括：

1. 各种类数据成员的初始化；

相关源码：ncvision.h, ncinit.cpp

1. 图像处理函数：用于调用NCVision类中的NCImgPro类成员的类函数，进行对应任务图像的预处理；

相关源码：ncvision.h, ncvision\_basic.cpp

1. 缓存处理函数：对于某些任务的判断，单取一帧图像进行判断会有很大误差，所以笔者开设了对应任务目标的缓存空间，将若干帧图像的判断结果存储到vector中，并用自定义的Update系列函数、Process系列函数、Average系列函数处理相关任务的缓存处理函数；

相关源码：ncvision.h, ncvision\_tmppro.cpp

1. 分析函数（重要）：以该帧预处理后的二值化图像为基础，进行分析，并得出对应任务的结果，将结果存储到视觉信息结构体VisionInfo中；主要使用轮廓分析；

相关源码：ncvision.h, ncvision\_analyse.cpp

1. 记录函数：用于记录图像分析结果；

相关源码：ncvision.h, ncvision\_record.cpp

注：实际跑工程的时候也确实使用了这一部分，但由于当时VisionInfo内容快速扩充，导致笔者完全没有经历保持对该部分函数的更新，所以该部分在我们使用的时候比较鸡肋；但在调试中出现很多问题都可以用该函数的记录功能，方便调试时问题的分析，所以这是一个很值得保留并发扬的功能；

1. 控制台打印函数：调试时在控制台一直cout信息，输出对应任务的数据信息；基本上可以说是对VisionInfo结构体中数据的字符串处理；

相关源码：ncvision.h, ncvision\_coutinfo.cpp

1. 显示函数：调试时实时显示图像处理情况的函数，在图像上会显示出分析函数得出的结果，并显示出该时刻对应任务的状态情况；可以说是控制台打印函数更加直观的反映，基本上是对VisionInfo结构体中数据结合OpenCV在图像中显示出来的操作；

相关源码：ncvision.h, ncvision\_draw.cpp

1. 控制图像捕获：调用NCCamera成员的类函数，对两个NCCamera对象DownCamera, FwdCamera进行图像的捕获；

相关源码：ncvision.h, ncvision\_basic.cpp

1. 图像数据传递：调用ImgPro成员的set/get系列类函数，将ImgPro的数据成员传递给NCVision的类成员，用于后续的图像分析；

相关源码：ncvision.h, ncvision\_basic.cpp

1. 各种set/get函数：不再赘述，看一眼源码秒懂……

相关源码：ncvision.h, ncvision\_basic.cpp

1. 视觉分析信息VisionInfo及其他宏定义数据：

VisionInfo中存储了所有视觉分析后得到的信息，包含了所有任务对应的分析结果（当然针对不同任务，其他任务的相关数据都为空）；

其他宏定义数据主要包含了任务判断依据，并以宏定义的形式处理；

另外对每个任务，笔者都设定了任务状态编号，用枚举enum的形式定义；（注：当时笔者对枚举的理解并不深刻，所以枚举的使用并不算正确，这一点师弟们吸取经验，日后有待改进）

相关源码：ncheader.h

# 后文：

江山代有才人出，各领风骚数百年。

哥老了，是你们的时代了。

附上一群被你们拍死在沙滩上的老人们的合影，请各位师弟师妹记住老朽们怨念且充满期待的眼神……





彩蛋时间：

琦少虽然已老，但依旧骚气且浪逼～