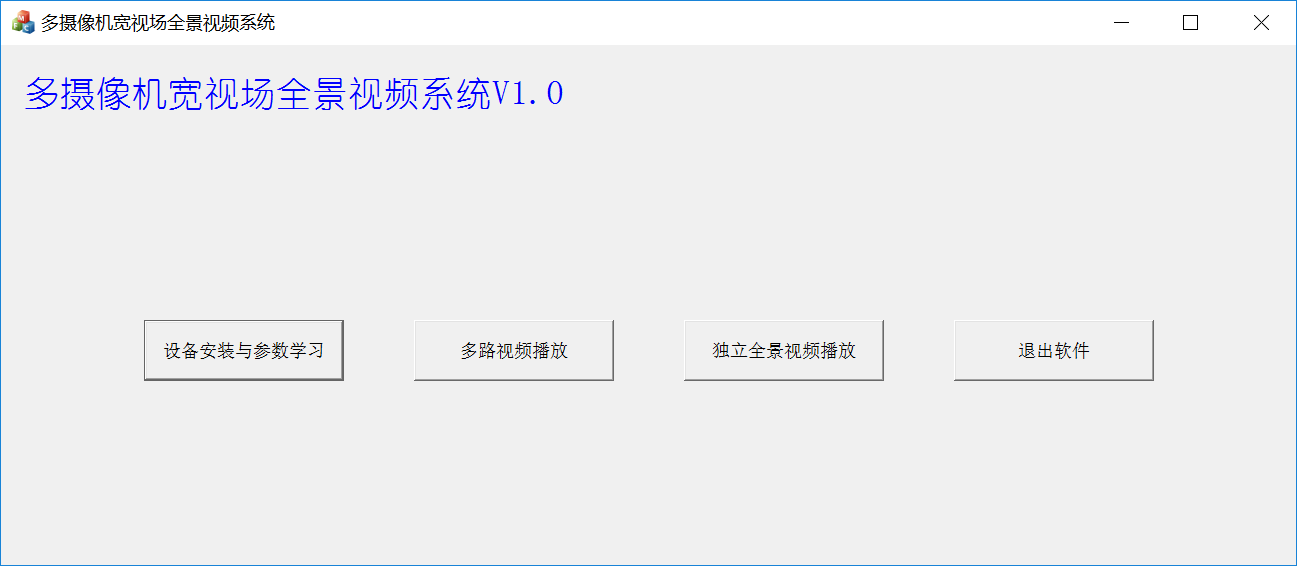
# 1前面的主要工作

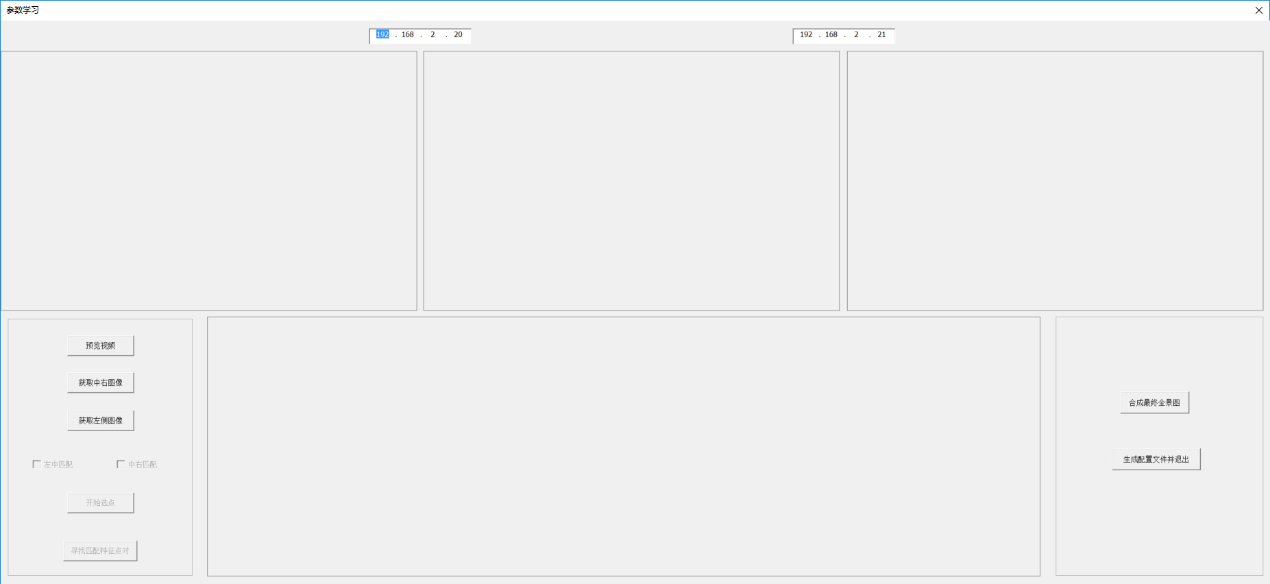
* 配合保千里方面人员搭建全景拼接系统，改进多摄像机宽视场全景视频软件并修复了一些BUG，优化性能；
* 继续开发弹性标定与配准软件，完成了Qt + OpenCV + 海康SDK环境搭建，研究了Qt中的Drag and Drop，目的是实现对标定图像的拖拽；
* 撰写《基于多摄像头的宽视场互信息配准与交互式全景视频拼接方法》专利材料。

# 2 改进多摄像机宽视场全景视频软件

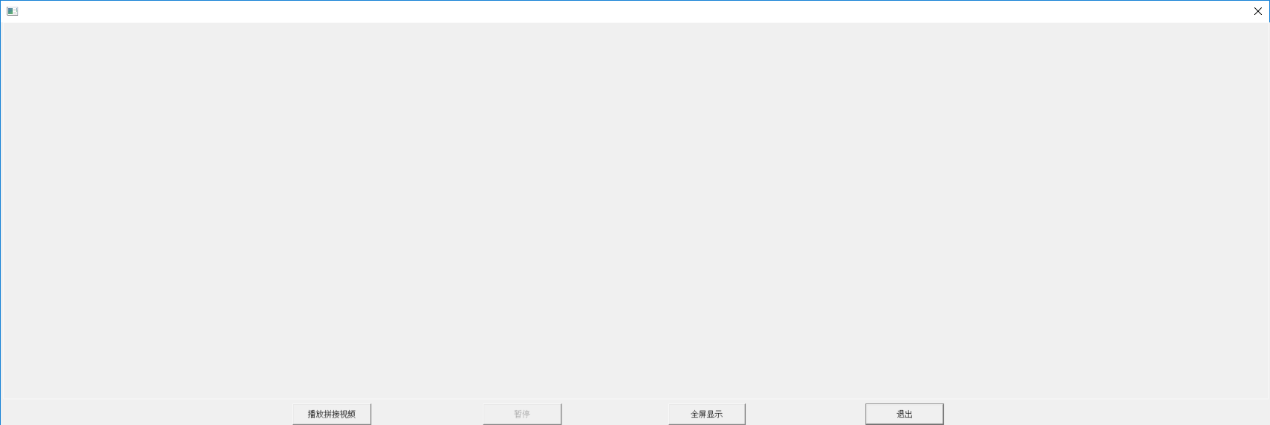
修复了视频播放画面错乱、按键逻辑错误、视频播放卡顿、软件退出时报错、内存泄露等问题，使用GPU加速改进视频拼接性能，同时实现软件全屏显示。

将软件打包成安装程序交付给保千里公司**，**软件主要界面如下**：**









# 3弹性标定与配准软件开发



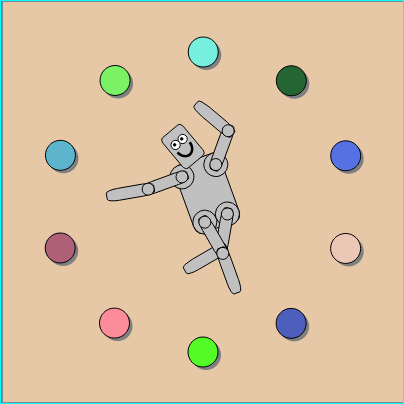
弹性标定与配准软件的优势是使用人机交互和互信息配准来对摄像头视场进行标定，基本的设计思路是：

* 对于待配准图像和基准图像，首先使用互信息配准手动选取四对匹配点；
* 在一个软件窗口中，将待配准图像投影到基准图像上，可观察到互信息配准后的结果；
* 对粗配置结果，可使用拖拽图像顶点的方式，对待配准图像精确配准到基准图像上，得到精确匹配时的投影变换参数。

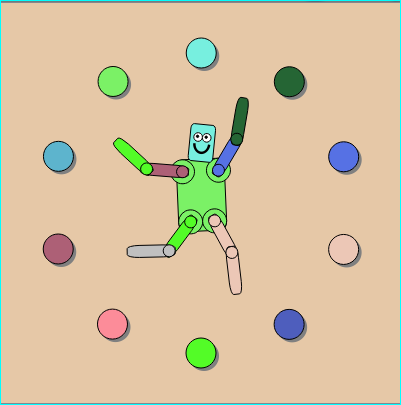
该软件的设计重点和难点在于实现拖动待配准图像实现精准匹配的过程。该过程称为：Drag & Drop的效果，即拖拽效果，例如在实现拼图游戏的时候，我们就需要能够用鼠标拖拽拼图块，然后在我们想要停下的地方松开鼠标左键，从而将该图块放置在我们指定的地方。

在选择开发平台时，虽然之前用的是MFC，但是网上可供参考的MFC对Drag & Drop的支持和类似的拖拽图片应用实例基本没有，而Qt对Drag & Drop提供了强大的支持。因此选用Qt来开发弹性标定软件界面。拖放提供一个应用程序之间传递信息的一种可视化机制。拖放机制类似于剪切和粘贴机制。拖放操作同时被QT项目视图和图形视图框架支持。

以下是Qt提供的一个换装机器人Demo，可以把机器人四周的颜色拖动到机器人的各个部位，比如说头，臂，身躯等，然后这个部位就会变成相应的颜色，类似于换装小游戏。



经过拖动颜色使其简略换装后的样子：



# 4撰写《基于多摄像头的宽视场互信息配准与交互式全景视频拼接方法》专利材料

# 5 六路摄像头实时显示

将海康摄像头取流的相关函数封装成一个类，由于从摄像头出来的数据是YV12格式（和尚涛师兄之前中间件捕获的数据格式相同），因此需要完成YV12 -> YUV -> RGB格式的操作。类中海包含必要的海康SDK函数，如登录网络摄像头、基本的摄像头参数配置函数等，从而实现实时取流。

基于以上提到的类，实现动态视频的全屏显示。



如图中所示，改变了之前视频帧像素直接叠加到底图上的做法，使用mask直接替换目标区域的像素。

同时，配置摄像头并开启日夜切换模式：白天输出正常的彩色视频，黑夜时切换至红外模式，可输出较清晰的灰度视频。



# 6 使用OpenCV + OpenCL 实现并行运算和GPU加速

使用OpenCL的主要目的是，利用GPU强大的并行能力代替CPU进行运算。

将1080p的海康摄像头码流数据进行处理并实时显示，涉及大量的运算，包括浮点运算。在此前实现的全屏实时播放软件的运行过程中，出现CPU负荷重、播放过程卡顿严重等情况，这是因为在对1920 \* 1080分辨率图像进行投影变换时耗时较长，且只使用CPU来运算。针对此场景，改进了工程代码，加入并行运算和GPU加速，OpenCV的OCL模块提供了这一功能。

在OCL模块中， GPU进行运算之前需要把内存中的图片数据转成GPU可以直接调用的显存。而在GPU上的运算结束后，我们还需要将在GPU显存上的数据转移到CPU可用的内存上。这两个操作在oclMat中定义为两个成员函数，分别为oclMat::download和oclMat::upload。

概括地说，使用OCL模块时有以下几个过程：

* 注册全局OpenCL设备。 // 此步可以省去，新版本的OpenCV会自动注册OpenCL设备
* 把内存的数据上传到显存。// 把Mat转化成oclMat
* 在OpenCL设备上进行计算。// 调用ocl模块函数
* 把显存的数据下载到内存。// 把oclMat转化回Mat
* 调用cv::函数进行剩余的运算。

**注：根据OpenCV文档，在新的OpenCV 3.x版本中，对执行在Intel CPU上的图形算法做了优化，在Windows x86和x64平台默认使用IPP来实现加速。但是目前项目组使用的OpenCV版本暂时还不是3.x版本。**

目前的效果是，在接收节点处使用GPU加速视频播放，虽无卡顿但仍然存在帧率偏低的情况。同样的程序在实验室台式机上运行则帧率正常，这是因为接收节点的性能比台式机要弱（虽然同为Intel i5处理器，但结点使用的是低功耗版本，功率仅为台式机CPU的1/2不到，主频也低了1/3）。

# 7接下来的主要工作

继续优化全景视频显示和投影变换处理的性能；

继续开发弹性标定与配准软件。