Belt Position Offset Detection with NI EVS-1464 vs PXI System

**NISH Application Engineer 鲍海翔（Haixiang Bao）**

**1 要求**

产线带子会由机械带着水平移动，比如每次平移150mm，镜头对准红色框出的“有纹路”和“无纹路”的交界点（附近）,见图1。

如果以镜头为参考0点，那么“交界点”的理想X轴座标是0，但事实上总会有一些水平位置的偏差。

客户希望，每一次图像采集加识别出结果的总时间控制在10ms以内，检测出交接点位置的偏移量。



图1. 待测样本

**2验证说明**

**硬件**

1、EVS平台

EVS-1464 (Intel Core Duo 1.66 GHz,2GB RAM, Real-Time)  
IEEE 1394 (firewire) camera (Basler scA1600-14fm)

Computar 16mm 焦距镜头

2个台灯光源

2、PXI平台(带模拟输出)

PXIe-1071机箱

PXIe-8108控制器(2.53 GHz dual-core Intel Core 2 Duo T9400, Real-Time)

IEEE 1394 (firewire) camera (Basler scA1600-14fm)

Computar 16mm 焦距镜头

PXIe-6251(M系列多功能数据采集板卡，用来输出模拟波形)

SHC68-68-EPM Cable(电缆)、BNC-2120(接线盒)、示波器

PXI-8252(IEEE 1394接口板卡)

2个台灯光源

**软件环境**

**1、EVS平台**

LabVIEW 2010, MAX 5.0 , NI-IMAQ驱动, Vision Acquisition 2010, Vision Assistant 2010

**2、PXI平台**

LabVIEW 2010, MAX 5.0 , NI-IMAQ驱动, Vision Acquisition 2010, Vision Assistant 2010、NI-DAQmx



图2. EVS实验平台

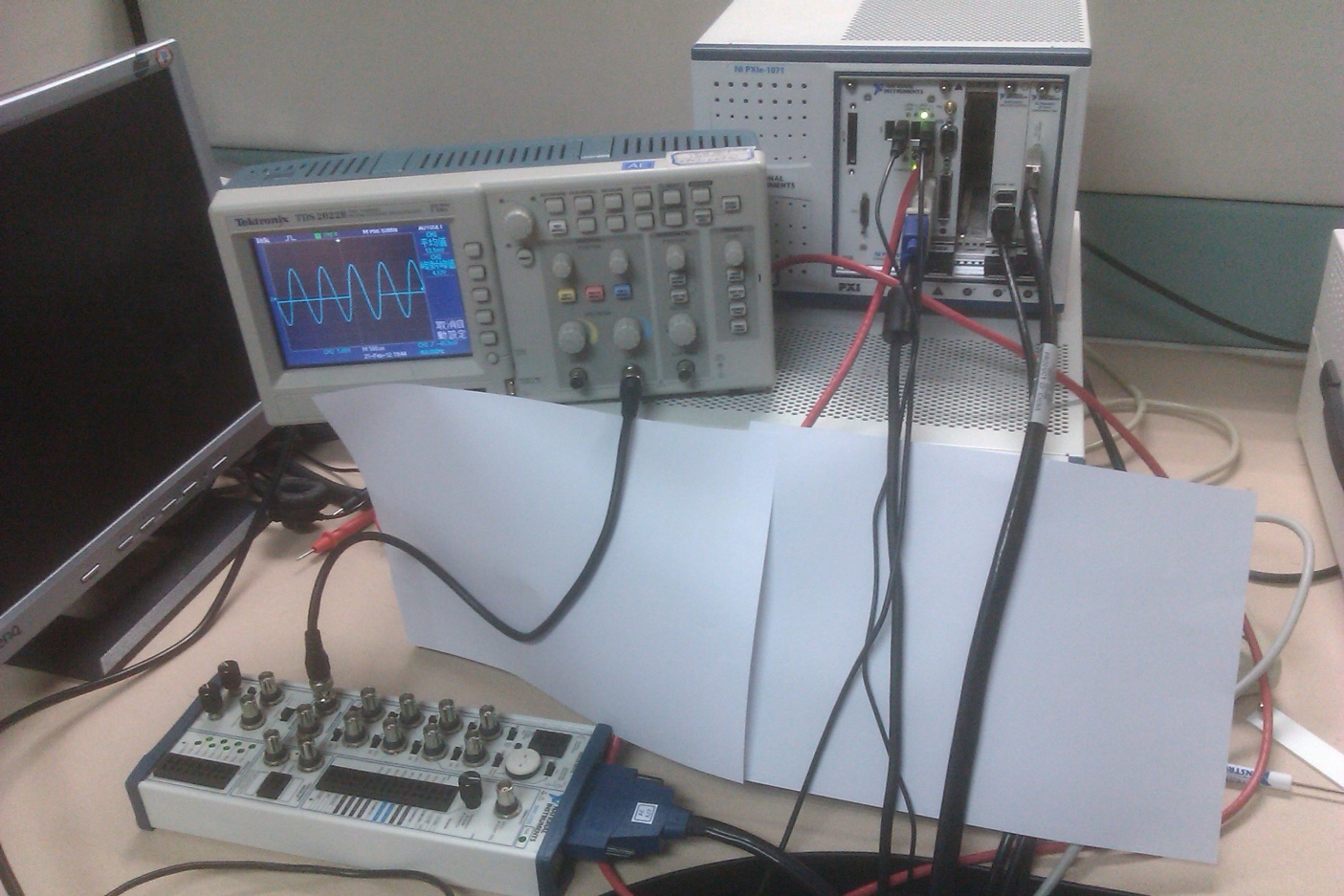


图3. PXI实验平台

**3验证**

**数据源的设置**

实验中，由于没有找到现成的环形光源，所以使用两个台灯相对放置的方式，尽量避免待测物体影子对测量效果的影响。通过调整相机高度以及镜头的焦距，使得待测物能够清晰的通过MAX的远程系统显示出来。在MAX中，可以看到，相机分辨率可以选择为1626×1236（默认），1600×1200，1280×960，640×480等等，考虑到本验证主要考虑的因素是采集以及处理的速度，所以分辨率较低且帧率较高的设置，总体时间消耗应该更小，但是因为考虑到过于小的分辨率会导致待测物移动距离的限制，移动后可能移出镜头可视区域，所以可以选用640×480的配置，帧率选择为30fps（已为最大值，后面将有不同设置时间消耗的比较）。

**图像处理算法（EVS与PXI一样）**

客户的要求是计算出“交界点”偏移镜头中心位置横坐标的量值，所以考虑为只需要获得交界点位置的横坐标就可以了，后续只需要把这个横坐标减去镜头中心位置的横坐标，而镜头中心位置的横坐标已经得知了（比如640×480的图片，整个横轴为640个像素，所以中心处为320个像素）。交界点位置横坐标可以使用“模板匹配函数”加上“设定坐标系”这两个步骤得到，“设定坐标系”可以轻松获得当前模板中心位置，也就是交界点的横坐标。为了尽可能减小算法复杂性，不再使用图像效果处理的步骤，步骤如下图所示：

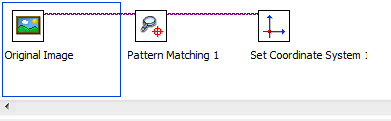


图4. 图像处理步骤

1. 获取图像
2. 待测物体交界点处模板匹配
3. 将参考坐标系设定在模板上，从而获得当前坐标值

为了计算从获取图像到处理图像的时间，使用如下图的编程方式实现：

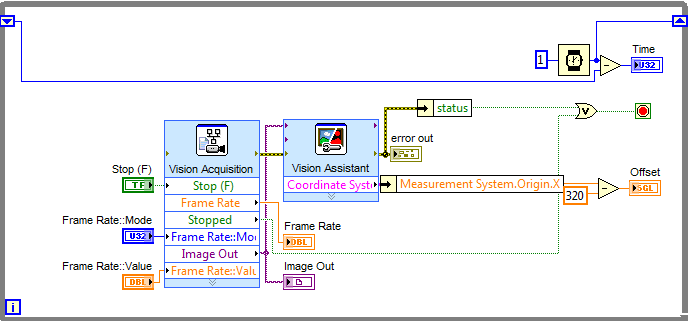


图5. EVS程序框图

注：Time表示当前的循环所使用的时间

Offset表示交界点横坐标偏移量，单位为像素

Vision Acquisition的模式设置为640×480 Mone8 30fps。

使用Vision Acquisition是因为其在调整相机分辨率方面想对比较方便，同时经过测试（在1626×1236分辨率下测试），使用Vision Acquisition与直接编程的方式时间上基本无差别。

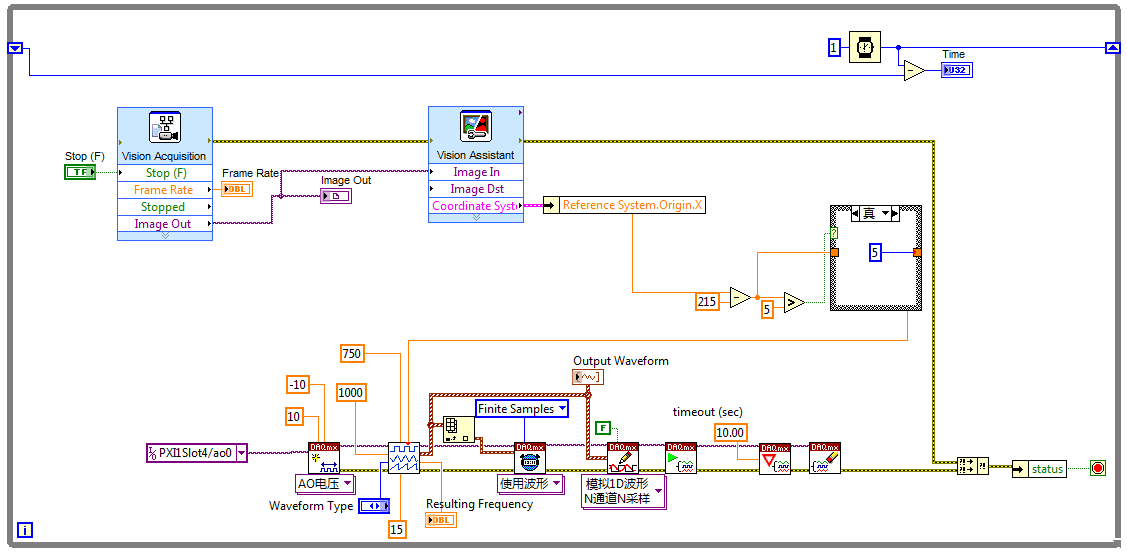


图6. PXI\_Vision程序框图

注：PXI平台下图像获取以及处理的算法同EVS下情况

PXI平台下将原来的EVS处理算法后将模板匹配的“交界点”当前偏移量作为模拟电压信号送入下方一系列的模拟输出程序中(这里我设置的是输出偏移量为模拟输出正弦波的幅值)。因为偏移量可能会因为Vision系统不准确的计算，导致输入到模拟输出板卡-PXIe6251的信号大于该板卡输出的范围(正负10V之间)，所以在这里使用一个条件结构，将模拟输出的控制量限制在5V以内，确保板卡的正常工作（10V以内即可，主要考虑不在满状态下运行）。

**开发流程**

A．VBAI或Vision Assistant交互式地进行算法建模。

B．将程序转化为LabVIEW的VI放在EVS-1464终端或者PXI的RT系统上面运行。

C．初步获得运行时间的初步参数后，在范例“Benchmark Project”上面进行准确的时间标定。

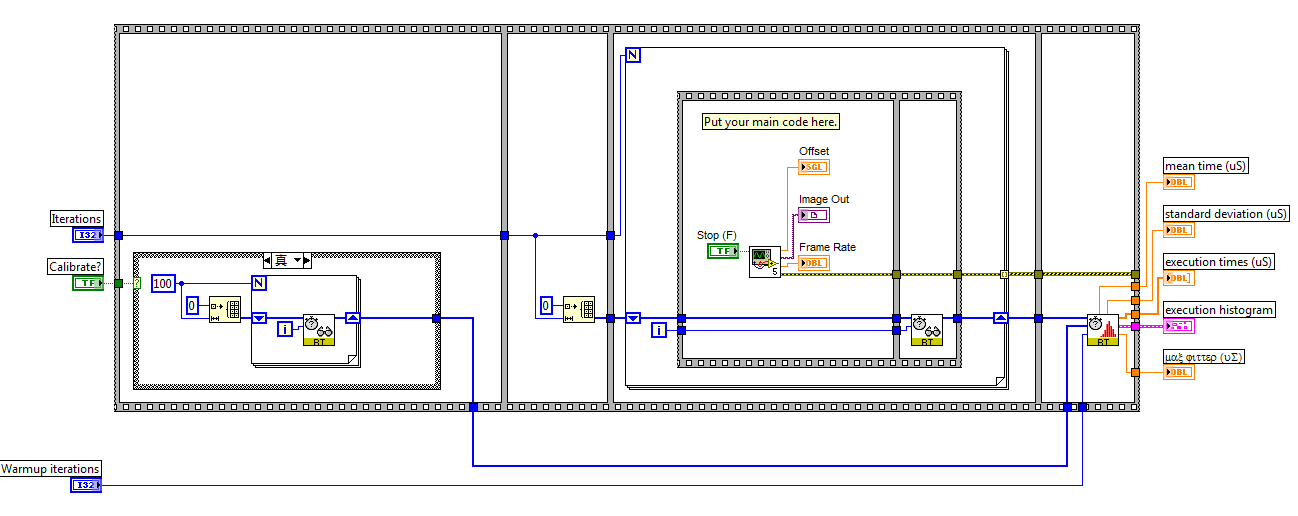


图7. 运行消耗时间标定

**测试结果**



表1. EVS-1464在不同像素下处理5000次取平均时间



表2. EVS-1464在不同帧速下处理5000次取平均时间



表3. PXI\_Vision在帧速为30fps下处理5000次取平均时间

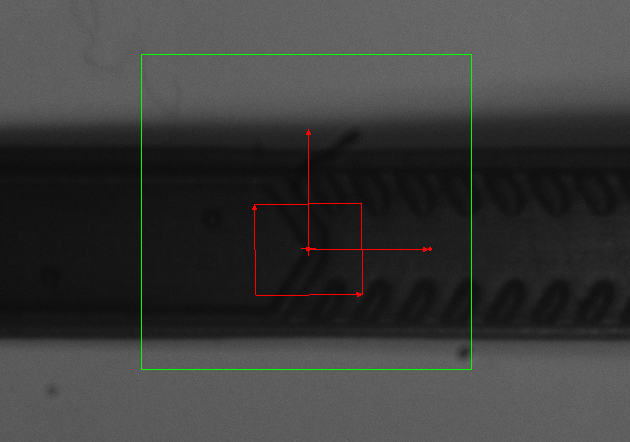
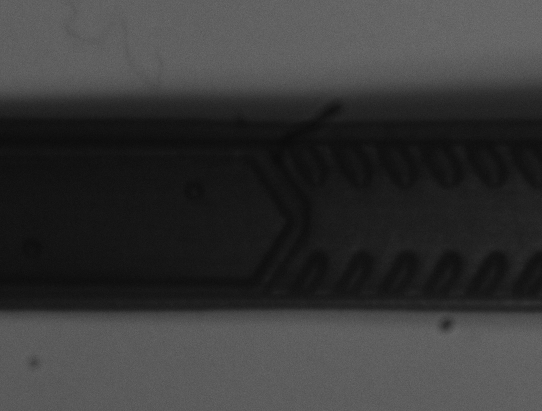


图8. Pattern Match前后对比

从表1与表2中可以看出，整个系统消耗时间主要由帧速决定，受分辨率的影响很小，所以实验前认为的分辨率与帧速共同决定的观点并不正确。另外一方面，在相同帧速以及相同相机分辨率情况下，软件算法对于图像模板的识别以及处理计算时间消耗在us级别（最大的为500us），产生这种结果的原因考虑是因为算法简化的效果，另外就是在RT系统下运行达到的us级别的精度。综合上面情况可以看出，软件识别方面完全可以满足10ms规定时间内完成“交界点”位置的测量。

注： 在本案例中，测试了grab方式与ring方式对测试结果的影响，发现效果并不明显，说明此案例中由于相机帧速的局限，将时间消耗主要定格在帧速影响上面（30fps对应的相机采集消耗时间为33.33ms）。如果使用帧速率较高的相机，可以通过使用ring的方式进一步提高相机的采集速度。IEEE1394相机可以通过提高packet size的大小来提高图像从相机到主机的传输速度，但是packet size的大小与所选择的模式有关。

在本案例中，由于只使用了一台相机，如果使用多台相机同步采集的话，就要考虑带宽限制的因素，不能在帧速上面要求过高，而有可能需要降低帧速率来平衡各个相机对带宽的占用率方面。

从表3中可以看出，对于PXI\_Vision系统，软件算法加上控制模拟电压输出，共计的消耗时间仍达到了us的级别的精度。

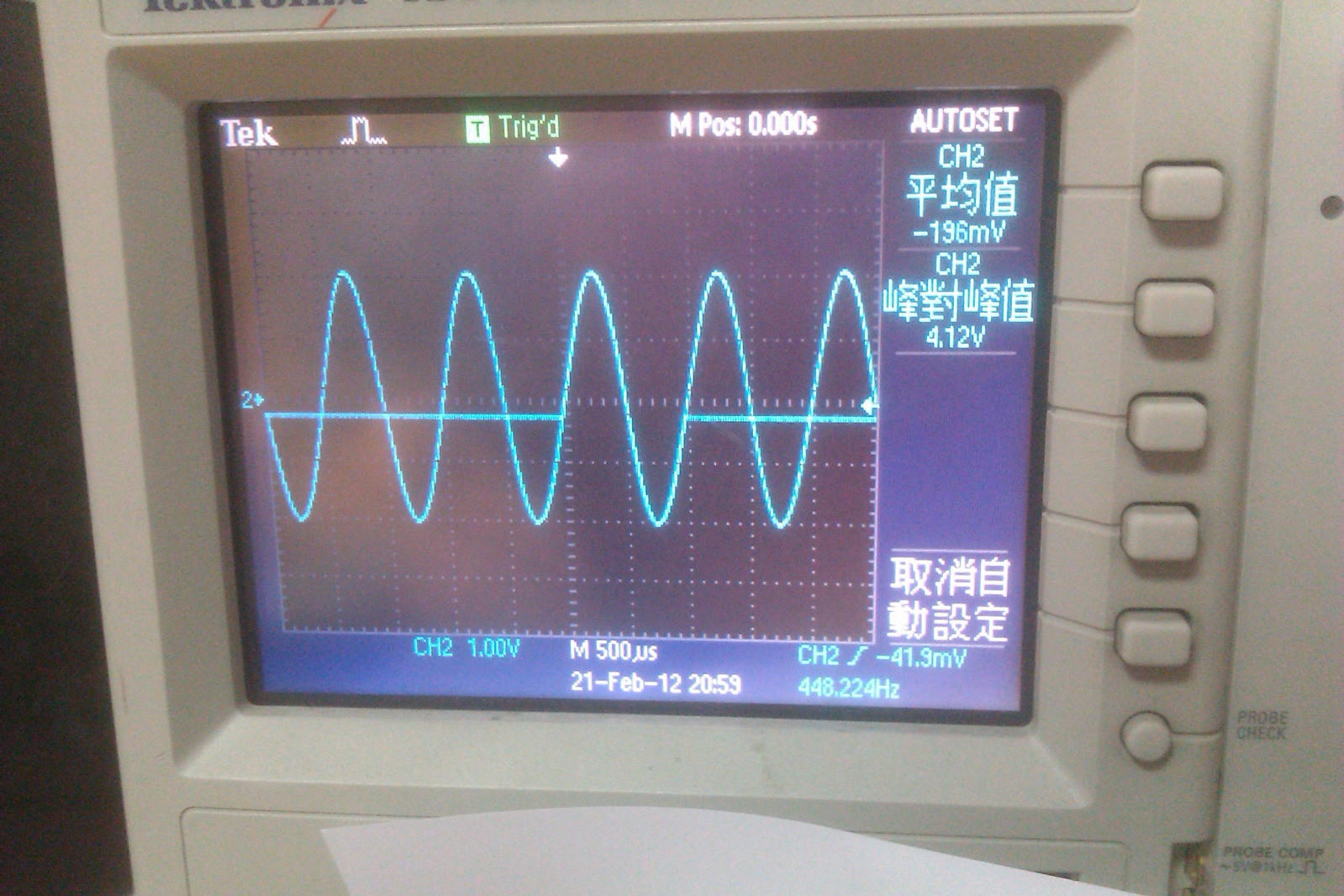


图9. 示波器显示的模拟输出波形图

**程序界面**

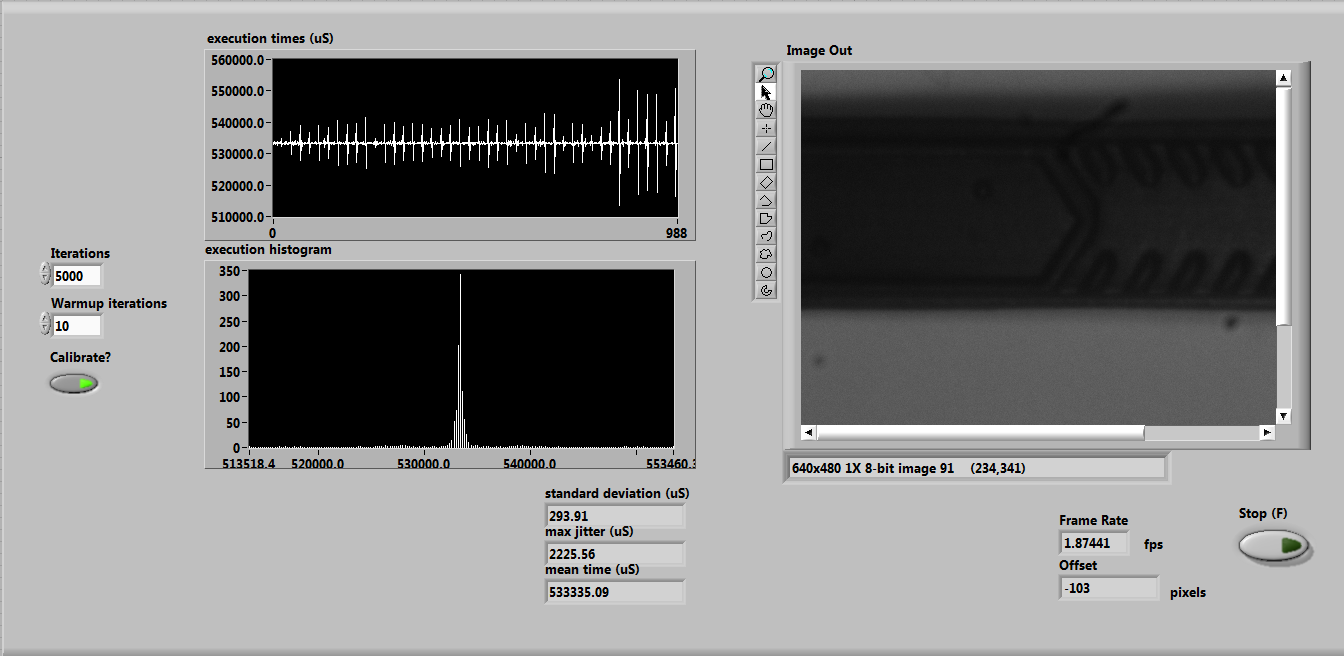


图10. EVS-1464上测试时的程序界面——image processing-time fix.vi

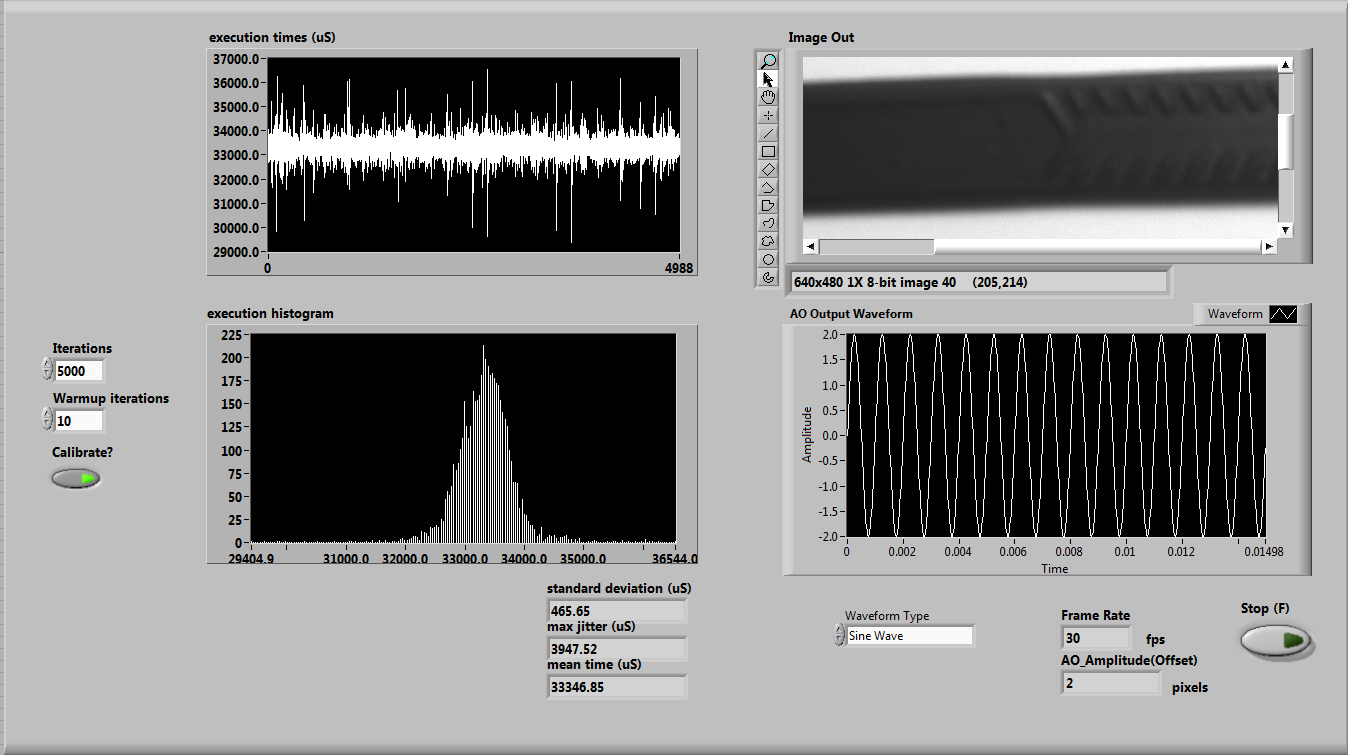


图11. PXI\_Vision上测试时的程序界面——image processing-time fix.vi

**4 结论**

本次实验主要因为相机帧速的影响，导致采集到处理的时间被限制在30ms左右，不能进一步提高，所以不论是EVS还是PXI\_Vision需要改进的方面是在硬件上，使用帧速更高的相机（在100fps以上）可以实现客户的要求，即采集到处理的时间加起来在10ms以内。PXI\_Vision即使加上了模拟输出的编程控制，仍然将时间的消耗控制在us的级别。另外，实验中由于没有环形光源，采集的图像效果不是很好，如果做更高要求的图像分析，考虑使用更好的专用光源。

软件上，考虑程序的对于客户的可读性，使用快速VI，测试时发现在RT系统下运行，将Vision Acquisition更改为代码编写的程序，对于系统时间影响不大。对于以后进一步提高相机采样速度，可以考虑使用Ring的方式。

使用Snap抓取图片也经过了测试，由于第一次开启相机需要过多的时间，所以该方式被放弃。