【20190302】光电校靶精度实验报告

2019-03-02

易思维（天津）科技有限公司

# 概述

## 1.1实验目的

根据光电校靶技术指标，设计对应的实验测试陀螺仪单轴精度、三轴合成精度、时间漂移大小、相机测角精度，并通过对比实验分析对整体精度影响较大的环节。

## 1.2实验组织

测试人员： 张楠楠 李志宇

测试时间： 20190215-20190226

报告整理人：李志宇

# 实验概要

## 2.1实验背景

通过了解柯马公司提出的基于轮廓的模板匹配的实现六自由度机器人引导解决方案的视频，在不通过已知特征点的世界坐标系的情况下，是可以实现位姿估计的。我们假设在模板匹配的时候可以找到大量的在失矫图像和标准模板图像中的对应点，事实上通过Halcon的标准例程可以发现确实能够找到大量对应点。

## 2.2实验工具

Matlab工具下仿真。

## 2.3实验测试原理

实际的情况是在相机在标准姿态下获取图像并建立模板，然后目标物体的位置做少许改变，也就是相机的位置相对目标物体成为失矫位置。为了分析方便，我们假定目标物体不做变化，相机移动到另一位置。我们的目标就是求得两个相机坐标系下的空间转换矩阵M = [R T]。

如图2-1所示，设在相机坐标系-(后面记为相机坐标系1)的坐标为()’，在相机坐标系-(后面记为相机坐标系2)的坐标为()’。根据针孔相机模型可以得到：



图2-1





其中 ，是尺度因子， 是相机的内参矩阵，两个相机坐标系中的空间坐标转换矩阵为M，也就是有：



根据公式(1)能够得到，，同样得到，带入公式(3)可以得到如下：



由公式(4)可以看出来，未知数的个数为：2n+6,等式约束个数为：3n，那么由不等式 可以得出 ,那么至少知道6个匹配点就可以进行迭代运算。

# 实验数据与分析

首先我们利用genPoints.m函数生成所需要的10组图像坐标点，genPoints.m所需要的数据包含10组相机坐标1的坐标()，相机内参A，空间转换矩阵M = [R T]。从而通过公式(1)—(4)可以计算得到其他的数据。设置的空间转换矩阵如下：



实验验证的脚本文件是myTest.m,其中调用的主要函数就是非线性最优化函数lsqnonlin。在targetFunc.m和targetFunc2.m中的生成最优化函数的目标函数向量。目标函数建立时，已知条件为：10组 和10组，相机内参矩阵A。其他参数均为待求解的变量。

**测试一：**

目标函数文件targetFunc.m是完全按照分析原理编写的函数，相应的其他参数均设置为待求解的未知变量。在运行时发现在 和 的迭代初值与标准值存在偏差的情况下，相应的迭代优化结果与在genPoints.m的设置的转换矩阵M=[R T]存在很大的偏差。我们将初设置如下，其中前九个是R，接着三个是T，接着10个位，最后10个是。r = [ 1 0 0 ,0 1 0, 0 0 1,0 0 0 ,1,1,1,1,1,1,1,1,1,1 ,1,1,1,1,1,1,1,1,1,1];

经过迭代后RR =

[1.0000 0.0000 -0.0000

-0.0000 1.0000 -0.0000

0.0000 0.0000 1.0000]

TT =

1.0e-13 \*[-0.1697 -0.2685 -0.5271]

发现空间转换矩阵基本没有进行收敛迭代，而Matlab中迭代结束的信息为lsqnonlin stopped because the final change in the sum of squares relative to its initial value is less than the default value of the function tolerance.默认的function tolerance为1.0e-6，说明在没有给定较为准确的 和的情况下，并不能得到可接受的结果。

**测试二：**

我们将初值设为一下形式

r = [1 0 0 ,0 1 0 ,0 0 1 ,0 0 1,

1200,1900,1560,1140,1530,1450,1360,1290,1120,1010 ...

2225,2847,2807,2317,2483,2484,2387,2214,2176,2119];

得到如下结果：

RR =

[ 0.9607 -0.0381 0.2750

-0.0979 0.8803 0.4641

-0.2598 -0.4728 0.8420]

TT =1.0e-10 \*[0.0004 -0.0022 0.1146]

旋转矩阵R与真实值得偏差比较小。

综合以上两种情况分析可知，由于测试一缺少尺度因子难以迭代求出准确的结果，测试二给出较为准确的尺度因子的初值，旋转矩阵就能比较好的收敛到真实值附近。由此我们猜想如果将平移向量的一个分量给出，那么是否也可以得到测试二的类似的结果。

**测试三：**

该测试使用的目标函数文件是targetFunc2.m该文件相对targetFunc.m增加了一个约束条件f(PointsNum\*3+7,1) = r(12)-r\_set，其中r\_set是一个平移向量的分量Tz的一个初值。注意该约束条件本身比较强，所以惩罚因子默认为1就行。这里我们将r\_set =1300,

得出以下结果:

RR =

0.9891 -0.0002 0.1472

-0.0616 0.9077 0.4151

-0.1338 -0.4196 0.8978

TT =

1.0e+03 \*

0.1686 0.0267 1.2980

对比真实值，RR-R=[0.0036 0.0452 -0.0164

-0.0337 0.0006 -0.0049

0.0338 -0.0012 0.0051]

相对误差，[0.0037 -0.9946 -0.1004

1.2062 0.0007 -0.0117

-0.2018 0.0028 0.0058]

主对角线上的元素差别较小.

TT’-T= [-3.0090, -11.7310, -33.3334]’,相对误差[-0.0175,-0.3050, -0.0250]’

由于在实际情况中Tz的值是个可以相对不太大的值，是可以将其约束在一个很小的范围内的，所以该测试在实际的情况下是有意义的。单次调整相机位置并不能达到一个标准位置，但可以通过2-3次调整达到一个误差容许的范围内。

**测试四：给对应点叠加正态分布的噪声**

生成正态分布随机数Nor1 = normrnd(0.1,0.2,[2 10])，Nor2 = normrnd(0.1,0.2,[2 10])，并进行叠加，PI1(1:2,:) = PI1(1:2,:)+Nor1，PI2(1:2,:) = PI2(1:2,:)+Nor2;迭代结果如下：

RR’ =

0.9891 -0.0008 0.1474

-0.0612 0.9075 0.4156

-0.1341 -0.4201 0.8975

TT’ =1.0e+03 \*

0.1686 0.0259 1.2978

RR’ - RR= 1.0e-03 \*

[-0.0257 -0.6386 0.2090

0.4252 -0.2266 0.5305

-0.3177 -0.5075 -0.2905]

相对误差[-0.0000 0.7615 0.0014

-0.0070 -0.0002 0.0013

0.0024 0.0012 -0.0003]

TT’-TT = [0.0369 0.8404 0.1623]

相对误差[0.0002 0.0315 0.0001]

我们再测试叠加均匀分布的误差，Nor1 = 0.1\*rand(2,10)，Nor2 = 0.1\*rand(2, 10)

结果如下：

RR =

0.9891 -0.0003 0.1472

-0.0615 0.9077 0.4152

-0.1338 -0.4197 0.8978

TT =

1.0e+03 \*

0.1687 0.0266 1.2980

由此可见由于量化误差和测量误差导致对应点像素定位不准，对求解空间转换矩阵的影响不大。

# 总结

由于仿真采用的数据是自己随意给出与实际的数据有很大的不同，所以在后期可以尝试从生产中获得较为真实的数据进行仿真。“基于轮廓对应点”的位姿估计并不能一次完成，可以在2-3次的校准后得到较为合理的位置。