多视角图像序列的三维重建MATLAB工具箱

目录

摘要............................................................................................................. 3.

1.总览

1.1简介...................................................................................................... 3.

1.2两个简例....................................................................................................

2.用户手册

2.1相机标定工具箱..........................................................................................

2.2立体几何函数库..........................................................................................

2.3相机模型函数库..........................................................................................

2.4 三维重建基本函数库...................................................................................

2.5 可视化函数库...............................................................................................

3. 更多示例.......................................................................................................

4. 附录

摘要

三维重建是机器视觉的重要组成部分，也是近年来图像技术方面的的门课题之一。 目前基于C/C++语言的平台有OpenCV等.MATLAB作为一种广泛使用的优秀的科学计算编程语言,近些年来也在图像处理和机器视觉上有所发展，但还没有出现针对三维重建的工具箱。该工具箱主要在MATLAB上提供一个三维重建的开发平台（包括相机标定工具箱、立体几何函数库、相机模型函数库、三维重建基本函数库、可视化函数库等），用于进行相关课题的学习、教学演示、程序开发、高级算法开发等。工具箱中还包含了已经利用该平台初步开发出的几种不同的三维重建解决方案（已知控制点、未知控制点、以曲线为对象的情况），适用于不同场合的三维重建。

**1.总览**

**1.1简介**

什么是多视角图像序列的三维重建？

多视角图像序列的三维重建是指根据一个物体的不同角度的数码相片还原出物体的三维模型，这类课题是机器视觉中重要的组成部分。三维重建技术可从平面图像中提取空间信息，补充平面图像的不足。

该工具箱解决了什么问题？能在哪些方面使用？

该工具箱主要在MATLAB上提供一个三维重建的开发平台（丰富的函数库、数据库、可视化等功能）,用于进行相关课题的学习、教学演示、程序开发、高级算法开发等。工具箱中还包含了已经利用该平台初步开发出的几种不同的三维重建解决方案，适用于不同对象的三维重建。

该工具箱作为开发平台具体有那些功能？

开发平台的功能包括

（1）改进的相机标定工具箱(包含GUI界面)——对普通的非测量图像设备进行标定，矫正图像扭曲、中心偏移等误差；

（2）立体几何函数库——处理空间点、矢量、射线、直线、平面、曲面等基本的几何对象之间的各种复杂关系（如改变其位置、求交点、求距离、求位置关系等）；

（3）相机模型函数库——建立相机的投影模型，对各种坐标系（本地、相机、相片）之间的基本几何对象进行转换及处理。

（4）三维重建基本函数库——提取特征点；根据已知坐标的控制点定位相机；管理原始数据及中间步骤产生数据的数据并自动存档等。

（5）可视化函数库——对前面功能进行大量的3D可视化实现，几乎每个计算过程的可视化都能完美呈现。

利用这些现成的功能, 用户可以方便地实现三维重建相关的程序的调试、修改、演示，也可以用于该方面课题的学习、教学等环节。

已利用该平台初步开发出的重建程序有哪些功能？

作者利用该平台已经初步开发出两类可实际应用的三维重建程序。

（1）当相片中含有空间坐标已知的控制点时(对于小物体的三维重建,通常用一张棋盘作为背景，在棋盘上建立空间坐标系),可利用这些控制点先精确定位相机（每张相片对应相机的一个位置和朝向），再进一步根据相片上的对应特征点逆向求解出每个特征点的空间坐标，最后根据算出的空间点进行模型的美化。

经过实际试验,这种方法的精度非常高,仅一个848\*480分辨率的摄像头可以把小物体上一点确定到0.01mm.

（2）在（1）的条件下，进行了相机定位以后，如果由于物体表面的性质（例如非常光滑）只能分辨出曲线，可以在各张图上选取对应的曲线，生成3D曲线以确定模型的形状，再对模型进行美化。

（3）当相片中没有空间坐标已知的控制点(例如建筑物等)时，

该工具箱能完美实现三维重建的所有过程吗？

目前该工具箱侧重于数据处理方面的算法（二维数据还原为三维数据的过程）。至于图像识别方面的算法，如特征点识别和配对、边界和角点检测、运动检测等图像（视频）识别方面的功能，不在该工具箱的目标范围。然而近几年来MATLAB在这些方面已经逐步增加了新的工具箱（例如Computer Vision Toolbox 和 Image Processing Toolbox）由于作者能力有限，未能及时将这些工具箱的内容整合到该工具箱中来，但程序中提供了数据接口和手动获取图像特征的友好界面。在本文的示例中，特征点（或曲线）由手动获取而得。

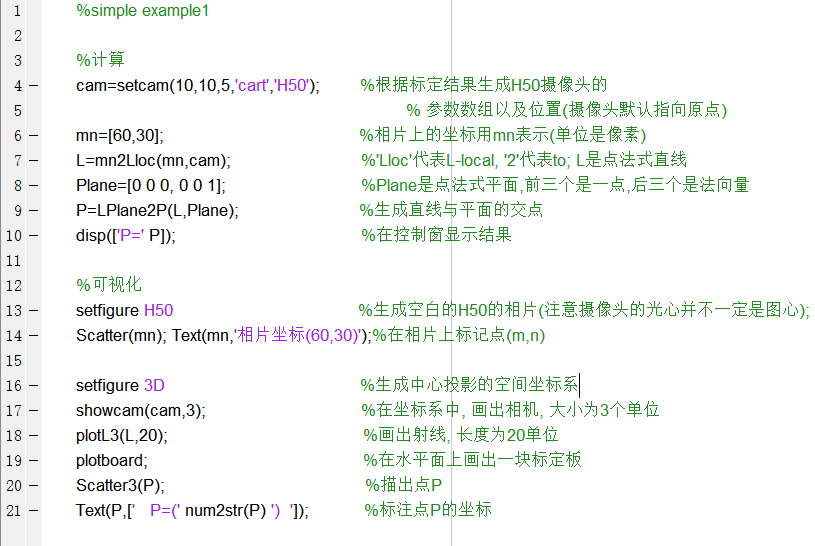
**1.2简例**

下面给出两个简例,例1是使用开发平台的简单实例, 例2是简单的三维重建实例.

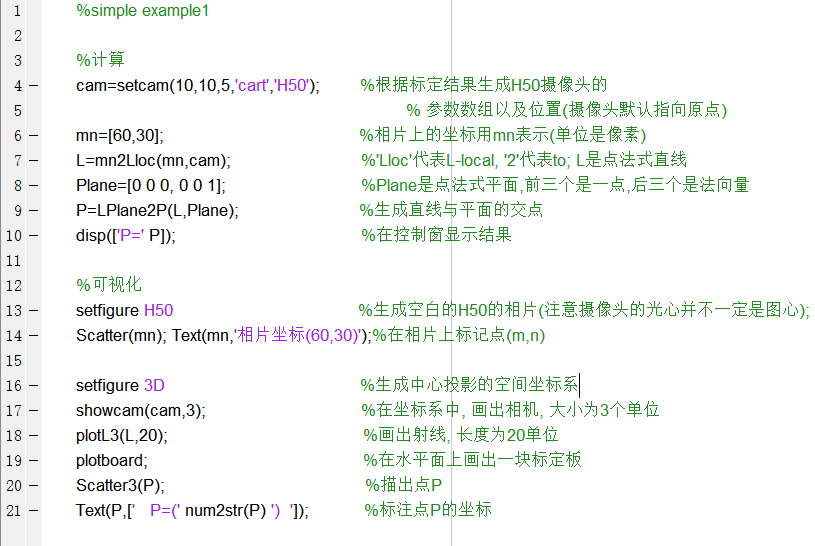
**简例1**

已标定好的H50(极速牌)摄像头在位置坐标为(x,y,z)=(10,10,5)水平俯视放置,主光轴通过原点,已知水平面上一点P在相片上的坐标(m,n)=(60,30),求点P的坐标,并且把结果可视化.

计算过程如下:



可视化编程如下:



说明:

以上的程序中,除了disp函数,使用的都是工具箱中的函数.在工具箱中,经常用到的数据已经过命名和规格标准化,例如cam是6\*3的数组,前3行分别是相机坐标系的3个单位向量,第4行是相机位置,第5,6行是中心偏移、图像大小、焦距等。

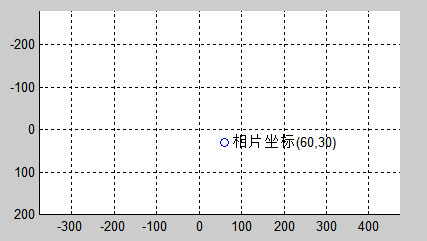
另外,函数的命名在工具箱中也遵循很好的规律。例如空间点表示为P，矢量为v,图像坐标为mn,相机为cam,本地坐标系为loc等。'2'通常用来表示转换关系. 例如函数mn2Lloc可根据图像上的点生成对应的空间射线, 又如vcam2vloc把相机坐标系中的矢量转换为本地坐标系中的矢量.

为了维持matlab可进行数组运算的优势,几乎所有的工具箱函数都支持数组运算. 例如mn坐标的数组规定为N\*2,直线(射线)L的数组为N\*6.

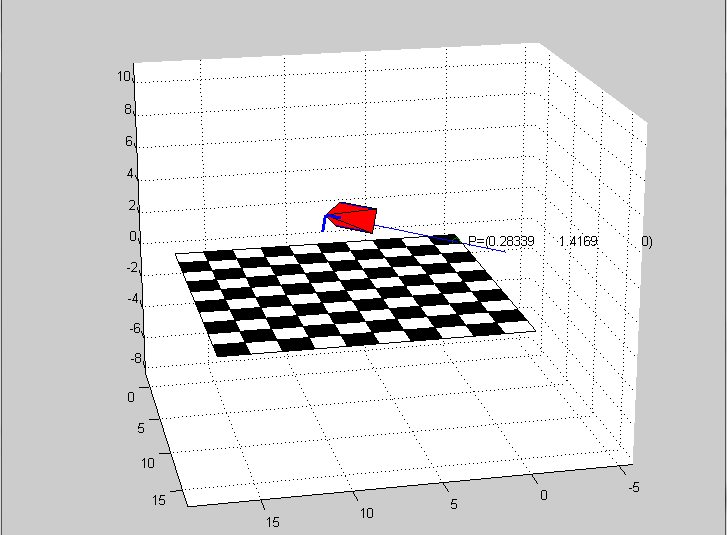
运行结果:

P=[0.28339 1.4169 0]

生成的图像如下



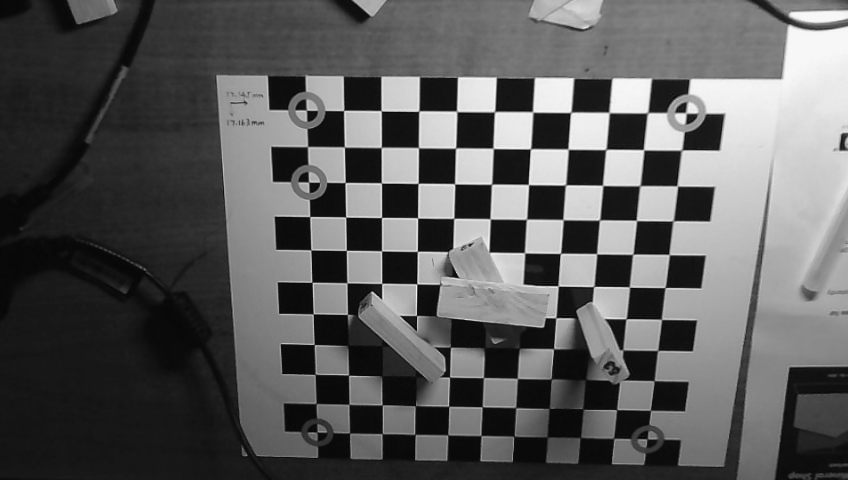
生成的3D视图如下



**简例2: 积木的三维重建**

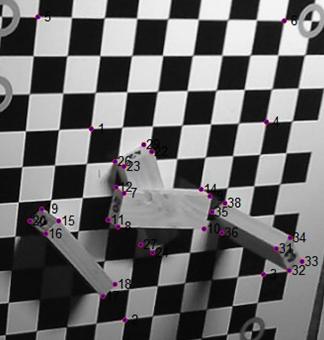
1.标定相机(详见相关章节)

2.在标定板上任意摆放积木并拍照(共七张)



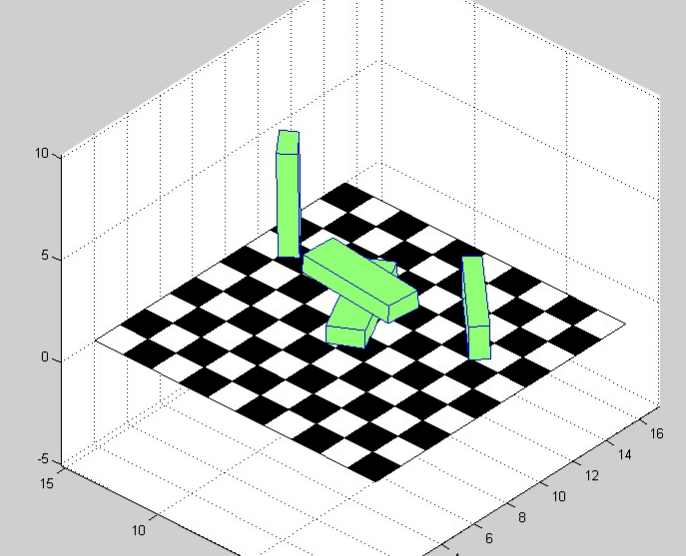
3.特征点的选取

该工具箱提供外部的数据接口(可利用MATLAB的Computer VisionToolbox或者Image ProcessingToolbox的相关功能)和方便的手动选取界面.选取完以后,程序会自动在文件夹中生成相关的数据文件供以后调用,查看等.

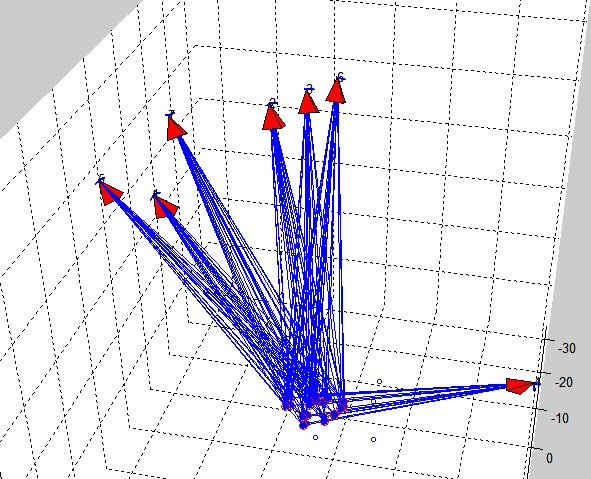


(特征点选取情况显示)

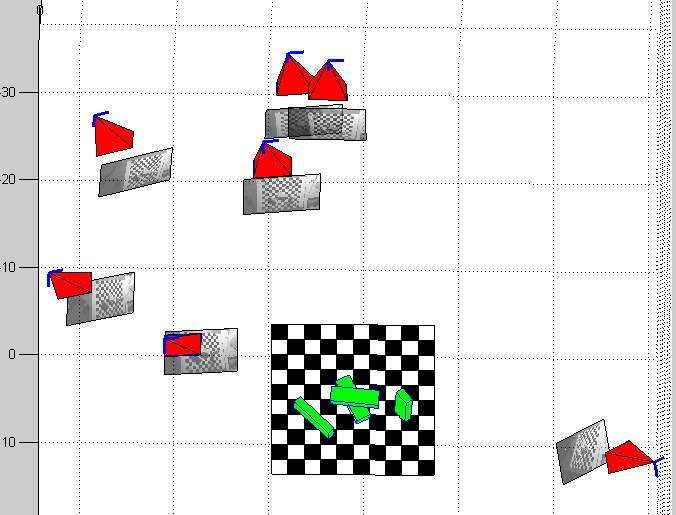
4.运行程序next.m和next\_check.m

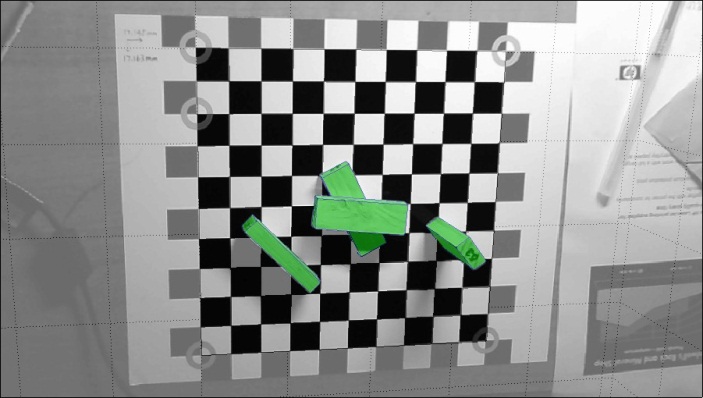
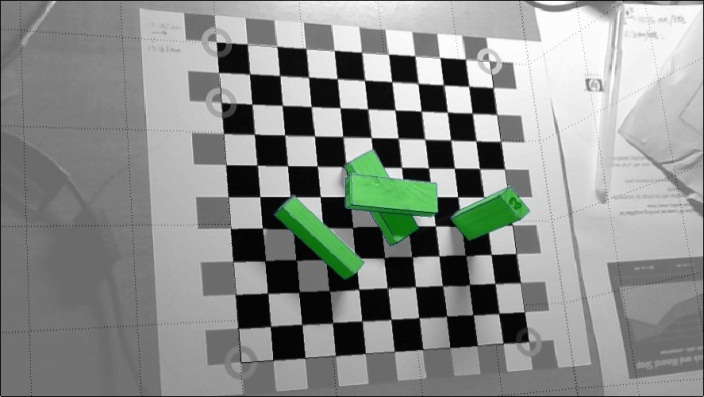
运行结果:

交叉检验（用于误差分析，每个空间点放大后会显示详细数据）:



重叠投影检验:把生成的3D模型在各个对应的相机位置观测，并透过半透明图像重叠观察，以检验结果的正确性.





**2.用户手册**

**2.1相机的标定**

**2.1.1改进的相机标定工具箱**

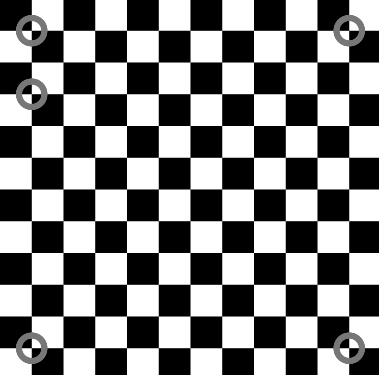
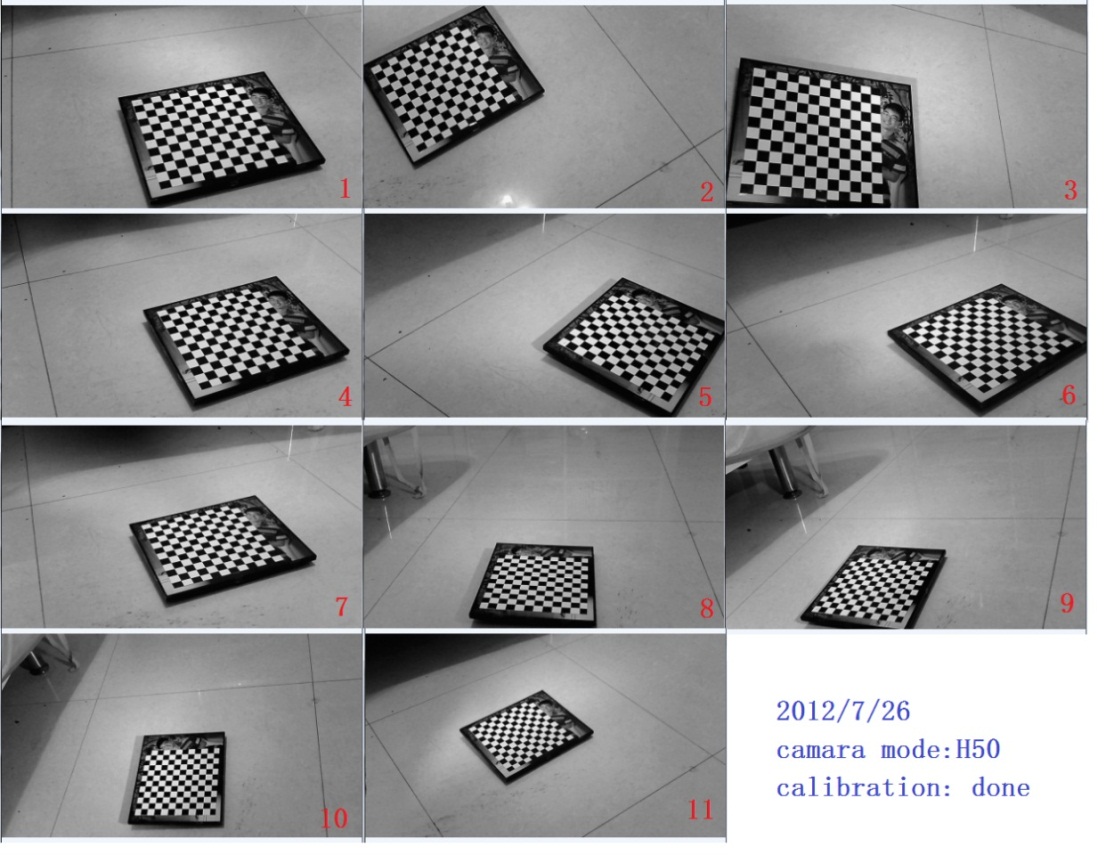
1.改进的相机标定工具箱

相机标定工具箱是根据网上的一个开源工具箱修改整合而来,改进的部分实现了跟该作品之间很好的兼容. 该工具箱可以根据一张打印的格子纸来标定相机,修正相片的形变,同时获取相机的参数(如视角,焦距等).

原工具箱网址(http://robots.stanford.edu/cs223b04/JeanYvesCalib/)

**2.1.2标定步骤**

1.打印格子纸,用摄像头拍照(建议10张左右,如下图a)

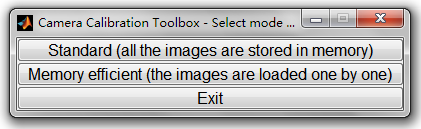


2.对标定板拍照

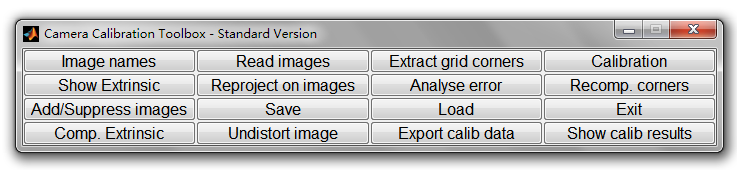
把相片按照[名字+序号]的方式命名,存入一个文件夹.例如:"calib1.bmp", "calib2.bmp"等等(相片格式支持'.bmp'、 '.jpg'、'.tif'、'.ras'、'.pgm'、'.ppm')

3.运行工具箱

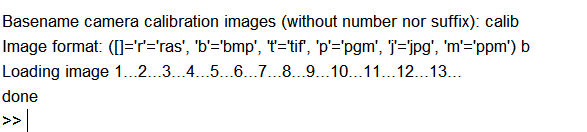
在MATLAB命令窗输入calib,启动工具箱



选择Standard



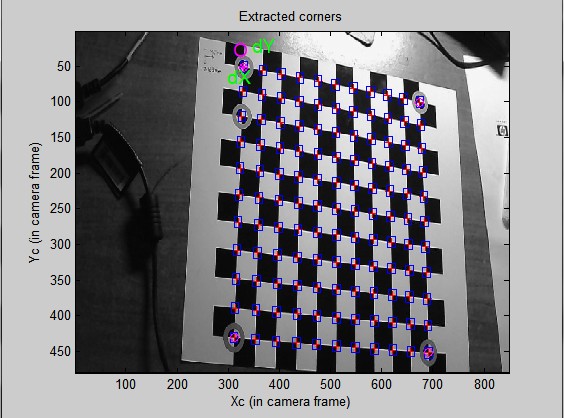
选择Image names,按提示输入相片名,后缀名



继续选择 Read images,结果如下



选择Extract grid corners,手动或者自动在图上选取标定板的四个角(若有个别无法自动识别的将提示手动选取),每一张图选完四个角以后,会自动选取剩下的角点.

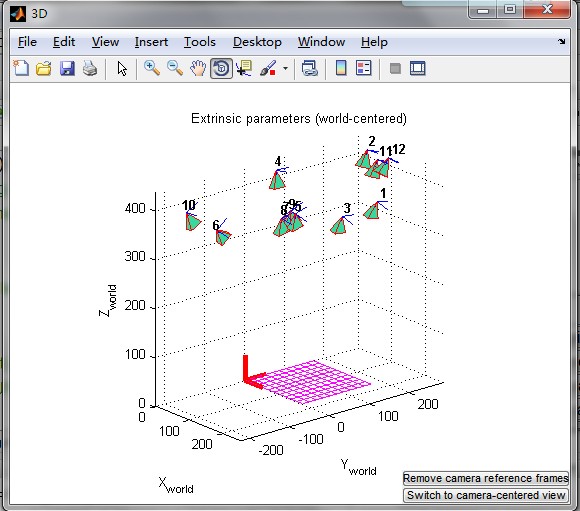
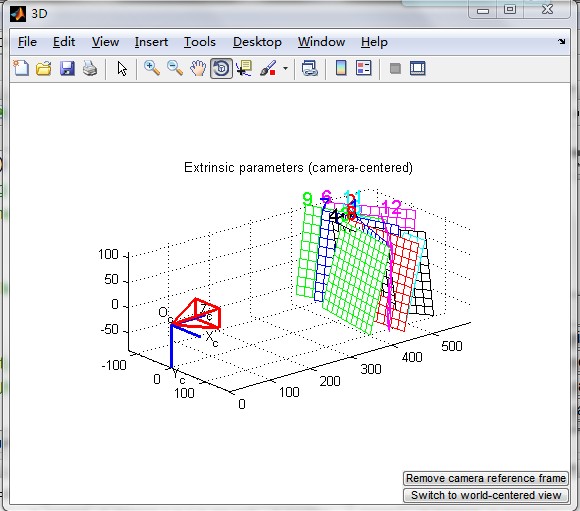


在界面中继续点击Calibration, 标定程序将自动运行.

运行结果:

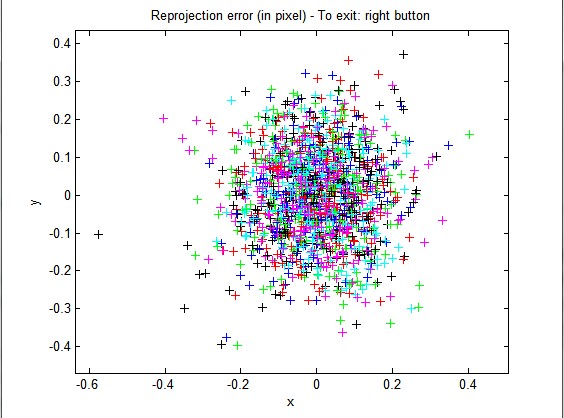
分别在界面中选择Show Extrinsic可显示标定板与相机的位置关系

(左图为相机参考系中标定板的位置,右图为标定板参考系中相机的位置)



选择Analysis Error进行误差分析

误差结果如下图(单位:像素)



对任意一点点击左键,可在控制窗口显示该点所在的相片以及该点的坐标, 以便查看该点的选取是否有误.

标定完相机以后, 程序会把相机的各种参数(如焦距,视角,中心等)以及误差分布保存至Calib\_Result.mat文件,以后每一张相片都可以根据结果矫正,生成矫正后的照片.

**2.2几何函数库**

几何对象的格式标准

矢量v:N\*3 double

每一行为一个矢量的三个直角坐标(x,y,z)。

空间点P: N\*3 double

每一行为一个点的三个直角坐标(x,y,z)。

射线（或直线）L：N\*6 double

每一行的前3列为射线的端点(直线上一点),后3列为射线的单位方向向量。

带误差的射线 Lerr2: N\*8 double

每一行的前6列为L，第7列为端点的误差估计（Perr），第8列为方向向量的角误差估计（angerr）

平面Plane: N\*6 double

每一行的前三列为平面上一点,后三列为平面的单位法向量。

关于矢量的函数

函数

PlotV3

格式

PlotV3(V)

功能

在空间坐标系中画出矢量V

函数

vang

格式

angle=vang(V1,V2)

size(V1)=size(V2)=[N,3];size(angle)=[N,1]

功能

求出两个矢量的夹角(弧度).区间[0 pi].支持数组运算

函数

vlaw

格式

V=vlaw(V1,V2)

size(V1)=size(V2)=size(V)=[N,3]

功能

求出两个矢量确定的平面的单位法向量(根据右手定则).支持数组运算

函数

vmag

格式

A=vmag(V)

size(V)=[N,3];size(A)=[N,1];

功能

求出矢量的模长.支持数组运算

函数

vunit

功能

把矢量化为单位矢量

Vu=vunit(V)

函数

vturn

格式

(1)Vt=vturn(V,A1,A2);(1)

(2)V1=turn(V,A,theta);(2)

size(V)=size(Vt)=[N,3]; size(A)=size(A1)=size(A2)=[1 3];

size(theta)=[1 1];

功能

(1)把空间矢量V从矢量A1方向转到矢量A2方向（绕A1叉乘A2的方向）。

(2)把V以A为轴,以右手定则旋转theta角(单位弧度)。

关于直线

函数

L22h

格式

h=L22h(L1,L2)

size(L1)=size(L2)=[N,6]; h=[N,1]

功能

计算两条直线的距离

函数

L22P

格式

[P,h,t1,t2]=L22P(L1,L2)

size(L1)=size(L2)=[N,6]; size(t1)=size(t2)=size(h)=[N,1]; size(P)=[N,6];

功能

计算两条直线的近似交点(最短的连线的中点)

函数

Ln2P

格式

[P,Hs]=Ln2P(Lerr2)

size(Lerr2)=[N,8]; size(P)=[N,3]; size(Hs)=[N,1]

功能

求出n条带误差射线的加权最优交点. 根据误差决定权重,具体表达式可在文件中修改."最优"是指选择交点P,使P到每条射线的距离(Hs)的平方和最小,用最小二乘法进行计算.

函数

LP2H

格式

H=LP2H(L,P)

size(L)=[N,6]; size(P)=[N,3]; size(H)=[N,1]

功能

给定一条射线L和一点P,求出距离H

函数

LP2P

格式

[P1,R]=LP2P(L,P)

size(L)=[N,6]; size(P)=size(P1)=[N,3]; size(R)=[N,1]

功能

给定一条射线L和一点P; 求L上的P1是P1离P最近, 并且求出P1离射线端点的距离R

函数

P22L

格式

L=P22L(P1,P2)

size(P1)=size(P2)=[N,3]; size(L)=[N,6];

功能

给定两点P1,P2,求出以P1为端点,过P2的射线L

函数

plotL3

格式

plotL3(L,Len,varargin)

size(L)=[N,6]; size(Len)=[1,1]或[N,1];

varargin是([属性名],[属性值],....)的参数对,用于控制线的属性,用法与MATLAB的plot函数一致

功能

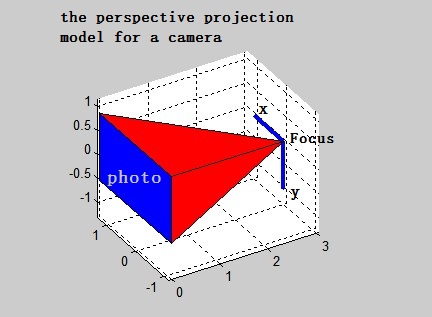
画出指定长度(Len)的射线L

**2.3 相机模型函数库**

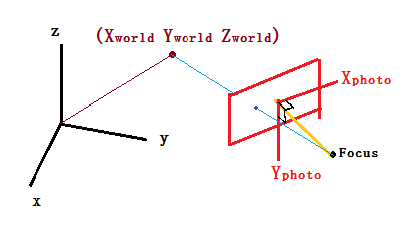
**2.3.1相机模型**

相机模型如下方图c所示,蓝色的平面是图像,四棱锥的顶点是焦点,空间中的一点与焦点连线与图像上交点就是图像中的对应点。

相机成像的数学模型（中心投影）



工具箱中的相机模型



**2.3.2程序中用到的坐标系**

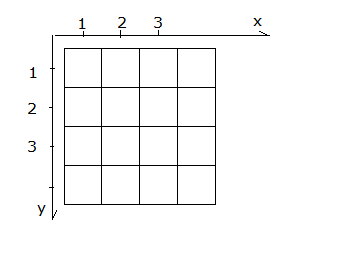
本地坐标系:与物体保持相对静止的坐标系 (在其他文献中也称世界坐标系)

相机坐标系:与相机保持静止,以主光轴为z轴,相机右方为x轴,相机下方为y轴的坐标系(如图).

图像坐标系:光轴上的一点在相片上的投影点作为原点(而不是相片的中心),相片右方为x轴,相片下方为y轴的平面坐标系.相片中的坐标在程序中以'mn'表示（size(mn)=[N,2]).原点的位置在标定相机使自动算出,储存在

Calib\_Results.mat的cc变量中.（size(mn)=[N,2]).

spatial坐标系:由于相片在MATLAB中通常以矩阵的形式保存,所以还有一种与矩阵行标和列标一致的坐标系(但允许使用小数),在程序中通常用'xy'表示,示意图如下(图中的每个方格代表一个像素).



下面是程序中对特定数组的格式规定

相机基本数组 cam：size(cam)=[N,6];

图片坐标数组 mn: size(mn)=[N,2];

spatial坐标数组 xy:size(xy)=[N,2];

矢量数组 V:size(V)=[N,3];

函数

setcam

格式

(1)cam=setcam(camera)

(2)cam=setcam(x,y,z,'cart',camera)

(3)cam=setcam(phi,theta,r,sph\_cart,camera)

函数

vcam2vloc

格式

vloc=vcam2vloc(vcam,cam)

size(vloc)=[N,3]; size(vcam)=[N,3]; size(cam)=[6,3];

（未完成）

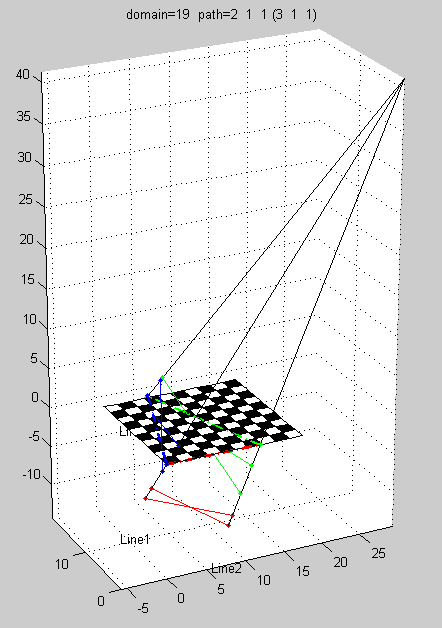
**2.4三维重建基本函数库**

该部分主要实现三维重建中一些基本的问题，如相机定位的几种方式，相机朝向的最优化等

**2.4.1三棱锥相机定位法(原创)**

三棱锥相机定位法可根据图片上空间坐标已知的极少几个控制点(大于3)实现相机的定位.

数值解三棱锥过程



(未完成)

**3.更多示例 (未完成)**

线条模型

生成一个四驱车模型外壳的线条

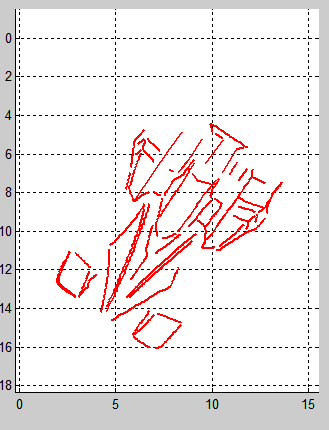
(1)用标定过的相机对车壳进行拍照(以标定板作为背景)

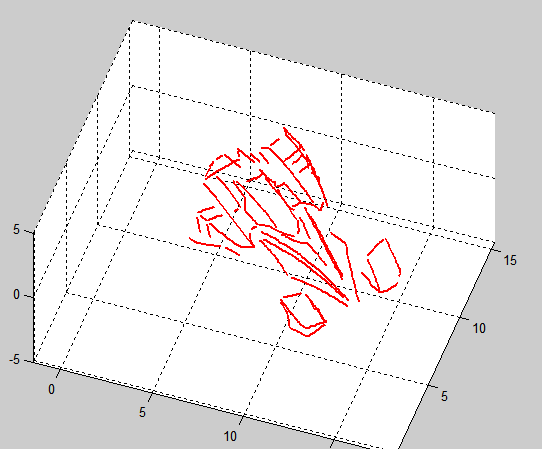


(2)利用featureline程序根据提示获取特征曲线.



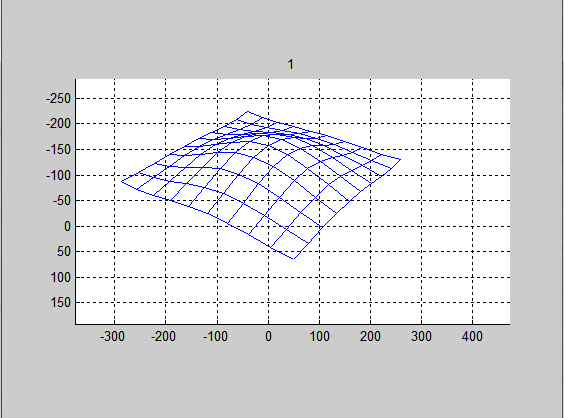
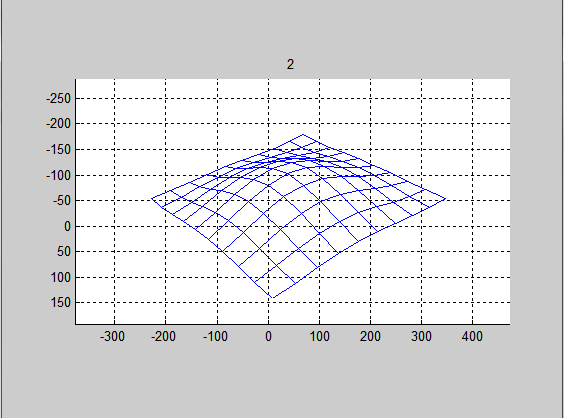
运行结果(只选取了部分主要曲线)





没有控制点情况下的模拟实验

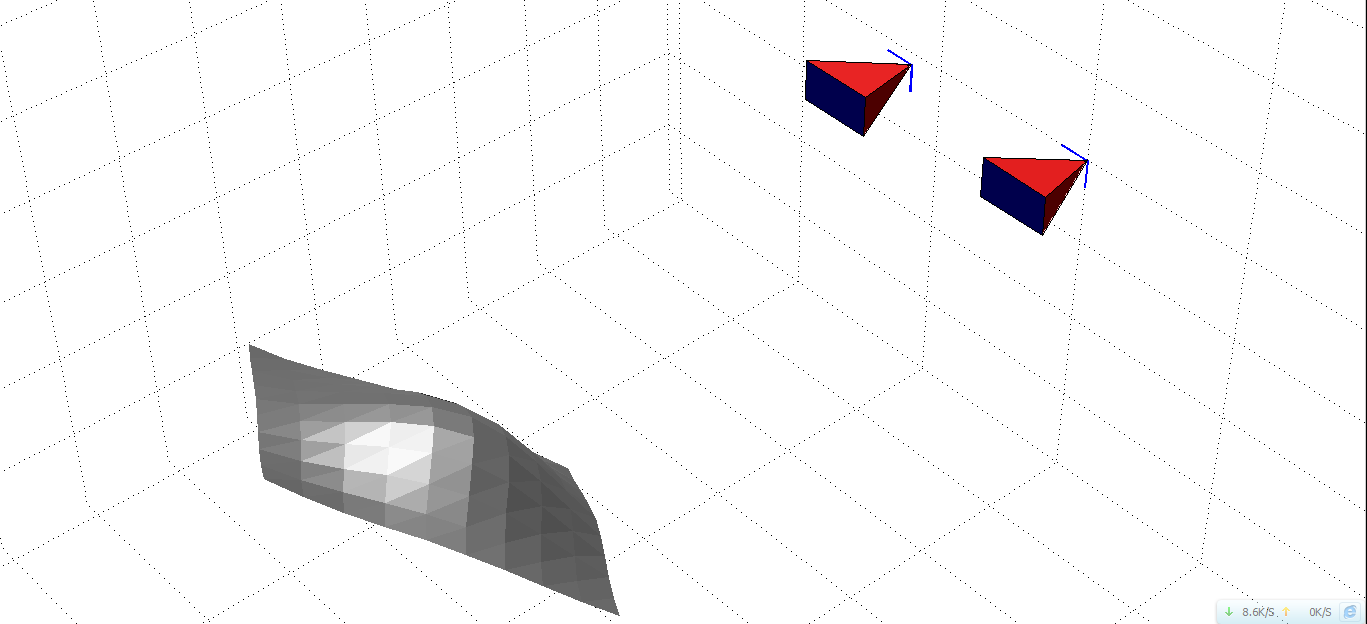
sinc函数生成的一张空间曲面网格在两个不同位置的相机中投影出的图像.





运行结果.

渲染后的效果.



**4. 附录**