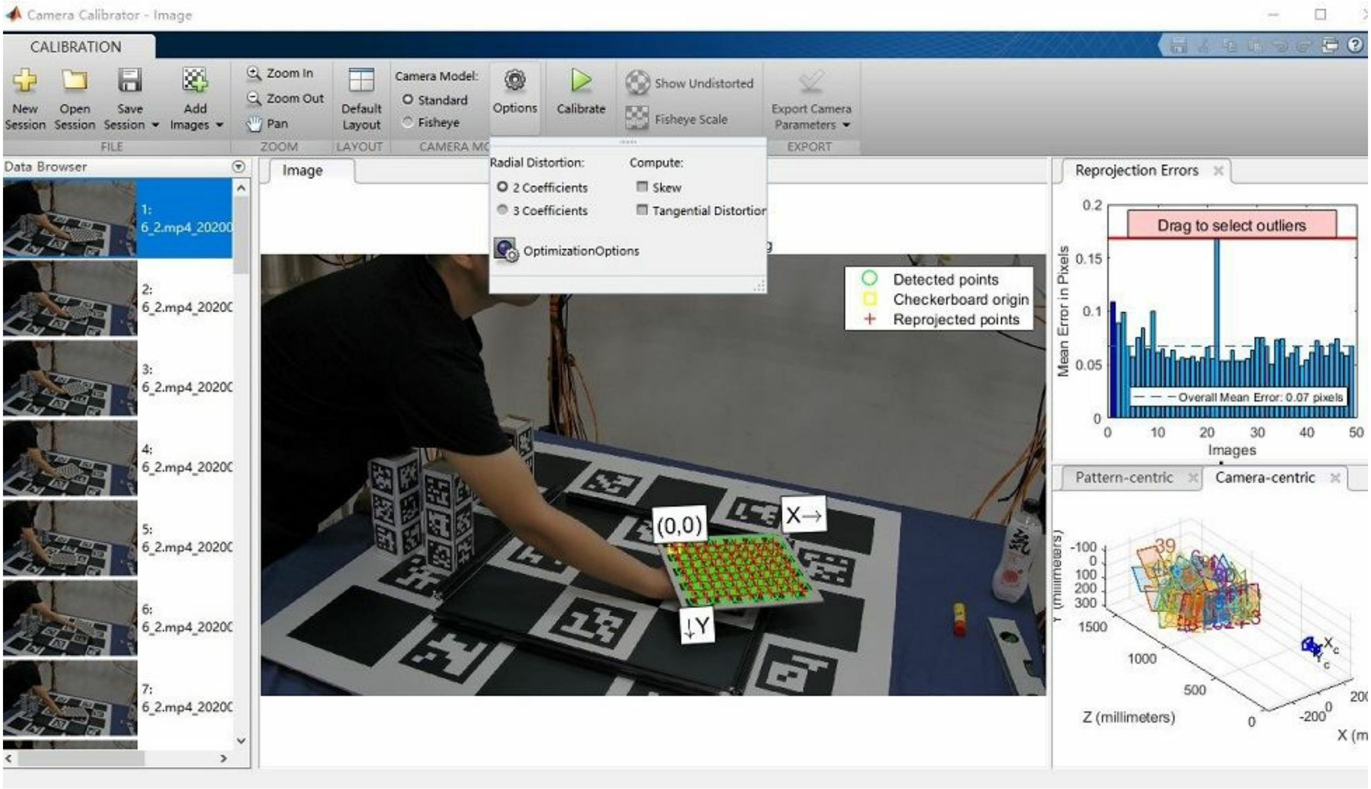
如何基于matlab相机标定导出xml文件

1 参数选择 径向畸变3个参数还是两个参数

默认两个参数

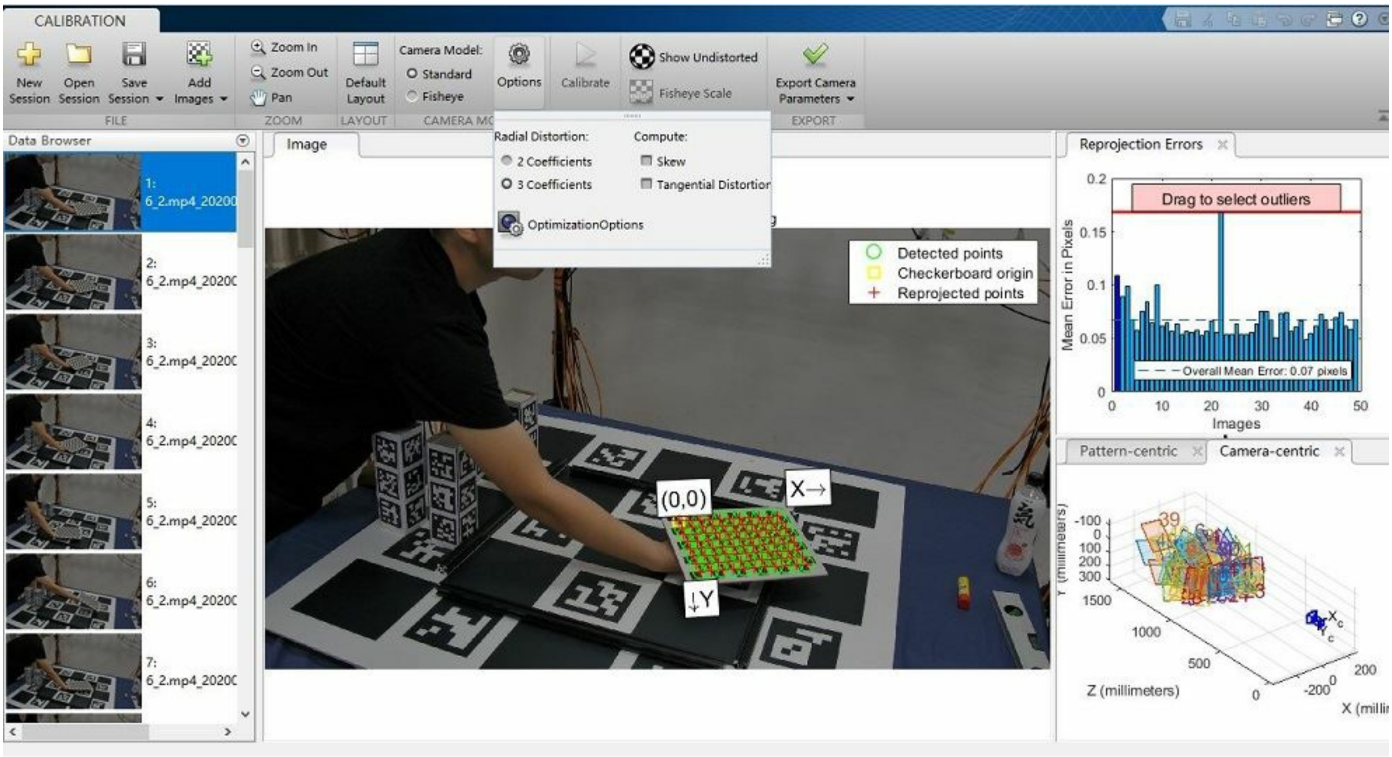


Camera Model: ● Standard ○ Fisheye

Radial Distortion: ● 2 Coefficients ○ 3 Coefficients

Compute: □ Skew □ Tangential Distortion

如果是三个参数

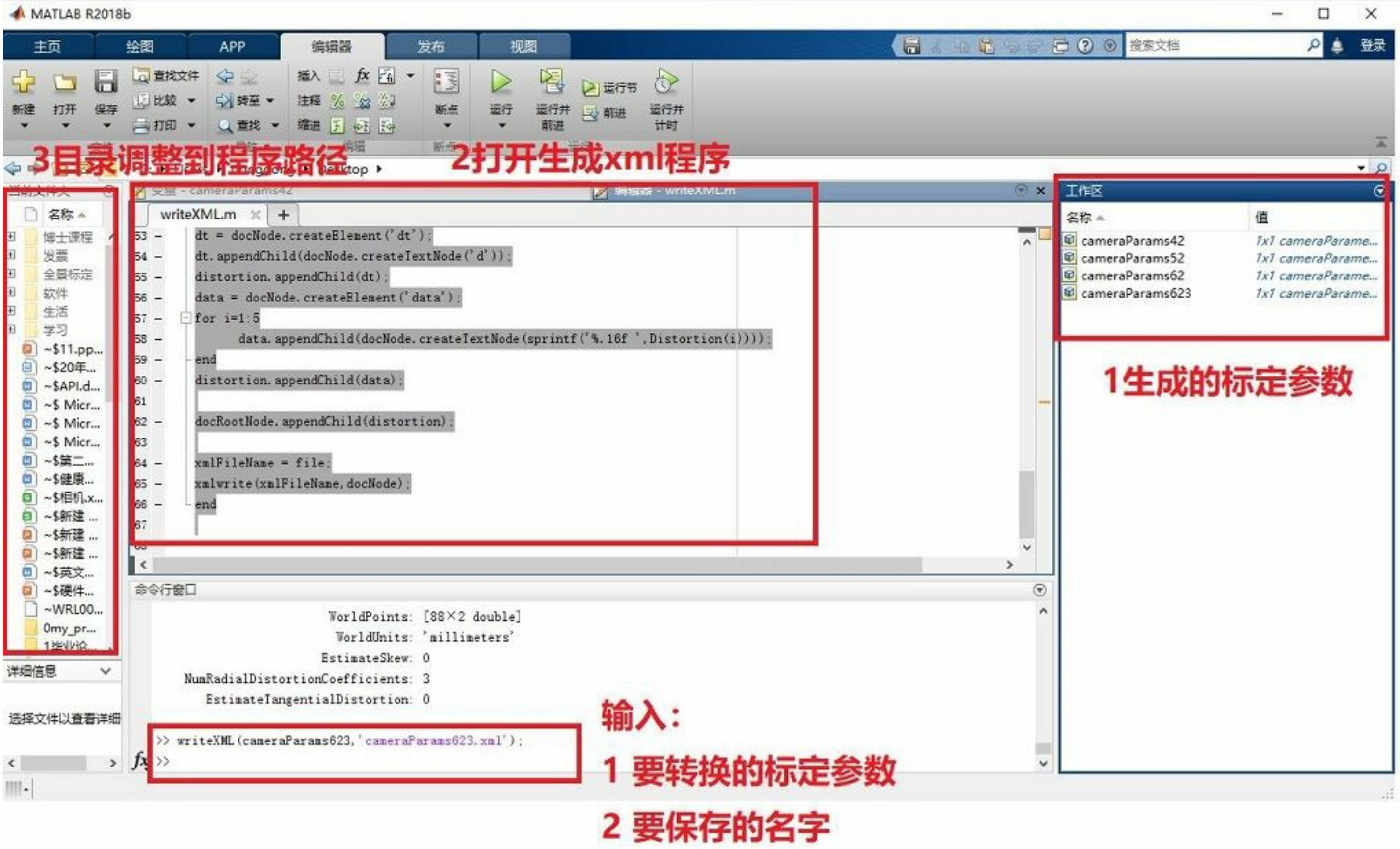


Camera Model: ● Standard ○ Fisheye

Radial Distortion: ○ 2 Coefficients ● 3 Coefficients

Compute: □ Skew □ Tangential Distortion

2准备转化生成结果



3目录调整到程序路径

2打开生成xml程序

writeXML.m

```
dt = docNode.createElement('dt');
dt.appendChild(docNode.createTextNode('d'));
distortion.appendChild(dt);
data = docNode.createElement('data');
for i=1:5
        data.appendChild(docNode.createTextNode(sprintf('%.16f ',Distortion(i))));
end
distortion.appendChild(data);

docRootNode.appendChild(distortion);

xmlFileName = file;
xmlwrite(xmlFileName,docNode);
end
```

名称	值
cameraParams42	1x1 cameraParame...
cameraParams52	1x1 cameraParame...
cameraParams62	1x1 cameraParame...
cameraParams623	1x1 cameraParame...

1生成的标定参数

```
                     WorldPoints: [88×2 double]
                      WorldUnits: 'millimeters'
                    EstimateSkew: 0
NumRadialDistortionCoefficients: 3
      EstimateTangentialDistortion: 0

>> writeXML(cameraParams623,'cameraParams623.xml');
>>
```

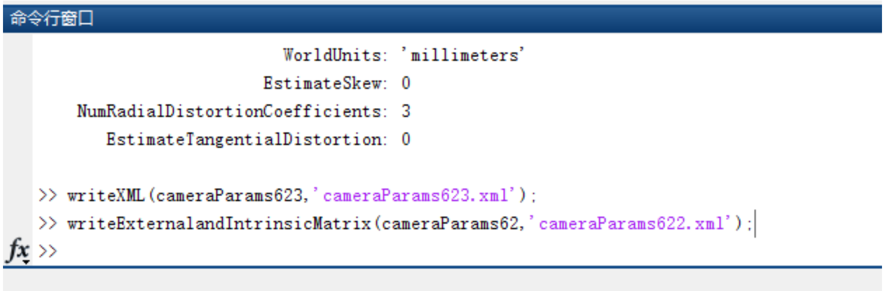
输入：

1 要转换的标定参数

2 要保存的名字

二参数的转化代码

writeExternalandIntrinsicMatrix(cameraParams62,'cameraParams622.xml');



```
                      WorldUnits: 'millimeters'
                    EstimateSkew: 0
NumRadialDistortionCoefficients: 3
      EstimateTangentialDistortion: 0

>> writeXML(cameraParams623,'cameraParams623.xml');
>> writeExternalandIntrinsicMatrix(cameraParams62,'cameraParams622.xml');
>>
```

function writeExternalandIntrinsicMatrix(cameraParams,file)
%writeXML(cameraParams,file)

docNode = com.mathworks.xml.XMLUtils.createDocument('opencv_storage');
docRootNode = docNode.getDocumentElement;
IntrinsicMatrix = ((cameraParams.IntrinsicMatrix)');

TangentialDistortion =cameraParams.TangentialDistortion;
%Distortion = [cameraParams.RadialDistortion(1:2),TangentialDistortion, cameraParams.RadialDistortion(3)];
Distortion = [cameraParams.RadialDistortion(1:2),TangentialDistortion,0];
FocalLength = cameraParams.FocalLength;
camera_matrix = docNode.createElement('IntrinsicCam'); %锟斤拷锟斤拷mat锟节碉拷
camera_matrix.setAttribute('type_id','opencv-matrix'); %锟斤拷锟斤拷mat锟节碉拷锟斤拷锟斤拷
rows = docNode.createElement('rows'); %锟斤拷锟斤拷锟叫节碉拷
rows.appendChild(docNode.createTextNode(sprintf('%d',3))); %锟斤拷锟斤拷锟侥憋拷锟节点，锟斤拷锟斤拷为锟叫碉拷锟接节碉拷
camera_matrix.appendChild(rows); %锟斤拷锟叫节碉拷锟斤拷为mat锟接节碉拷

cols = docNode.createElement('cols');
cols.appendChild(docNode.createTextNode(sprintf('%d',3)));

```
camera_matrix.appendChild(cols);

dt = docNode.createElement('dt');
dt.appendChild(docNode.createTextNode('d'));
camera_matrix.appendChild(dt);

data = docNode.createElement('data');
for i=1:3
for j=1:3
data.appendChild(docNode.createTextNode(sprintf('%.16f ',IntrinsicMatrix(i,j))));
end
data.appendChild(docNode.createTextNode(sprintf('\n')));
end
camera_matrix.appendChild(data);
docRootNode.appendChild(camera_matrix);

distortion = docNode.createElement('DistortionCam');
distortion.setAttribute('type_id','opencv-matrix');
rows = docNode.createElement('rows');
rows.appendChild(docNode.createTextNode(sprintf('%d',1)));
distortion.appendChild(rows);

cols = docNode.createElement('cols');
cols.appendChild(docNode.createTextNode(sprintf('%d',5)));
distortion.appendChild(cols);

dt = docNode.createElement('dt');
dt.appendChild(docNode.createTextNode('d'));
distortion.appendChild(dt);
data = docNode.createElement('data');
for i=1:5
data.appendChild(docNode.createTextNode(sprintf('%.16f ',Distortion(i))));
end
distortion.appendChild(data);
docRootNode.appendChild(distortion);

focalLength = docNode.createElement('FocalLength');
focalLength.setAttribute('type_id','opencv-matrix');
rows = docNode.createElement('rows');
rows.appendChild(docNode.createTextNode(sprintf('%d',1)));
focalLength.appendChild(rows);
cols = docNode.createElement('cols');
cols.appendChild(docNode.createTextNode(sprintf('%d',1)));
focalLength.appendChild(cols);
dt = docNode.createElement('dt');
dt.appendChild(docNode.createTextNode('d'));
focalLength.appendChild(dt);
data = docNode.createElement('data');
for i=1:1
data.appendChild(docNode.createTextNode(sprintf('%.16f ',FocalLength(i))));
end
focalLength.appendChild(data);
docRootNode.appendChild(focalLength);

% distortion = docNode.createElement('Pmatrix');
% distortion.setAttribute('type_id','opencv-matrix');
% rows = docNode.createElement('rows');
% rows.appendChild(docNode.createTextNode(sprintf('%d',1)));
% distortion.appendChild(rows);
%
% cols = docNode.createElement('cols');
% cols.appendChild(docNode.createTextNode(sprintf('%d',4)));
% distortion.appendChild(cols);
%
% dt = docNode.createElement('dt');
% dt.appendChild(docNode.createTextNode('d'));
% distortion.appendChild(dt);
% data = docNode.createElement('data');
% for i=1:4
% data.appendChild(docNode.createTextNode(sprintf('%.16f ',Distortion(i))));
% end
% distortion.appendChild(data);
% docRootNode.appendChild(distortion);

xmlFileName = file;
xmlwrite(xmlFileName,docNode);
end
```

二参数的保存结果

```
<?xml version="1.0" encoding="utf-8"?>
<opencv_storage>
<IntrinsicCam type_id="opencv-matrix">
<rows>3</rows>
<cols>3</cols>
<dt>d</dt>
<data>1558.5669994681102253 0.0000000000000000 821.5211092415044050
0.0000000000000000 1557.8077127262038175 460.9748043702705331
0.0000000000000000 0.0000000000000000 1.0000000000000000
</data>
</IntrinsicCam>
<DistortionCam type_id="opencv-matrix">
<rows>1</rows>
<cols>5</cols>
<dt>d</dt>
<data>-0.1873006682834817 0.0171597428423078 0.0000000000000000 0.0000000000000000 0.0000000000000000 </data>
</DistortionCam>
<FocalLength type_id="opencv-matrix">
<rows>1</rows>
<cols>1</cols>
<dt>d</dt>
<data>1558.5669994681102253 </data>
</FocalLength>
</opencv_storage>
```

三参数的转化代码

```
function writeXML(cameraParams,file)
%writeXML(cameraParams,file)
%功能：将相机校正的参数保存为xml文件
%输入：
%cameraParams：相机校正数据结构
%file：xml文件名
%说明在xml文件是由一层层的节点组成的。
%首先创建父节点 fatherNode，
%然后创建子节点 childNode=docNode.createElement(childNodeName)，
%再将子节点添加到父节点 fatherNode.appendChild(childNode)
docNode = com.mathworks.xml.XMLUtils.createDocument('opencv_storage'); %创建xml文件对象
docRootNode = docNode.getDocumentElement; %获取根节点

IntrinsicMatrix = (cameraParams.IntrinsicMatrix)'; %相机内参矩阵
RadialDistortion = cameraParams.RadialDistortion; %相机径向畸变参数向量1*3
TangentialDistortion =cameraParams.TangentialDistortion; %相机切向畸变向量1*2
Distortion = [RadialDistortion(1:2),TangentialDistortion,RadialDistortion(3)]; %构成opencv中的畸变系数向量[k1,k2,p1,p2,k3]

camera_matrix = docNode.createElement('camera-matrix'); %创建mat节点
camera_matrix.setAttribute('type_id','opencv-matrix'); %设置mat节点属性
rows = docNode.createElement('rows'); %创建行节点
rows.appendChild(docNode.createTextNode(sprintf('%d',3))); %创建文本节点，并作为行的子节点
camera_matrix.appendChild(rows); %将行节点作为mat子节点

cols = docNode.createElement('cols');
cols.appendChild(docNode.createTextNode(sprintf('%d',3)));
camera_matrix.appendChild(cols);

dt = docNode.createElement('dt');
dt.appendChild(docNode.createTextNode('d'));
camera_matrix.appendChild(dt);

data = docNode.createElement('data');
for i=1:3
for j=1:3
data.appendChild(docNode.createTextNode(sprintf('%.16f ',IntrinsicMatrix(i,j))));
end
data.appendChild(docNode.createTextNode(sprintf('\n')));
end
camera_matrix.appendChild(data);
docRootNode.appendChild(camera_matrix);

distortion = docNode.createElement('distortion');
distortion.setAttribute('type_id','opencv-matrix');
rows = docNode.createElement('rows');
rows.appendChild(docNode.createTextNode(sprintf('%d',5)));
distortion.appendChild(rows);

cols = docNode.createElement('cols');
cols.appendChild(docNode.createTextNode(sprintf('%d',1)));
distortion.appendChild(cols);

dt = docNode.createElement('dt');
dt.appendChild(docNode.createTextNode('d'));
distortion.appendChild(dt);
data = docNode.createElement('data');
for i=1:5
```

```
data.appendChild(docNode.createTextNode(sprintf('%.16f ',Distortion(i))));
end
distortion.appendChild(data);

docRootNode.appendChild(distortion);

xmlFileName = file;
xmlwrite(xmlFileName,docNode);
end
```

三参数的转化保存结果

```
<?xml version="1.0" encoding="utf-8"?>
<opencv_storage>
<camera-matrix type_id="opencv-matrix">
<rows>3</rows>
<cols>3</cols>
<dt>d</dt>
<data>1558.6100144620272658 0.0000000000000000 821.6453269280840459
0.0000000000000000 1557.8120286433929778 460.8682816753835141
0.0000000000000000 0.0000000000000000 1.0000000000000000
</data>
</camera-matrix>
<distortion type_id="opencv-matrix">
<rows>5</rows>
<cols>1</cols>
<dt>d</dt>
<data>-0.1840928673709393 -0.0328189923757994 0.0000000000000000 0.0000000000000000 0.2205440258401062 </data>
</distortion>
</opencv_storage>
```

您可能感兴趣的文章:MATLAB数学建模之画图汇总将pycharm配置为matlab或者spyder的用法说明基于python实现matlab filter函数过程详解matlab、python中矩阵的互相导入导出方式Matlab及Java实现小时钟效果使用matlab 判断两个矩阵是否相等的实例对Matlab中共轭、转置和共轭装置的区别说明matlab中二维插值函数interp2的使用详解matlab画三维图像的示例代码(附demo)