https://stackoverflow.com/questions/13760110/using-opencv-fitellipse-for-circle-fitting

für eine einfachere Auswahl der geeignete Ellipse, messen wir manuell den Radius

fitellipse 算法对找圆来讲没什么不能用的，因为算的是global minimum误差

OpenCV的优点：几行代码就可以实现一个功能，因为算法都集成在自带的函数里了，。。。等等等等再找找别的优点

Timer.cancel: 滚到三分之一位置

https://medium.com/@nullbyte.in/ros2-from-the-ground-up-part-7-achieving-reliable-communication-in-ros-2-with-qos-configurations-83c534c3aff5

现实里旋转选择0.00008的原因是因为考虑到greifer的那根线的beweglichkeit? 或者长度不够。转多了会断。 但是根本来讲应该转的更多或者最好转一周，为了更好的interpolieren?

Python version :3.10.6??学校的是几阿

Starting with Python 3.1, Python (on most systems) is now able to choose the shortest of these and simply display 0.1.

有要求一定要参考多少本书吗

with open https://blog.csdn.net/weixin\_39613951/article/details/110785735

连续多个open

误差： 考虑数据向量是frame到frame的，要加上visual的高度吗，再与原来的作对比，看看原来的gripper会顶到rot plate吗

图像坐标转显示坐标： ！！！

https://juejin.cn/s/opencv%20%E5%9B%BE%E5%83%8F%E5%9D%90%E6%A0%87%E8%BD%AC%E6%8D%A2%E7%8E%B0%E5%AE%9E%E5%9D%90%E6%A0%87

300dpi是多少像素/英寸

http://www.bestnw.com/article\_read\_127.html

照相机标定。。。

返回的contours是整数吗

看看别的编程类的论文

问老师：

之前的是urdf现在是xacro， 用xacro来写没问题把

wir, unser 的使用

link frame ist entschied in cad?

最后讲论文的时候要在那台电脑上安装不同的python库

写的时候直接看 + 找外文链接

在本论文中，由于远心镜头的使用，实际上检测到的是一个圆，但本文用了椭圆检测，所以可能存在误差，但本文没有继续探究圆的检测效果。

yaml warning:

https://github.com/yaml/pyyaml/wiki/PyYAML-yaml.load(input)-Deprecation

yaml:

[https://python.land/data-processing/python-yaml#Writing\_or\_dumping\_YAML\_to\_a\_file](https://python.land/data-processing/python-yaml" \l "Writing_or_dumping_YAML_to_a_file)

[https://realpython.com/python-yaml/#tweak-the-formatting-with-optional-parameters](https://realpython.com/python-yaml/" \l "tweak-the-formatting-with-optional-parameters)

暂时没用到：

https://pyyaml.org/wiki/PyYAMLDocumentation

ruamel.yaml:

https://yaml.readthedocs.io/en/latest/basicuse.html

Will the function fitellipse() automatically remove points with large errors?

https://cseweb.ucsd.edu//classes/sp02/cse252/foerstner/foerstner.pdf

您可以使用 OpenCV 的函数 cv2.getRectSubPix() 来获取图像中某个浮点坐标位置的像素值。

图像坐标，像素坐标：

https://qiy.net/2020/06/09/IAP-Cam-calibration/

亚像素：

https://www.sohu.com/a/478650708\_121119003

先\*1000 再/1000：

https://blog.csdn.net/weixin\_41616991/article/details/126158231

opencv fitellise函数会自动剔除误差大的点吗?

fit points to circles:

https://juejin.cn/s/opencv%20fit%20circle%20to%20points

what are contours? :

https://docs.opencv.org/3.1.0/d4/d73/tutorial\_py\_contours\_begin.html

拟合圆： https://blog.csdn.net/yx123919804/article/details/108745727

可以查看velocity at that point ，但是也可以看ros2 interface show trajectory\_msgs/msg/JointTrajectory

http://wiki.ros.org/robot\_mechanism\_controllers/JointTrajectoryActionController

节点的初始化及退出，但是已经可以rclpy.destory\_node了

https://www.cnblogs.com/HaoQChen/p/11048616.html

自动对焦：

https://github.com/huchunxu/ros\_exploring/blob/master/robot\_marm/marm\_planning/scripts/trajectory\_demo.py

https://answers.ros.org/question/50610/the-meaning-of-velocities-accelerations-and-time\_from\_start-in-jointtrajectorypointmsg/

https://www.cnblogs.com/feifanrensheng/p/15579537.html

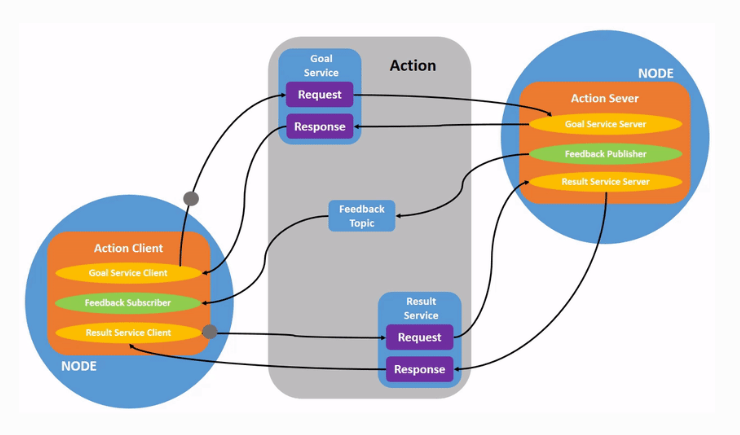
https://www.programcreek.com/python/example/123227/trajectory\_msgs.msg.JointTrajectoryPoint

In dieser Arbeit wird (Trajektorie) verwendet, um den Roboter zu steuern. 和forwaord position对比的优劣？In ROS 2 gibt es insgesamt drei Typen von Interfaces um die Kommunikation zwischen Nodes zu erreichen: Topic, Service und Action. “Topic” ist eine Hauptmethode, mit denen die Daten zwischen den Nodes übertragen werden. Sie verwendet ein “Publisher-Subscriber”-Modell um einen Datenfluss zu erreichen. Die Nodes mit gleichen Namen der “Topics” kommunizieren miteinander. Eine Node kann die Daten zu einer beliebigen Anzahl von “Topics” publizieren und auch eine beliebige Anzahl von “Topics” abonnieren. “Service” basieren auf einem “Call-and-Response”-Modell. Anders als einen kontinuierlichen Datenfluss von der “Topic”-Kommunikation, stellt die “Service”-Kommunikation die Daten nur dann bereit, wenn sie ausdrücklich von einem Client aufgerufen werden. Es kann viele Clients existieren , die einen selben Service benutzten, aber für jeder Service kann es nur ein einzelner Server existieren. “Action” ist für lang laufende Aufgaben gedacht. Sie bestehen aus drei Teilen: einem Ziel, Feedbacks und einem Ergebnis. Die “Action”-Kommunikation benutzt ein “Client-Server”-Modell, das auf “Topic” und “Service” basiert. Ein „Action”-Client sendet ein Ziel an einen „Action”-Server, der das Ziel bestätigt und einen Feedbackstrom und ein Ergebnis zurückgibt. Eine Haupteigenschaft der “Action”-Kommunikation ist, dass sie abgebrochen während der Ausführung werden kann. Außerdem kann die “Action”-Kommunikation stetige Feedbacks während der Ausführung neben der einzelnen endlichen Antwort des Ergebnisses geben.

Um die Trajektorien an den Controller zu senden kann man zwei Mechanismen davon benutzten: “Action Interface” oder “Topic Interface”. “Topics” werden normalerweise für kontinuierliche Datenströme wie Sensordaten, Roboterstatus usw. verwendet, wobei der Datenfluss unidirektional ist. “Actions” werden meistens für diskretes Verhalten verwendet, das zur Steuerung eines Roboters oder zum Laufen über einen längeren Zeitraum. Während der Ausführung der “Actions” werden die Feedbacks gegeben. Im unseren Fall wird ein Präzisionsmontageroboter gesteuert und die Feedbacks während dem Prozess sind erwünscht, deswegen wird in dieser Arbeit die Steurung des Roboter durch die Kommunikation mit “Action” erreicht.

Unten ist eine grafische Darstellung der “Action”-Kommunikation.

Der “Action Client” fordert ein Request an den “Action Server” an und der “Action Server” sendet eine Antwort zurück, um es aufzuweisen, ob das Request akzeptiert wird. Wenn das Request erfolgreich akzeptiert wird, wird das Request bzw. die “Action” ausgeführt. Im Anschluss an die “Request”-Anforderung stellt der Client eine asynchrone Anforderung für das Ergebnis. Während das Ziel ausführt wird, gibt der “Action Server” die kontinuierlichen Feedbacks an den “Action Client”. Und letztendlich gibt der “Action Server” ein einmaliges Feedback, um zu deuten ob die “Action” erfolgreich beendet wird. 然后也许可以结合一下我们的程序，带进去讲一下



控制greifer运行到一个位置，是有一定偏差的，很微小，可以到时候看看是小数点后几位，可能影响到最后的精度 allow\_partial\_joints\_goal?

讲trajectory代码的时候继续用这个链接：

http://wiki.ros.org/joint\_trajectory\_controller

Trajectories are specified as a set of waypoints to be reached at specific time instants, 但是我们这里是简单的运动，所以就不刻画多个点了。

http://wiki.ros.org/joint\_trajectory\_controller

Topic vs action:

https://docs.ros.org/en/humble/How-To-Guides/Topics-Services-Actions.html

trajectory看这个：

<http://design.ros2.org/articles/actions.html> 这个是档案

http://wiki.ros.org/joint\_trajectory\_controller

http://wiki.ros.org/pr2\_controllers/Tutorials/Moving%20the%20Head

参考文献可以看这个

http://wiki.ros.org/ros2\_controllers/Tutorials

action也许可以看一下：

https://github.com/ros2/examples/blob/humble/rclpy/actions/minimal\_action\_client/examples\_rclpy\_minimal\_action\_client/client\_cancel.py

trajectory例子：

https://github.com/huchunxu/ros\_exploring/blob/master/robot\_marm/marm\_planning/scripts/trajectory\_demo.py

trajectory：

http://wiki.ros.org/Robots/TIAGo/Tutorials/trajectory\_controller

报错：eg：path error：

http://docs.ros.org/en/noetic/api/control\_msgs/html/action/FollowJointTrajectory.html

clock不需要调，直接用服务端的：

http://wiki.ros.org/Clock

clock：

https://design.ros2.org/articles/clock\_and\_time.html

周五：

two ways...

http://wiki.ros.org/joint\_trajectory\_controller#ROS\_API

http://wiki.ros.org/ros2\_controllers/Tutorials

vel,acc = 0 , stabil:

https://answers.ros.org/question/50610/the-meaning-of-velocities-accelerations-and-time\_from\_start-in-jointtrajectorypointmsg/

背景：（其他的也可以写背景）

https://zhuanlan.zhihu.com/p/467256942

jointpoint单位：

https://docs.ros2.org/foxy/api/trajectory\_msgs/msg/JointTrajectoryPoint.html

python opencv找椭圆

https://zhuanlan.zhihu.com/p/603629524

https://blog.csdn.net/Useless\_csdn/article/details/102616576

https://zhuanlan.zhihu.com/p/438542999

https://zhuanlan.zhihu.com/p/603629524

https://blog.csdn.net/Useless\_csdn/article/details/102616576

椭圆拟合3种方法：

https://blog.csdn.net/weixin\_41616991/article/details/126158231

Mode: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

https://docs.opencv.org/3.4/d3/dc0/group\_\_imgproc\_\_shape.html

看获得的图片是几通道的：

https://stackoverflow.com/questions/19062875/how-to-get-the-number-of-channels-from-an-image-in-opencv-2

https://numpy.org/doc/stable/reference/generated/numpy.ndarray.ndim.html#numpy-ndarray-ndim

注意调参：

https://bbs.huaweicloud.com/blogs/251166

https://blog.51cto.com/u\_15362367/3888720

https://blog.csdn.net/HaoZiHuang/article/details/106437761

circles = cv2.HoughCircles(

edges, cv2.HOUGH\_GRADIENT, 1, 50, param1=100, param2=30, minRadius=5, maxRadius=100)

会 return none

https://stackoverflow.com/questions/31140386/python-opencv-hough-circles-returns-none

https://www.google.com/search?q=cv2.HoughCircles+return+None&client=ubuntu-sn&hs=MAz&channel=fs&ei=xfxUZM7ANdCIxc8Puoi58As&ved=0ahUKEwiOt4nwnN7-AhVQRPEDHTpEDr4Q4dUDCA8&oq=cv2.HoughCircles+return+None&gs\_lcp=Cgxnd3Mtd2l6LXNlcnAQDDIGCAAQFhAeOhMIABCPARDqAhC0AhCMAxDlAhgBOhMILhCPARDqAhC0AhCMAxDlAhgBSgQIQRgAUJoKWJoKYK0RaAFwAXgAgAE-iAE-kgEBMZgBAKABAaABArABCsABAdoBBAgBGAo&sclient=gws-wiz-serp

https://stackoverflow.com/questions/50568668/understanding-houghcircles-in-python-opencv-cv2

if circles is not None:。。。

cv2.画圆要是int：https://docs.opencv.org/2.4/modules/core/doc/drawing\_functions.html#circle

op. py: waitkey(1)会显示黑的图片

图片坐标：https://blog.csdn.net/oqqENvY12/article/details/71933651

https://blog.csdn.net/lz0499/article/details/80978433

findcontur 可能因为原图不圆导致质心不是圆心，但是可以调教优化原图，但尽管如此如果原图的圆存在缺陷就不行了。（例如有灰尘遮挡） 有对比截图

cv2.moments: https://xie.infoq.cn/article/c61a7c1d9547813a319042879

fincontur返回值：在看别的

https://juejin.cn/s/cv2.findcontours%20%E8%BF%94%E5%9B%9E%E5%80%BC

使用Python的OpenCV获取二值图像轮廓及中心点坐标的代码：chat机器人

经过滤波之后的casd 坐标与asd不太一样 。。那用或不用别的呢，有啥影响

\*\*\* 主要是如何评判哪个结果好啊 不知道 \*\*\*

urdf的修改 摄像头？ teil\_change里1k 2k是直接改数据的， 想一想阿 想不起来了

in opencv gibt es 圆检测的函数， but我们不用， 用轮廓检测

放不放大的图片 点坐标不一样

moments 解释:

https://docs.opencv.org/4.x/dd/d49/tutorial\_py\_contour\_features.html

https://en.wikipedia.org/wiki/Image\_moment

相同的图片 houghcirlce和 findcontur作对比： 精度应该差不多，但是 findcontur更稳定

findcontur会修改图像输入.. https://www.cnblogs.com/zyly/p/9327425.html

3.2以后就不会了..

！！！ https://blog.csdn.net/u012566751/article/details/54379290

以及下面的下载文件

高精度定位

这个参数还行：

https://blog.csdn.net/kingroc/article/details/86167222

根据要求知道要修改的是什么标签了，但是在编写程序之前，我们需要先探究一下改动这些属性，对其他urdf标签有什么影响。以及相邻的link joint

http://wiki.ros.org/Papers/TePRA2013\_Foote?action=AttachFile&do=get&target=TePRA2013\_Foote.pdf

前景motivation: studienarbeit纸上allgeimeins.

Die Aufgabe ist…… Das erste zu kalibrierte Batuteil ist 1K…

‘frame’… + Bild in ros/xml

Ein URDF(xacro) Dateil in ros ist xml sepzif… http://wiki.ros.org/urdf/XML

(Es besetht aus…) Die in (unsere) xacro Dateil verwendete(n) Elemente sind: (‘’)robot, link, joint, gazebo. Die Elemente link, joint, gazebo sind gekapselt in dem Element robot. Das Element link beschreibt die Eigenschaften der Bauteile eines Roboters. Das Element gazebo (ist URDF robot description format), und beschreibt die Performance im Simulator Gazebo, deswegen werden diese zwei Elemente nicht in Betracht gezogen.

(Lassen wir diskutieren..)Das Element joint ‘(describes the kinematics and dynamics of the joint and also specifies the safety limits of the joint. )’. Ein ‘joint’ Element hat zwei Attribute: name und type. ‘name’ gibt einen eindeutigen Name von joint an und ‘type’ gibt den Bewegungstyp an. Es besteht aus den folgenden untergeordneten Elenmenten: origin, parent, child, axis, calibration , dynamics, limit, mimic und safety\_controller. Die zu unserem Roboter relevant sind: origin, parent, child, axis und limit. In ‘parent’ Element wird der name des übergeordneten Links(parent link) geschrieben. In ‘child’ Element wird der name des untergeordneten Links(child link) geschrieben. Das ‘origin’ Element beschreibt die Transformation vom parent (Link) zum child Link. Es enthält zwei Attribute: xyz und rpy. ‘xyz’ repräsentiert den x-, y-, z-Offset und rpy repräsentiert die Rotation um die x-, y-, z-Achse. 'axis' gibt das Bezugssystem an. ‘limit’ legt die Obergrenze, die Untergrenze, die maximale Arbeit und die maximale Geschwindigkeit eines ‘joint’ fest.

In dieser Aufgabe wird ‘frame’ von den Bauteilen angepasst, dementsprechend ist der zu veränderte Teil das ‘origin’ Element. Werden die Elemente in einer URDF Dateil betrachtet, kann man daraus finden, dass die Elemente in ‘joint’ beeinflusst die Element in ‘link’ nicht(….) Deswegen sind nur die entsprechende Teile zu modifizieren, und die andere relevante Elemente nicht betrachten zu können.

Es ist ungünstig wenn wir TF verwenden. (urdf to graphiz darstellen)

(Zuerst) wird der 1K\_Dispenser betrachtet. Er ist ein ‘link’, der gleiten aber sich nicht drehen kann.

Dann wird … betrachtet

Ein urdf Dateil enthält die folgend

找椭圆里用a = []

b = 0

c = 0求平均的原因： 找到的2个或可能多个的椭圆非常相似，角度也基本一样，所以求个平均。 如果找到的椭圆相差有点大，或者说角度相差大(意思可能不是同一个椭圆了)，那就不要求平均了。所以这里还是求一下。

让椭圆转的话，一方面要尽可能的让他到正中间，另一方面不能让他转出镜头，但是可能会与中心点偏差很小，所以应该不会转出去。

局限性： 图像处理获取的图片一开始便由整数像素表示，所以不知道一开始的轮廓的具体亚像素坐标。后面的fitellipse基于整数的像素计算出小数中心等参数是不准确的。但是后面用二分法尽量准确。

Contoller 的位置控制认为joint到了也是微小误差的一个叠加。

反思： 所有的圆心组成的圆确定后，找到中心(rot\_plate) 移动 rot\_plate 到中心，如果转动之后loch中心仍在一个圆上，则准确。

某些小程序，小代码的位置可能放的不对，比如放在写的别的函数里了。

有的代码可能不是最优化的，可能有更好的解或者更简洁的代码。

某些代码的使用不恰当，比如 global，self.rr 之类的定义

用parameter作条件有些投机取巧，threading之类的可能更有意义

set\_parameter 也在回调函数里，会一直执行，有点不 sinnvoll

程序也许可以把条件设置成自动多次执行的，这样可以直接多次执行，然后从中取对的值

程序可能要多执行几次，(但是每次结果不一样可能是因为fitellipse及fitcircle本身的检测结果不一样)，来确保一个准确的结果。

fit\_ellpise其实是找最小外接矩形，所以会受加工开口影响

canny()我们直接用的是里面的sobel，没有用canny的另一种输入参数dx, dy的函数：sobel：https://docs.opencv.org/4.x/d2/d2c/tutorial\_sobel\_derivatives.html

这有一点不准，scharr更准，但是因为我们精度要求不那么高，也许并不需要。

本文前提是认为anlage的所有其他部件是准确的，比如unterkamera。如果把这些也考虑进去，也是一个误差。

图像不能有那个加工缺损。因为我们加上了灯光，所以那个缺损被照模糊了，所以在检测的时候椭圆中偏的也许没有那么明显。这是用findcontour()的一个缺陷。👇

fitellipse()找的是全局最小值?。。

再看看fitellpise论文

讲的时候把那个没有打灯光的照片加上去，说明一下实际是存在小缺口的。但是这个很小，我们用个大的测试一下：然后用那个不规则椭圆的图检测的效果放进论文。

展望： 编程不太行，编程思想可能不太对

本文中的椭圆检测方法及精度不够，所以有误差，另外还有本文未涉及到的校准误差的做法没做，也会导致一些误差吧

ros为什么用四元数

最小二乘法拟合椭圆： https://www.cnblogs.com/cv-pr/p/4625122.html

先乘1000再除1000： https://blog.csdn.net/weixin\_41616991/article/details/126158231

找椭圆中的angle：

https://datascience.stackexchange.com/questions/85064/where-is-the-rotated-angle-actually-located-in-fitellipse-method

畸变比用 a, b 来比较，和别的一样这样取平均计算也简单点 (搞个圆被拉成椭圆的图？，然后说直接用b计算比较简单) 关于最后用算出的中心点乘以这个比例，也许可以在一开始就用那些很多的点计算会更准一些，但是这样太麻烦了(komplex, von weniger Bedeutung)，所以直接用最后一个点计算简单。（我们是用检测到的所有中心来计算b的畸变的，不直接用最后算出来的椭圆，因为用这些数据量也不大，并且可能更准一点。。）

同时，也不能保证检测到的椭圆的b就是我们要的b，就是说测到的b不一定垂直图中的中点。

画个图： 成像wie凸透镜，往右边移动的话上下没有太变，左右边缘射过来的光线变窄了。

论文算法的讲解从括号内向外一步一步讲

um die kreise zu detektieren, wird in dieser Arbeit die Funktion ellipse() von OpenCV. Wenn 椭圆的长短轴足够接近的话，我们就认为它是一个圆。

fit\_ellipse:

In Python ist der Gleitkommazahlentyp ‘float’ eine 64-Bit-Darstellung einer Gleitkommazahl mit doppelter Genauigkeit, was bedeutet, dass der Datentyp ‘float64’ ist. Deswegen können die Punkte nicht direkt verwendet werden, da der Datentyp jetzt float64 ist. Wenn wir so machen, es wird:

cv2.error: OpenCV(4.7.0) /io/opencv/modules/imgproc/src/shapedescr.cpp:355: error: (-215:Assertion failed) n >= 0 && (depth == CV\_32F || depth == CV\_32S) in function 'fitEllipseNoDirect' , (图像深度 image depth) was bedeutet, dass der Typ CV\_32F oder CV\_32S sein muss. CV\_32F und CV\_32S sind zwei Typen der Zahl in OpenCV, die die‚float32‘ und ‚int32‘ in Pthon repräsentieren.因此要先转换一下。ist… die vorher in diesem Programm verwendet ist schon std::vector …

+链接-

https://note.nkmk.me/en/python-sys-float-info-max-min/

https://gist.github.com/yangcha/38f2fa630e223a8546f9b48ebbb3e61a

# 类型错误问题看下这个： https://codeantenna.com/a/yuQYvN9qTT

# https://docs.ros.org/en/humble/Concepts/About-ROS-Interfaces.html

fitellipse: 由于精度要求不那么高，三种fitellipse算法的进一步探讨在本文不再进行

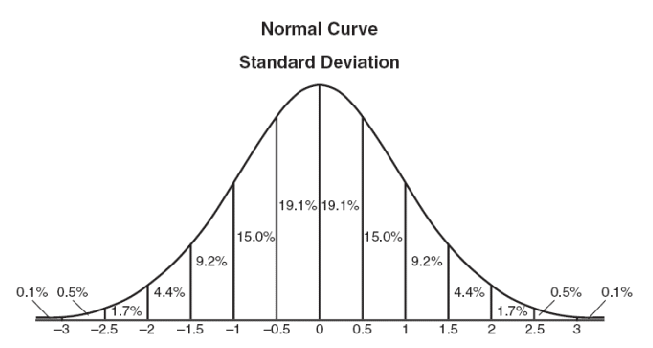
time.sleep(2) 一方面保证了第一个椭圆的检测，另一方面给了检测kamera的时间，然后再self.get\_logger().info('Waiting for kamera...')

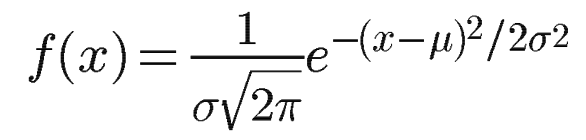
Gauß-Filter: 最后可能不用高斯的原因：因为canny()的双阈值控制，那些孤立的点不会被算作边缘，这也正是canny的优点。

啊？。。。(更低频或者有时在相同的位置检测不出来)： 所以说高斯让图片平滑后，让图片更难检测出轮廓了？

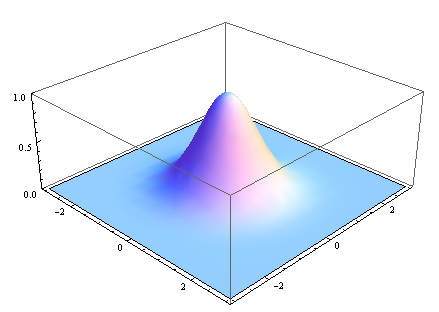
In der Bildverarbeitung werden Gauß-filter häufig verwendet, um Bildinhalte zu glätten oder weicher zu machen. Damit gehen die kleinere Strukturen verloren und bleiben die gröbere Strukturen erhalten, und kann es eine Verminderung des Bildrauschens erreicht werden. Der Gauß-filter kann das Rauschen mit der Normalverteilung(Gauß-Verteilung) gut entfernen. Eine standardmäßige eindimensionale Normalverteilung ist wie das Bild 6 unten gezeigt. Ihrer mathematischer Ausdruck ist wie…, wobei μ der Durchschnitt von x und σ^2 die Varianz ist. Da in der standardmäßigen eindimensionalen Normalverteilung der Mittelpunkt bei der Berechnung des Durchschnittswerts der Ursprungspunkt ist, ist μ gleich 0.

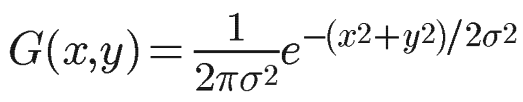
[https://de.wikipedia.org/wiki/Gau%C3%9F-Filter#Bildverarbeitung](https://de.wikipedia.org/wiki/Gauß-Filter" \l "Bildverarbeitung)





In der Bildverarbeitung, da der Graph zweidimensional ist, benötigen wir eine zweidimensionale Normalverteilung. Sie ist wie das Bild ... dargestellt. Der entsprechende mathematische Ausdruck ist wie...





https://www.pixelstech.net/article/1353768112-gaussian-blur-algorithm

In OpenCV, die entsprechende Funktion ist … 讲讲函数参数，然后讲高斯covolution怎么运作的，用上面这个网址

um den Miettelpunkt auszurechnen, gibt es in OpenCV eine andere Methode: Moment, aber 图有缺口的话，算的就不准了。

选canny的原因。。。Canny Edge Detection ist ein beliebter Kantenerkennungsalgorithmus. Es wurde von John F. Canny entwickelt. In OpenCV, neben der Standradfunktion Canny(), gibt es noch eine „overloaded member function“. Eine „overloaded member function“ kann denselben Name wie die originale Funktion besitzen aber mit unterschiedlichen Argumenttypen oder unterschiedlicher Anzahl von Argumenten. Der Unterschied zwischen der Standradfunktion und der „overloaded funktion“ von Canny() liegt darin, dass die Eingabeparameter sind unterschiedlich. Die Standardfunktion verwendet ein Bild als die Eingabe, während die „overloaded member function“ die x- und y-Ableitung des Bildes verwendet. Im unseren Fall werden die Bilder direkt aufgenommen von der Kamera, deswegen benutzen wir die Standardfunktion von Canny() und berechnen wir nicht mehr die x- und y-Ableitung von Bildern. Die Canny() Standardfunktion hat insgesamt 6 Parameter: „image“, „edges“, „threshold1“, „threshold2“, „apertureSize“ und „L2gradient“. „image“ ist das Eingabeparamter, das ein 8-bit-Bild anfordert. ‚edges‘ ist das Ausgabeparameter, das ein Einzelkanal-8-bit-Kantenbild ausgibt. ,L2gradient’ 。。。Die anderen drei Parameter sind die Parameter des Canny-Edge-Detection-Algorithmus, die die Performance dieses Algorithmus beschreiben. ‚threshold1‘ und ‚threshold2‘ sind die zwei Schwellwerte für das Hystereseverfahren. Das wird im folgenden weiter besprochen werden. „apertureSize“ stellt die (Blendengröße) für den Sobel-Operator fest. Sobel…

sobel算子结合了gauss平滑，所以也具有一定的抵抗噪声的作用。然后canny的高低阈值进一步(更有效的抵抗了噪声，因为两个阈值的作用)

the larger kernels

give a better approximation to the derivative because the smaller kernels are very sen-

sitive to noise. 论文148页， 还有sobel可以用来计算导数也在这个里面

论文：

https://d1wqtxts1xzle7.cloudfront.net/35756777/OReilly\_Learning\_OpenCV-libre.pdf?1417149048=&response-content-disposition=inline%3B+filename%3DOPEN\_CV.pdf&Expires=1689441541&Signature=SDAaYfU-idkOxzARnNfMwkUaUOkuJXAhtkzNr4WVbuDWLNX2ltKi8uJgvSYGtL-O0EAFzdHsmHzS5e7S-JN9vpGdZO5CTb7vozV6XynUK5FawDs-43xrrGf9RKqWB8rN7Pun9ikXkr500pZZRtsK4Wq3KVjegVHa2CS7ur-cQ~DN9ya-0R~bwIrMWey~TNxy~Th6DwpR09yThB1ErNf28vXx3enFpCKMzTHnEMiilvDu6dCVfLK17mfU6QLyY8P-SF0xqxzPm12~M1J3oNhj6YJN-ib7lH7AGkL9BPoxf1Uf4C~G4uqWyeDYtgql519S~VvLAqHufrmFfsHYUXnO9g\_\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

sobel和后面的图片对比一起讲吧，可能会用到下面这个链接：

https://pyimagesearch.com/2021/05/12/image-gradients-with-opencv-sobel-and-scharr/

https://www.edureka.co/blog/function-overloading-in-cpp/

Die Canny() Funktion findet die Kanten in den Eingabebildern mit dem Canny Algorithmus. Canny-Edge-Detection- Algorithmus ist ein mehrstufiger Algorithmus, der die folgenden Stufen enthält(下面名字都用德语吧): **Noise Reduction, Finding Intensity Gradient of the Image, Non-maximum Suppression und Hysteresis Thresholding.**

由于canny的优点：不容易受噪声干扰 --→ 不用用滤波也许

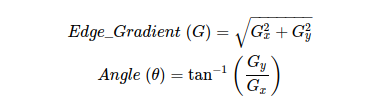
**Noise Reduction:**

Da die Kantenerkennung anfällig für Bildrauschen ist, besteht die erste Stufe darin, das Bildrauschen zu entfernen. In der Canny Edge Detection wird ein 5x5-Gaußfilter verwendet.

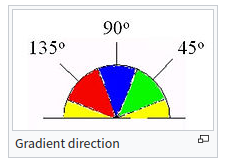
也许可以先讲讲canny的参数啥的

**Finding Intensity Gradient of the Image:**

Der Gradient eines Bildes ist einer der Grundbestandteile der Bildverarbeitung und Computer Vision. Er repräsentiert eine Richtungsänderung der Intensität oder Farbe in einem Bild. In diesem Schritt wird das vorher geglättete Bild mit einem Sobel-Kernel gefiltert in beide horizontaler und vertikaler Richtung. Die erste Ableitung in der horizontalen und vertikalen Richtung ist damit ausgerechnet. Der Gradient und die Richtung der Kante für jedes Pixel sind wie das folgende (Bild) gezeigt:



Die Richtung des Gradienten ist immer senkrecht zu den Kanten und sie wird auf einen von vier Winkeln gerundet: die vertikale, horizontale und zwei diagonale Richtungen. Eine grafische Darstellung ist wie folgend gezeigt. Die Winkel in [0°, 22,5°] oder [157,5°, 180°] wird auf 0° abgebildet.



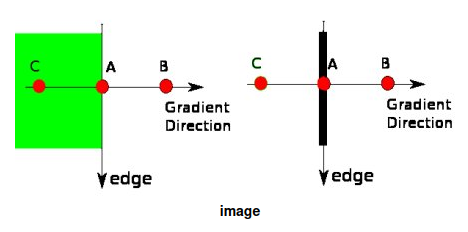
https://en.wikipedia.org/wiki/Image\_gradient

theata 正负详解：

https://pyimagesearch.com/2021/05/12/image-gradients-with-opencv-sobel-and-scharr/

**Non-maximum Suppression: 重要：比sobel得到的更光滑（ppt里用）：https://medium.com/@haidarlina4/sobel-vs-canny-edge-detection-techniques-step-by-step-implementation-11ae6103a56a**

Wie der Name dieses Schritts werden im diesen Schritt die schärfsten Gradienten beibehalten und die anderen werden verworfen, womit eine Kantenverdünnung erreicht wird. Nachdem die Größe und Richtung des Gradienten ermittelt wurde, wird ein vollständiger Scan des Bildes durchgeführt, um alle unerwünschten Pixel zu entfernen, die möglicherweise nicht die Kante bilden. Dazu wird bei jedem Pixel geprüft, ob es in seiner Nachbarschaft in Gradientenrichtung ein lokales Maximum gibt, siehe Abbildung 几.



Der Punkt A liegt auf einer Kante. Die Richtung des Gradienten ist vertikal zu der Kante. B und C sind zwei Punkte, die auch auf der Richtung des Gradienten liegen. In diesem Fall wird der Punkt A mit Punkt B und C überprüft, um zu sehen, ob er ein lokales Maximum bildet. Wenn ja, bleibt der Punkt A erhalten in die nächste Stufe, andernfalls wird es unterdrückt (auf Null gesetzt).

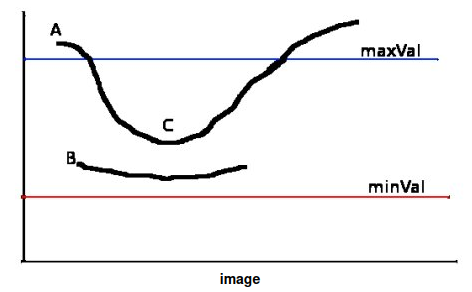
Deswegen 正如前面参数部分所说的输出是一个而值化的图。In short, the result you get is a binary image with "thin edges".

https://en.wikipedia.org/wiki/Canny\_edge\_detector

Hysteresis Thresholding:

Auf dieser Stufe wird entschieden, welche Kanten wirklich Kanten sind und welche nicht. Das ‚Hysteresis’ hier bedeutet die Abhängigkeit des Zustands eines Systems von seiner Geschichte. In der Canny() Funktion kann es als die Abhängigkeit eines Pixels von seinen Nachbarn verstanden werden. Das wird im folgenden erklärt werden. Um zu unterscheiden, ob es sich um eine reale Kante handelt werden in Canny() zwei Schwellenwerte verwendet werden: minVal und maxVal, welche bedeuten der minimale Wert und der maximale Wert. Die Kante mit einem Intensitätsgradienten größer als maxVal werden als reale Kanten betrachtet und diejenigen unterhalb von minVal werden keine Kanten betrachtet und werden daher verworfen. Diejenigen, die zwischen diesen beiden Schwellenwerten liegen, werden nach ihrer Konnektivität als Kanten oder Nichtkanten klassifiziert. d.h. wenn sie mit „sicheren Kanten“-Pixeln verbunden sind, werden sie als Teil von Kanten betrachtet. Andernfalls werden sie ebenfalls verworfen.

Siehe das Bild unten:



Die Kante A liegt über dem maxVal und wird daher als reale Kanten betrachtet. Obwohl Kante C unter maxVal liegt, ist sie mit Kante A verbunden, deswegen wird sie auch als gültige Kante betrachtet. Ebenso ist der obere rechte Teil von Kante C, der über dem maxVal liegt. Daher erhalten wir die vollständige Kurve. Jetzt betrachten wir die Kante B. Obwohl sie über minVal liegt und sich im gleichen Bereich wie Kante C befindet, ist sie mit keinen realen Kanten verbunden und wird folglich verworfen. Daher ist es von wesentlicher Bedeutung, dass wir das geeignete minVal und maxVal auswählen, um das richtige Ergebnis zu erhalten. Außerdem werden in dieser Phase die kleine Pixelrauschen auch entfernt, da es davon ausgegangen wird, dass es sich bei den Kanten um lange Linien handelt. Und was wir endlich erhalten sind starke Kanten im Bild.

因此不需要去椒盐噪声了

https://docs.opencv.org/4.x/dd/d1a/group\_\_imgproc\_\_feature.html#ga04723e007ed888ddf11d9ba04e2232de

https://docs.opencv.org/4.x/da/d22/tutorial\_py\_canny.html

https://bbs.elecfans.com/jishu\_2314292\_1\_1.html

https://indiantechwarrior.com/canny-edge-detection-for-image-processing/

https://en.wikipedia.org/wiki/Hysteresis

可能用到，自动调阀值的：

https://news.sangniao.com/p/1069340831

canny默认参数不滤波：

https://stackoverflow.com/questions/63543033/does-cv2-canny-perform-a-gaussian-blur

https://stackoverflow.com/questions/44646176/set-variance-of-gaussian-function-in-opencv-canny-function

gauss优点。。说吗？

Gauss: https://blog.csdn.net/assjaa/article/details/103536420

https://www.pixelstech.net/article/1353768112-gaussian-blur-algorithm

http://www.ruanyifeng.com/blog/2012/11/gaussian\_blur.html

https://computergraphics.stackexchange.com/questions/39/how-is-gaussian-blur-implemented

https://aryamansharda.medium.com/image-filters-gaussian-blur-eb36db6781b1

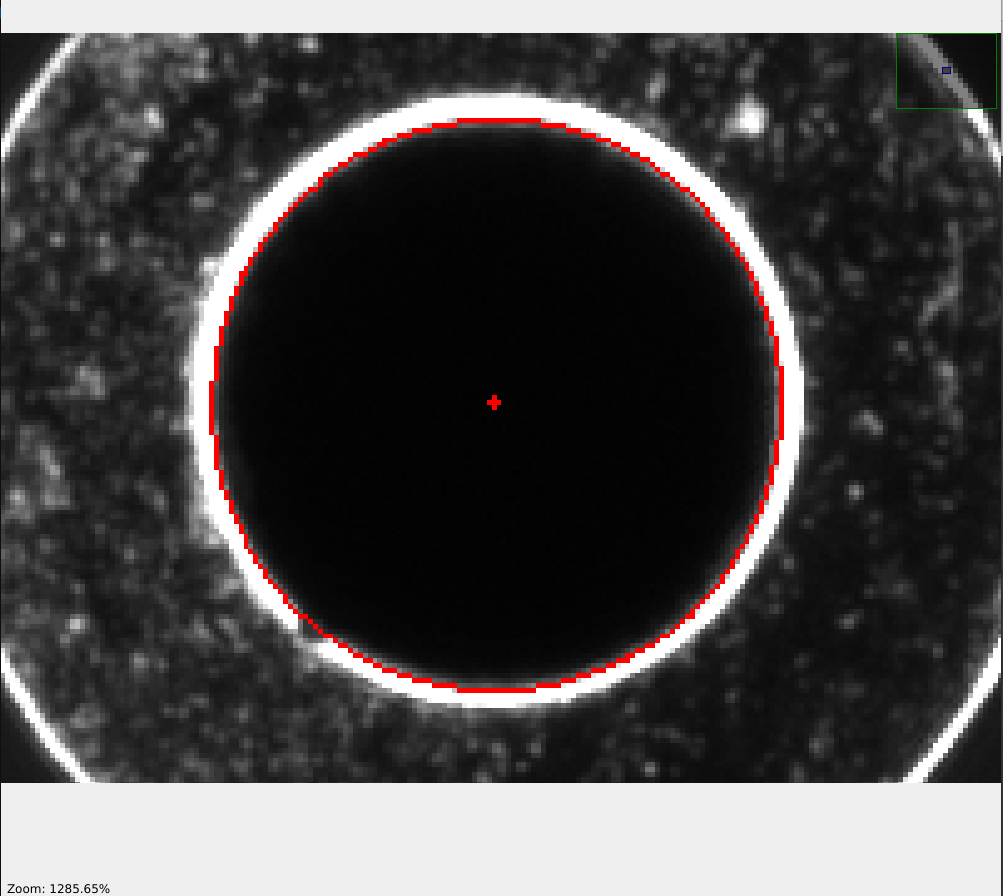
https://docs.opencv.org/4.x/dc/dd3/tutorial\_gausian\_median\_blur\_bilateral\_filter.html

https://docs.opencv.org/4.7.0/da/d22/tutorial\_py\_canny.html

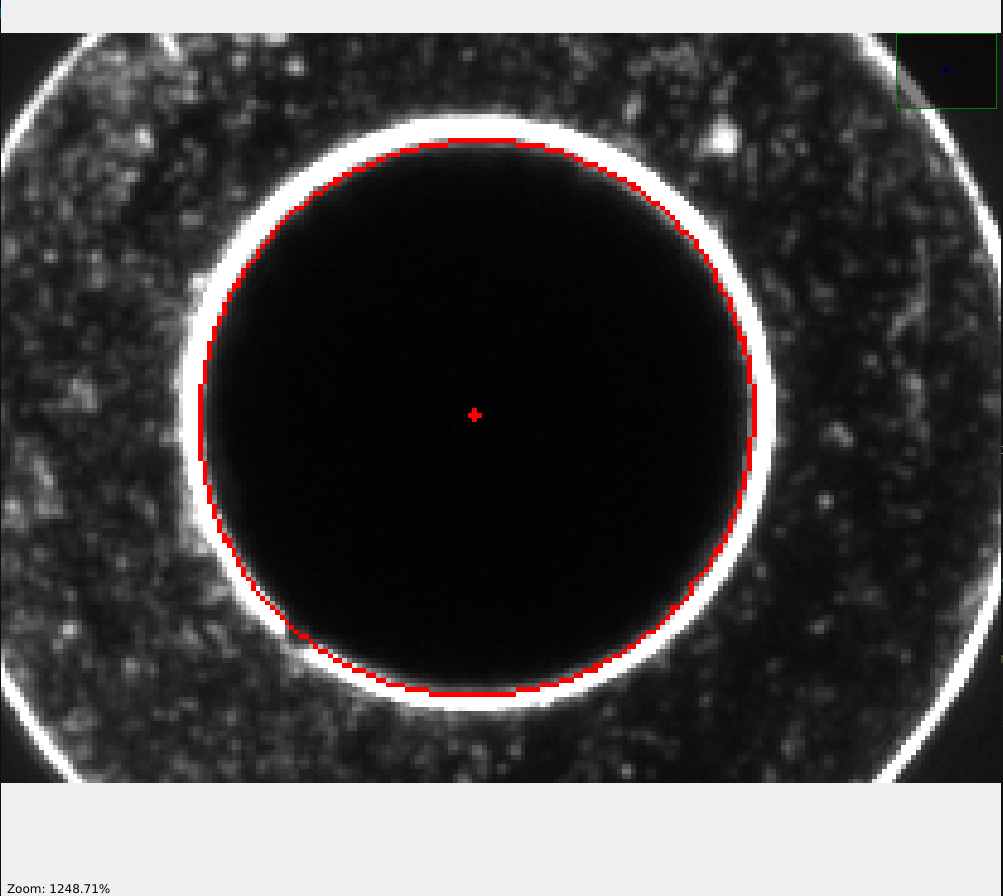
https://docs.opencv.org/4.7.0/da/d5c/tutorial\_canny\_detector.html

#cv2.error: OpenCV(4.7.0) /io/opencv/modules/imgproc/src/canny.cpp:846: error: (-206:Bad flag (parameter or structure field)) Aperture size should be odd between 3 and 7 in function 'Canny'

roh\_bild:



gauss(5,5)



Das erste zu kalibrierte Bauteil ist der Greifer. (greifer\_type) Der Grund fuer die ungenauigkeit liegt dafuer , dass wegen die montage zwischen dem Greifer und Gripper\_rot\_plate durch die Schrauben sind, sind diese zwei Bauteile nicht ganz konzentrisch, sondern gibt es eine Verschiebung in der x-y Ebene. Wie im Bild gezeigt.(用gazebo歪的greifer截个图) (die parallelitaet muessen wir nicht betrachten?) Deshalb wenn die rot\_plate sich dreht, liegt der Greiferkopf nicht genau am einen Punkt sondern sihc dreht auf einem Kreis, dessen Mittelpunkt auf der Drehachse liegt.

Basierend auf diesem Phaenomen wird es in dieser Arbeit eine entsprechende Kalibriermethode entwickelt. Die Grundidee ist die Verfolgung des Greifers waehrend der Rotation und die Beschreibung der Trajektorie des Greifers, mit der den Mittelpunkt von dem rot\_plate bzw. die Postion der Rotationsachse berechnet wird(不知道rot\_plate的中心位置). Es wird opencv als Erkennungsmethode… Die Verfolgung des Greifers erfolgt durch die Detection des runden Greiferkopfs. Zur einen Kamera(找文献论证), wenn ein Kreis parallel zur Optik der Karema (先讲一下不parallel的也是椭圆) aber nicht der Kamera genau gegenueber steht, wird das in der kamera photographierte Bild keinen Kreis sondern eine Ellipse. Obwohl der Fehler zwischen Kreis und Ellipse in unserem Fall ziemlich klein ist, wird es in dieser Arbeit davon ausgegangen, dass die photographierte Form eine Ellipse ist. Das ist eine Untenansicht des Greiferkopfs photographiert durch die Kalibrierkamera(gazebo里的截个图). Die interessiert Zone ist der innere Teil des Greiferkopfs.

Die verwendete opencv Version ist … OpenCV (Open Source Computer Vision Library) ist eine Open-Source-Bibliothek, die mehrere Hundert Computer Vision-Algorithmen enthält.(7.7.0 : <https://docs.opencv.org/4.7.0/d1/dfb/intro.html>)

先讲contour 和 hierarchy 然后 Bisher haben wir die kotur und Hierarchie知道了， 现在我们讲findcontour()

我们可以用moment算质心，但是有缺口就不行了，所以还是用fitellipse。

Einer der Ausgabeparamter von findContours() ist ‘contours’. Konturen können einfach als eine Kurve erklärt werden, die alle zusammenhängenden Punkte entlang der Grenze mit derselben Farbe oder Intensität verbindet. Der Python-Typ von diesem Output ‘list’. Es enthält alle Konturen, die von dieser Funktion detektiert werden. Jede individuelle Kontur ist repräsentiert durch einem Numpy-Array von (x,y)-Koordinaten aller einzelnen Punkte des detektierten Objektes. Für eine bessere Genauigkeit der Detektion werden immer die Binärbilder gebraucht. Aber darüber muss man nicht viel nachdenken, weil das Output von Canny() Funktion ist schon...

Contours: ...So remember, object to be found should be white and background should be black.

https://docs.opencv.org/4.x/d4/d73/tutorial\_py\_contours\_begin.html

画轮廓也是上面这个链接

Hierarchie: (hiecharchie 也许用官网或者别的图讲解一下)

https://docs.opencv.org/4.x/da/d0a/tutorial\_js\_contours\_hierarchy.html

Hierarchie: https://docs.opencv.org/4.7.0/d9/d8b/tutorial\_py\_contours\_hierarchy.html

dao di jige shuchu a

Ein anderes Output von findContours() ist ‚hierarchy‘. Wenn wir die Objekte in einem Bild detektieren, in manchen Fällen liegen einige Formen innerhalb anderer Formen. In diesem Fall bezeichnen wir das Äußere als Eltern und das Innere als Kind. Auf diese Weise stehen die Konturen in einem Bild in einer gewissen Beziehung zueinander. Und wir können angeben, wie eine Kontur miteinander verbunden ist, z. B. ob sie einer anderen Kontur untergeordnet ist oder ob sie ein übergeordnetes Element ist usw. Die Darstellung dieser Beziehung wird als Hierarchie bezeichnet. In OpenCV ist diese Beziehung zwischen Konturen als ein Array aus vier Werten repräsentiert: [Next, Previous, First\_Child, Parent]. ‚Next‘ bezeichnet den Index der nächsten Kontur auf derselben Hierarchieebene. ‚Previous‘ bezeichnet den Index der vorherigen Kontur auf derselben Hierarchieebene. ‚First\_Child‘ bezeichnet den Index seiner ersten untergeordnete Kontur und ‚Parent‘ bezeichnet den Index seiner übergeordneten Kontur. Wenn eine solche Kontur existiert nicht, wird dieses Feld als -1 angenommen. (bild?)

我们的论文中只对其中一个圆(轮廓)感兴趣，因此等级关系及其相应的研究不需要考虑

Es gibt zwei zentrale Funktionen in OpenCV um die Konturen zu detektieren: findContours() und findContoursH(). Die Funktion findContoursH() ruft Konturen aus dem Binärbild ab und berechnet deren Hierarchie, die nicht relevant in dieser Arbeit sind. Deswegen wird diese Funktion nicht verwendet. Die zentrale Funktion ist findContours(). Mit dieser Funktion werden die Konturen aus dem Binärbild abgerufen. Das Eingabebild ist ein 8-Bit-Einkanalbild. Pixeln ungleich Null werden als Einsen behandelt. Nullpixeln bleiben Nullen, daher wird das Bild als binär behandelt. Die andere Parameter dieser Funktion sind ‚contours‘, ‚hierarchy‘, ‚mode‘, ‚method‘ und ‚offset‘, wobei ‚contours‘ und ‚hierarchy‘ Ausgaben sind. Der Parameter ‚mode‘ legt den Modus des Konturabrufs. In opencv 4.7.0 gibt es vier Kontur Retrieval Modes(Modus?): RETR\_EXTERNAL, RETR\_CCOMP, RETR\_LISTund RETR\_TREE. Das RETR\_EXTERNAL Mode gibt nur die extreme äußere Flags zurück. Im unseren Fall wird es nich in Betracht kommen. Das RETR\_CCOMP Mode ruft alle Konturen ab und ordnet sie in einer zweistufigen Hierarchie an. Die äußere Konturen des Objekts (dh seine Grenze) werden in Hierarchie-1 platziert. Und die Konturen der Löcher im Objekt, bzw. die innere Konturen (sofern vorhanden) werden in Hierarchie 2 platziert. Wenn sich darin ein Objekt befindet, wird dessen Kontur wieder nur in Hierarchie-1 platziert. Und es ist ein Loch in Hierarchie 2 und so weiter. Das RETR\_LIST Mode ruft alle Konturen ab, ohne hierarchische Beziehungen herzustellen(这里面retrieve的意思是abrufen吗，啥意思阿)， d.h. alle Konturen gehoeren zur gleichen Hierarchieebene. Hier ist also der 3. und 4. Term im Hierarchie-Array immer -1. Wie das RETR\_LIST Mode ruft das RETR\_TREE Mode alle Konturen ab. Daneben erstellt es eine vollständige hierarchische Beziehungen. (….本文使用的是哪种方法) 都行，因为这个 im unseren Fall nicht so relevant ist

Der Parameter legt die Methode zur Konturnäherung fest. Die vorhandenen Methoden sind: CHAIN\_APPROX\_NONE, CHAIN\_APPROX\_SIMPLE, CHAIN\_APPROX\_TC89\_L1, CHAIN\_APPROX\_TC89\_KCOS. Das Methode CHAIN\_APPROX\_NONE speichert alle Konturpunkte.

Die Methode CHAIN\_APPROX\_TC89\_L1 und CHAIN\_APPROX\_TC89\_KCOS sind verwendet, wenn die Formen gekrümmt und keine einfachen Polygone sind, wobei Berechnungen mit höherer Präzision erfordern. In unserem Fall ist die zu detektierte Form eine einfache Ellipse, wobei die Methode CHAIN\_APPROX\_NONE, CHAIN\_APPROX\_SIMPLE schon reichen,deswegen werden diese zwei Methoden nicht weiter betrachtet.

https://cvexplained.wordpress.com/2020/06/03/finding-and-drawing-contours/

https://docs.opencv.org/4.7.0/d9/d8b/tutorial\_py\_contours\_hierarchy.html

https://docs.opencv.org/4.7.0/d3/dc0/group\_\_imgproc\_\_shape.html

chain simple 举例子图用这个：

https://docs.opencv.org/4.7.0/d4/d73/tutorial\_py\_contours\_begin.html

‘Offset’ ist ein optionaler Parameter, mitdem jeder Kontuepunkt verschoben werden kann. Es ist verwendet um die Konturen, die aus dem Bild-ROI extrahiert werden, in dem Kontext des gesamten Bild zu analysieren. Dieser Parameter ist für diese Aufgabe nicht relevant, deswegen wird er in dieser Arbeit nicht weiter berücksichtigt.

真的要去噪吗 ellipse文献里第一段最后三行说了二乘法关于噪声的看下

https://users.cs.cf.ac.uk/Paul.Rosin/resources/papers/ellipse3.pdf

椭圆检测，但是一开始也就没有移动到最中间而是一个大概的位置，所以一开始也是一个椭圆。

image\_processing.py 里 fit\_ellipse里 self.list的第一个一定是检测到的第一个椭圆的中心，也就是初始位置，因为在程序前面已经规定过 m1,m2 != 0了