目录

[相机标定 2](#_Toc2595653)

[定义 2](#_Toc2595654)

[分类 2](#_Toc2595655)

[张正友经典标定法 3](#_Toc2595656)

[内参的约束条件 4](#_Toc2595657)

[内参求解 5](#_Toc2595658)

[最大似然估计 6](#_Toc2595659)

[角点检测 7](#_Toc2595660)

[OpenCV的实现 7](#_Toc2595661)

[极几何(epipolar geometry) 7](#_Toc2595662)

[定义 7](#_Toc2595663)

[极线约束 8](#_Toc2595664)

[本质矩阵(Essential Matrix) 8](#_Toc2595665)

[基础矩阵(Fundamental Matrix) 9](#_Toc2595666)

[极线约束(epipolar constraint)的公式表达 10](#_Toc2595667)

[共面点成像(Planar Homography) 10](#_Toc2595668)

[定义 10](#_Toc2595669)

[单应性矩阵的求解 11](#_Toc2595670)

[从单应性矩阵分解出T矩阵和R矩阵（待补充） 12](#_Toc2595671)

[单应性的应用 12](#_Toc2595672)

[对极约束和单应性的比较 13](#_Toc2595673)

[Bundle Adjustment 14](#_Toc2595674)

[定义 14](#_Toc2595675)

[计算方法 15](#_Toc2595676)

[最速下降法 15](#_Toc2595677)

[Newton型方法 15](#_Toc2595678)

[Gauss-Newton方法 16](#_Toc2595679)

[LM方法 16](#_Toc2595680)

[三维重建中的BA 16](#_Toc2595681)

[Ceres Solver 17](#_Toc2595682)

# 相机标定

## 定义

获得相机的内部参数以及外部参数。

## 分类

目前标定方法可以大致分为传统标定方法、主动标定方法和自标定方法三类。

#### 传统摄像机标定方法

传统的摄像机标定方法需要形状规则，集几何信息已知的标定模板。通过对一幅以上图像特征点的提取和处理，简历标定物和拍摄图像之间的约束关系，形成关于摄像机内外参数的标定方程。目前，在传统摄像机标定中，比较典型的方法有：

1. 最优化的摄像机标定方法

该方法考虑了标定过程中各种失真的情况，成像模型非常复杂，很难控制结果的精度。同时成像模型的初始值对于结果的精度也有很大的影响。根据选择不同的标定模型，比较典型的最优化标定法可以分为两种，分别是基于摄影测量学的传统标定法和直接线性变换法。

1. 利用透视变换矩阵的标定方法

在摄影测量学中，可以用摄像机的内部参数和外部参数共同构成的非线性方程来描述二维图像坐标系和三维空间坐标系之间的相互关系。如果忽略摄像机镜头畸变的影响，将一组三维空间点和相应的图像点作为已知，而将透视变换矩阵中的元素作为未知，这样可以利用线性的方法来求解透视变换矩阵当中的各个元素。

从某种意义上来讲，该方法与直接线性标定方法没有本质的区别。利用透视变换矩阵的标定方法，不需要通过最优化的方法来求解摄像机的参数，可以提高标定的速度，而且能够实时的获取标定的结果，但同样也存在不足，因为在标定过程当中忽略了非线性畸变的影响，从而影响了结果的精度。

1. 考虑畸变影响的两步法

为了提高摄像机标定结果的精度，可以利用以上两种方法(直接线性标定方法和利用摄像机透视变换矩阵的标定方法)的标定结果作为初值，同时考虑非线性畸变的影响，然后再利用最优化方案，进一步提高精度，这样既可以克服初值对标定结果的影响，同时还能兼顾非线性畸变因素的影响，这种标定方法结合了前面两类方法的思想，将每一种标定方法的优势与不足之处结合到一起，形成了一种新颖的摄像机标定方法，这便是两步法的标定思想。

目前，经典的两部法是由Tsai所提出来的，该方法先用径向约束，求解模型中的大部分参数，再用非线性方程求解畸变系数，有效焦距等。

#### 自标定方法

自标定方法的目的是想让摄像机在未知场景和沿任意方向运动等一般情形下完成摄像机标定。自标定方法最大的优势在于不需要制作特定的标定模版来实现标定，它是一种对环境具有很强适应性的标定方法，可以利用环境的刚体变换，通过对比分析多幅图像中的对应点来获取摄像机参数。目前典型的关于摄像机自标定方法大致可以分为基于绝对二次曲线的自标定法、基于绝对二次曲面的自标定法、分层逐步标定法和其他改进的摄像机自标定技术。

#### 基于主动视觉的自标定法

基于主动视觉的摄像机自标定方法要求摄像机作特殊的运动来获得多幅图像，因此摄像机将被精确的安装在控制平台上，然后利用摄像机可控制的运动参数和图像来求解摄像机的内部参数和外部参数。其中比较典型的就是让摄像机作纯旋转运动和正交平移的运动。

## 张正友经典标定法

张正友于1998年在论文："A Flexible New Technique for Camera Calibration"提出了基于单平面棋盘格的相机标定方法。该方法介于传统的标定方法和自标定方法之间，使用简单实用性强，有以下优点：

* 不需要额外的器材，一张打印的棋盘格即可。
* 标定简单，相机和标定板可以任意放置。
* 标定的精度高。

设为场景中的一点，在针孔相机模型中，经过以下变换最终得到像素图像上的像点:

* 从世界坐标系通过刚体变换到相机坐标系
* 从相机坐标系通过透视变换得到成像平面上的点
* 通过仿射变换得到像素坐标系上的点

## 内参的约束条件

将上述过程整理成变换矩阵的形式：

令相机的内参矩阵为K：

在张氏标定法中，用于标定的棋盘格是三维场景中的一个平面，其在成像平面的像是另一个平面π，知道了两个平面的对应点的坐标，就可以用8点法求解得到两个平面的单应矩阵H。

设棋盘格所在的平面为世界坐标系中的平面，这样棋盘格的任一角点的世界坐标可表示为，根据小孔相机模型：

得到单应矩阵H和相机矩阵（包含内参和外参）的关系：

将旋转矩阵R的各个列向量和平移向量t使用H的列向量表示，

又由于R是正交矩阵，其任意两个列向量的内积为0，列向量的模为1。故有：

则对于一幅棋盘标定版的图像（一个单应矩阵）可以获得两个对内参数的约束等式：

只要能通过单应矩阵H求解内参矩阵K，就能完成相机标定。

## 内参求解

令，因为B是对称矩阵包含6个未知项，写成向量的形式：

将用替换，整理后得到

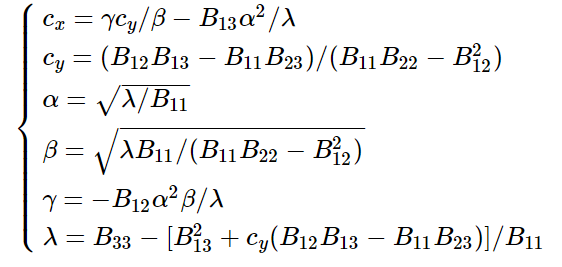
将两个约束条件合并成一个矩阵形式，从一副图像的单应矩阵中可得到：

假设有n幅图像，则有：

其中为的矩阵，所以有

* 当n≥3，可以得到b的唯一解;
* 当n=2，则可以假设扭曲参数γ=0作为额外的约束条件
* 当n=1，则值能计算两个相机的内参数

对于方程可以使用SVD求得其最小二乘解，从而可以得到相机的各个内参数：



## 最大似然估计

上面使用最小二乘法得到估计得到的解，并没有物理上的实际意义，。为了进一步增加标定结果的可靠性，可以使用最大似然估计(Maximum likelihood estimation)来优化上面估计得到的结果。

假设同一相机从n个不同的角度的得到了n幅标定板的图像，每幅图像上有m个像点。表示第i幅图像上第j个像点对应的标定板上的三维点，则通过下式计算出像点的位置：

其中，表示第i幅图像对应相机的旋转矩阵和平移向量。则像点的概率密度函数是

构造似然函数：

要使得L最小，就要最小化下列值：

这是一个非线性优化问题，可以使用Levenberg-Marquardt的方法，利用上面得到的解作为初始值，迭代得到最优解。

## 角点检测

### OpenCV的实现

在OpenCV中，findChessboardCorners函数能在输入的图像中定位出标定板上的角点。

阅读源码，可以把整个过程归纳为：

* 直方图均衡化，增大图像的对比度。
* 二值化，根据参数不同，有固定阈值和自适应阈值两种实现方法。
* 图像膨胀，将黑色的棋盘格相互分离。
* 添加轮廓，便于定位那些被截断的矩形。
* 检测四边形轮廓，排除干扰轮廓。
* 分类并判断，将相邻的角点归为一类，依据棋盘的规格判断每个类是否为棋盘。
* 亚像素精确化，对于获得的整数坐标进行数学计算，得到亚像素级的坐标。

其中为了定位的鲁棒性，自适应阈值二值化和图像膨胀所采用核的大小不能唯一，代码中采用多次循环，每次使用不同的大小的核对棋盘图像处理。

对于轮廓提取后可以得到的轮廓进行多边形拟合cvApproxPoly，排除不是矩形的轮廓，利用矩形的其他性质，再排除一些干扰轮廓。这些工作主要由icvGenerateQuads函数完成。

寻找每个方格的相邻方格，并记相邻方格的个数，连同相邻方格的信息存在相应CvCBQuad结构体中。找到相邻的方格之后，计算出二值图像在膨胀前的公共点，用公共点替代膨胀后分开的点。这主要由icvFindQuadNeighborhors函数完成。

对所有“方格”(包括被误判的)分类，分类的原则是类内所有方格是相邻的。由icvFindConnectedQuads函数完成。

根据已知所求的角点个数，判别每个类中方格是否为所求的棋盘方格，并对棋盘方格排序。在这个过程中，可以添加每类方格总缺少的方格，也可以删除每类方格中多余的方格。icvOrderFoundConnetedQuads函数完成该过程。

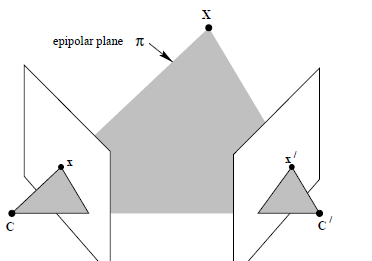
icvCleanFoundConnectedQuads函数、icvCheckQuadGroup函数根据已知棋盘的方格个数(由棋盘的角点数计算出来)确认方格位置及个数是否正确，并确定粗略强角点的位置(两个方格的相连位置)。icvCheckBoardMonotony再次检验棋盘方格是否提取正确。

以上如果有一步所有方格都不符合要求，则进入一个新的循环。若所有循环结束，还未找到符合要求的方格，则棋盘定位失败，退出函数。

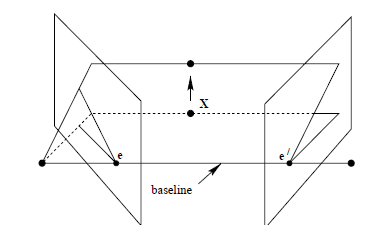
# 极几何(epipolar geometry)

## 定义

对于双目视觉系统，即有两个摄像机，定义两个摄像机的光学中心点为C、C‘，在三维空间中存在一个场景点X，这个点与两个摄像机光学中心点共同构成的平面就是极平面π，每个摄像机都有一个图像平面，分别为Image1和Image2，CX交Image1于x点，C'X交Image2于x'点，而CC'连线分别交两个图像平面于e和e'，这两个点称为极点，CC'称为基线。极平面与图像平面相交于两条极线l和l'，这两条极线的关系是对应的，而x、e、x'、e'分别位于l和l'上。

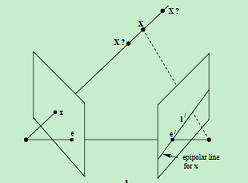


随着三维场景点的移动，极平面将绕着基线转动，这些极平面共同构成一个极平面束，这些极平面与图像平面所交汇成的极线族分别都交于两个极点e和e‘。

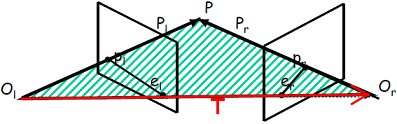


## 极线约束

假如我们只知道X投射在图像平面Image1上的投射点x，我们如何去获知在另一个图像平面上（也就是Image2）x的相应点x'呢，这个相应点x'符合什么样一种几何规则呢？我们知道，极平面是由基线和xX共同构成的，从上面的介绍我们知道了这个相应点（现在还是未知点）也一定位于极平面π上，因此可以得出x'点位于极平面π与另一个图像平面Image2的交线l'上，也即l'是投射点x的反向投影在第二个视角（第二个图像平面）上的图像。



## 本质矩阵(Essential Matrix)



由图可得，其中, 为场景点分别在左右相机坐标系中的坐标，为旋转矩阵，为平移矩阵。

因为是正交矩阵，可得，代入上式得到

因为向量共面，他们的混合积为0,得到

因为向量的叉积又可表示为矩阵与向量的乘积，令，

最后得到，取有，E即为Essential Matrix。

令，为场景点在两个图像平面上的图像点的齐次坐标，由三角形相似可以得到

## 基础矩阵(Fundamental Matrix)

由于矩阵E并不包含摄像头内参信息，且E是面向摄像头坐标系的。实际上我们更感兴趣的是在图像像素坐标系上去研究一个像素点在另一视图上的对极线，需要用到摄像机的内参信息将摄像头坐标系和图像像素坐标系联系起来。

假设场景点在左右图像上的像素坐标值分别为和，相机内参矩阵分别为和。

代入式子得

令得到

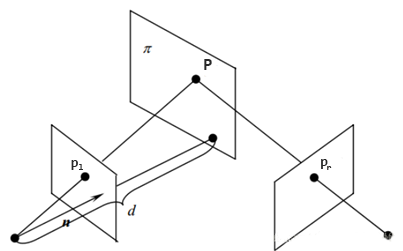
F就是基础矩阵，不光包含了旋转和平移的信息，还包含了相机的内参信息，使用的是像素坐标。

## 极线约束(epipolar constraint)的公式表达

直线上的一点，可以用矩阵表示成，因此对于，我们可以得到点在所确定的极线上

# 共面点成像(Planar Homography)

## 定义



上图表示场景中的平面在左右两相机的成像，设平面π在第一个相机坐标系下的单位法向量为，平面到第一个相机光心（假设为坐标原点）的距离为d，分别为场景点P在左右两相机坐标系中的三维坐标，则平面可表示为下式：

整理得到

右侧的相机坐标可由左侧的坐标经过平移旋转得到，即

代入式子，得

我们得到了两个相机坐标系之间的单应性矩阵H’

考虑到像素坐标更为使用，我们需要把相机坐标系的点用图像上的像素点表示，两者的关系如下

代入式子得

最终的得到了两幅像素图像之间的单应性矩阵H

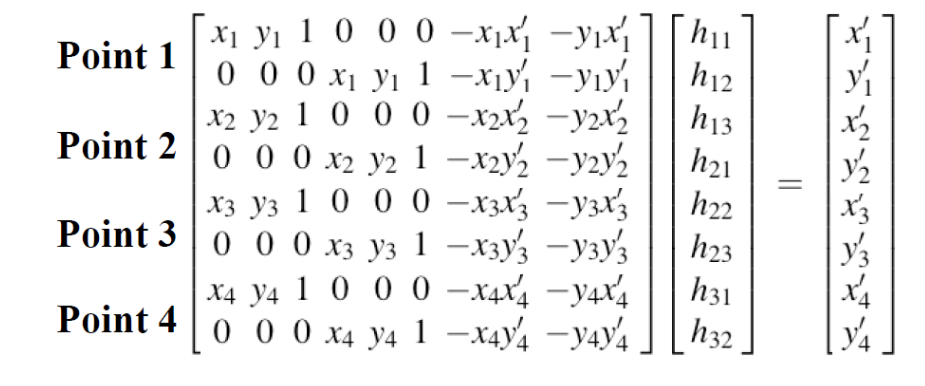
单应矩阵的适用场景为：当场景中的特征点都落在同一平面上，比如墙、地面等，此时可用单应性估计运动。

## 单应性矩阵的求解

单应性矩阵H满足下式

假设，得

整理得



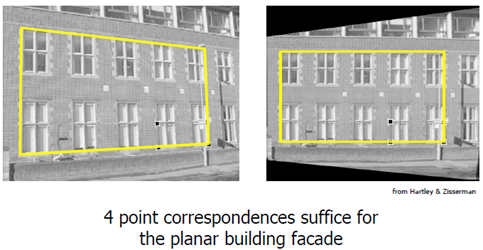
注意：以上做法只有当 h33 即最后一个元素不为零是才能得到正确的解！

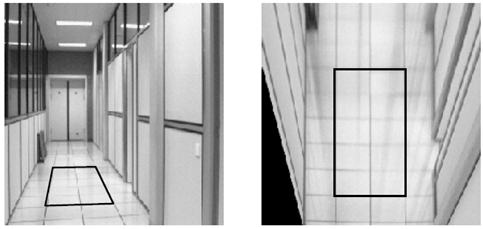
以上可以看出，从一对匹配点可以得到两项约束，于是自由度为8的单应矩阵可以通过4对匹配点进行求解。

## 从单应性矩阵分解出T矩阵和R矩阵（待补充）

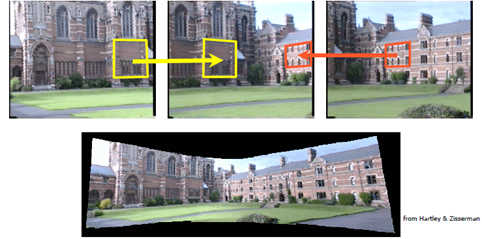
## 单应性的应用

1.图像校正





2.图像拼接



## 对极约束和单应性的比较

两幅图像之间的对极约束和场景的结构无关，即对于任意场景结构的图像都是成立的，他不能给出两幅图像上的点的一一对应关系，只能将点映射到线。而单应矩阵可以将点对应到点。

单应矩阵，不像对极约束那样完全不限制场景的结构，单应矩阵要求场景中的点都在同一平面上。

当相机只有旋转没有平移时，可使用单应矩阵估计运动，因为此时平移为0，计算得到的本质矩阵也为0，进而旋转也为0，得到了错误的解，而使用单应性依然能够正确计算。

# Bundle Adjustment

## 定义

Bundle Adjustment（以下简称BA），中文可译为光束法平差。所谓bundle，来源于bundle of light，其本意就是指的光束，这些光束指的是三维空间中的点投影到像平面上的光束。

BA的本质是非线性优化算法，它的原型是：

其中是我们需要优化的参数，一般称为代价函数（Cost Function），为损失函数（Loss Function）。其中的返回值可能是一个向量，因此总的代价取该向量的2-范数。

## 计算方法

### 最速下降法

梯度方向是函数上升最快的方向，而此时我们需要解决的问题是让函数最小化，那就顺着梯度的负方向去迭代，寻找使函数最小的变量值。

梯度下降法就是用的这种思想，用数学表达的话

为步长。

最速下降法保证了每次迭代中函数都是下降的，在初始点离最优点很远的时候，下降的速度非常快，但是接近最优点时收敛非常非常的慢。

### Newton型方法

对于一个给定的开口向上的一元二次函数，对该函数求导，导数为0处就是函数最小处。

Newton型方法也就是这种思想，首先将函数利用泰勒展开到二次项：

**f**(**x**+*δ***x**)≈*φ*(*δ***x**)=**f**(**x**)+**J**(**x**)*δ***x**+**21**​*δ***xTH**(**x**)*δ***x**

其中为Jacobi矩阵，对矩阵函数求一次偏导而来，梯度也是对向量函数求一次偏导而来。为Hessian矩阵，也就是二次偏导矩阵。

也就是说Newton型方法将函数局部近似成一个二次函数进行迭代，然后令在δx方向上迭代直至收敛，接下来对这个函数求导：

*φ***′**(*δ***x**)=**J**+**H***δ***x**=**0**⟹*δ***x**=−**H**−**1J**

Newton型方法收敛的时候特别快，尤其是对于二次函数而言一步就可以得到结果。但是该方法有个最大的缺点就是Hessian矩阵计算实在是太复杂了，并且Newton型方法的迭代并不像最速下降法一样保证每次迭代都是下降的。

### Gauss-Newton方法

既然Newton型方法计算Hessian矩阵太困难了，那有没有什么方法可以不计算Hessian矩阵呢？将泰勒展开式的二次项也去掉好像就可以避免求Hessian矩阵了吧，就像这样：

**f**(**x**+*δ***x**)≈**f**(**x**)+**J**(**x**)*δ***x**

这好像变成了一个线性函数了啊，线性函数如果要最小化的话好像是需要增加其他的约束条件的啊。那这里有没有其他的约束条件呢？仔细思考一下，我们需要最小化的是重投影误差，它的最小值是0。所以这个时候就不应该求导了，而是直接令函数为0。此时，令f(x)=ε有

*ε*+**J***δ***x**=**0**⟹**JTJ***δ***x**=−**JT***ε*

**x**=**x**+*δ***x**

Gauss-Newton方法就避免了求Hessian矩阵，并且在收敛的时候依旧很快。但是依旧无法保证每次迭代的时候函数都是下降的。

### LM方法

LM方法就是在以上方法基础上的改进，通过参数的调整使得优化能在最速下降法和Gauss-Newton法之间自由的切换，在保证下降的同时也能保证快速收敛。

## 三维重建中的BA

对于三维重建中的BA，代价函数往往是重投影误差。

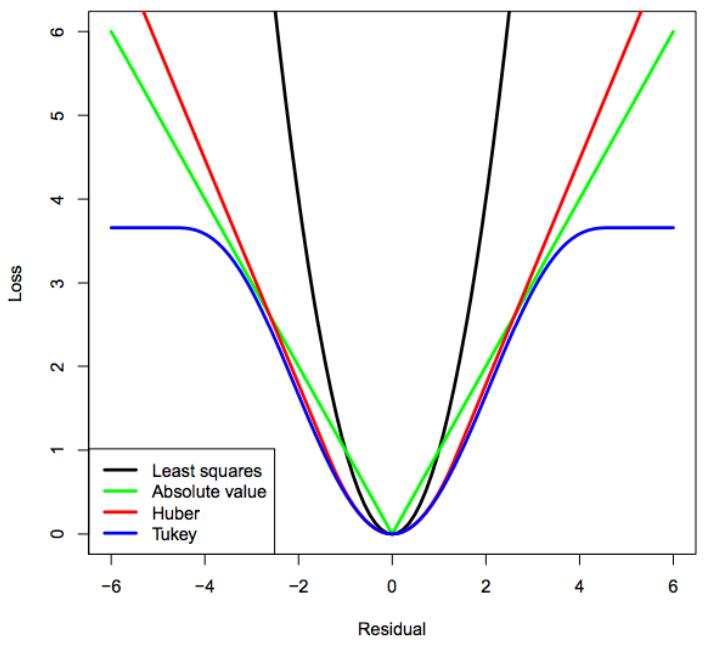
重投影指的是，在通过图像进行三维重建的过程中，利用我们计算得到的三维点的坐标和计算得到的相机位姿进行的第二次投影。

重投影误差，指的就是真实三维空间点在图像平面上的投影和基于计算的重投影之间的差值。

比如我们需要优化的参数有相机的内参、外参和点云，设图像的内参为，外参为和，点云中某一点的空间坐标为，该点在图像中的像素坐标为，则重投影误差的表示为

上式中的和均为齐次坐标，为投影函数。

而损失函数的目的是为了增强算法的鲁棒性，使得算法不易受离群点的影响，常见的有Huber函数、Tukey函数等，这些函数的图像如下



若不使用损失函数，即，代价(Cost) 如上图中的黑色曲线所示，随误差以二次幂的速度增长。使用其他的损失函数，可以发现，随着误差的增大，要么代价的增长是趋于线性的(Huber)，要么干脆趋于不变(Tukey)，这样就降低了误差较大的点对总代价的影响。

## Ceres Solver

求解BA的总体思想是使用梯度下降，比如高斯-牛顿迭代、Levenberg-Marquardt算法等，由于BA还有自己的一些特殊性，比如稀疏性，在实现时还有很多细节需要处理。好在现在有很多用于求解非线性最小二次问题的库，比如Google的一个开源项目——Ceres Solver.

Ceres Solver专为求解此类问题进行了大量的优化，有很高的效率，尤其在大规模问题上，其优势更加明显。并且，Ceres内置了一些常用的函数，比如对坐标的旋转以及各类损失函数，使其在开发上也比较高效。

使用Ceres求解非线性优化问题，一共分为三个部分：

1、 第一部分：构建代价函数，也就是寻优的目标式。这个部分需要使用仿函数（functor）这一技巧来实现，做法是定义一个cost function的结构体，在结构体内重载（）运算符。

2、 第二部分：通过代价函数构建待求解的优化问题。

3、 第三部分：配置求解器参数并求解问题，这个步骤就是设置方程怎么求解、求解过程是否输出等，然后调用Solve方法。

Level-set提高像素精度

向量场 确定参考点 计算向量 匹配算法

自上而下的卷积网络：从整体到细节