

# CS131 Lecture06: 边缘检测

by: 斯坦福大学计算机科学系

github: [https://github.com/zhaoxiongjun/CS131\\_notes\\_zh-CN](https://github.com/zhaoxiongjun/CS131_notes_zh-CN) (包含中英文版课件及相关课程视频)

## 1 介绍

本课程包括边缘检测、Hough 转换和 RANSAC。边缘检测提供了有意义的语义信息，有助于理解图像。这有助于分析元素的形状、提取图像特征，以及了解所描绘场景的属性变化，例如深度不连续、材质类型和光照等。我们将探讨 Sobel 和 Canny 边缘检测技术的应用。下一节将介绍用于检测图像中参数模型的 Hough 变换；例如，通过 Hough 变换可以检测由两个参数定义的线性线。此外，这种技术可以推广到检测其他形状（如圆）。然而，正如我们将看到的，使用霍夫变换在拟合具有大量参数的模型时并不有效。为了解决这个模型拟合问题，在最后一节中引入了随机抽样一致性（RANSAC）；这种非确定性方法重复抽样数据子集，根据如何将剩余的数据点分类为“内点”或“异常值”，并使用它们来拟合模型。它们可以通过拟合模型来解释（即它们与模型的接近程度）。结果用于最终选择用于实现最终模型拟合的数据点。最后一节对 RANSAC 变换和 Hough 变换进行了比较。

## 2 边缘检测

### 2.1 边缘检测的目的

边缘检测与哺乳动物的眼睛非常相关。大脑中的某些神经元擅长识别直线。来自这些神经元的信息被整合到大脑中，用于识别物体。事实上，边缘对于人类的识别非常有用，线条图几乎和原始图像一样容易识别（图 1）。我们希望能够提取信息、识别对象以及恢复图像的几何和视点。

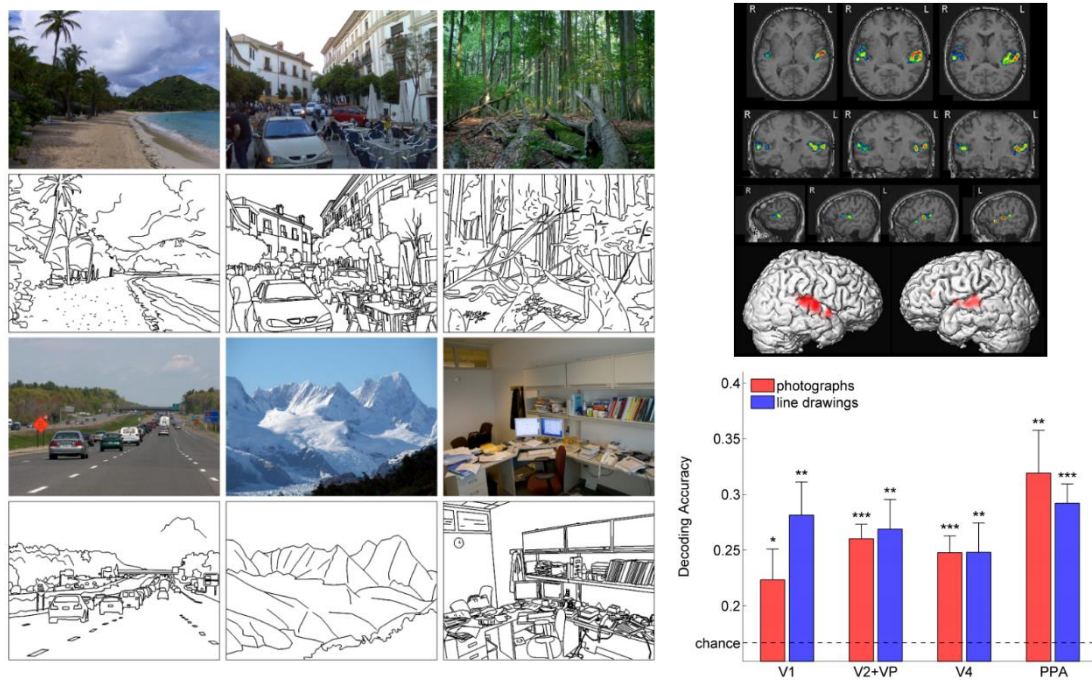


图 1：大脑的某些区域对边缘作出反应；线条图和原始图像一样容易识别

## 2.2 边缘基础知识

图像中有四种可能的边缘来源：表面法向不连续（表面急剧改变的方向）、深度不连续（一个表面在另一个表面后面）、表面颜色不连续（单个表面改变颜色）、光照（光强）不连续（阴影/照明）。这些不连续性如图 2a 所示；不同类型的边缘如图 2b 所示：

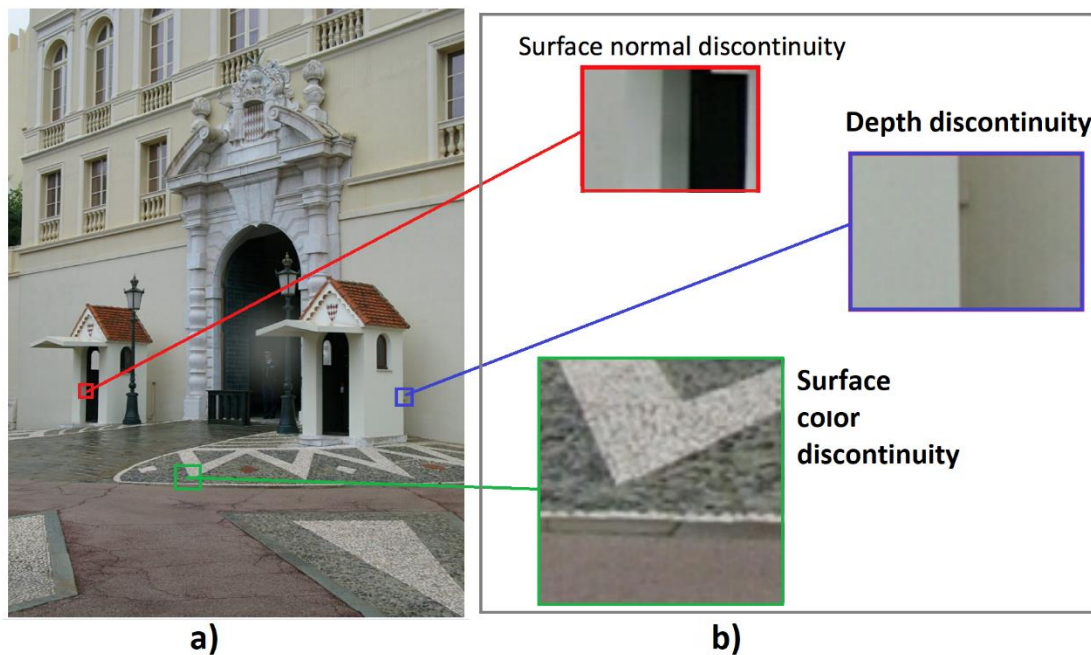


图 2。由于表面颜色、表面深度和表面法向的不连续性导致不同类型的边缘

当梯度的幅度较大时，图像中会出现边缘。

## 2.3 求解梯度

为了求梯度，我们必须首先求  $x$  和  $y$  方向的导数。

### 2.3.1 离散导数

$$\begin{aligned}\frac{df(x)}{dx} &= \lim_{\delta x \rightarrow 0} \frac{f(x) - f(x - \delta x)}{\delta x} = f'(x) \\ \frac{df(x)}{dx} &= \frac{f(x) - f(x - 1)}{1} = f'(x) \\ \frac{df(x)}{dx} &= f(x) - f(x - 1) = f'(x)\end{aligned}$$

有三种不同的求导方式（后向，前向，中心）（上一节也提到了）：

- Backward:  $f'(x) = f(x) - f(x - 1)$
- Forward:  $f'(x) = f(x + 1) - f(x)$
- Central:  $f'(x) = \frac{f(x+1) - f(x-1)}{2}$

每一个都可以表示为一个过滤器（将过滤器与图像卷积得到导数）

- Backward:  $f'(x) = f(x) - f(x - 1) \rightarrow [0, 1, -1]$
- Forward:  $f'(x) = f(x) - f(x + 1) \rightarrow [-1, 1, 0]$
- Central:  $f'(x) = f(x + 1) - f(x - 1) \rightarrow [1, 0, -1]$

梯度计算如下：

$$\begin{aligned}\nabla f(x, y) &= \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix} \\ &= \begin{bmatrix} f_x \\ f_y \end{bmatrix}\end{aligned}$$

也可以计算梯度的大小和角度：

$$|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$$

$$\theta = \tan^{-1}(f_y/f_x)$$

## 2.4 减少噪音

噪声会干扰梯度，导致使用简单的方法无法找到边缘，即使边缘仍然可以被眼睛检测到。解决方法是先平滑图像。设  $f$  为图像， $g$  为平滑核。因此，为了找到平滑的梯度，我们必须计算（1 维示例）：

$$\frac{d}{dx}(f * g)$$

根据导数卷积定理：

$$\frac{d}{dx}(f * g) = f * \frac{d}{dx}g$$

这种简化为我们节省了一次操作。平滑虽然消除了噪音，但使边缘变模糊。不同粒径的平滑可以检测不同尺度的边缘。

## 2.5 索贝尔(Sobel)噪声检测

该算法利用了 2 个 3\*3 的核，与图像卷积，近似于原始图像的  $x$  和  $y$  导数。

$$\mathbf{G}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \mathbf{G}_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

这些矩阵表示平滑和微分的结果。

$$\mathbf{G}_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} [1 \quad 0 \quad -1]$$

Sobel 滤波器存在许多问题，包括定位精度不好。Sobel 过滤器也倾向于水平和垂直边缘，而不是倾斜边缘。

## 2.6 Canny 边缘检测器

Canny 边缘检测器有五个算法步骤：

- 抑制噪声
- 计算梯度大小和方向
- 应用非极大值抑制
- 滞后阈值
- 检测边缘的连通性分析

### 2.6.1 抑制噪声

我们既可以抑制噪声，也可以使用类似于 Sobel 滤波器的方法计算 x 和 y 方向的导数。

### 2.6.2 计算梯度大小和方向

$$|\nabla f(x, y)| = \sqrt{f_x^2 + f_y^2}$$
$$\theta = \tan^{-1}(f_y/f_x)$$

### 2.6.3 应用非极大值抑制

这部分算法的目的是确保边缘是特定的（唯一的）。因此，我们假设边缘发生在梯度达到极大值时。我们抑制任何具有非极大渐变的像素。基本上，如果像素不是三个像素中最大的一个，并且与渐变方向相反，则设置为 0。此外，所有梯度都四舍五入到最接近的 45°。

### 2.6.4 滞后阈值

所有剩余的像素都受到滞后阈值的影响。本部分使用两个值作为高阈值和低阈值。每个值高于高阈值的像素都被标记为强边。低于低阈值的每个像素都设置为 0。两个阈值之间的每个像素都标记为弱边。

### 2.6.5 检测边缘的联通性分析

最后一步是连接边缘。所有标记为强边的像素都是边。对于弱边缘像素，只有和强边缘像素毗邻的弱边缘像素是边。这部分使用 BFS 或 DFS 来查找所有边缘。

## 3 Hough（霍夫）变换

### 3.1 霍夫变换介绍

Hough 变换是一种检测图像中特定结构的方法，即直线。然而，霍夫变换可以用来检测任何已知参数方程的结构。它在噪声和部分遮挡下提供了一个强大的探测器。

### 3.2 霍夫变换检测线的目标

Hough 变换可以用来检测图像中的线条。要做到这一点，我们需要定位组成图像中直线的像素集。这项工作的目的是在应用边缘检测器后检测图像中的线条，以获得仅边缘的像素（因此，我们发现哪些像素集构成了直线）。

### 3.3 在 A、B 空间使用 Hough 变换检测直线

对于一个像素点  $(x_i, y_i)$ ，有无限的线可以穿过这一点。我们可以定义一条穿过这个像素  $(x_i, y_i)$  的线：

$$y_i = a * x_i + b$$

这样，我们可以将每个像素转换为 A, B 空间，方法是将此公式重新编写为：

$$b = -a * x_i + y_i$$

这个等式表示 A, B 空间中的一条线，并且每个 (A, B) 点代表通过我们的点  $x_i, y_i$  的一条可能的线。

因此，对于每个像素  $x_i, y_i$ ，在我们的边缘像素集合中，我们将其转换成 A, B 空间以获得一条直线。



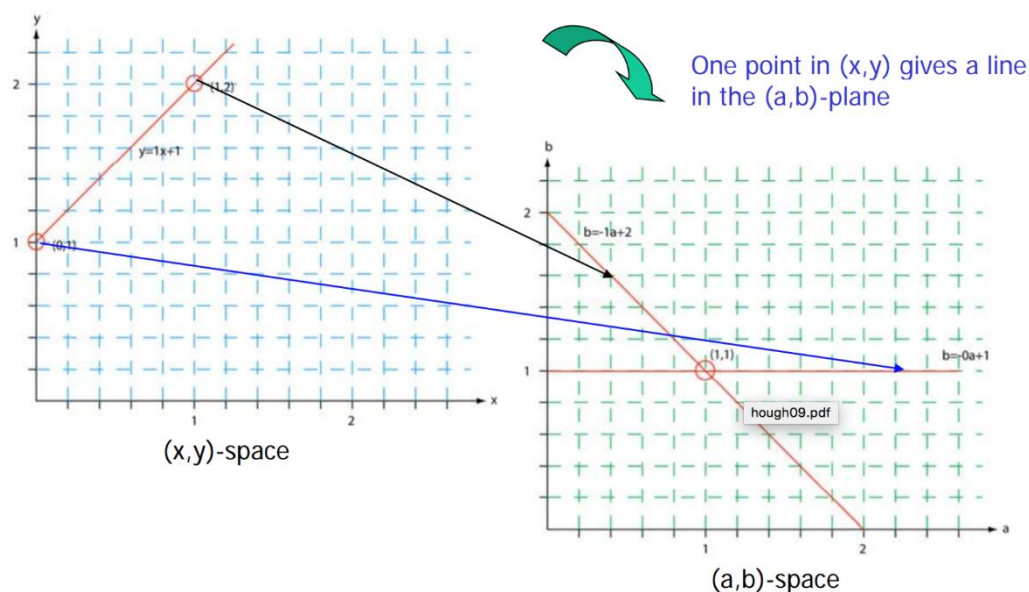


图 3：从原始空间到霍夫空间的转换

A 和 B 空间中的线的交点表示这些线对通过这些点的线  $Y_i = AX + B$  进行折叠。例如：假设我们有两点  $x_1, y_1 = (1, 1)$  和  $x_2, y_2 = (2, 3)$ 。我们将这些点转换成 A, B 空间，其中  $B = -A \cdot 1 + 1$  和  $B = -A \cdot 2 + 3$ 。求解这两条线的交点，得到  $A = 2$  和  $B = -1$ 。

(A, B) 空间中的交叉点为通过 x, y 空间中两个点的线提供的值。

(简而言之，这里想表达的是：**hough 变换**相当于一种映射变换，只不过是从一种表达方式换为另外一种。其根本思想是将原有的  $y = kx + b$  中的 x, y 看做固定值，而 k, b 看成变量来运算的。x-y 坐标和 k-b 坐标有点——线的对偶性。这里的对偶性指的是 x-y 系坐标中的点对应 k-b 坐标系中的线，x-y 坐标系中的线对应 k-b 坐标系中的点。)

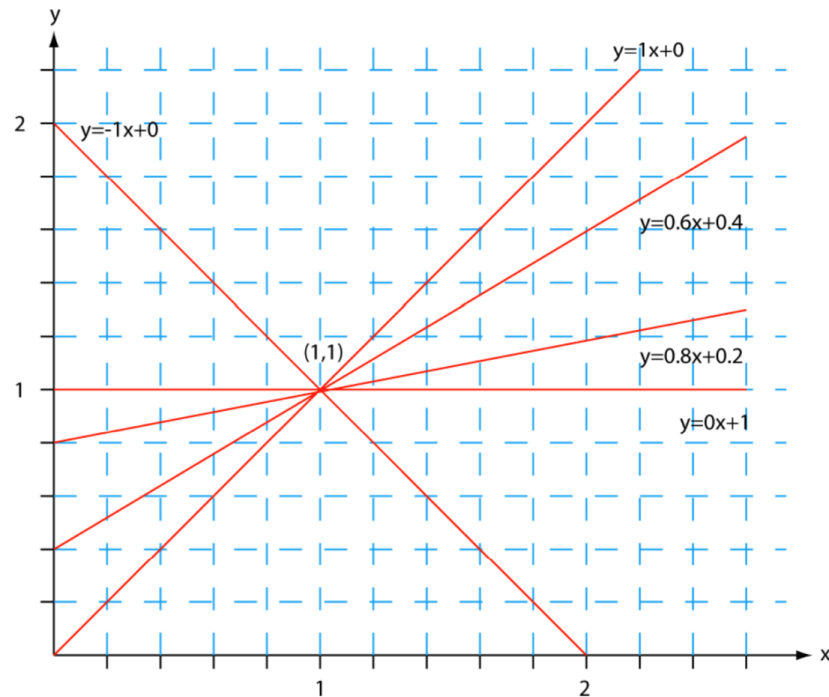


图 4. 穿过原始空间点的线条

<https://blog.csdn.net/mysteryrat/article/details/12495425> (关于霍夫变换好的解释)

### 3.4 “蓄电池单元”

为了得到“最佳”的线条，我们将 A, B 空间量化为单元。对于我们的 a, b 空间中的每一行，我们向它通过的每个单元格添加一个“投票”或一个计数。我们对每一行都这样做，所以在最后，“投票”最多的单元格具有最多的交叉点，因此应该与图像中的实线相对应。A, B 空间中的 hough 变换算法如下：

## Algorithm for Hough transform

- Quantize the parameter space (a b) by dividing it into cells
- This quantized space is often referred to as the accumulator cells.
- Count the number of times a line intersects a given cell.
  - For each pair of points  $(x_1, y_1)$  and  $(x_2, y_2)$  detected as an edge, find the intersection  $(a', b')$  in (a, b) space.
  - Increase the value of a cell in the range  $[[a_{\min}, a_{\max}], [b_{\min}, b_{\max}]]$  that  $(a', b')$  belongs to.
  - Cells receiving more than a certain number of counts (also called ‘votes’) are assumed to correspond to lines in (x, y) space.



图 5: Hough 变换算法

## 实现算法描述

摘取一篇博客的算法描述:

1. 初始化 $(\theta, \rho)$ 空间,  $N(\theta, \rho) = 0$  ( $N(\theta, \rho) = 0$  表示在该参数表示的直线上的像素点的个数)
2. 对于每一个像素点 $(x, y)$ , 在参数空间中找出满足 $x \cos \theta + y \sin \theta = \rho$  的 $(\theta, \rho)$ 对, 然后令 $N(\theta, \rho) = N(\theta, \rho) + 1$
3. 统计所有 $N(\theta, \rho)$ 的大小, 取出 $N(\theta, \rho) > \tau$ 的参数 ( $\tau$ 是预设的阈值)

但我觉得这并不是十分完整的算法流程。所以我将其改进描述如下

1. 读取原始图并转换成灰度图, 采用边缘检测算子 (如Canny) 转换成二值化边缘图像
2. 然后对该图像进行霍夫变换
3. 先使用峰值检测函数, 找到大于阈值的霍夫变换单元 (局部最大值应该最可能是线, 步长和量化会影响效果)
4. 将上述识别出的一组候选峰, 需要确定与其相关的线段及其起始点和终止点 (这需要一定的算法, 很多论文对此都做了改进, 诸如蝴蝶形状宽度, 峰值走廊)
5. 然后描绘于原图 (或结果图) 上

## 3.5 极坐标系下的霍夫变换

参考: 霍夫变换——神奇的特征提取算法:

<https://zhuanlan.zhihu.com/p/47649796>

## 3.6 总结

Hough 变换的优点是概念简单 (只需变换和在 Hough 空间中找到交叉点)。它也相当容易实现, 并且可以很好地处理丢失和阻塞的数据。另一个优点是, 只要结构具有参数方程, 它就可以找到除直线以外的其他结构。

缺点包括在参数越多, 计算上越复杂。它也只能同时查找一种结构 (因此不能将线和圆放在一起)。也不能检测到线段的长度和位置。它可以被“明显”的线条欺骗, 并且不能同线性的线段作区分。

## 4 RANSAC (随机抽样一致)

随着模型复杂度的增加 (如参数数量), hough 变换失去了其有效性; 本节详细介绍了随机样本一致性 (ransac) 技术[1]的设计, 该技术提供了一种在图像中拟合模型的计算有效方法。首先介绍了 RANSAC 的基本思想, 然后介绍了其算法。

## 4.1 RANSAC 基础

RANSAC 算法用于估计图像中模型的参数（即模型拟合）。RANSAC 的基本思想是使用随机选择的最小数据子集多次解决拟合问题，并选择性能最佳的拟合。为了实现这一点，RANSAC 尝试迭代地识别与我们试图拟合的模型相对应的数据点。

图 7a 说明了将线性模型（即 2 个参数）拟合到数据上的示例；当大多数数据点拟合线性模型时，右上角的两个点会显著影响整体拟合的精度（如果它们包含在拟合中）。RANSAC 算法旨在通过识别数据中的“内点”和“异常点”来解决这一挑战。

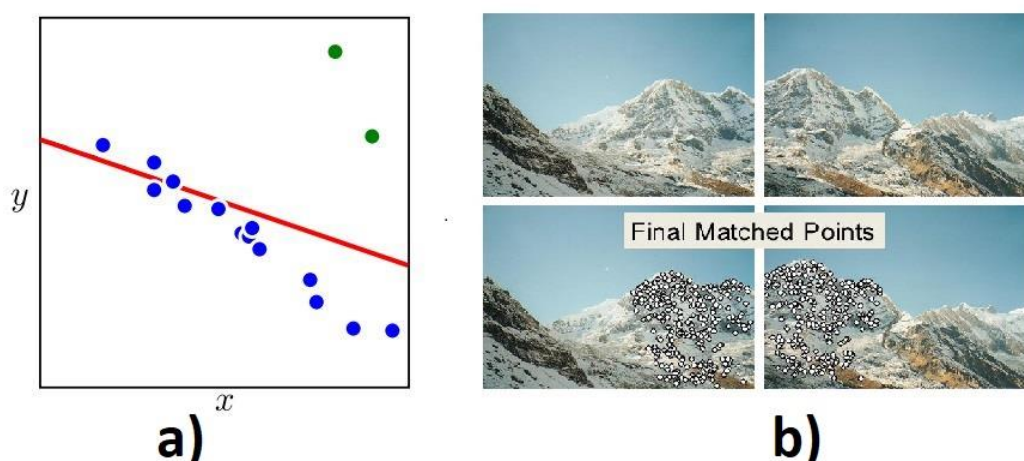


图 7： a) RANSAC 检测异常值以改进参数估计； b) RANSAC 图像拼接应用

## 4.2 应用

RANSAC 算法可用于估计不同模型的参数；这在图像拼接（图 7b）、离群值检测、车道检测（线性模型估计）和立体摄像机计算中被证明是有益的。

## 4.3 算法

RANSAC 算法对原始数据的名义子集进行迭代采样（例如，2 个点用于直线估计）；将模型拟合到每个样本，并计算与此拟合对应的“内点”数；这包括接近拟合模型的数据点。接近阈值的点（例如，2 个标准偏差或预先确定的像素数）被视为“内点”。如果大部分数据视为该拟合的“内点”，则认为拟合模型是好的。在拟合良好的情况下，使用所有的内点重新拟合模型，并丢弃异常值。重复此过程，并比较具有足够大的内点分数（例如，大于预先指定的阈值）的模型估计，以选择性能最佳的拟合。图 8 说明了线性模型及其三个样本的这一过程。第三个样本（图 8c）提供了最佳拟合，因为它包含了最多数量的内点。

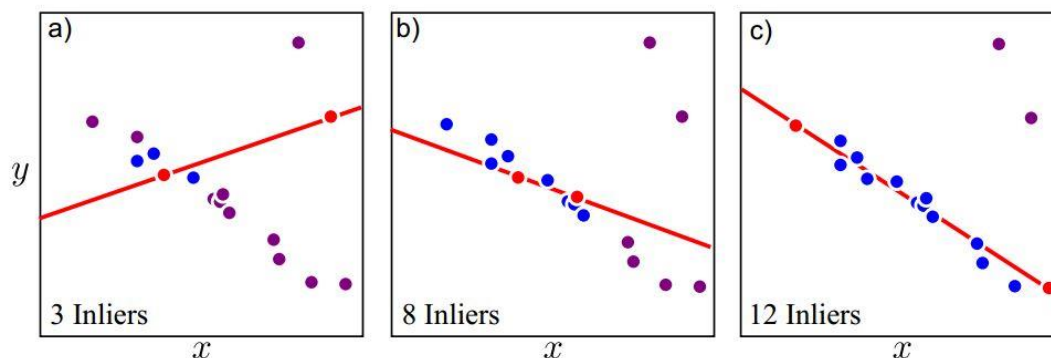


图 8：线性模型估计和三个随机样本的 RANSAC 算法证明

```

Determine  $n$ ,  $t$ ,  $d$  and  $k$  as above
Until there is a good fit or  $k$  iterations have occurred
    draw a sample of  $n$  points from the data
    uniformly and at random

    fit to that set of  $n$  points

    for each data point outside the sample

        test the distance from the point to the line
        against  $t$ ; if the distance from the point to the line
        is less than  $t$ , the point is close

    end
    if there are  $d$  or more points close to the line
    then there is a good fit. Refit the line using all
    these points, and terminate
end

```

图 9：RANSAC 伪代码

RANSAC 如图 9 所示。RANSAC 回路中包括的主要步骤：

- 1：从数据中随机选择种子组。
- 2：使用所选种子组执行参数估计。
- 3：确定内点（接近估计模型的点）。
- 4：（如果存在足够多的内点，）使用所有内点重新估计模型。
- 5：重复步骤 1-4，最后将估计值保持在最大的内点和最佳的匹配状态。

## 4.4 需要多少样品？

RANSAC 是一种非确定性的模型拟合方法；这意味着样本数量必须足够大，以提供对参数的高置信估计。所需样本的数量取决于 1) 拟合参数的数量和 2) 噪声的数量。图 10 列出了基于  $p=0.99$  和样本大小变化（即参数数量）和异常值分数

(即噪声)所需的最小样本数量。估计更大的模型和噪声数据需要更多的样本。需要选择足够多的样本 (k)，以确保只看到坏样本 (pf) 的概率较低：

$$P_f = (1 - W^n)^k = 1 - p$$

其中 w 和 n 分别是模型拟合所需的内点分数和点数。最小样本数：

$$k = \frac{\log(1 - p)}{\log(1 - W^n)}$$

## RANSAC: Computed k (p=0.99)

Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

图 10：噪声总体和模型尺寸的不同选择的样本数

### 4.5 优势、局限性和考虑因素

RANSAC 的优点在于其实现简单，在模型拟合领域的应用范围广。其他优点包括计算效率；采样方法为所有可能的特征组合提供了更好的解决问题的替代方法。在某些情况下，使用 hough 变换而不是 ransac 会更有效：

- 1：参数数目较少，例如，使用 Hough 变换可以有效地实现线性模型估计（2 个参数），而图像拼接则需要更节省的计算方法，如 RANSAC。
- 2：如果噪声总体很高，正如我们前面看到的，噪声的增加需要更广泛的采样方法（更高的采样数），从而增加了计算成本。增加的噪声降低了参数估计的正确率和内分类的准确性。

高噪声数据中的低性能是 RANSAC 的主要限制；这一点尤其重要，因为现实问题的异常值率很高。

(可参考：RANSAC <https://zhuanlan.zhihu.com/p/45532306> )