

公告

Since 2017.05.21:

Visitors

CN 127,594	GB 975
US 5,440	DE 728
TW 2,660	AU 626
HK 1,972	SG 616
JP 1,406	CA 541

Pageviews: 241,370

FLAG counter

昵称：桂。  
园龄：1年6个月  
粉丝：237  
关注：16  
[+加关注](#)

< 2018年7月 >						
日	一	二	三	四	五	六
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

最新随笔

- 1. Mathematica简介
- 2. 概率论02
- 3. 概率论01
- 4. 小数延迟滤波器的诸多工程问题及改进措施
- 5. coon's patch
- 6. system generator学习笔记【02】
- 7. CZT变换 ( chirp z-transform )
- 8. system generator学习笔记【01】
- 9. 硬件资源拆解
- 10. 基础008\_定浮点转化[floating point IP]

随笔分类

- 00-C 语言(2)
- 01-MATLAB(12)
- 02-Python(9)
- 03-前端相关(1)
- 10-HTK(1)
- 11-Linux(3)
- 12-Tensorflow(2)
- 13-scikit-learn(4)

随笔-242 文章-7 评论-157

随机抽样一致算法 ( Random sample consensus , RANSAC )

作者：桂。  
时间：2017-04-25 21:05:07  
链接：<http://www.cnblogs.com/xingshansi/p/6763668.html>



前言

仍然是昨天的问题，别人问到最小二乘、霍夫变换、RANSAC在直线拟合上的区别。昨天梳理了霍夫变换，今天打算抽空梳理一下RANSAC算法，主要包括：

- 1 ) RANSAC理论介绍
- 2 ) RANSAC应用简介；

内容为自己的学习记录，其中很多地方借鉴了别人，最后一起给出链接。

一、RANSAC理论介绍

普通最小二乘是保守派：在现有数据下，如何实现最优。是从一个整体误差最小的角度去考虑，尽量谁也不得罪。

RANSAC是改革派：首先假设数据具有某种特性（目的），为了达到目的，适当割舍一些现有的数据。

给出最小二乘拟合（红线）、RANSAC（绿线）对于一阶直线、二阶曲线的拟合对比：

14-工具使用(16)  
 15-音频分析工具(2)  
 16-硬件(39)  
 20-音频信号(30)  
 21-信号处理(55)  
 22-图像(3)  
 23-模式识别(18)  
 24-矩阵(18)  
 25-测向算法(32)  
 26-电磁(2)  
 27-数学(2)  
 待归档(5)  
 读书(26)  
 工作记录(12)  
 随手记(9)

## 随笔档案

2018年6月 (4)  
 2018年5月 (22)  
 2018年4月 (4)  
 2018年3月 (2)  
 2018年2月 (12)  
 2018年1月 (4)  
 2017年12月 (3)  
 2017年11月 (11)  
 2017年10月 (13)  
 2017年9月 (14)  
 2017年8月 (18)  
 2017年7月 (14)  
 2017年6月 (16)  
 2017年5月 (37)  
 2017年4月 (29)  
 2017年3月 (25)  
 2017年2月 (10)  
 2017年1月 (4)

## 积分与排名

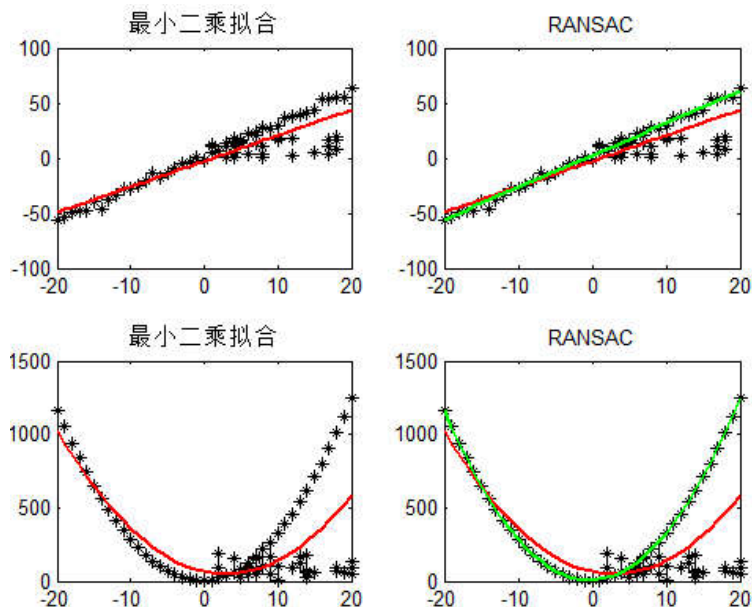
积分 - 142061  
 排名 - 2143

## 最新评论

1. Re:信号处理——EMD、VMD的一点小思考  
 @Matti太久没用都记不清了。不过可以这样：计算每个IMF的瞬时频率，sort排序，人为统一起来。...

--桂。

2. Re:信号处理——EMD、VMD的一点小思考



可以看到RANSAC可以很好的拟合。RANSAC可以理解作为一种采样的方式，所以对于多项式拟合、混合高斯模型 (GMM) 等理论上都是适用的。

RANSAC的算法大致可以表述为 (来自wikipedia)：

Given:

data - a set of observed data points  
 model - a model that can be fitted to data points  
 n - the minimum number of data values required to fit the model  
 k - the maximum number of iterations allowed in the algorithm  
 t - a threshold value for determining when a data point fits a model  
 d - the number of close data values required to assert that a model fits well to data

Return:

bestfit - model parameters which best fit the data (or nul if no good model is found)

iterations = 0

bestfit = nul

besterr = something really large

while iterations < k {

    maybeinliers = n randomly selected values from data

    maybeinliers = model parameters fitted to maybeinliers

    alsoinliers = empty set

    for every point in data not in maybeinliers {

        if point fits maybeinliers with an error smaller than t

            add point to alsoinliers

    }

我看到你EMD和VMD对比图中，VMD分解的各个IMF是按照频率由大到小排列，而EMD则是反的。但是我之前自己做EMD的时候IMF也是频率由大到小，我对EMD了解的不是太多，想问下EMD和VMD分解的.....

--Matti

3. Re:空间谱专题10：MUSIC算法

@桂。好的~多谢回复...

--7yen

4. Re:空间谱专题10：MUSIC算法

@7yen你的情况我不确定。可以先分析下采集的数据，验证一下自己的假设是否合理...

--桂。

5. Re:空间谱专题10：MUSIC算法

博主，你好，有个问题想请教一下，混响情况下MUSIC算法性能不好便是因为你上面说的相干情况存在吗？

--7yen

#### 阅读排行榜

1. python音频处理用到的操作(21199)
2. 音频特征提取——librosa工具包使用(15280)
3. 信号处理——Hilbert变换及谱分析(13057)
4. PCA算法(11138)
5. MATLAB ( 2 ) ——小波工具箱使用简介(10854)

#### 推荐排行榜

1. 信号处理——EMD、VMD的一点小思考(7)
2. 音频特征提取——常用音频特征(6)
3. python音频处理用到的操作(4)
4. 霍夫变换(3)
5. 混合高斯模型 ( GMM ) 推导及实现(3)

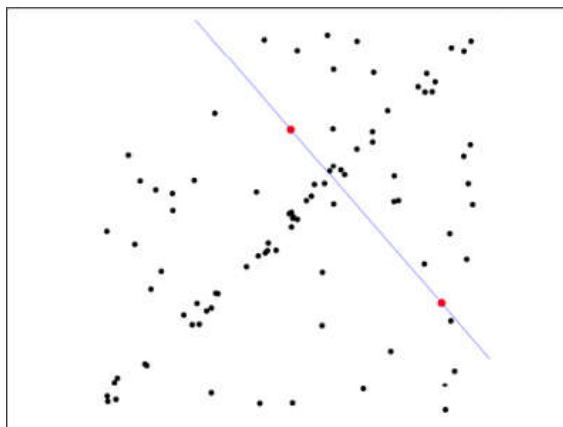
```

if the number of elements in alsoinliers is > d {
    % this implies that we may have found a good model
    % now test how good it is
    bettermodel = model parameters fitted to all
points in maybeinliers and alsoinliers
    thiserr = a measure of how well model fits these
points
    if thiserr < besterr {
        bestfit = bettermodel
        besterr = thiserr
    }
    increment iterations
}
return bestfit

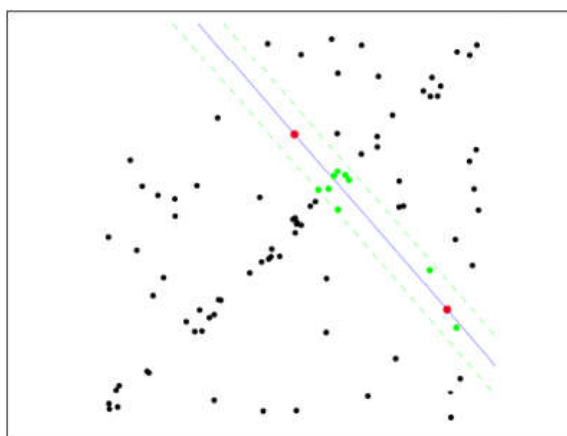
```

RANSAC简化版的思路就是：

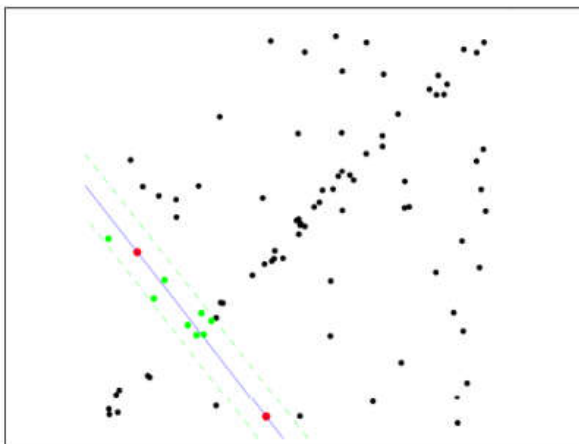
**第一步：**假定模型（如直线方程），并随机抽取Nums个（以2个为例）样本点，对模型进行拟合：



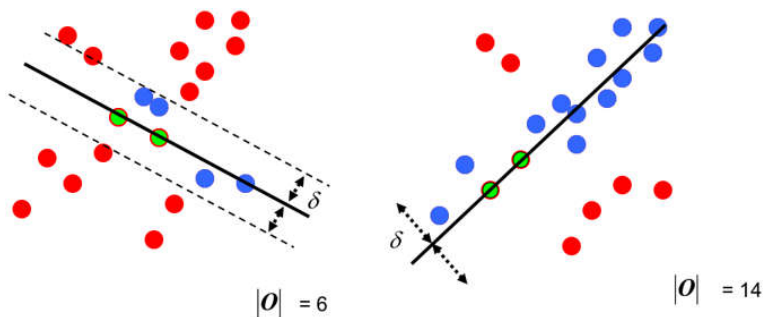
**第二步：**由于不是严格线性，数据点都有一定波动，假设容差范围为： $\sigma$ ，找出距离拟合曲线容差范围内的点，并统计点的个数：



**第三步：**重新随机选取Nums个点，重复第一步~第二步的操作，直到结束迭代：



第四步：每一次拟合后，容差范围内都有对应的数据点数，找出数据点个数最多的情况，就是最终的拟合结果：



至此：完成了RANSAC的简化版求解。

这个RANSAC的简化版，只是给定迭代次数，迭代结束找出最优。如果样本个数非常多的情况下，难不成一直迭代下去？其实RANSAC忽略了几个问题：

- 每一次随机样本数**Nums**的选取：如二次曲线最少需要3个点确定，一般来说，Nums少一些易得出较优结果；
- 抽样迭代次数**Iter**的选取：即重复多少次抽取，就认为是符合要求从而停止运算？太多计算量大，太少性能可能不够理想；
- 容差**Sigma**的选取：sigma取大取小，对最终结果影响较大；

这些参数细节信息参考：[维基百科](#)。

**RANSAC的作用有点类似：将数据一切两段，一部分是自己人，一部分是敌人，自己人留下商量事，敌人赶出去。RANSAC开的是家庭会议，不像最小二乘总是开全体会议。**

附上最开始一阶直线、二阶曲线拟合的code(只是为了说明最基本的思路，用的是**RANSAC的简化版**)：

一阶直线拟合：

```
1  clc;clear all;close all;
2  set(0,'defaultfigurecolor','w');
3  %Generate data
4  param = [3 2];
5  npa = length(param);
6  x = -20:20;
```

```
7 y = param*[x; ones(1,length(x))]+3*randn(1,length(x));
8 data = [x randi(20,1,30);...
9         y randi(20,1,30)];
10 %figure
11 figure
12 subplot 221
13 plot(data(1,:),data(2:,:), 'k*');hold on;
14 %Ordinary least square mean
15 p = polyfit(data(1,:),data(2,:),npa-1);
16 flms = polyval(p,x);
17 plot(x,flms,'r','linewidth',2);hold on;
18 title('最小二乘拟合');
19 %Ransac
20 Iter = 100;
21 sigma = 1;
22 Nums = 2;%number select
23 res = zeros(Iter,npa+1);
24 for i = 1:Iter
25     idx = randperm(size(data,2),Nums);
26     if diff(idx) ==0
27         continue;
28     end
29     sample = data(:,idx);
30     pest = polyfit(sample(1,:),sample(2,:),npa-1);%parameter est
31     res(i,1:npa) = pest;
32     res(i,npa+1) = numel(find(abs(polyval(pest,data(1,:))-data(2
33     end
34     [~,pos] = max(res(:,npa+1)));
35     pest = res(pos,1:npa);
36     fransac = polyval(pest,x);
37 %figure
38 subplot 222
39 plot(data(1,:),data(2:,:), 'k*');hold on;
40 plot(x,flms,'r','linewidth',2);hold on;
41 plot(x,fransac,'g','linewidth',2);hold on;
42 title('RANSAC');
```

二阶曲线拟合：

[+ View Code](#)

## 二、RANSAC应用简介

RANSAC其实就是一种采样方式，例如在图像拼接（Image stitching）技术中：

**第一步：**预处理后（据说桶形变换，没有去了解过）提取图像特征（如SIFT）



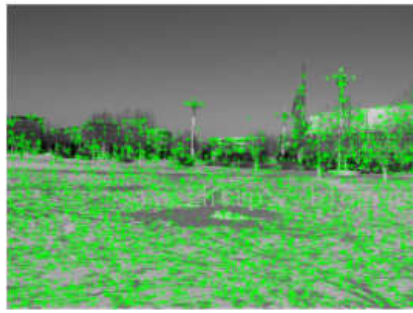


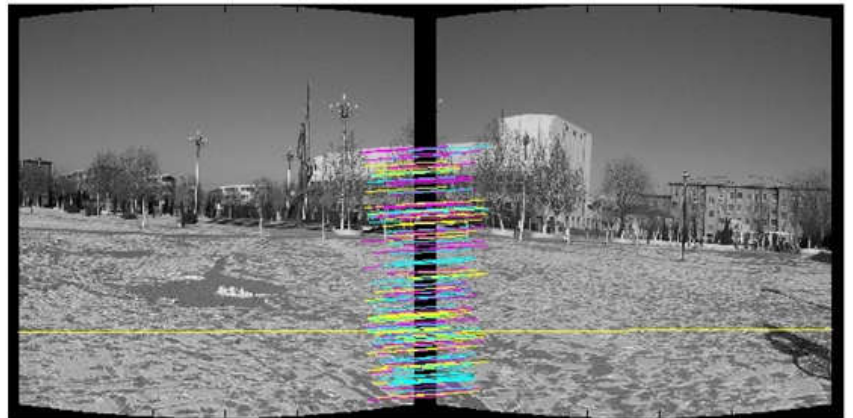
图 1 的 sift 特征



图 2 的 sift 特征

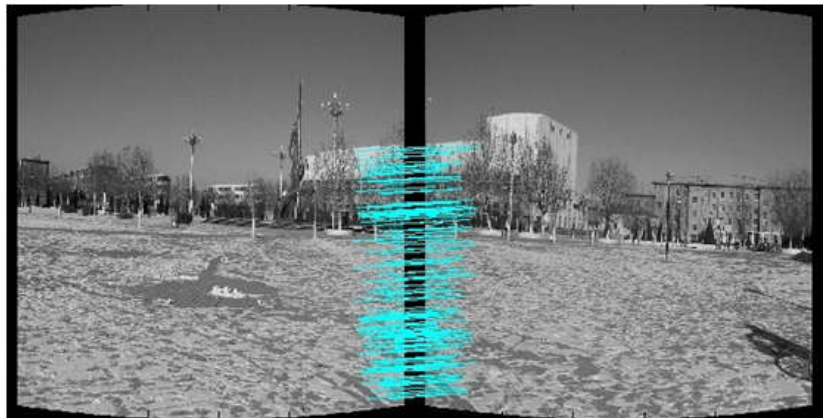
**第二步：**特征点进行匹配，可利用归一化互相关(Normalized Cross Correlation method, NCC)等方法。

但这个时候会有很多匹配错误的点：



这就好比拟合曲线，有很多的误差点，这个时候就想到了RANSAC算法：我不要再兼顾所有了，每次选取Nums个点匹配 → 计算匹配后容差范围内的点数 → 重复实验，迭代结束后，找出点数最多的情况，就是最优的匹配。

利用RANSAC匹配：



**第三步：**图像拼接，这个就涉及拼接技术了，直接给出结果：



参考：

- RANSAC :  
[https://en.wikipedia.org/wiki/Random\\_sample\\_consensus](https://en.wikipedia.org/wiki/Random_sample_consensus)
- 图像拼接 :  
<http://blog.csdn.net/xiaoch1222/article/details/53510895>

分类: [22-图像](#), [21-信号处理](#)

标签: [RANSAC](#), [随机采样一致](#)

好文要顶

关注我

收藏该文



桂。

关注 - 16

粉丝 - 237

+加关注

20

« 上一篇: [霍夫变换](#)  
» 下一篇: [统计学习方法：核函数 \( Kernel function \)](#)  
posted @ 2017-04-25 22:13 桂。 阅读(4929) 评论(1) 编辑 收藏

评论列表

#1楼 2018-04-10 15:05 没事敲敲代码  
非常直观，感谢分享  
支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库！

【推荐】腾讯云新注册用户域名抢购1元起

【大赛】2018首届“顶天立地”AI开发者大赛

 腾讯云

40+热门云产品0门槛体验

云服务器、云存储等  
最长免费畅享180天

立即体验