



南京航空航天大学

NANJING UNIVERSITY OF  
AERONAUTICS AND ASTRONAUTICS

# 机器视觉测量与建模

Machine vision based surveying and modelling



李明磊

南京航空航天大学 电子信息工程学院

E-mail: minglei\_li@nuaa.edu.cn

1



## 4.特征提取与匹配

4.1 梯度信息提取

4.2 点特征提取

4.3 特征匹配技术



• 特征提取与匹配

- 剩下的问题
  - 特征点怎么来？
    - 特征点提取算法（Harris、SIFT、SURF、ORB等）
  - 特征提取算法还会计算每个特征点周围的图像统计信息
    - 描述子（Descriptor）
  - 描述子能够区别每个关键点——可以用于匹配

- Harris角点检测
- SIFT
- **SURF**
- **FAST**
- **ORB**



• 特征提取与匹配

- **SURF**

Speeded Up Robust Features  
针对SIFT做的改进算法

Herbert Bay 等人（ETH），  
2006年发表在ECCV大会上  
on a GPU, SURF can be extracted for video  
size images @ 100Hz with ease

- SIFT： 使用**DoG 近似 LoG** 建立尺度空间
- SURF： 使用**Box Filter 近似 LoG**

## • 特征提取与匹配

### • SURF

Speeded Up Robust Features (2006, ECCV)

SURF采用海森矩阵（Hessian Matrix）**H**的行列式的近似值建立影像差分空间。

Hessian 矩阵是一个多元函数的二阶偏导数构成的方阵，描述了函数的局部曲率，由德国数学家Ludwin O. Hessian于十九世纪提出。

连续函数 $f(x, y)$ 的二阶微分Hessian矩阵为：
$$\mathbf{H}_{f(x,y)} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

定理：

矩阵主对角线上的元素和等于矩阵特征值之和。

•海森矩阵也可以理解为多元函数泰勒展开后的二阶导系数矩阵

$$\text{Laplace} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ = D_{xx} + D_{yy}$$

Laplace算子 = Hessian矩阵的特征值之和

黑塞矩阵常用于[牛顿法](#)解决优化问题，利用黑塞矩阵可判定多元函数的[极值](#)问题。

李明磊@nuua

5

## • 特征提取与匹配

### • SURF

Speeded Up Robust Features

连续函数 $f(x, y)$ 的二阶微分Hessian矩阵为：
$$\mathbf{H}_{f(x,y)} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

•如果函数 $f$ 在 $D$ 区域内二阶连续可导, 那么 $f$ 的Hessian矩阵（海森矩阵） $\mathbf{H}_f$ 在 $D$ 内为**对称矩阵**。原因是: 如果函数 $f$ 的二阶偏导数连续, 则二阶偏导数的求导顺序没有区别, 即

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial}{\partial x} \left( \frac{\partial f}{\partial y} \right) = \frac{\partial}{\partial y} \left( \frac{\partial f}{\partial x} \right)$$

**正定矩阵:**

设 $M$ 是 $n$ 阶方阵, 如果对任何非零向量 $z$ , 都有 $z^T M z > 0$ , 其中 $z^T$ 表示 $z$ 的转置, 就称 $M$ 为正定矩阵。

若 $A$ 的特征值均为正数, 则 $A$ 是正定的; 若 $A$ 的特征值均为负数, 则 $A$ 为负定的。

李明磊@nuua

6

## • 特征提取与匹配

### • SURF

SURF采用海森矩阵（Hessian Matrix）

**H**的行列式的近似值建立影像差分空间。

Hessian矩阵描述函数的局部曲率。

设 $f$ 为多元实函数，在点 $\mathbf{p}$ 的邻域内有二阶连续偏导。 $\frac{\partial f}{\partial x_i} |_{\mathbf{p}(x_1, \dots, x_i, \dots)} = 0$ 。

当 $\mathbf{H}_f$ 为正定矩阵时， $\mathbf{p}$ 点处是极小值；

当 $\mathbf{H}_f$ 为负定矩阵时， $\mathbf{p}$ 点处是极大值；

当 $\mathbf{H}_f$ 为非定矩阵时， $\mathbf{p}$ 点不是极值点；

正定矩阵：复习线性代数

连续函数 $f(x, y)$ 的二阶微分  
Hessian矩阵为：

$$\mathbf{H}_{f(x,y)} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

利用Hessian矩阵的行列式  
来近似寻找差分极值：

$$\det(\mathbf{H}) = \frac{\partial^2 f}{\partial x^2} \frac{\partial^2 f}{\partial y^2} - \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2$$

李明磊@nuaa

7

### • SURF

步骤：

- 进行高斯滤波，再构造Hessian矩阵

$$\mathbf{H}(X, \sigma) = \begin{pmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{pmatrix}$$

$$L_{xx} = \frac{\partial^2 G(x, y, \sigma)}{\partial x^2} * I(x, y)$$

$$\det(\mathbf{H}_{approx}) = L_{xx}L_{yy} - (wL_{xy})^2$$

其中 $w$ 是因为 $\text{box filters}$ 是高斯二阶差分的近似，为了使行列式的值大致相等，乘以权值，取0.9。

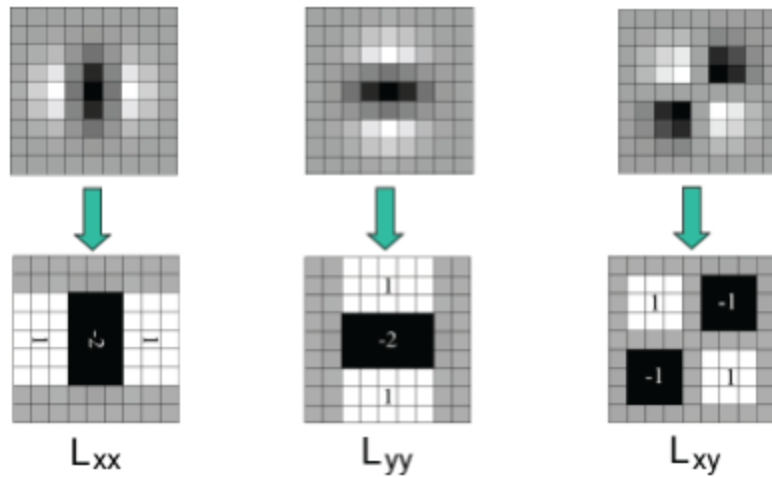
- 构造尺度空间  $\longrightarrow$  不同尺度的盒子滤波器与原图像卷积
- 在影像空间中找极值点，定位特征点
- 确定特征点主方向  $\longrightarrow$  统计特征点领域内的Harr小波特征
- 构造特征描述子  $\longrightarrow$  在特征点周围取 $4 \times 4$ 的正方形框，每个小区域有4个值  $4 \times 4 \times 4 = 64$ 维

李明磊@nuaa

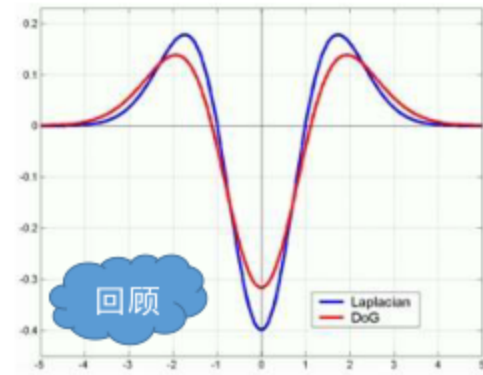
8

# • SURF

使用了盒式滤波器（Box-filter）来近似替代来近似处理高斯二阶滤波器，提高运算速度



高斯滤波器 ( $L_{yy}$ 、 $L_{xy}$ ) 与盒式滤波器 ( $D_{yy}$ 、 $D_{xy}$ ) 的比较

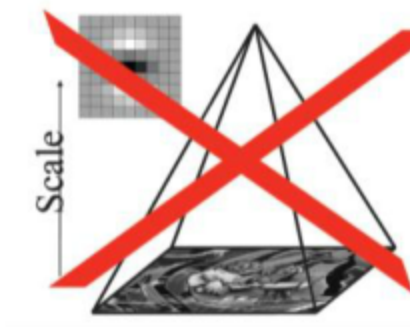


高斯拉普拉斯和高斯差分的比较

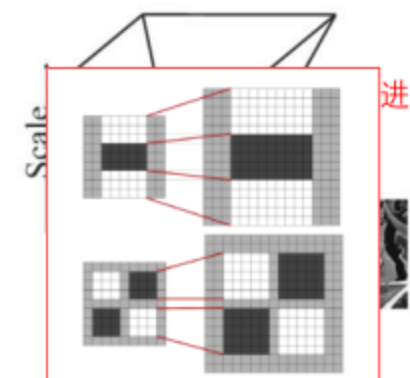
李明磊@nuaa

9

# • SURF



SIFT算法中，尺度空间影像是高斯函数与原影像函数的卷积，高斯函数的 $\sigma$ 越大，则尺度越大，尺度越大则影像越模糊。在SIFT中，尺度影像的金字塔每一层的建立都要在上一层结束之后才开始，所以这样非常没有效率，导致运行速度很慢。



SURF算法构建的尺度空间不改变影像尺寸的大小，而是通过改变模板的大小，对影像进行滤波构造尺度空间。同时可以对金字塔中的每层进行处理，大大地节省了时间。

计算行列式的时候，scale增加，采样的间隔提高一倍。例如，Octave1，每个点都计算；Octave2 隔点计算。（用增大模板大小，对图像上的点采样计算的方法，等同于实现对图像进行下采样并改变模板尺度的大小。）

李明磊@nuaa

10

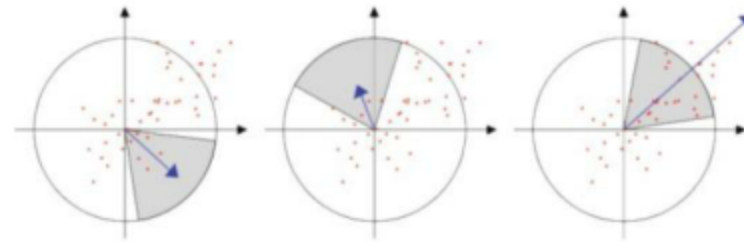
• SURF

提取主方向→旋转不变性

统计特征点圆形邻域内的Haar小波特征。以特征点为中心，计算半径为 $6s$ （ $s$ 为特征点所在的尺度值）的圆形邻域内的点在 $x$ 、 $y$ 方向的Haar小波响应（Haar小波边长取 $4s$ ），Haar小波模板如图所示：



统计特征点圆形邻域内的Haar小波特征，即在特征点的圆形邻域内，统计60度扇形内所有点的水平、垂直Haar小波特征总和。然后扇形以0.2弧度大小的间隔，进行旋转并再次统计该区域内Haar小波特征值。最后将值最大的那个扇形的方向作为该特征点的主方向。



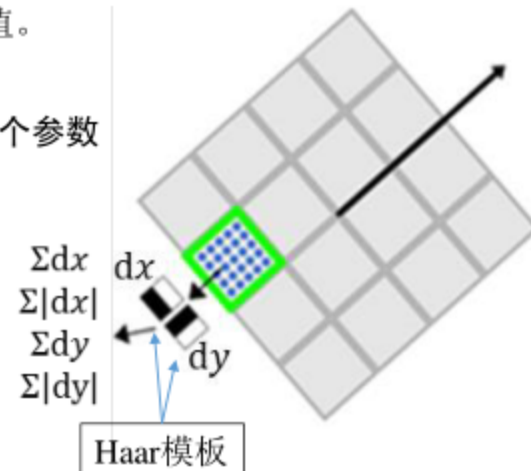
李明磊@nuaa

11

• SURF

- **SIFT算法**：提取特征点周围 $4 \times 4$ 个区域块，统计每小块内8个梯度方向→ $4 \times 4 \times 8 = 128$ 维向量。
- **SURF算法**：在特征点周围一个方形区域内进行的，大小为 $20s \times 20s$ ，但是所取得矩形区域方向是与特征点主方向平行的。周围分成 $4 \times 4$ 个子块，每个子块包含 $5s \times 5s$ 。统计每个子块内的像素点的Haar模板沿主方向和垂直方向上的响应，并统计响应值。
- 有16个子块
- 每个子块统计 $\Sigma dx$ 、 $\Sigma |dx|$ 、 $\Sigma dy$ 和 $\Sigma |dy|$  4个参数
- 特征点描述子由64维向量构成

Haar小波特征为水平方向值之和、垂直方向值之和、水平方向值绝对值之和以及垂直方向绝对值之和4个方向。



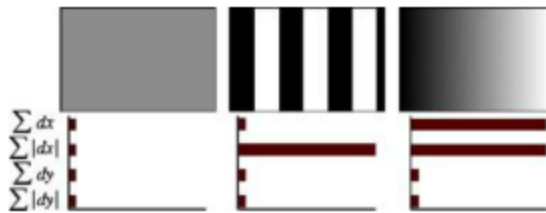
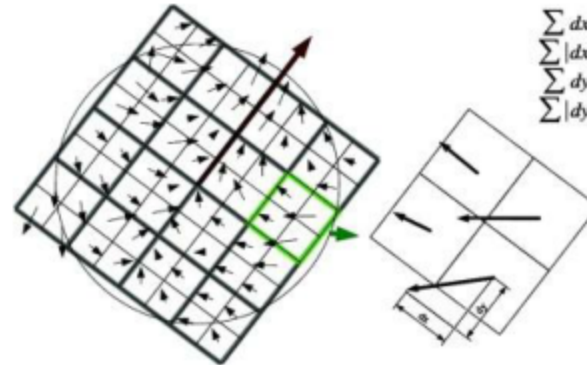
李明磊@nuaa

12





- SURF



The left image shows the properties of the descriptor for three distinctively different image-intensity patterns within a sub-region. One can imagine combinations of such local intensity patterns, resulting in a distinctive descriptor.

论文原文<https://www.sciencedirect.com/science/article/pii/S1077314207001555>

13



- 特征提取与匹配

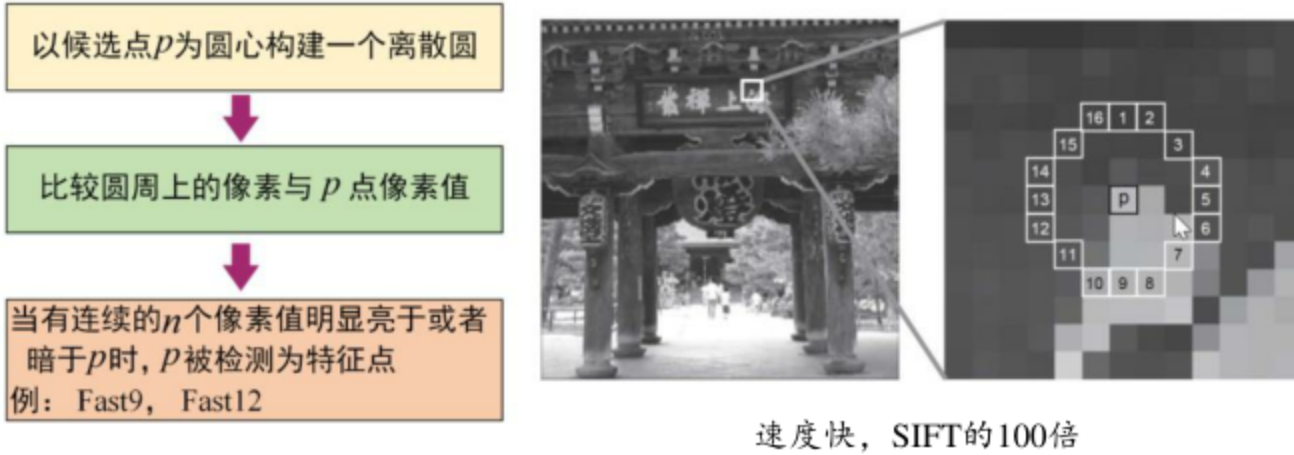
- FAST

FAST(Features from Accelerated Segment Test)

由Edward Rosten和Tom Drummond在2006年首先提出

## FAST

通过检测局部像素灰度变化来确认特征点的位置



假如选取的圆上, 有连续的  $N$  个点的亮度大于  $I_p + T$  或小于  $I_p - T$ , 那么像素  $p$  可以被认为是特征点 (一般  $N$  取 9 或 12)

李明磊@nuua

15

## FAST

原始的FAST角点经常出现“扎堆”的现象。

非极大值抑制

——解决从邻近的位置选取了多个特征点的问题

- 为每一个检测到的特征点计算它的响应大小  $V$ 。  $V$  定义为点  $P$  和它周围 16 个像素点的绝对偏差的和
- 考虑两个相邻的特征点, 并比较它们的  $V$  值
- $V$  值较低的点将会被删除



**FAST角点不具有尺度和旋转不变性!**

李明磊@nuua

16





## • ORB

Oriented FAST (特征点) 和 Rotated BRIEF (特征描述)

**ORB** = **Oriented FAST** + **Rotated BRIEF**

改进  
FAST

尺度不变性 → 构建**图像金字塔**，在金字塔每一层上检测关键点

旋转不变性 → 通过**灰度质心法**确定特征点方向

**灰度质心法**是假设某特征点的灰度与该邻域重心之间存在偏移，通过这个特征点到重心的向量，就能算出该特征点的主方向

将邻域矩 (moment) 定义为:  $C_{mn} = \sum_{x,y \in B} x^m y^n I(x,y) \quad m,n = \{0,1\}$

邻域质心为:  $\mathbf{o} = \left( \frac{C_{10}}{C_{00}}, \frac{C_{01}}{C_{00}} \right) \quad \theta = \arctan\left(\frac{C_{01}}{C_{10}}\right)$

李明磊@nuaa

17



## • ORB

### **BRIEF**

Binary Robust Independent Elementary Features

Calonder 在ECCV2010上提出的BRIEF描述子采用二进制码串(每一位非1即0)作为描述子向量，论文中考虑长度有128,256,512几种

由于BRIEF特征描述子是非常简单的0和1串，算法过程简单，可以用汉明距离 (Hamming distance) 来计算特征之间的相似度实现匹配搜索。

Hamming distance: 一个串变成另一个串所需要的最小替换次数

**原始的BRIEF描述子并不支持旋转**

汉明距离(Hamming Distance)表示两个(相同长度)字符串对应位不同的数量。

李明磊@nuaa

18



## • ORB

### BRIEF

Binary Robust Independent Elementary Features

- 先平滑影像，然后在特征点周围选择一个子图块（Patch），例如 $S \times S$ （ $48 \times 48$ ）大小
- 在这个Patch内，通过一种选定的方法（论文里提供了5种），来挑选出来 $n$ 个点对 $(p, q)$ 点对
- 对于每一个点对比较亮度值，如果 $I(p) > I(q)$ ，则这个点对生成了二值化向量中一个的值为1，否则否则为0
- 所有 $n$ 个点对，都进行比较之间，就生成了一个二进制串

原始的BRIEF描述子并不支持旋转

李明磊@nuaa

19



## • ORB

ORB = Oriented FAST + Rotated BRIEF

改进BRIEF描述子 Steer BRIEF → 给BRIEF加上旋转不变性

对于一个点 $(x_i, y_i)$ 上由 $n$ 个二值测试生成的任一特征集，定义一下size为 $2 \times n$ 的矩阵  $S = \begin{pmatrix} x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{pmatrix}$

该图像块的方向 $\theta$ 为 计算图像块的方向（数学中的矩 moment）。  
In mathematics, a moment is a specific quantitative measure, used in both mechanics and statistics, of the shape of a set of points.

利用特征点的方向角 $\theta$  形成旋转矩阵 $R_\theta$

$$R_\theta = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \quad S_\theta = R_\theta S$$

图 5-10 ORB 特征点提取



在新的点集位置上比较点对的大小形成二进制串的描述符，即BRIEF描述子

李明磊@nuaa

20



## 数学中的矩 moment

### 图像的矩

#### 零阶矩:

$$M_{00} = \sum_I \sum_J V(i, j)$$

这里的图像是单通道图像,  $V(i, j)$  表示图像在  $(i, j)$  点上的灰度值。

我们可以发现, 当图像为二值图时,  $M_{00}$  就是这个图像上白色区域的总和, 因此,  $M_{00}$  可以用来求二值图像 (轮廓, 连通域) 的面积。

#### 一阶矩:

$$M_{10} = \sum_I \sum_J i \cdot V(i, j)$$

$$M_{01} = \sum_I \sum_J j \cdot V(i, j)$$

当图像为二值图时,  $V(i, j)$  只有0 (黑), 1 (白) 两个值。  $M_{10}$  就是图像上所有白色区域 x 坐标值的累加。因此, 一阶矩可以用来求二值图像的重心:

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}$$

李明磊@nuaa

21



### 二阶矩

$$M_{20} = \sum_I \sum_J i^2 \cdot V(i, j)$$

$$M_{02} = \sum_I \sum_J j^2 \cdot V(i, j)$$

$$M_{11} = \sum_I \sum_J i \cdot j \cdot V(i, j)$$

二阶矩可以用来求物体形状的方向。

$$\theta = \frac{1}{2} \text{fastAtan2}(2b, a - c)$$

其中:  $a = \frac{M_{20}}{M_{00}} - x_c^2$ ,  $b = \frac{M_{11}}{M_{00}} - x_c y_c$ ,  $c = \frac{M_{02}}{M_{00}} - y_c^2$

fastAtan2() 为 opencv 的函数, 输入向量, 返回一个 0-360 的角度。

这里修改一下, 我之前在看二阶矩求物体形状方向的时候, 有公式是这么写的:

$$\theta = \frac{1}{2} \arctan\left(\frac{2b}{a - c}\right)$$

李明磊@nuaa

22

### BRIEF特征描述子

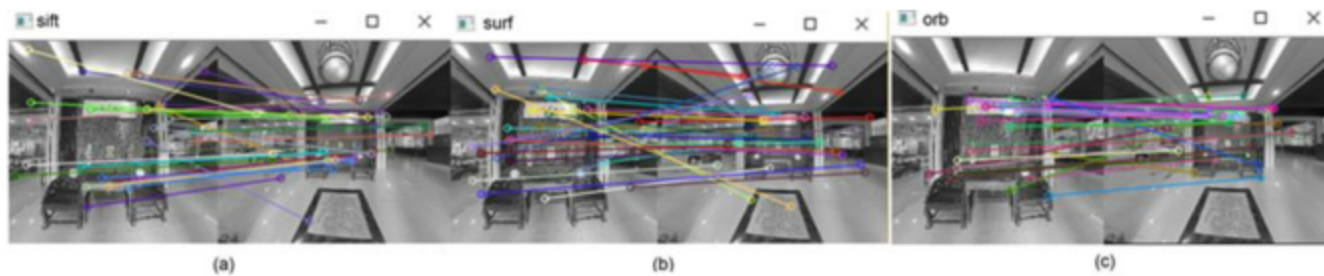
原始的特征描述子使用浮点数，内存占用：

SIFT 128维，1个点占用512字节

SURF 64维，1个点占用256字节

BRIEF 如果取 $N=128$ ，每个特征点需要 $128/8=16$ 个字节内存

- 两个特征编码对应bit位上相同元素的个数小于64的，一定不是配对的。
- 一幅图上特征点与另一幅图上特征编码对应bit位上相同元素的个数最多的特征点配成一对。



来源网络图片

李明磊@nuua

23

## 4.特征提取与匹配

### 4.1 梯度信息提取

### 4.2 点特征提取

### 4.3 特征匹配技术

李明磊@nuua

24

## 4.3 特征匹配技术

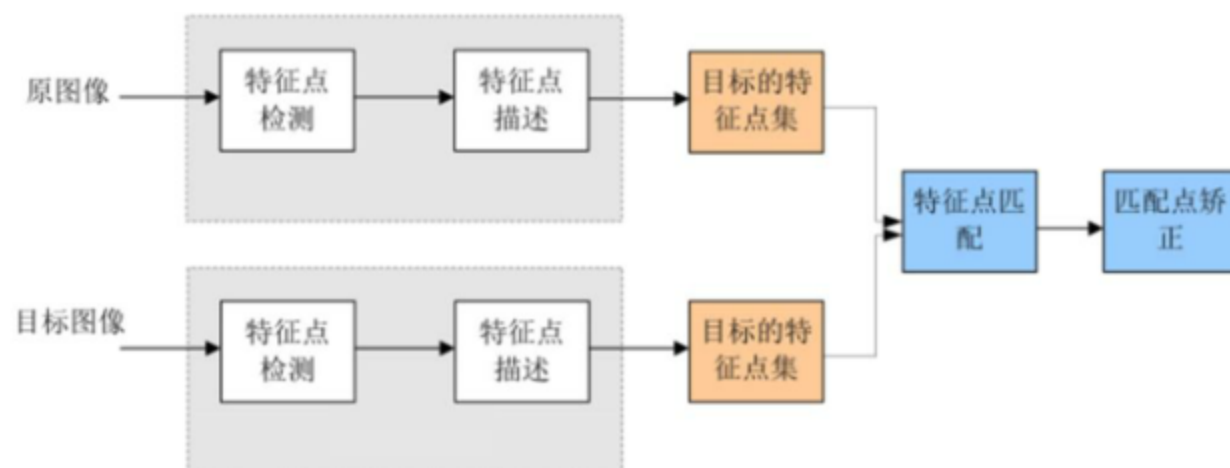


*If we know where the good features are,  
how do we match them?*

李明磊@nuaa

25

## 4.3 特征匹配技术



- 目的：通过描述子的差异把两张图中相同的特征点一一对应起来

李明磊@nuaa

26



## 特征匹配

暴力匹配  
(BF)

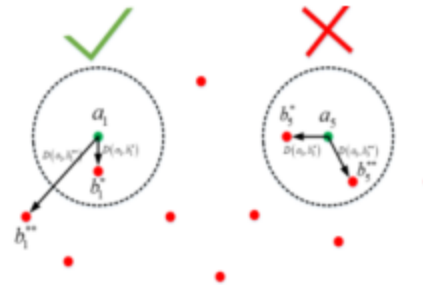
对左图中的每一个特征点，计算它和右图所有特征点的相似程度（特征描述子的距离），然后按照相似程度进行排序，取最相似的那个作为匹配点。

计算复杂，误匹配率高

## 握手匹配

在暴力匹配基础上又用右图每个特征点和左图的所有特征点进行匹配，取这两个匹配对网的交集。

计算复杂度至少是暴力匹配的2倍

快速近似  
最近邻匹配

$$b^* = \arg \min_{b \in B} D(a, b), D(a, b^*) < \beta$$

$$\frac{D(a, b^*)}{D(a, b^{**})} < \alpha$$

最近邻距离和次近邻距离比小于一定值

李明磊@nuaa

27

## Comparing image regions

## Compare intensities pixel-by-pixel



差异性测度

Dissimilarity measures

Sum of Square Differences

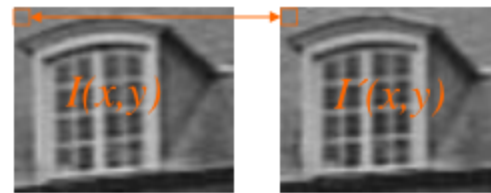
$$SSD = \iint_W [I'(x, y) - I(x, y)]^2 dx dy$$

李明磊@nuaa

28



## Compare intensities pixel-by-pixel



相似性测度

## Similarity measures

### Zero-mean Normalized Cross Correlation

归一化互相关

$$NCC = \frac{N(I', I)}{\sqrt{N(I', I')N(I, I)}}$$

$$\text{其中 } N(I, I') = \iint_W (I(x, y) - \bar{I})(I'(x, y) - \bar{I}') dx dy$$

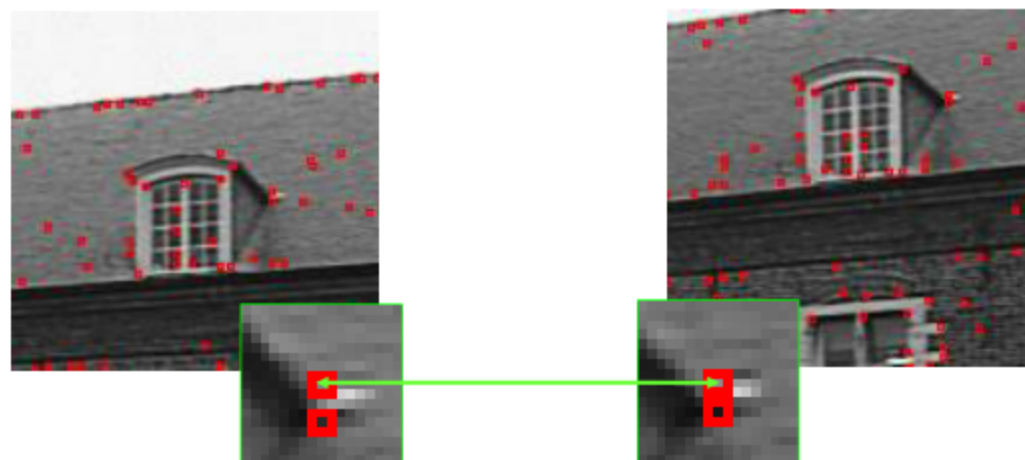
李明磊@nuaa

29



## Simple matching

- For each corner in image 1, find the corner in image 2 that is most similar (using SSD or NCC) and vice-versa
- Only compare geometrically compatible points
- Keep mutual best matches

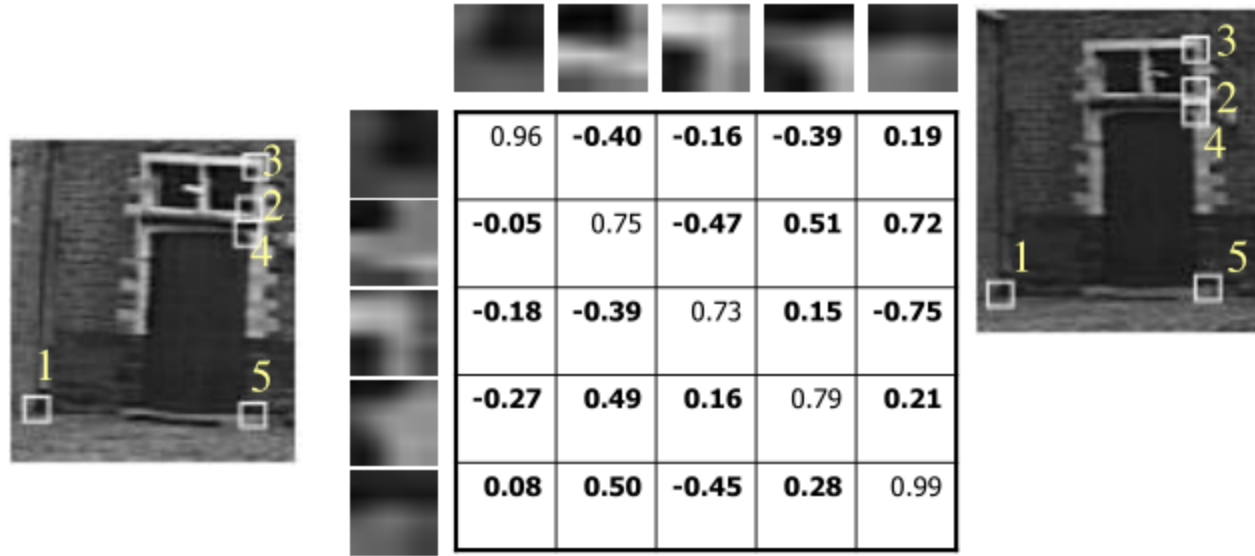


李明磊@nuaa

30



# Feature matching: example



Input images (zip 1.1Mb)



Output panorama 1



Output panorama 2 - Image (Panorama)