



尺度不变特征变换匹配算法 Scale Invariant Feature Transform (SIFT)

1. SIFT简介
2. SIFT算法实现细节
3. SIFT算法的应用领域
4. SIFT算法的扩展与改进

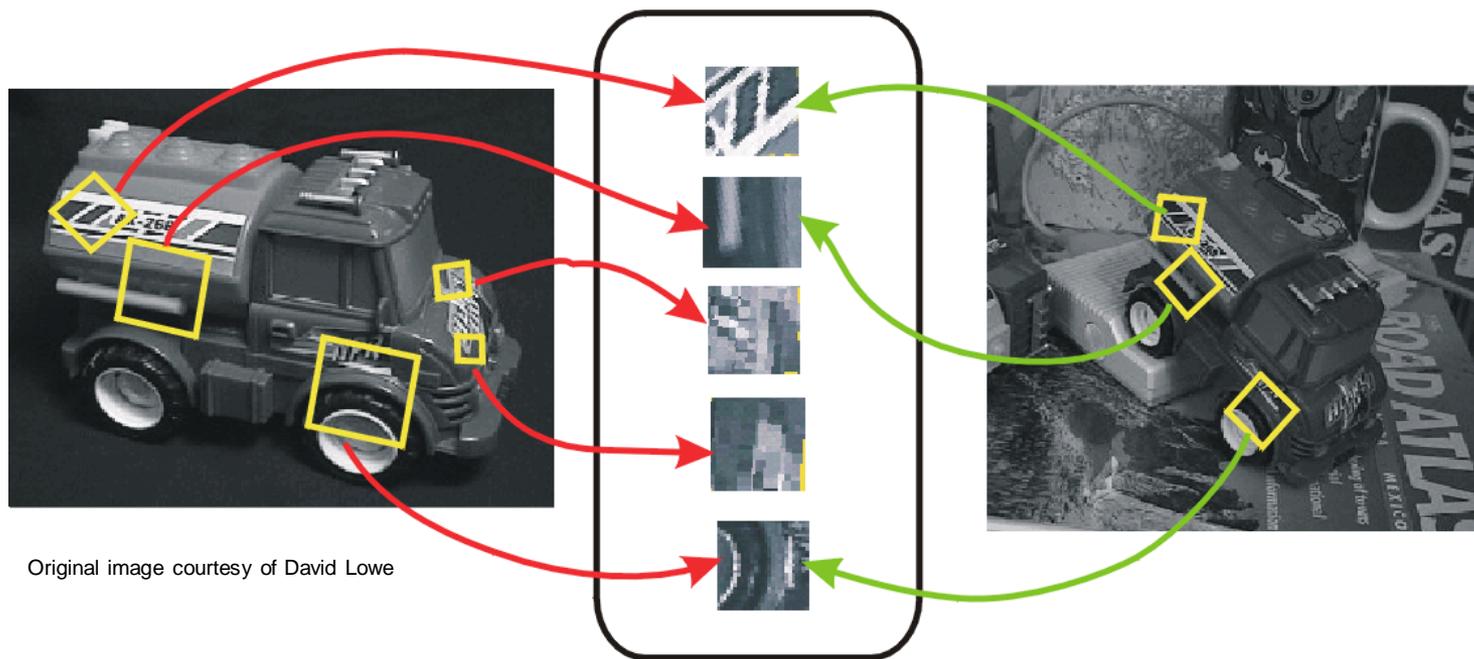
- 传统的特征提取方法
 - 成像匹配的核心问题是将同一目标在不同时间、不同分辨率、不同光照、不同位姿情况下所成的像相对应。传统的匹配算法往往是直接提取角点或边缘，对环境的适应能力较差，急需提出一种鲁棒性强、能够适应不同光照、不同位姿等情况下能够有效识别目标的方法。

- SIFT提出的目的和意义



David G. Lowe
Computer Science Department
2366 Main Mall
University of British Columbia
Vancouver, B.C., V6T 1Z4,
Canada
E-mail: lowe@cs.ubc.ca

- 1999年British Columbia大学大卫·劳伊 (David G. Lowe) 教授总结了现有的基于不变量技术的特征检测方法，并正式提出了一种基于尺度空间的、对图像缩放、旋转甚至仿射变换保持不变性的图像局部特征描述算子—SIFT (尺度不变特征变换)，这种算法在2004年被加以完善。



- 将一幅图像映射（变换）为一个局部特征向量集；特征向量具有平移、缩放、旋转不变性，同时对光照变化、仿射及投影变换也有一定不变性。

● SIFT算法特点

- SIFT特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性。
- 独特性(Distinctiveness)好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配。
- 多量性，即使少数的几个物体也可以产生大量SIFT特征向量。
- 经过优化的SIFT算法可满足一定的速度需求。
- 可扩展性，可以很方便的与其他形式的特征向量进行联合。

- SIFT算法可以解决的问题

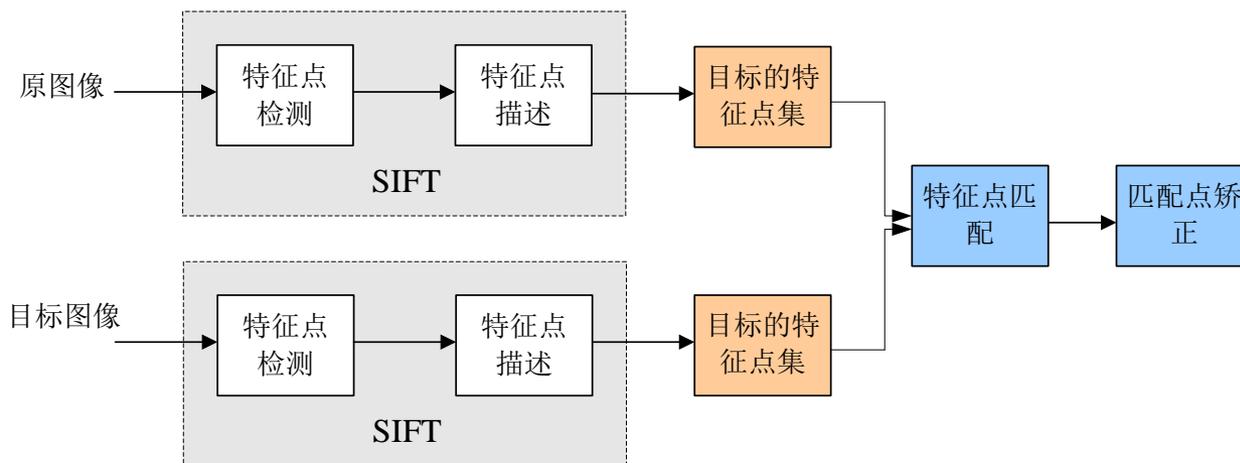
目标的自身状态、场景所处的环境和成像器材的成像特性等因素影响图像配准/目标识别跟踪的性能。而SIFT算法在一定程度上可解决：

- 目标的旋转、缩放、平移 (RST)
- 图像仿射/投影变换 (视点viewpoint)
- 光照影响 (illumination)
- 目标遮挡 (occlusion)
- 杂物场景 (clutter)
- 噪声

[Back](#)

● SIFT算法实现步骤简述

SIFT算法的实质可以归为在不同尺度空间上查找特征点（关键点）的问题。



SIFT算法实现物体识别主要有三大工序，1、提取关键点；2、对关键点附加详细的信息（局部特征）也就是所谓的描述器；3、通过两方特征点（附带特征向量的关键点）的两两比较找出相互匹配的若干对特征点，也就建立了景物间的对应关系。

- SIFT算法实现步骤

1. 关键点检测
2. 关键点描述
3. 关键点匹配
4. 消除错配点

1. 哪些点是SIFT中要查找的关键点（特征点）？

这些点是一些十分突出的点不会因光照条件的改变而消失，比如角点、边缘点、暗区域的亮点以及亮区域的暗点，既然两幅图像中有相同的景物，那么使用某种方法分别提取各自的稳定点，这些点之间会有相互对应的匹配点。

所谓关键点，就是在不同**尺度空间**的图像下检测出的具有方向信息的局部极值点。

根据归纳，我们可以看出特征点具有的三个特征：

尺度

方向

大小

2. 什么是尺度空间 (scale space) ?

我们要精确表示的物体都是通过一定的尺度来反映的。现实世界的物体也总是通过不同尺度的观察而得到不同的变化。

尺度空间理论最早在1962年提出，其主要思想是通过对原始图像进行尺度变换，获得图像多尺度下的尺度空间表示序列，对这些序列进行尺度空间主轮廓的提取，并以该主轮廓作为一种特征向量，实现边缘、角点检测和不同分辨率上的特征提取等。

尺度空间中各尺度图像的模糊程度逐渐变大，能够模拟人在距离目标由近到远时目标在视网膜上的形成过程。

尺度越大图像越模糊。

根据文献《Scale-space theory: A basic tool for analysing structures at different scales》我们可知，高斯核是唯一可以产生多尺度空间的核，一个图像的尺度空间， $L(x, y, \sigma)$ ，定义为原始图像 $I(x, y)$ 与一个可变尺度的2维高斯函数 $G(x, y, \sigma)$ 卷积运算。

高斯函数
$$G(x_i, y_i, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x-x_i)^2 + (y-y_i)^2}{2\sigma^2}\right)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

尺度是自然存在的，不是人为创造的！高斯卷积只是表现尺度空间的一种形式...

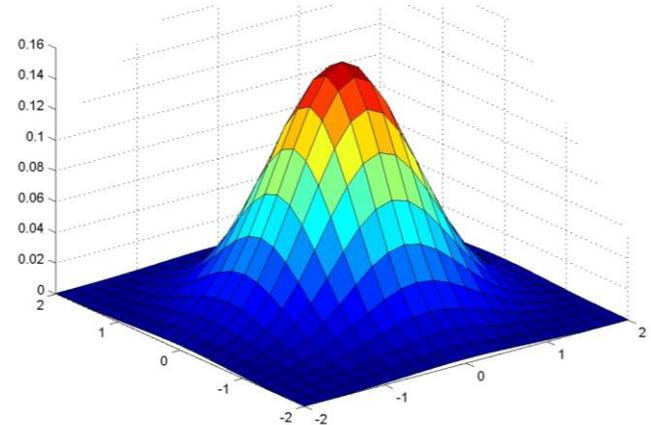
3. 高斯模糊

高斯模糊是在Adobe Photoshop等图像处理软件中广泛使用的处理效果，通常用它来减小图像噪声以及降低细节层次。这种模糊技术生成的图像的视觉效果是好像经过一个半透明的屏幕观察图像。



$$G(r) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

r为模糊半径， $r = \sqrt{x^2 + y^2}$



在减小图像尺寸の場合经常使用高斯模糊。在进行欠采样的时，通常在采样之前对图像进行低通滤波处理。这样就可以保证在采样图像中不会出现虚假的高频信息。

高斯模板大小的选择

高斯模板

0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00038771	0.01330373	0.11098164	0.22508352	0.11098164	0.01330373	0.00038771
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	0.00019117
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00000067	0.00002292	0.00019117	0.00038771	0.00019117	0.00002292	0.00000067

- 在实际应用中，在计算高斯函数的离散近似时，在大概 3σ 距离之外的像素都可以看作不起作用，这些像素的计算也就可以忽略。
- 通常，图像处理程序只需要计算 $(6\sigma + 1) \times (6\sigma + 1)$

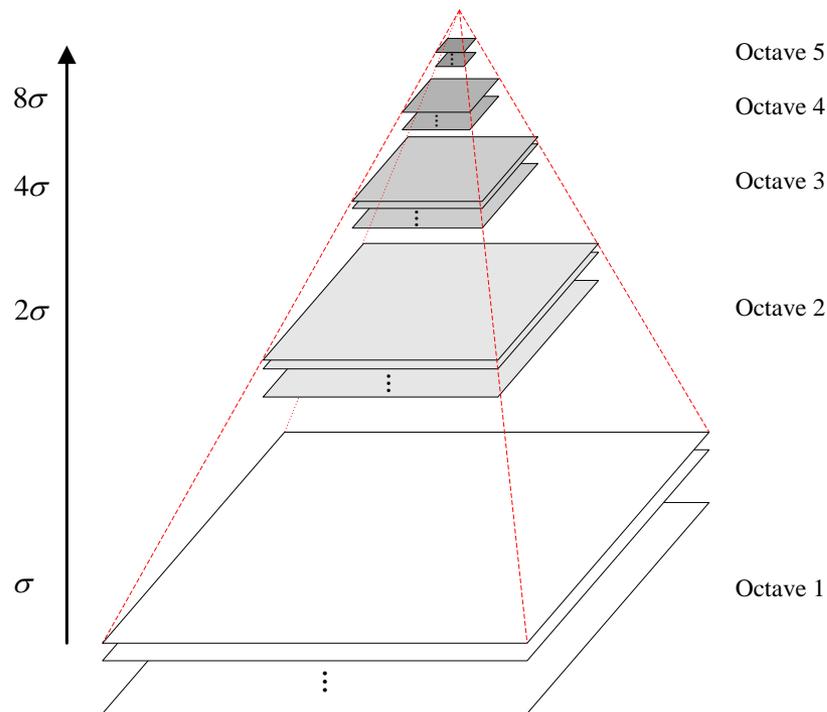
高斯模糊的性质

- 高斯模糊具有圆对称性。
- 高斯模糊具有线性可分的性质，也可以在二维图像上对两个独立的一维空间分别进行计算。这样可以大大减少了运算的次数。
- 对一幅图像进行多次连续高斯模糊的效果与一次更大的高斯模糊可以产生同样的效果，大的高斯模糊的半径是所用多个高斯模糊半径平方和的平方根。例如，使用半径分别为 6 和 8 的两次高斯模糊变换得到的效果等同于一次半径为 10 的高斯模糊效果， $\sqrt{6^2 + 8^2} = 10$
根据这个关系，使用多个连续较小的高斯模糊处理不会比单个高斯较大处理时间要少。

4. 高斯金字塔

- 高斯金字塔的构建过程可分为两步：
 - 对图像做高斯平滑；
 - 对图像做降采样。

为了让尺度体现其连续性，在简单下采样的基础上加上了高斯滤波。一幅图像可以产生几组（octave）图像，一组图像包括几层（interval）图像。



关键点检测的相关概念

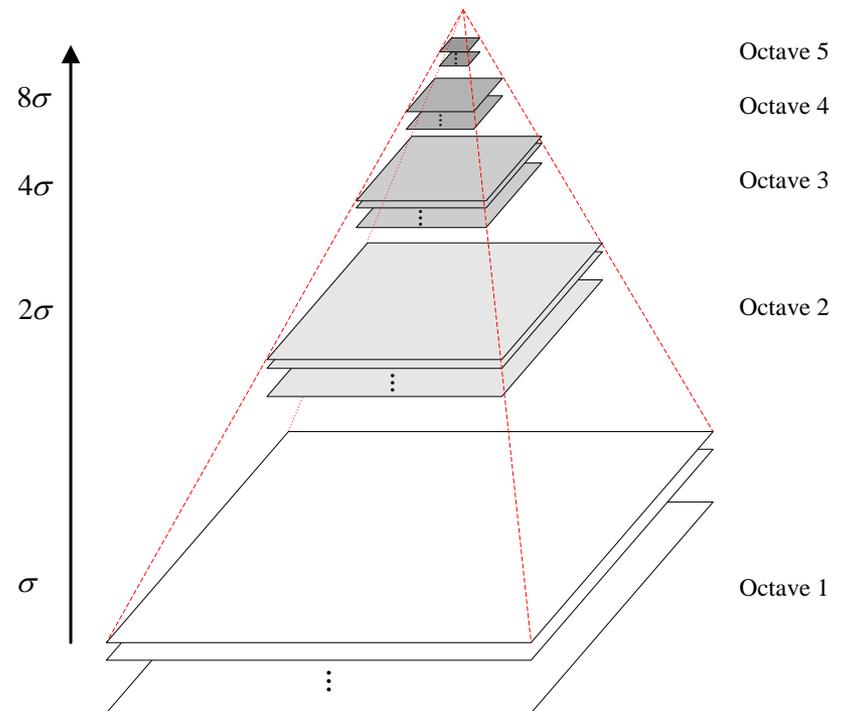
SIFT

Scale Invariant Feature Transform

- 高斯图像金字塔共o组、s层，则有：

$$\sigma(s) = \sigma_0 \cdot 2^{s/S}$$

- σ —— 尺度空间坐标；
- s —— sub-level层坐标；
- σ_0 —— 初始尺度；
- S —— 每组层数（一般为3~5）。



- 高斯金字塔的初始尺度

当图像通过相机拍摄时，相机的镜头已经对图像进行了一次初始的模糊，所以根据高斯模糊的性质：

$$\sigma_0 = \sqrt{\sigma_{init} \times \sigma_{init} - \sigma_{pre} \times \sigma_{pre}}$$

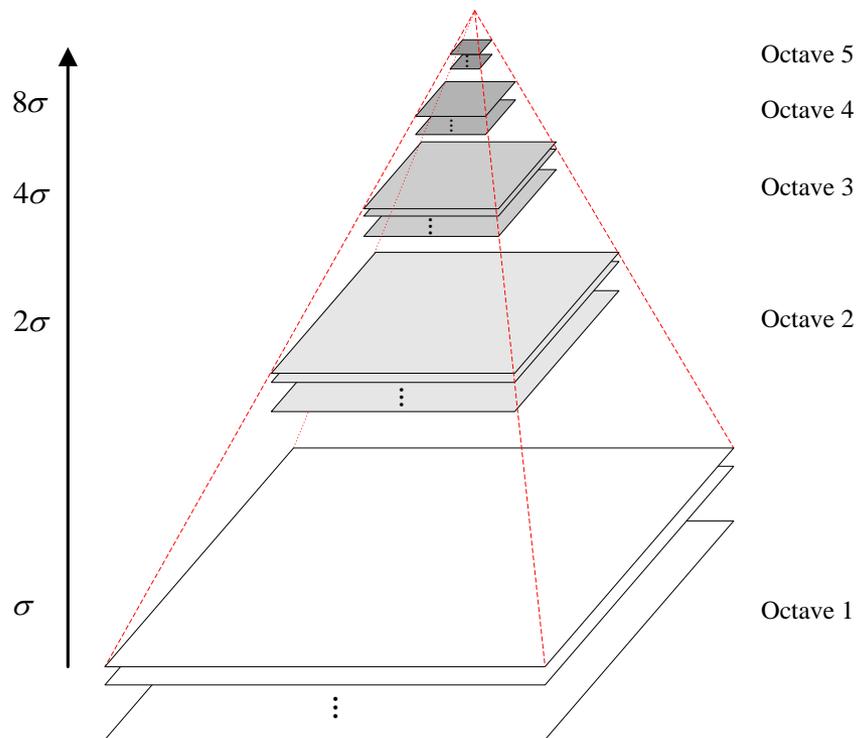
σ_{init} —— 第0层尺度

σ_{pre} —— 被相机镜头模糊后的尺度

- 高斯金字塔的组数

$$O = \lceil \log_2(\min(M, N)) \rceil - 3$$

M、N分别为图像的行数和列数



- 高斯金字塔的组内尺度与组间尺度

$$\sigma(s) = \sigma_0 \cdot 2^{s/s}$$

组内尺度是指同一组（**octave**）内的尺度关系，组内相邻层尺度化简为：

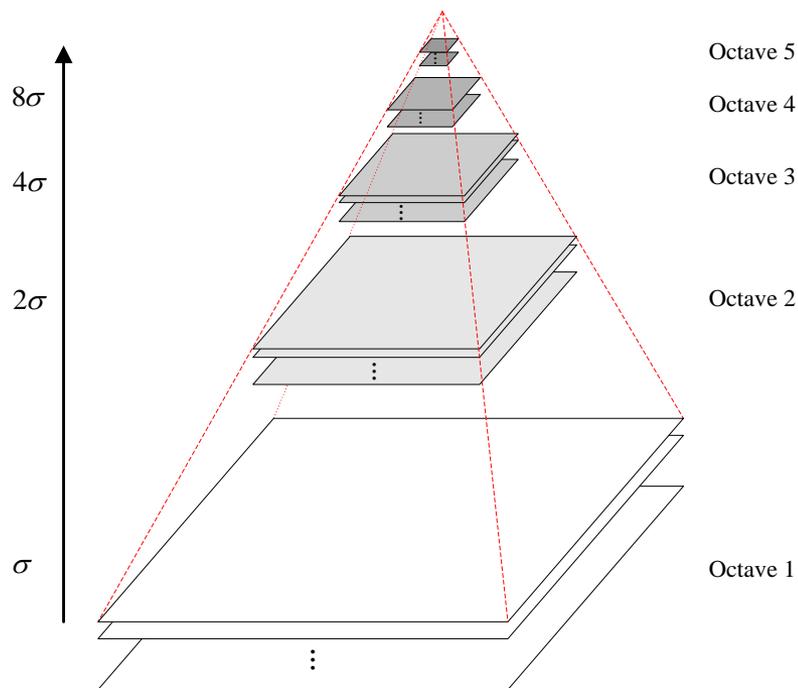
$$\sigma_{s+1} = \sigma_s \cdot 2^{1/s}$$

组间尺度是指不同组直接的尺度关系，相邻组的尺度可化为：

$$\sigma_{o+1}(s) = \sigma_o \cdot 2^{s+S/s}$$

$$\sigma_o \cdot 2^{s+S/s} = 2\sigma_o \cdot 2^{s/s}$$

由此可见，相邻两组的同一层尺度为**2倍**的关系



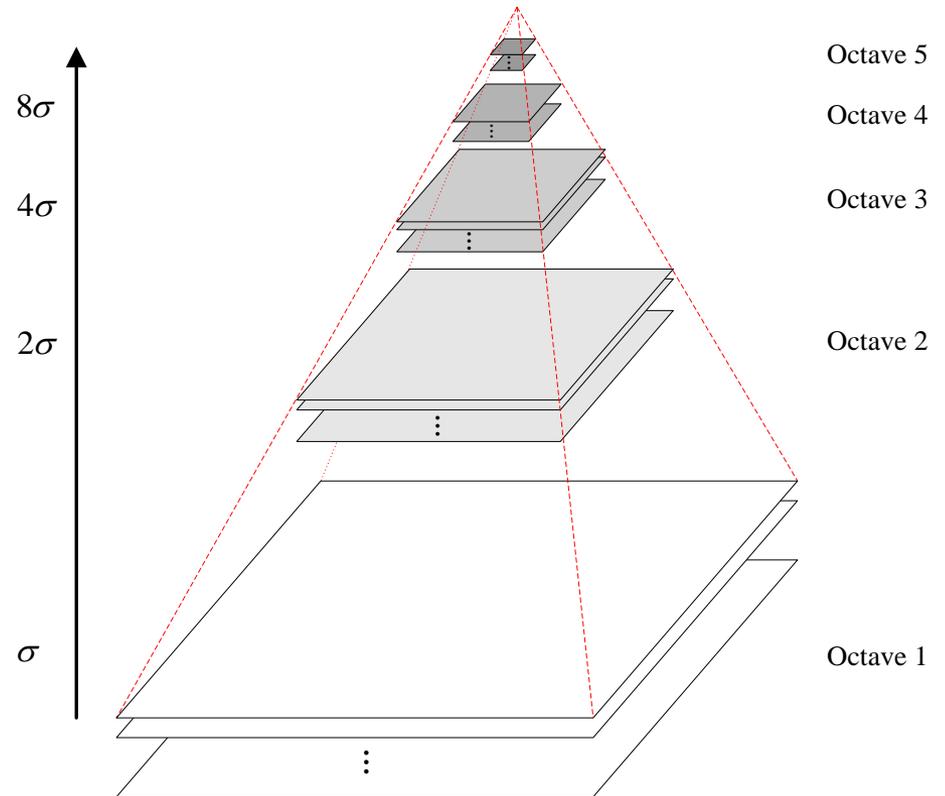
- 最后可将组内和组间尺度归为:

$$2^{i-1}(\sigma, k\sigma, k^2\sigma, \dots, k^{n-1}\sigma)$$

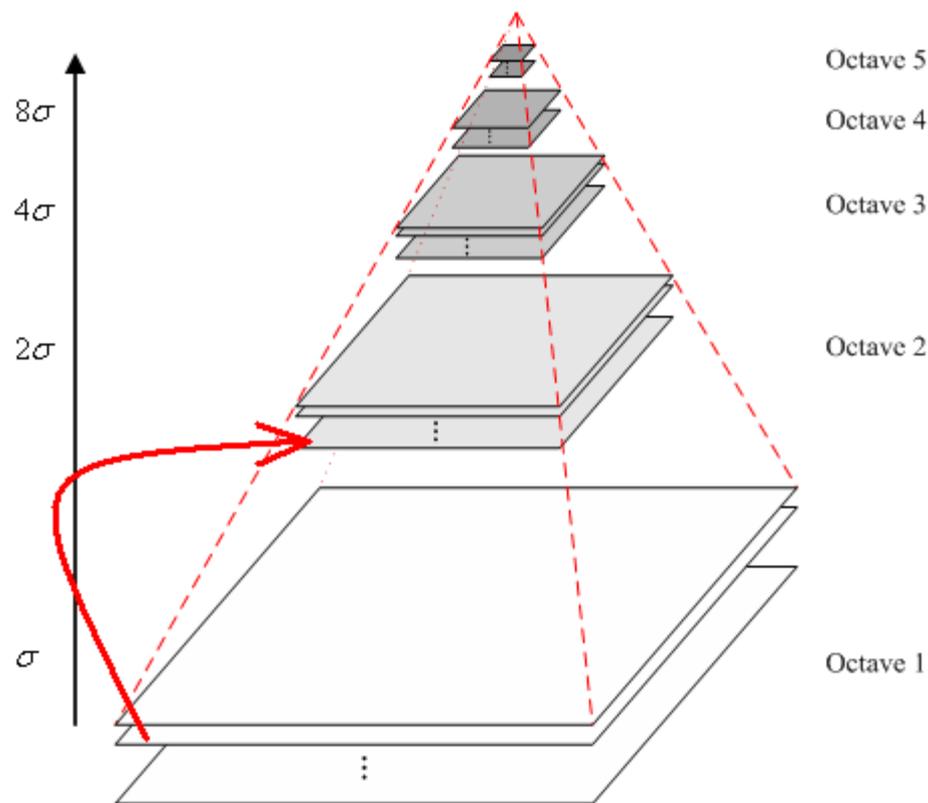
$$k = 2^{1/s}$$

i——金字塔组数

n——每一组的层数



- 上一组图像的底层是由前一组图像的倒数第二层图像隔点采样生成的。这样可以保持尺度的连续性。



通过研究Lowe教授的论文发现，所有特征点的检测都是基于了尺度不变的特性，特征点的检测占据了论文的大部分的篇章，**具有十分重要的意义！**

Lindeberg在文献《Scale-space theory: A basic tool for analysing structures at different scales》指出尺度规范化的LoG算子具有真正的尺度不变性。

LoG算子即（Laplacion of Gaussian），可以由高斯函数梯度算子GOG构建

尺度规范化的GoG算子

$$\nabla^2 G = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

尺度规范化的LoG算子

$$\sigma^2 \nabla^2 G$$

LOG算子与高斯核函数的关系

$$\nabla^2 G = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

$$LOG(x, y, \sigma) = \sigma^2 \nabla^2 G \approx \frac{Gauss(x, y, k\sigma) - Gauss(x, y, \sigma)}{\sigma^2 (k-1)}$$

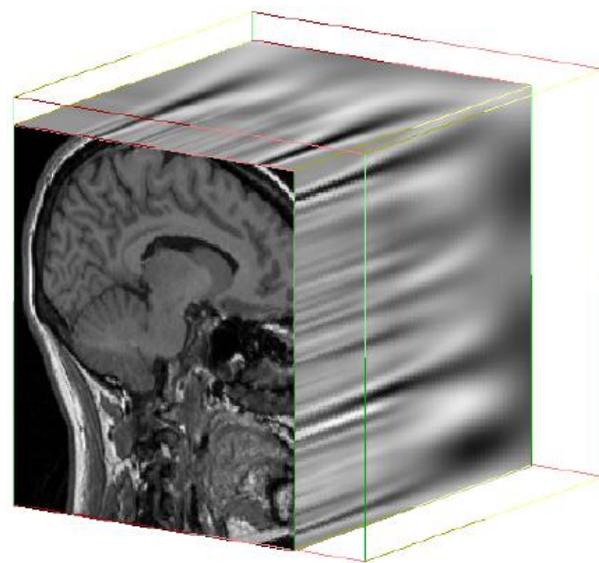
$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1) \sigma^2 \nabla^2 G$$

通过推导可以看出，LOG算子与高斯核函数的差有直接关系，由此引入一种新的算子DOG（Difference of Gaussians），即**高斯差分算子**。

DoG (Difference of Gaussian) 函数

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

$$\begin{aligned} D(x, y, \sigma) &= [G(x, y, k\sigma) - G(x, y, \sigma)] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned}$$

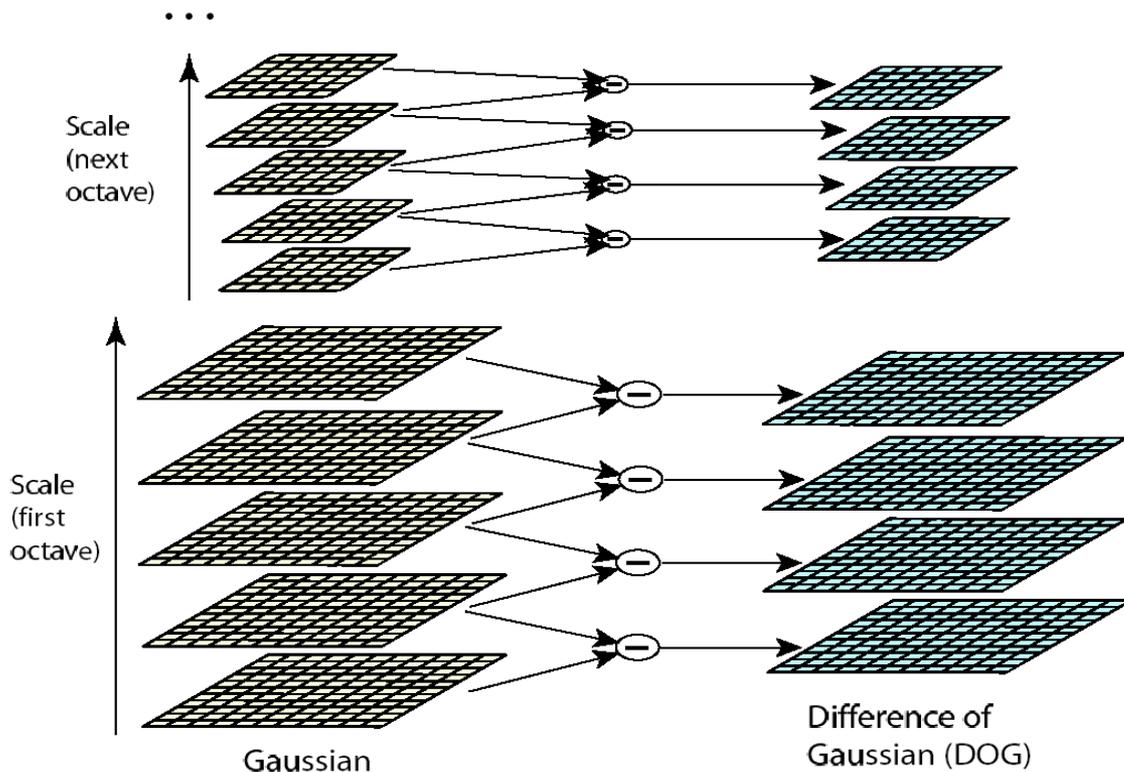


DoG在计算上只需相邻尺度高斯平滑后图像相减，因此简化了计算！

DoG高斯差分金字塔

对应DOG算子，我们要构建DOG金字塔

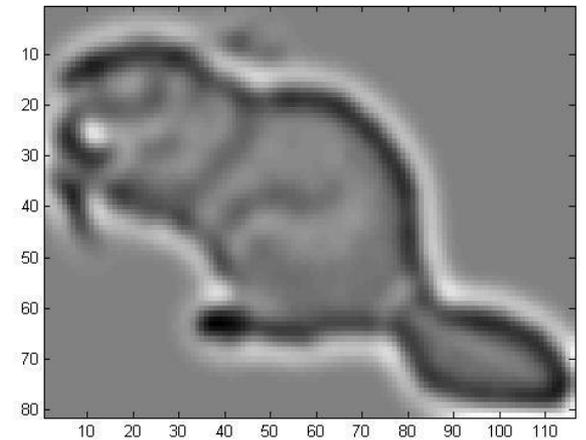
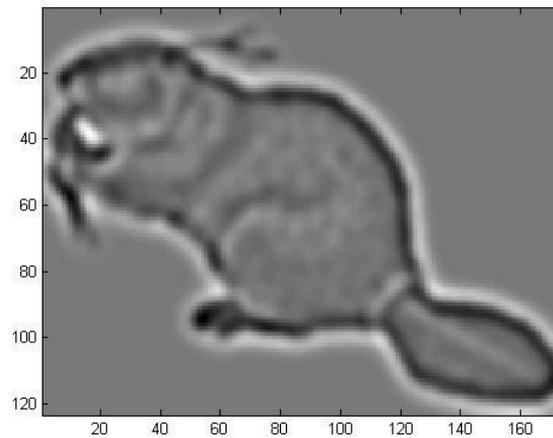
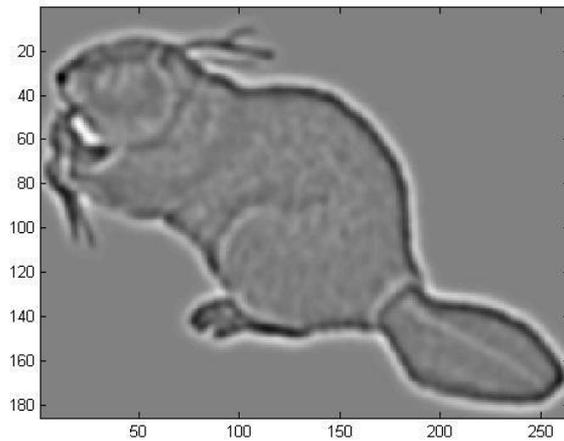
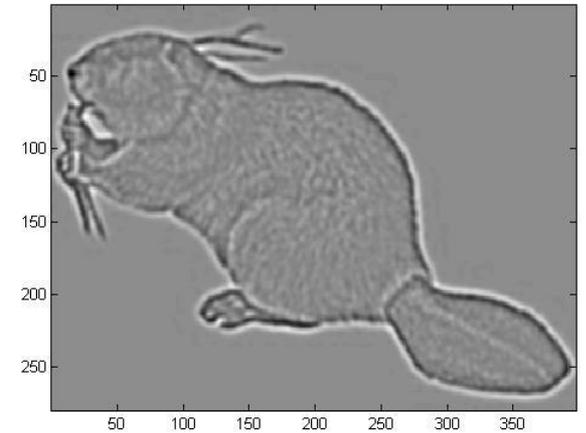
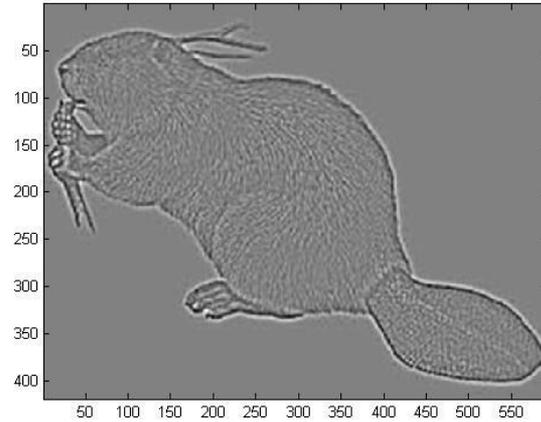
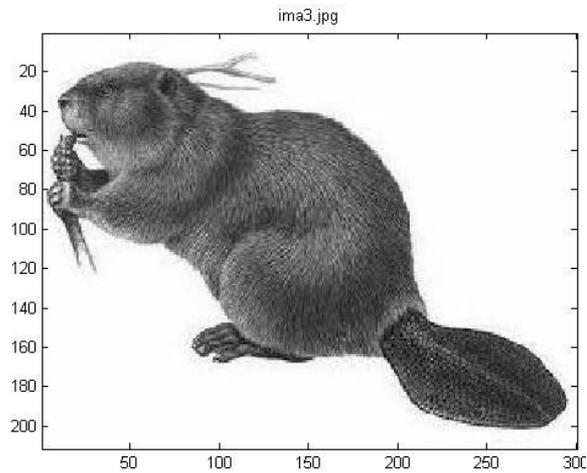
我们可以通过高斯差分图像看出图像上的像素值变化情况。（如果没有变化，也就没有特征。特征必须是变化尽可能多的点。）DOG图像描绘的是目标的轮廓。



关键点检测——DOG

SIFT

Scale Invariant Feature Transform



在Lowe的论文中，将第0层的初始尺度定为1.6，图片的初始尺度定为0.5，则图像金字塔第0层的实际尺度为

$$\sqrt{1.6 \times 1.6 - 0.5 \times 0.5} = 1.52$$

在检测极值点前对原始图像的高斯平滑以致图像丢失高频信息，所以Lowe建议在建立尺度空间前首先对原始图像长宽扩展一倍，以保留原始图像信息，增加特征点数量。

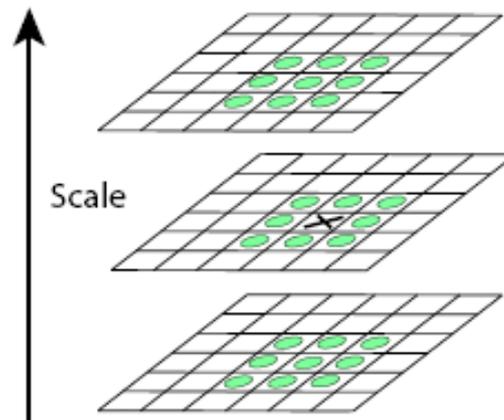
当对图像长宽扩展一倍时，便构建了-1层，该层尺度为

$$\sqrt{1.6 \times 1.6 - (2 \times 0.5) \times (2 \times 0.5)} = 1.25$$

P. S. : 图像插值时，选用的插值函数可以是多种多样的。

- DoG的局部极值点

关键点是由DOG空间的局部极值点组成的。为了寻找DOG函数的极值点，每一个像素点要和它所有的相邻点比较，看其是否比它的图像域和尺度域的相邻点大或者小。



中间的检测点和它同尺度的8个相邻点和上下相邻尺度对应的 9×2 个点共26个点比较，以确保在尺度空间和二维图像空间都检测到极值点。

DOG局部极值检测

SIFT

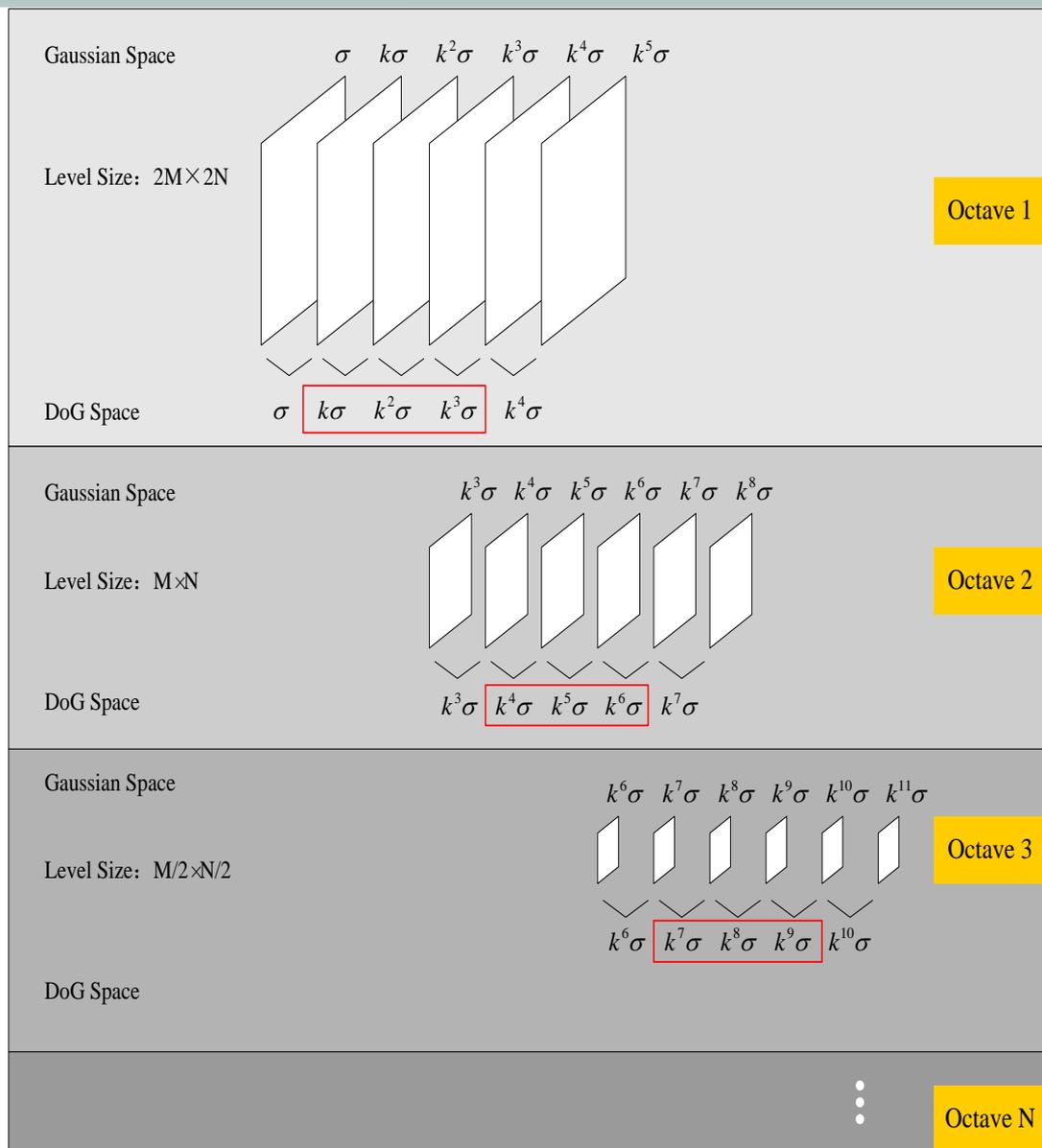
Scale Invariant Feature Transform

在极值比较的过程中，每一组图像的首末两层是无法进行极值比较的，为了满足尺度变化的连续性，我们在每一组图像的顶层继续用高斯模糊生成了3幅图像，高斯金字塔有每组 $S+3$ 层图像。DOG金字塔每组有 $S+2$ 层图像

右图为不同尺度不同层间极值检测示意图。

P. S. : 我们只牺牲了-1组的第0层和第N组的最高层

2012/3/27



- 关键点精确定位

由于DoG值对噪声和边缘较敏感, 因此, 在上面DoG尺度空间中检测到局部极值点还要经过进一步的检验才能精确定位为特征点。

为了提高关键点的稳定性, 需要对尺度空间DoG函数进行曲线拟合。

利用DoG函数在尺度空间的Taylor展开式:

$$D(X) = D + \frac{\partial D}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$$

其极值点 $\hat{X} = (x, y, \sigma)^T$

在计算过程中，分别对图像的行、列及尺度三个量进行了修正，其修正结果如下：

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$$

$$\text{求解得 } \hat{X} = -\frac{\partial D^T}{\partial X} \left(\frac{\partial^2 D}{\partial X^2}\right)^{-1}$$

\hat{X} 为修正值

在Lowe的程序中，对坐标进行了五次修正。

将修正后的结果代入式 $D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X$

$$D(\hat{X}) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} X$$

上式去除那些对比度较低的不稳定极值点。Lowe的试验显示，

所有

取值小于0.04的极值点均可抛弃（像素灰度值范围

2012/3/27

[0, 1])。

- 去除边缘响应

仅仅去除低对比度的极值点对于极值点的对于特征点稳定性是远远不够的。DOG函数在图像边缘有较强的边缘响应，因此我们还需要排除边缘响应。

DOG函数的（欠佳的）峰值点在横跨边缘的方向有较大的主曲率，而在垂直边缘的方向有较小的主曲率。主曲率可以通过计算在该点位置尺度的 2×2 的Hessian矩阵得到，导数由采样点相邻差来估计：

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

D_{xx} 表示DOG金字塔中某一尺度的图像x方向求导两次

D 的主曲率和 H 的特征值成正比，为了避免直接的计算这些特征值，而只是考虑它们的之间的比率。令为 最大特征值 α ， β 为最小的特征值，则

$$\alpha = r\beta \quad \frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r+1)^2}{r}$$

$$Tr(H) = D_{xx} + D_{yy}$$

$$Det(H) = D_{xx} \times D_{yy} - D_{xy} \times D_{xy}$$

$$(r+1)^2 / r$$

在两特征值相等时达最小，随 r 的增长而增长。Lowe论文中建议 r 取10。

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad \text{时将关键点保留，反之剔除}$$

通过尺度不变性求极值点，可以使其具有缩放不变的性质，利用关键点邻域像素的梯度方向分布特性，我们可以为每个关键点指定方向参数方向，**从而使描述子对图像旋转具有不变性。**

我们通过求每个极值点的梯度来为极值点赋予方向。

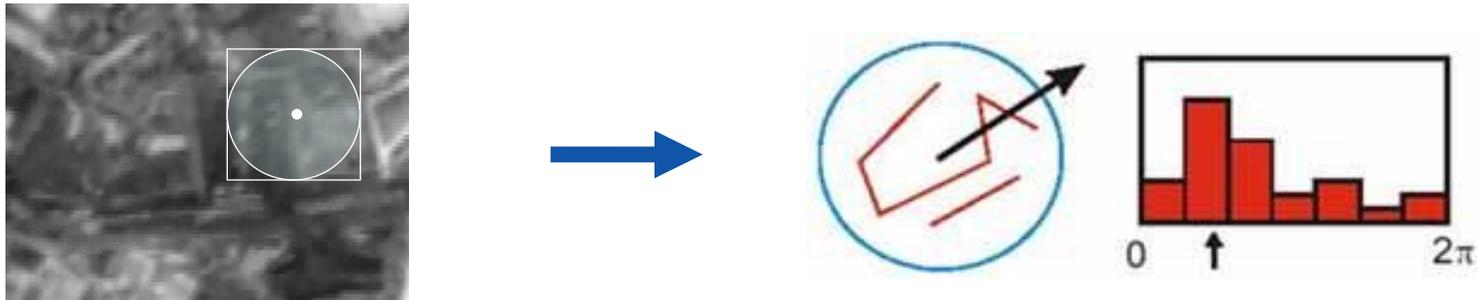
- 像素点的梯度表示

$$\text{grad}I(x, y) = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$$

梯度幅值:
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

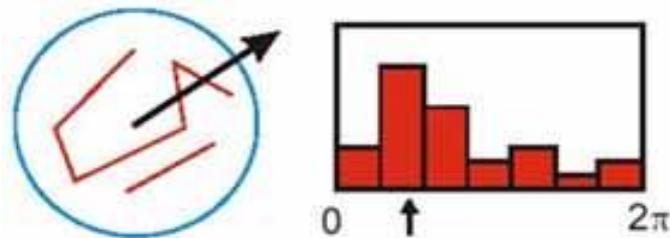
梯度方向:
$$\theta(x, y) = \tan^{-1} \left[\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right]$$

- 方向直方图的生成



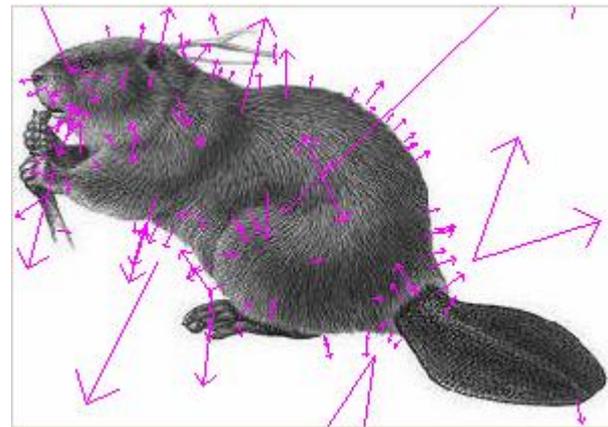
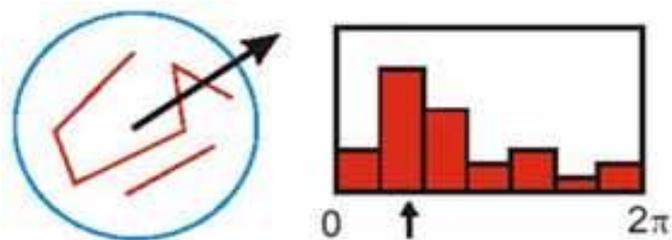
确定关键点的方向采用梯度直方图统计法，统计以关键点为原点，一定区域内的图像像素点对关键点方向生成所作的贡献。

关于方向直方图的一点说明



1. 直方图以每10度方向为一个柱，共36个柱，柱所代表的方向为像素点梯度方向，柱的长短代表了梯度幅值。
2. 根据Lowe的建议，直方图统计半径采用 $3 * 1.5 * \sigma$ 。
3. 在直方图统计时，每相邻三个像素点采用高斯加权，根据Lowe的建议，模板采用 $[0.25, 0.5, 0.25]$ ，并连续加权两次。

关键点的主方向与辅方向



- 关键点主方向：极值点周围区域梯度直方图的主峰值，也是特征点方向
- 关键点辅方向：在梯度方向直方图中，当存在另一个相当于主峰值80%能量的峰值时，增强该峰值的鲁棒性为该的关键点辅方向。大概有15%关键点具有多方向，但这些点对匹配的稳定性至为关键。

● 方向分配实现步骤

1. 确定计算关键点直方图的高斯函数权重函数参数 ；
2. 生成含有36柱的方向直方图，梯度直方图范围0~360度，其中每10度一个柱。由半径为图像区域生成；
3. 对方向直方图进行两次平滑；
4. 求取关键点方向（可能是多个方向）；
5. 对方向直方图的Taylor展开式进行二次曲线拟合，精确关键点方向；

图像的关键点已检测完毕，每个关键点有三个信息：位置、尺度、方向；同时也就使关键点具备平移、缩放、和旋转不变性。

[Back](#)

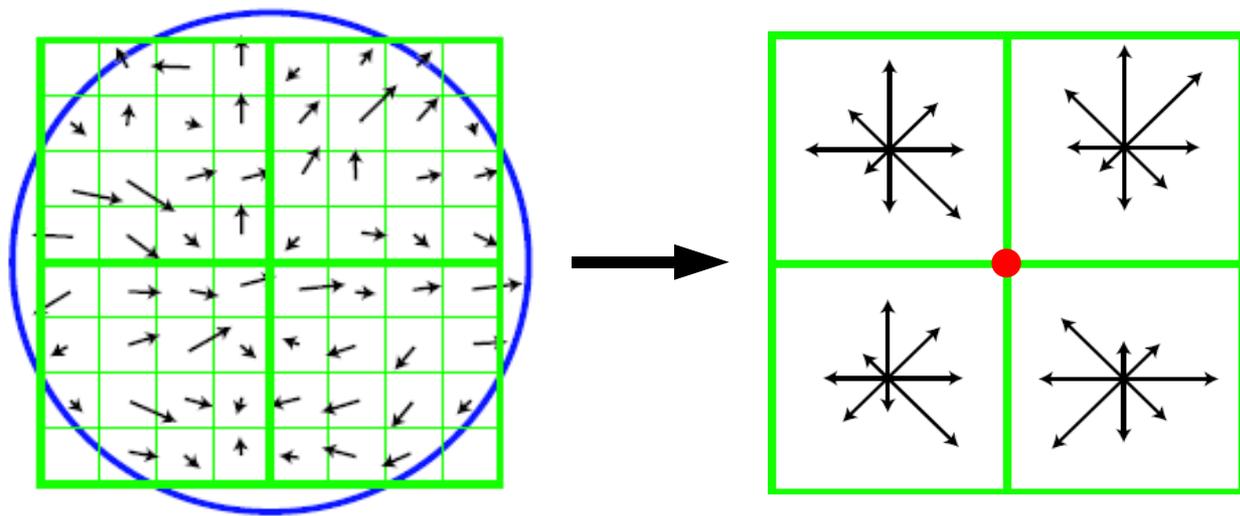
- 描述的目的

描述的目的是在关键点计算后，用一组向量将这个关键点描述出来，这个描述子不但包括关键点，也包括关键点周围对其有贡献的像素点。用来作为目标匹配的依据，也可使关键点具有更多的不变特性，如光照变化、3D视点变化等。

- 描述的思路

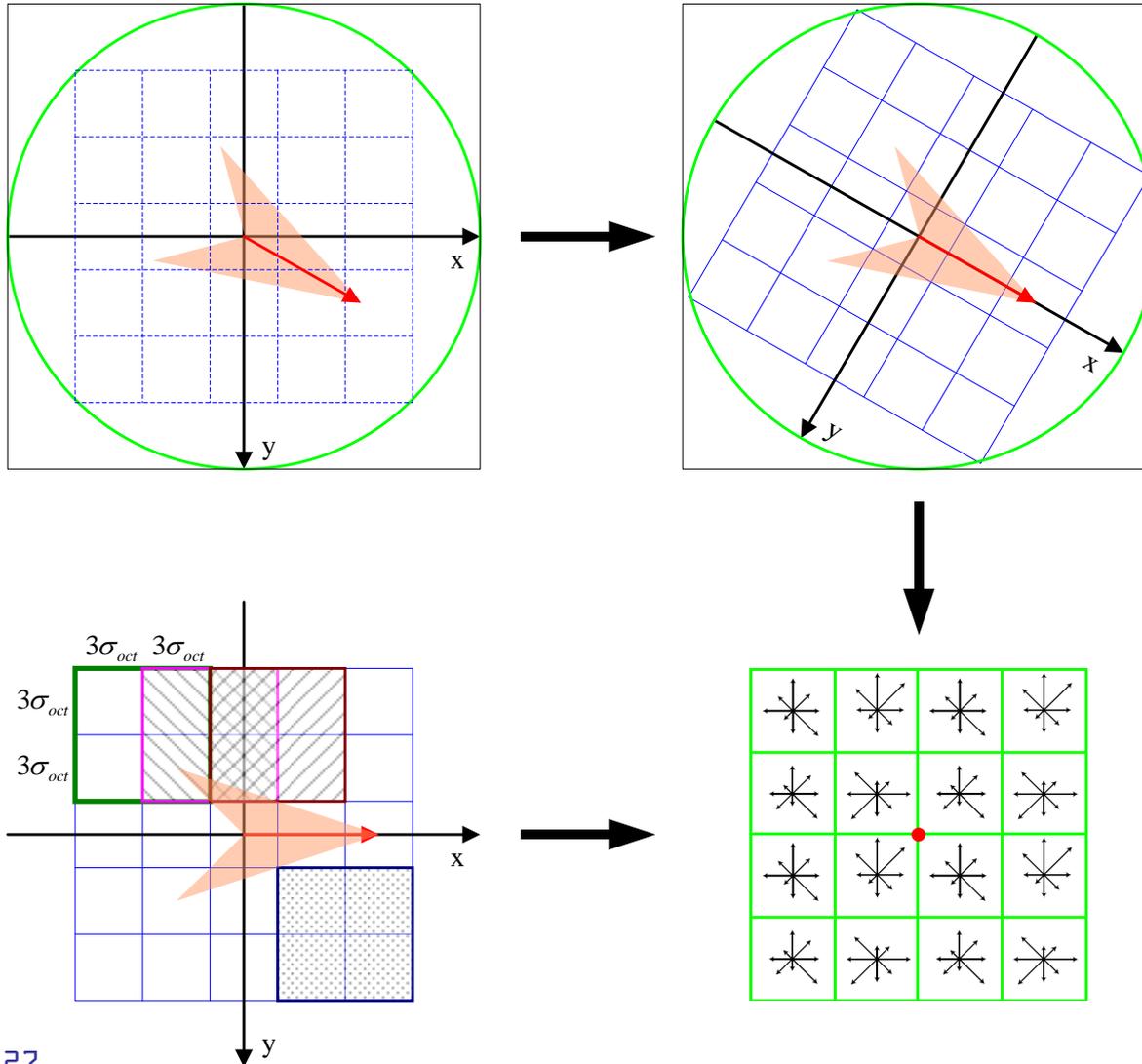
通过对关键点周围图像区域分块，计算块内梯度直方图，生成具有独特性的向量，这个向量是该区域图像信息的一种抽象，具有唯一性。

下图是一个SIFT描述子事例。其中描述子由 $2 \times 2 \times 8$ 维向量表征，也即是 2×2 个8方向的方向直方图组成。左图的种子点由 8×8 单元组成。每一个小格都代表了特征点邻域所在的尺度空间的一个像素，箭头方向代表了像素梯度方向，箭头长度代表该像素的幅值。然后在 4×4 的窗口内计算8个方向的梯度方向直方图。绘制每个梯度方向的累加可形成一个种子点，如右图所示：一个特征点由4个种子点的信息所组成。



关键点周围区域图像梯度

关键点描述子



Lowe实验结果表明：描述子采用 $4 \times 4 \times 8 = 128$ 维向量表征，综合效果最优（不变性与独特性）。

- 128维关键点描述子生成步骤

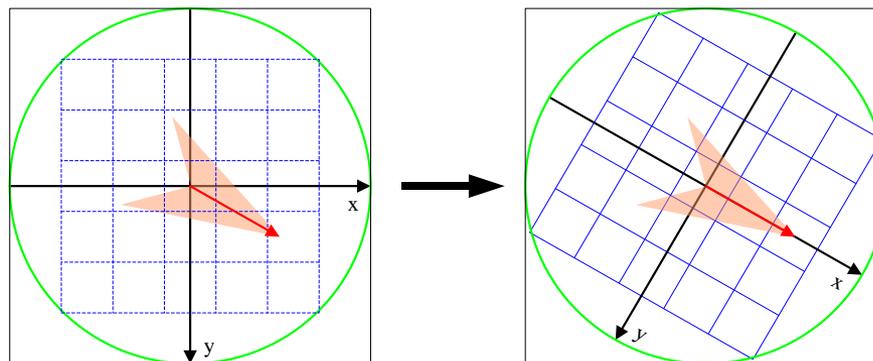
1. 确定计算描述子所需的图像区域

描述子梯度方向直方图由关键点所在尺度的模糊图像计算产生。图像区域的半径通过下式计算：

$$radius = \frac{3\sigma_{oct} \times \sqrt{2} \times (d+1) + 1}{2}$$

σ_{oct} 是关键点所在组（octave）的组内尺度，

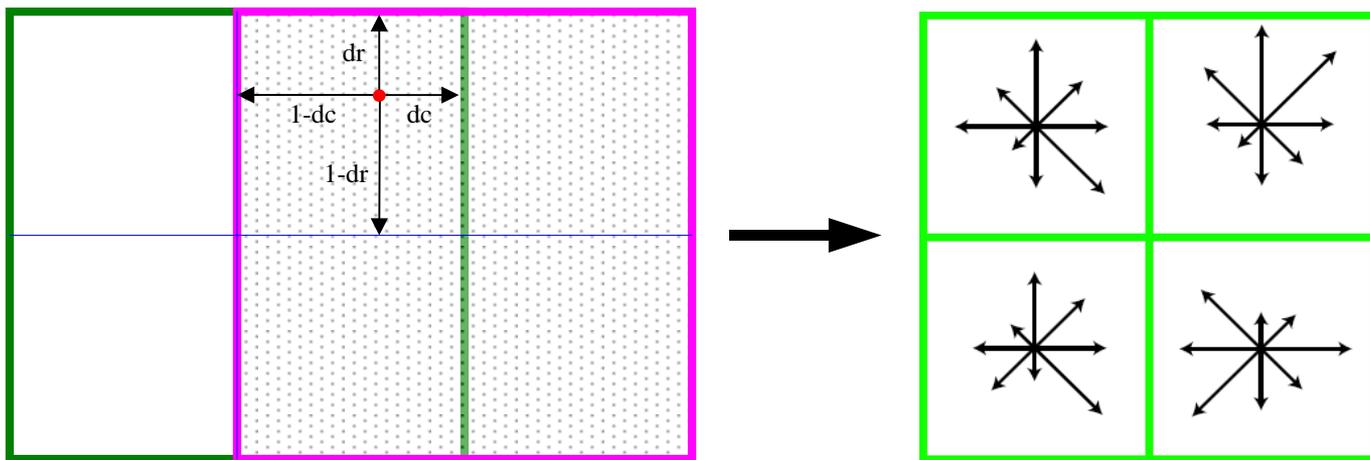
2. 将坐标移至关键点主方向



那么旋转角度后新坐标为：

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \times \begin{pmatrix} x \\ y \end{pmatrix}$$

3. 在图像半径区域内对每个像素点求其梯度幅值和方向，然后对每个梯度幅值乘以高斯权重参数，生成方向直方图。



$$weight = |grad(I_{\sigma}(x, y))| \times \exp\left(-\frac{x_k^2 + y_k^2}{2\sigma_w}\right) \times (1-d_r) \times (1-d_c) \times (1-d_o)$$

x_k : 该点与关键点的列距离;

y_k : 该点与关键点的行距离;

σ_w : 等于描述子窗口宽度 3σ \times 直方图列数 (取4) 的一半;

4. 在窗口宽度为2X2的区域内计算8个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点。然后再在下一个2X2的区域内进行直方图统计，形成下一个种子点，共生成16个种子点。

5. 描述子向量元素门限化及门限化后的描述子向量规范化。

描述子向量元素门限化：

方向直方图每个方向上梯度幅值限制在一定门限值以下（门限一般取0.2）。

描述子向量元素规范化：

$W = (w_1, w_2, \dots, w_{128})$ 为得到的128描述子向量，

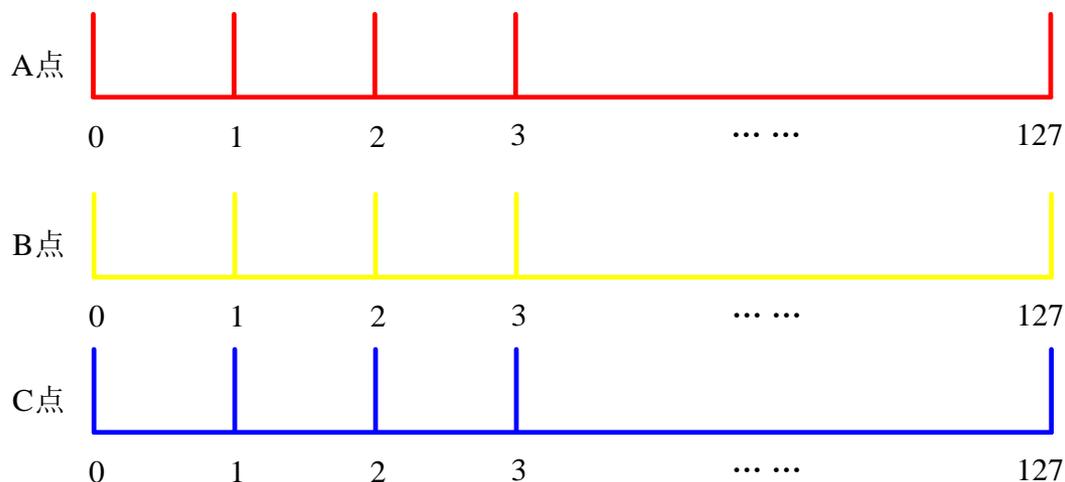
$L = (l_1, l_2, \dots, l_{128})$ 为规范化后的向量

$$l_j = w_j / \sqrt{\sum_{i=1}^{128} w_i} \quad j = 1, 2, \dots, 128$$

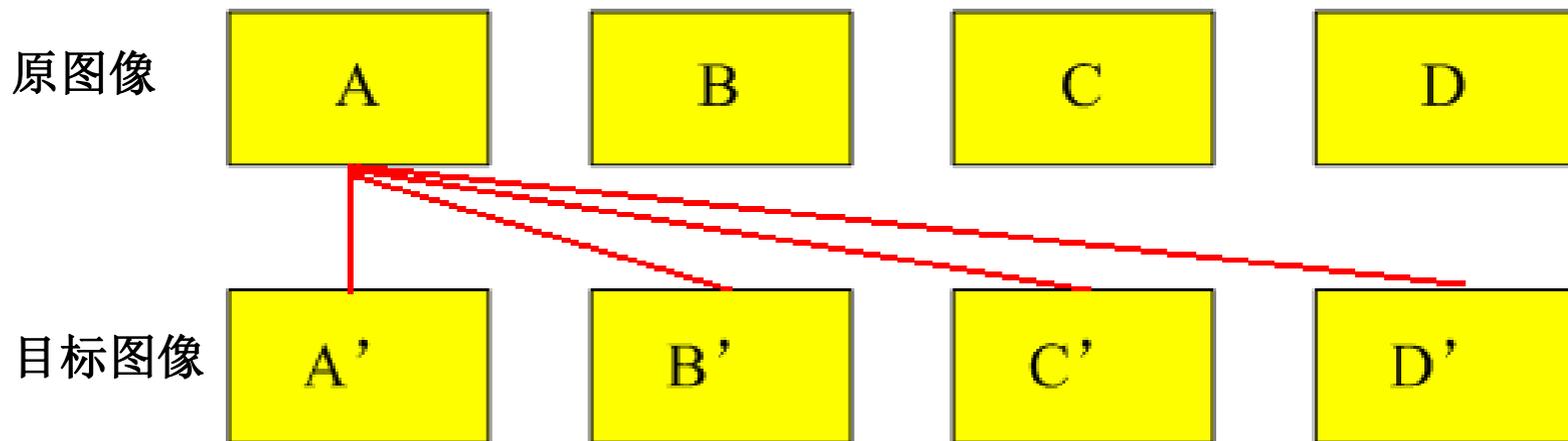
关键点描述子向量的规范化正是可去除满足此模型的光照影响。对于图像灰度值整体漂移，图像各点的梯度是邻域像素相减得到，所以也能去除。

[Back](#)

分别对模板图（参考图，reference image）和实时图（观测图，observation image）建立关键点描述子集合。目标的识别是通过两点集内关键点描述子的比对来完成。具有128维的关键点描述子的相似性度量采用欧式距离。



穷举匹配



- 模板图中关键点描述子: $R_i = (r_{i1}, r_{i2}, \dots, r_{i128})$
- 实时图中关键点描述子: $S_i = (s_{i1}, s_{i2}, \dots, s_{i128})$
- 任意两描述子相似性度量: $d(R_i, S_i) = \sqrt{\sum_{j=1}^{128} (r_{ij} - s_{ij})^2}$

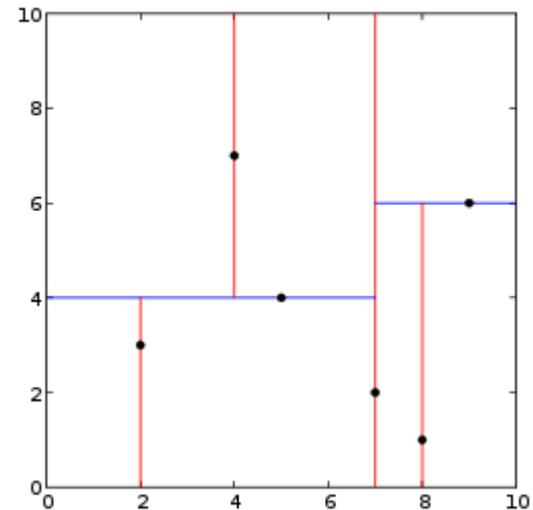
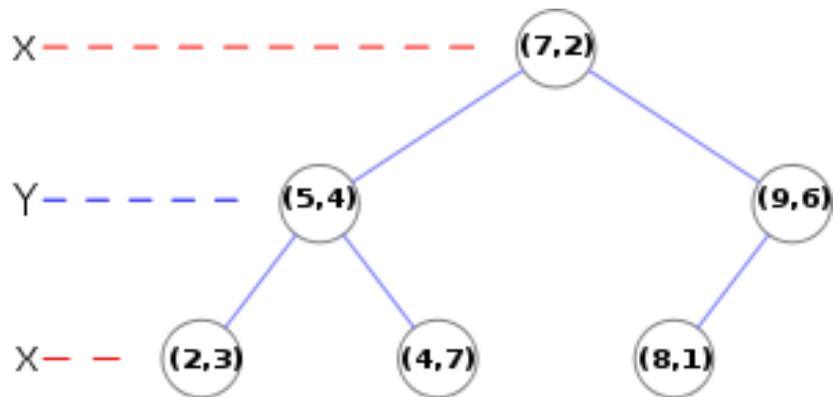
要得到配对的关键点描述子 (R_i, S_j)

需满足:

$$\frac{\text{实时图中距离 } R_i \text{ 最近的点 } S_j}{\text{实时图中距离 } R_i \text{ 的次最近点 } S_p} < \textit{Threshold}$$

- 关键点的匹配可以采用穷举法来完成，但是这样耗费的时间太多，一般都采用一种叫kd树的数据结构来完成搜索。搜索的内容是以目标图像的关键点为基准，搜索与目标图像的特征点最邻近的原图像特征点和次邻近的原图像特征点。

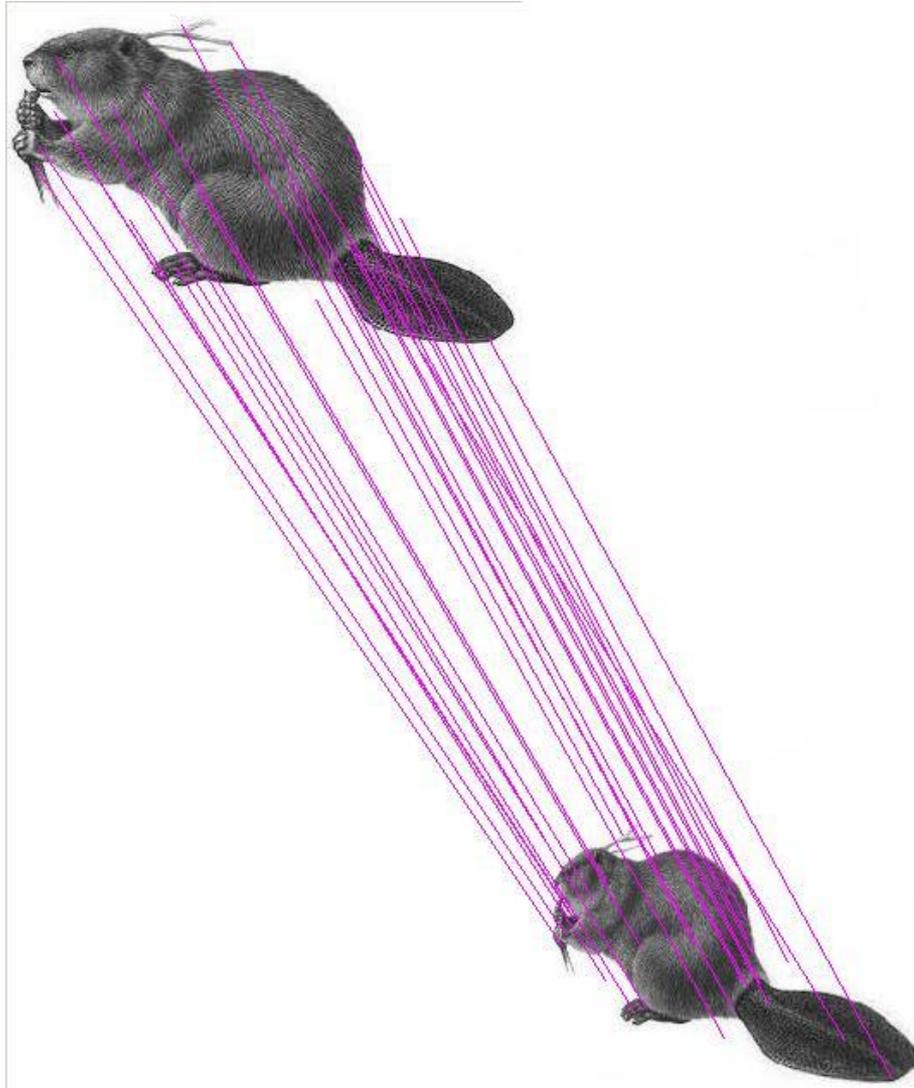
Kd树是一个平衡二叉树



关键点匹配

SIFT

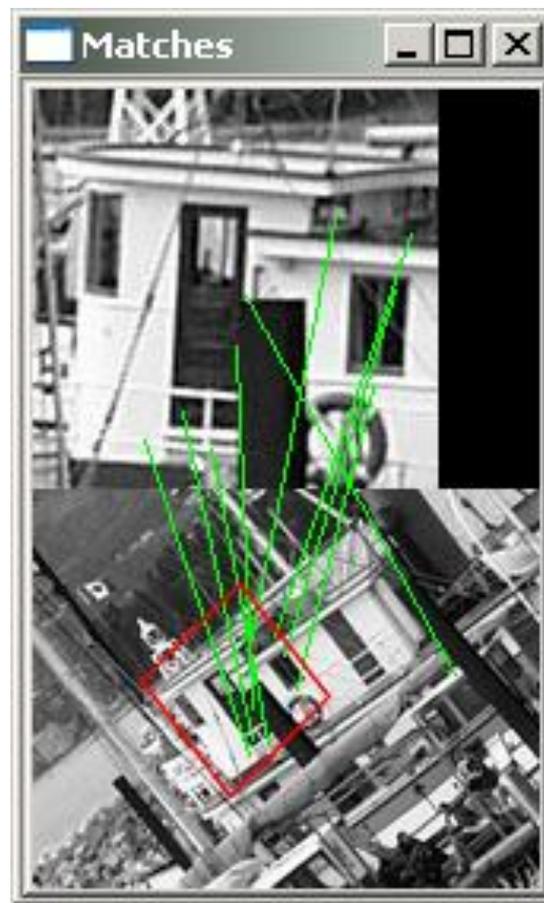
Scale Invariant Feature Transform



[Back](#)

- 关键点匹配并不能标志着算法的结束，因为在匹配的过程中存在着大量的错配点。

图中交叉的绿线为错配点



RANSAC简介

RANSAC (Random Sample Consensus, 随机抽样一致) 是一种鲁棒性的参数估计方法。

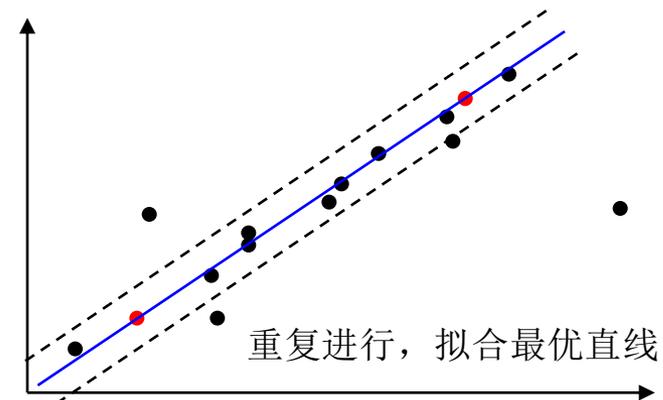
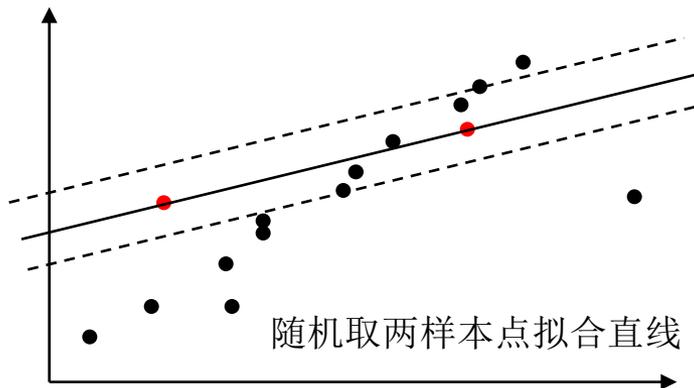
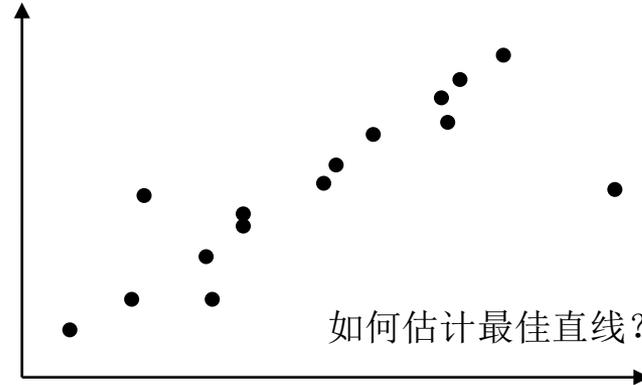
RANSAC实质上就是一个反复测试、不断迭代的过程。

RANSAC的基本思想:

首先根据具体问题设计出某个目标函数，然后通过反复提取最小点集估计该函数中参数的初始值，利用这些初始值把所有的数据分为“内点” (inlier) 和“外点” (outlier) ，最后用所有的内点重新计算和估计函数的参数。

- RANSAC事例

拟合直线: $y = kx + b$



[Back](#)

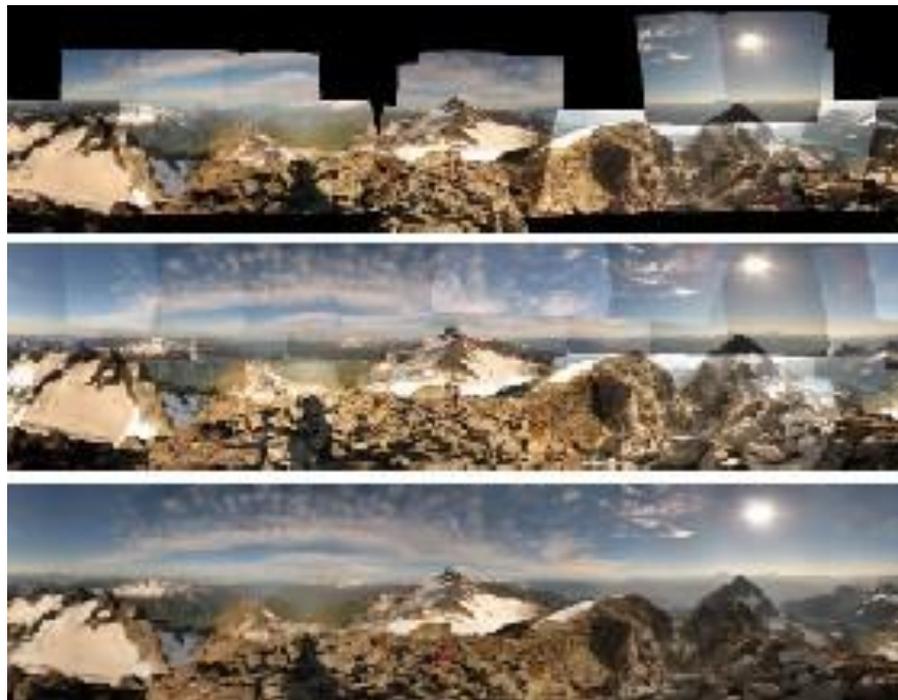
SIFT算法目前在军事、工业和民用方面都得到了不同程度的应用，其应用已经渗透了很多领域，典型的应用如下：

- 物体识别
- 机器人定位与导航
- 图像拼接
- 三维建模
- 手势识别
- 视频跟踪
- 笔记鉴定
- 指纹与人脸识别
- 犯罪现场特征提取
-

- 物体识别



- 图像拼接



[Demo](#)

- 三维建模

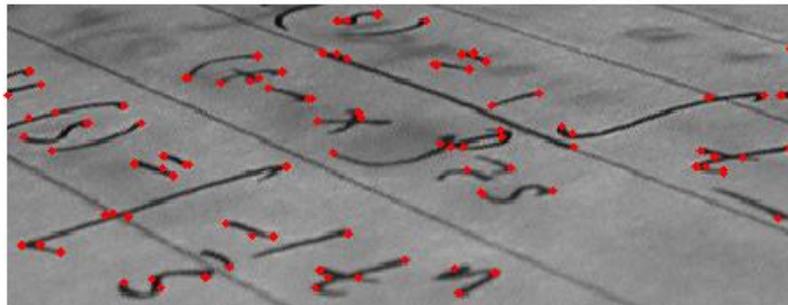
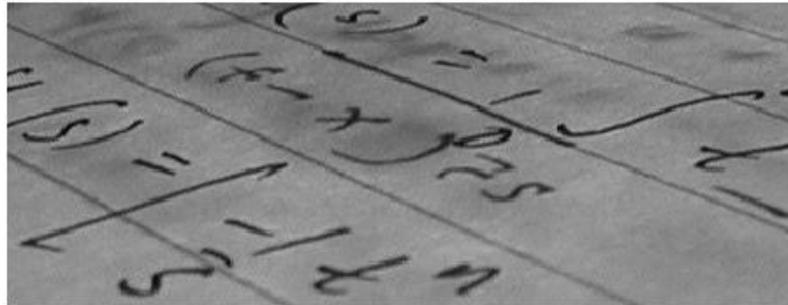


- 手势识别

目前，手势识别已应用于手机等设备上。



- 笔记鉴定



- 来自网友的创意
——周正龙的老虎

图1周正龙的华南虎照片与年画上的华南虎照片12点匹配

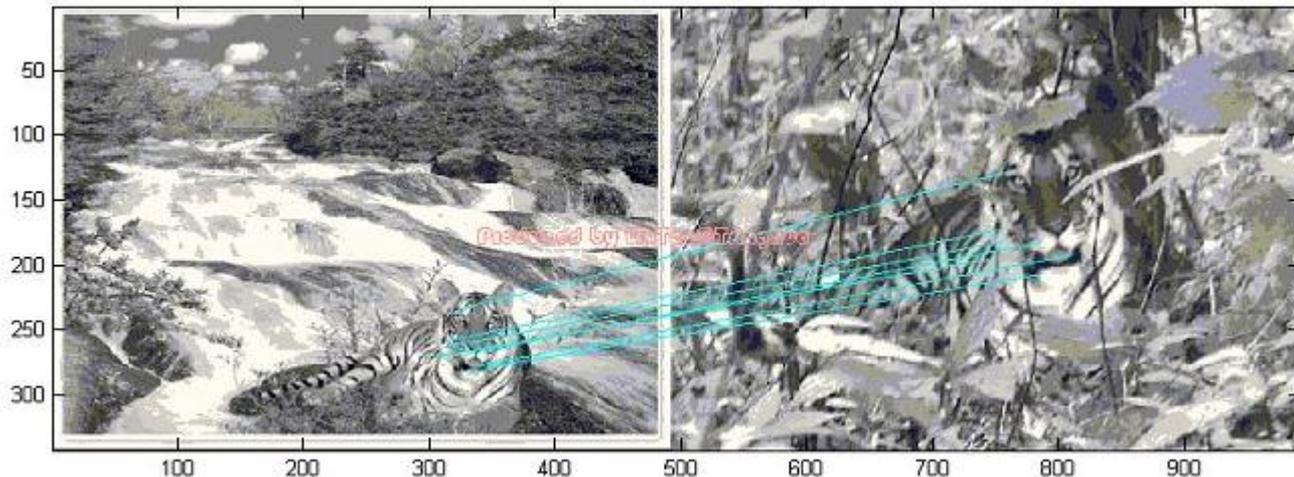
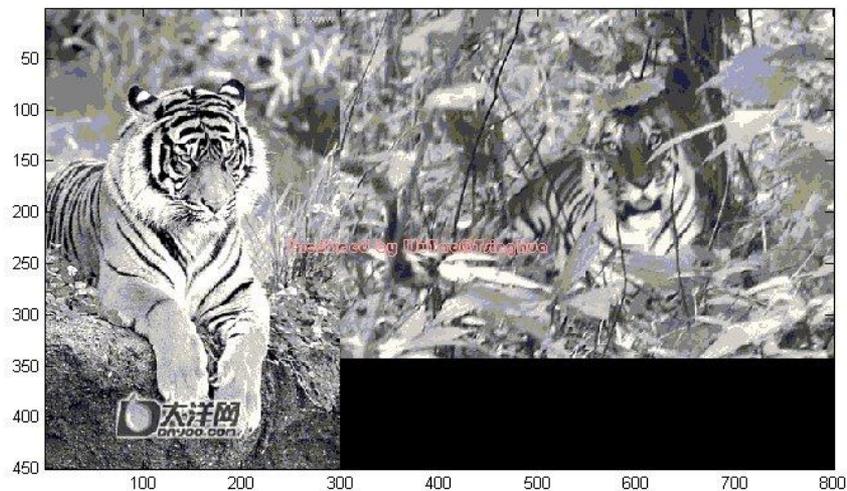
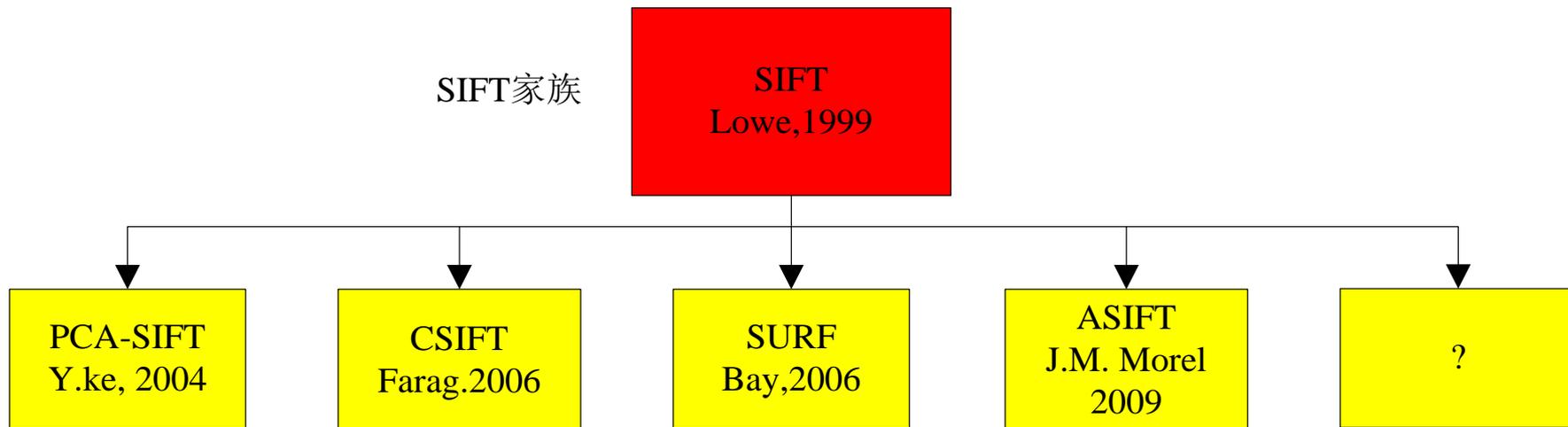


图2周正龙的华南虎照片与真实的华南虎照片0点匹配



[Back](#)

SIFT在图像的不变特征提取方面拥有无与伦比的优势，但其并不是完美的，仍然存在着实时性不高、有时特征点较少、对边缘模糊的目标无法准确提取特征点等缺陷。自从1999年，SIFT算法问世以来，人们从未停止对它的优化和改进。



- PCA-SIFT

PCA (Principal Component Analysis) ，即主成分分析，是一种数据降维技术。由Y. Ke 2004年提出。

通过降维技术，可有效化简SIFT算子的128维描述子。

- CSIFT

CSIFT (Colored scale invariant feature transform) 彩色尺度特征不变变换，可以针对彩色图像进行图像的不变特征提取。由 Farag 2006年提出。

- SURF

SURF (SURF Speeded Up Robust Features)，号称是SIFT算法的增强版，SURF算法的计算量小，运算速度快，提取的特征点几乎与SIFT相同，由Bay 2006年提出。

	SIFT	SURF
特征点检测	用不同尺度的图片与高斯函数做卷积	用不同大小的box filter与原始图像(integral image)做卷积，易于并行
方向	特征点邻接矩形区域内，利用梯度直方图计算	特征点邻接圆域内，计算x、y方向上的Haar小波响应
描述符生成	20*20(单位为pixel)区域划分为4*4(或2*2)的子区域，每个子域计算8bin直方图	20*20(单位为sigma)区域划分为4*4子域，每个子域计算5*5个采样点的Haar小波响应，记录 $\Sigma dx, \Sigma dy, \Sigma dx , \Sigma dy $ 。

- SIFT、PCA-SIFT 与 SURF 的对比

method	Time	Scale	Rotation	Blur	Illumination	Affine
Sift	common	best	best	common	common	good
PCA-sift	good	good	good	best	good	best
Surf	best	common	common	good	best	good

- ASIFT

ASIFT (Affine-SIFT) 抗仿射SIFT变换。

两个向量空间之间的一个仿射变换或者仿射映射由一个线性变换接上一个平移组成。ASIFT可以抵抗强仿射情况，提取的特征点远多于SIFT算法，由J.M. Morel 2009年提出。

[Demo](#)

谢谢！