

昵称：Brook_icv
园龄：5年10个月
粉丝：249
关注：10
+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 |

我的标签

- C++(8)
- FFmpeg(5)
- OpenCV(3)
- optimization(2)
- image retrieval(2)
- DIP(2)
- dx9(1)
- 其他(1)

随笔分类(58)

- 00-图像处理(13)
- 01-OpenCV(10)
- 02-CV (2)
- 03-SLAM(6)
- 04-FFmpeg(10)
- 09-C++(9)
- Optimization(2)
- Other(6)

积分与排名

积分 - 150707

SIFT特征详解

1.SIFT概述

SIFT的全称是Scale Invariant Feature Transform，尺度不变特征变换，由加拿大教授David G.Lowe提出的。SIFT特征对旋转、尺度缩放、亮度变化等保持不变性，是一种非常稳定的局部特征。

1.1 SIFT算法具的特点

- 1. 图像的局部特征，对旋转、尺度缩放、亮度变化保持不变，对视角变化、仿射变换、噪声也保持一定程度的稳定性。
- 2. 独特性好，信息量丰富，适用于海量特征库进行快速、准确的匹配。
- 3. 多量性，即使是很少几个物体也可以产生大量的SIFT特征
- 4. 高速性，经优化的SIFT匹配算法甚至可以达到实时性
- 5. 扩招性，可以很方便的与其他的特征向量进行联合。

1.2 SIFT特征检测的步骤

有4个主要步骤

- 1. **尺度空间的极值检测** 搜索所有尺度空间上的图像，通过高斯微分函数来识别潜在的对尺度和选择不变的兴趣点。
- 2. **特征点定位** 在每个候选的位置上，通过一个拟合精细模型来确定位置尺度，关键点的选取依据他们的稳定程度。
- 3. **特征方向赋值** 基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向，后续的所有操作都是对于关键点的方向、尺度和位置进行变换，从而提供这些特征的不变性。
- 4. **特征点描述** 在每个特征点周围的邻域内，在选定的尺度上测量图像的局部梯度，这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变换。

2. 尺度空间

在一定的范围内，无论物体是大还是小，人眼都可以分辨出来。然而计算机要有相同的能力却不是那么的容易，在未知的场景中，计算机视觉并不能提供物体的尺度大小，其中的一种方法是把物体不同尺度下的图像都提供给机器，让机器能够对物体在不同的尺度下有一个统一的认知。在建立统一认知的过程中，要考虑的就是在图像在不同的尺度下都存在的特征点。

2.1 多分辨率图像金字塔

在早期图像的多尺度通常使用**图像金字塔**表示形式。图像金字塔是同一图像在不同的分辨率下得到的一组结果，其生成过程一般包括两个步骤：

- 1. 对原始图像进行平滑
- 2. 对处理后的图像进行降采样（通常是水平、垂直方向的1/2）
降采样后得到一系列不断尺寸缩小的图像。显然，一个传统的金字塔中，每一层的图像是其上一层图像长、高的各一半。多分辨率的图像金字塔虽然生成简单，但其本质是降采样，图像的局部特征则难以保持，也就是无法保持特征的尺度不变性。

2.2 高斯尺度空间

我们还可以通过图像的模糊程度来模拟人在距离物体由远到近时物体在视网膜上成像过程，距离物体越近其尺寸越大图像也越模糊，这就是高斯尺度空间，使用不同的参数模糊图像（分辨率不变），是尺度空间的另一种表现形式。

我们知道图像和高斯函数进行卷积运算能够对图像进行模糊，使用不同的“高斯核”可得到不同模糊程度的图像。一副图像其高斯尺度空间可由其和不同的高斯卷积得到：

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



排名 - 1956

最新评论

1. Re:SLAM入门之视觉里程计(3) :
两视图对极约束 基础矩阵

@Brook_icv谢谢回复, 对极约束不是点与线的对应么, 左视图的点在右视图极线上一定可以找到, 但是怎么确定在极线的什么位置呢? 您说的搜索依据是什么? 还望请教...

--柏林熊猫
2. Re:SLAM入门之视觉里程计(3) :
两视图对极约束 基础矩阵

@柏林熊猫是可以的, 如果检测到了左视图的点, 可以通过对极约束, 再右视图的对应的对极线上, 搜索对应的点. ...

--Brook_icv
3. Re:SLAM入门之视觉里程计(3) :
两视图对极约束 基础矩阵

@Brook_icv是的, 我希望使用Harris等检测算法, 在左视图检测出某区域的角点, 计算右视图对应的像点进行匹配, 不知道这个方法是否可行? ...

--柏林熊猫
4. Re:SLAM入门之视觉里程计(3) :
两视图对极约束 基础矩阵

@柏林熊猫没有很明白是什么意思. 是说, 从左视图的像点, 直接利用基础矩阵来计算其在右视图对应的像点么...

--Brook_icv
5. Re:SLAM入门之视觉里程计(3) :
两视图对极约束 基础矩阵

您好, 我想问一问既然说基本矩阵指的是点到直线的映射, 那从左视图的匹配点是不是不可以直接利用基础矩阵来进行右视图对应点的计算呢?

--柏林熊猫

阅读排行榜

1. OpenCV2:Mat(40302)

其中, $G(x, y, \sigma)$ 是高斯核函数。

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

σ 称为尺度空间因子, 它是高斯正态分布的标准差, 反映了图像被模糊的程度, 其值越大图像越模糊, 对应的尺度也就越大。 $L(x, y, \sigma)$ 代表着图像的高斯尺度空间。
构建尺度空间的目的是为了检测出在不同的尺度下都存在的特征点, 而检测特征点较好的算子是 $\Delta^2 G$ (高斯拉普拉斯,LoG),

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

使用LoG虽然能较好的检测到图像中的特征点, 但是其运算量过大, 通常可使用**DoG (差分高斯, Difference of Gaussina)**来近似计算LoG[Marr and Hidreth]。
设 k 为相邻两个高斯尺度空间的比例因子, 则**DoG**的定义:

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

其中, $L(x, y, \sigma)$ 是图像的高斯尺度空间。
从上式可以知道, 将相邻的两个高斯空间的图像相减就得到了DoG的响应图像。为了得到DoG图像, 先要构建高斯尺度空间, 而高斯的尺度空间可以在图像金字塔降采样的基础上加上高斯滤波得到, 也就是对图像金字塔的每层图像使用不同的参数 σ 进行高斯模糊, 使每层金字塔有多张高斯模糊过的图像。降采样时, 金字塔上边一组图像的第一张是由其下面一组图像倒数第三张降采样得到。
易知, 高斯金字塔有多组, 每组又有多层。一组中的多个层之间的尺度是不一样的(也就是使用的高斯参数 σ 是不同的), 相邻两层之间的尺度相差一个比例因子 k 。如果每组有 S 层, 则 $k = 2^{\frac{1}{S}}$ 。上一组图像的最低层图像是由下一组中尺度为 2σ 的图像进行因子为2的降采样得到的(高斯金字塔先从底层建立)。高斯金字塔构建完成后, 将相邻的高斯金字塔相减就得到了DoG金字塔。
高斯金字塔的组数一般是

$$o = \lceil \log_2 \min(m, n) \rceil - a$$

o 表示高斯金字塔的层数, m, n 分别是图像的行和列。减去的系数 a 可以在 $0 - \log_2 \min(m, n)$ 之间的任意值, 和具体需要的金字塔的顶层图像的大小有关。
高斯模糊参数 σ (尺度空间), 可由下面关系式得到

$$\sigma(o, s) = \sigma_0 \cdot 2^{\frac{o+s}{S}}$$

其中 o 为所在的组, s 为所在的层, σ_0 为初始的尺度, S 为每组的层数。
在Lowe的算法实现中 $\sigma_0 = 1.6, o_{min} = -1, S = 3, o_{max} = 1$ 就是首先将原图像的长和宽各扩展一倍。
从上面可以得知同一组内相邻层的图像尺度关系

$$\sigma_{s+1} = k \cdot \sigma_s = 2^{\frac{1}{S}} \cdot \sigma_s$$

相邻组之间的尺度关系

$$\sigma_{o+1} = 2\sigma_o$$

2.3 高斯金字塔构建示例

以一个 512×512 的图像 I 为例, 构建高斯金字塔步骤:(从0开始计数, 倒立的金字塔)

3. 金字塔的组数, $\log_2 512 = 9$, 减去因子3, 构建的金字塔的组数为6。取每组的层数为3。
4. 构建第0组, 将图像的宽和高都增加一倍, 变成 1024×1024 (I_0)。第0层 $I_0 * G(x, y, \sigma_0)$, 第1层 $I_0 * G(x, y, k\sigma_0)$, 第2层 $I_0 * G(x, y, k^2\sigma_0)$
5. 构建第1组, 对 I_0 降采样变成 512×512 (I_1)。第0层 $I_1 * G(x, y, 2\sigma_0)$, 第1层 $I_1 * G(x, y, 2k\sigma_0)$ $I_1 * G(x, y, 2k^2\sigma_0)$



| |
|--|
| 2. OpenCV2:特征匹配及其优化(38984) |
| 3. OpenCV2:Mat属性type, depth, step(34242) |
| 4. Qt自适应大小显示图片, 添加菜单(33762) |
| 5. SIFT特征详解(30555) |

| |
|--|
| 评论排行榜 |
| 1. OpenCV2:特征匹配及其优化(16) |
| 2. OpenGL超级宝典visual studio 2013开发环境配置, GLTools(11) |
| 3. OpenCV2:图像的几何变换, 平移、镜像、缩放、旋转(1)(8) |
| 4. OpenCV2:Mat(7) |
| 5. FFmpeg学习3: 播放音频(7) |

| |
|---------------------------|
| 推荐排行榜 |
| 1. SIFT特征详解(27) |
| 2. 字符串模式匹配KMP算法(15) |
| 3. 图像处理基础(4): 高斯滤波器详解(15) |
| 4. OpenCV2:特征匹配及其优化(13) |
| 5. C++ 11 多线程--线程管理(11) |

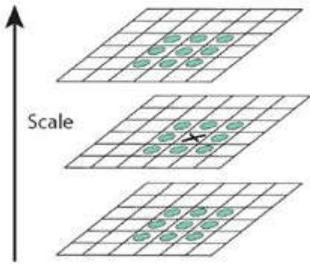
6. ∴
7. 构建第o组, 第s层 $I_o * G(x, y, 2^o k^s \sigma_0)$

高斯金字塔构建成功后, 将每一组相邻的两层相减就可以得到DoG金字塔。

3. DoG空间极值检测

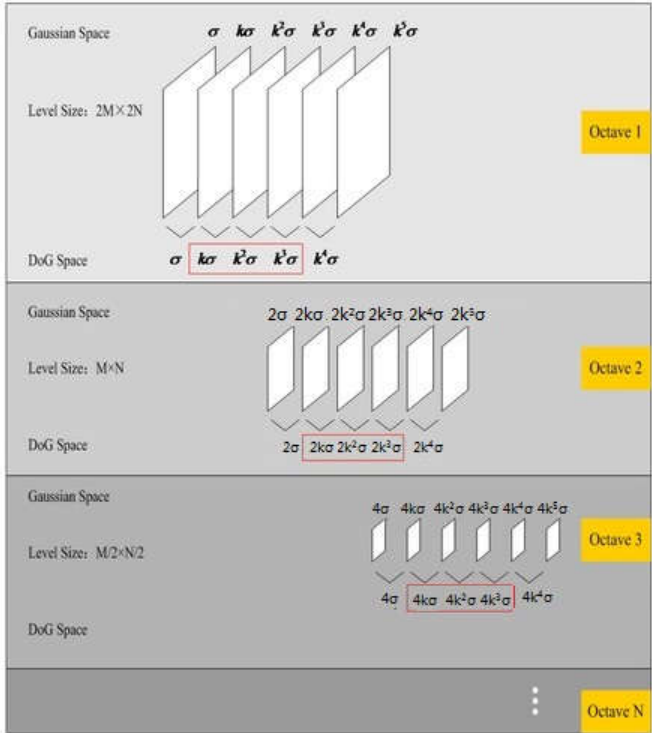
为了寻找尺度空间的极值点, 每个像素点要和其图像域(同一尺度空间)和尺度域(相邻的尺度空间)的所有相邻点进行比较, 当其大于(或者小于)所有相邻点时, 改点就是极值点。如图所示, 中间的检测点要和其所在图像的3 × 3邻域8个像素点, 以及其相邻的上下两层的3 × 3邻域18个像素点, 共26个像素点进行比较。

从上面的描述中可以知道, 每组图像的第一层和最后一层是无法进行比较取得极值的。为了满足尺度变换的连续性, 在每一组图像的顶层继续使用高斯模糊生成3幅图像, 高斯金字塔每组有S + 3层图像, DoG金字塔的每组有S + 2组图像。



3.1 尺度变化的连续性

设S = 3, 也就是每组有3层, 则 $k = 2^{\frac{1}{3}} = 2^{\frac{1}{3}}$, 也就是有高斯金字塔每组有(S - 1)3层图像, DoG金字塔每组有(S-2)2层图像。在DoG金字塔的第一组有两层尺度分别是σ, kσ, 第二组有两层的尺度分别是2σ, 2kσ, 由于只有两项是无法比较取得极值的(只有左右两边都有值才能有极值)。由于无法比较取得极值, 那么我们就需要继续对每组的图像进行高斯模糊, 使得尺度形成σ, kσ, k²σ, k³σ, k⁴σ, 这样就可以选择中间的三项kσ, k²σ, k³σ。对应的下一组由上一组降采样得到的三项是2kσ, 2k²σ, 2k³σ, 其首项2kσ = 2 · 2^{1/3}σ = 2^{4/3}σ, 刚好与上一组的最后一项k³σ = 2^{3/3}σ的尺度连续起来。



4. 删除不好的极值点 (特征点)

通过比较检测得到的DoG的局部极值点实在离散的空间搜索得到的, 由于离散空间是对连续空间采样得到的结果, 因此在离散空间找到的极值点不一定是真正意义上的极值点, 因此要设法将不满足条件的点剔除掉。可以通过尺度空间DoG函数进行曲线拟合寻找极值点, 这一步的本质是去掉DoG局部曲率非常不对称的点。

要剔除掉的不符合要求的点主要有两种:

1. 低对比度的特征点

2. 不稳定的边缘响应点

4.1 剔除低对比度的特征点

候选特征点 x ，其偏移量定义为 Δx ，其对比度为 $D(x)$ 的绝对值 $|D(x)|$ ，对 $D(x)$ 应用泰勒展式

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

由于 x 是 $D(x)$ 的极值点，所以对上式求导并令其为0，得到

$$\Delta x = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D(x)}{\partial x}$$

然后再把求得的 Δx 代入到 $D(x)$ 的泰勒展开式中

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

设对比度的阈值为 T ，若 $|D(\hat{x})| \geq T$ ，则该特征点保留，否则剔除掉。

4.2 剔除不稳定的边缘响应点

在边缘梯度的方向上主曲率值比较大，而沿着边缘方向则主曲率值较小。候选特征点的DoG函数 $D(x)$ 的主曲率与 2×2 Hessian矩阵 H 的特征值成正比。

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

其中， D_{xx}, D_{xy}, D_{yy} 是候选点邻域对应位置的差分求得的。

为了避免求具体的值，可以使用 H 特征值得比例。设 $\alpha = \lambda_{max}$ 为 H 的最大特征值， $\beta = \lambda_{min}$ 为 H 的最小特征值，则

$$\begin{aligned} Tr(H) &= D_{xx} + D_{yy} = \alpha + \beta \\ Det(H) &= D_{xx} D_{yy} - D_{xy}^2 = \alpha \cdot \beta \end{aligned}$$

其中， $Tr(H)$ 为矩阵 H 的迹， $Det(H)$ 为矩阵 H 的行列式。

设 $\gamma = \frac{\alpha}{\beta}$ 表示最大特征值和最小特征值的比值，则

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

上式的结果与两个特征值的比例有关，和具体的大小无关，当两个特征值想等时其值最小，并且随着 γ 的增大而增大。因此为了检测主曲率是否在某个阈值 T_γ 下，只需检测

$$\frac{Tr(H)^2}{Det(H)} > \frac{(T_\gamma + 1)^2}{T_\gamma}$$

如果上式成立，则剔除该特征点，否则保留。（Lowe论文中取 $T_\gamma = 10$ ）

5. 求取特征点的主方向

经过上面的步骤已经找到了在不同尺度下都存在的特征点，为了实现图像旋转不变性，需要给特征点的方向进行赋值。利用特征点邻域像素的梯度分布特性来确定其方向参数，再利用图像的梯度直方图求取关键点局部结构的稳定方向。

找到了特征点，也就可以得到该特征点的尺度 σ ，也就可以得到特征点所在的尺度图像

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

计算以特征点为中心、以 $3 \times 1.5\sigma$ 为半径的区域图像的幅角和幅值，每个点 $L(x, y)$ 的梯度的模 $m(x, y)$ 以及方向 $\theta(x, y)$ 可通过下面公式求得

$$m(x, y) = \sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x, y+1) - L(x, y-1)]^2}$$

$$\theta(x, y) = \arctan \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}$$

Fork me on GitHub

计算得到梯度方向后，就要使用直方图统计特征点邻域内像素对应的梯度方向和幅值。梯度方向的直方图的横轴是梯度方向的角度（梯度方向的范围是0到360度，直方图每36度一个柱共10个柱，或者没45度一个柱共8个柱），纵轴是梯度方向对应梯度幅值的累加，在直方图的峰值就是特征点的主方向。在Lowe的论文还提到了使用高斯函数对直方图进行平滑以增强特征点近的邻域点对关键点方向的作用，并减少突变的影响。为了得到更精确的方向，通常还可以对离散的梯度直方图进行插值拟合。具体而言，关键点的方向可以由主峰值最近的三个柱值通过**抛物线插值**得到。在梯度直方图中，当存在一个相当于主峰值80%能量的柱值时，则可以将这个方向认为是该特征点辅助方向。所以，一个特征点可能检测到多个方向（也可以理解为，一个特征点可能产生多个坐标、尺度相同，但是方向不同的特征点）。Lowe在论文中指出

15%的关键点具有多方向，而且这些点对匹配的稳定性很关键。

得到特征点的主方向后，对于每个特征点可以得到三个信息 (x, y, σ, θ) ，即位置、尺度和方向。由此可以确定一个SIFT特征区域，一个SIFT特征区域由三个值表示，中心表示特征点位置，半径表示关键点的尺度，箭头表示主方向。具有多个方向的关键点可以被复制成多份，然后将方向值分别赋给复制后的特征点，一个特征点就产生了多个坐标、尺度相等，但是方向不同的特征点。

6. 生成特征描述

通过以上的步骤已经找到了SIFT特征点位置、尺度和方向信息，下面就需要使用一组向量来描述关键点也就是生成特征点描述子，这个描述符不只包含特征点，也含有特征点周围对其有贡献的像素点。**描述子应具有较高的独立性，以保证匹配率。**

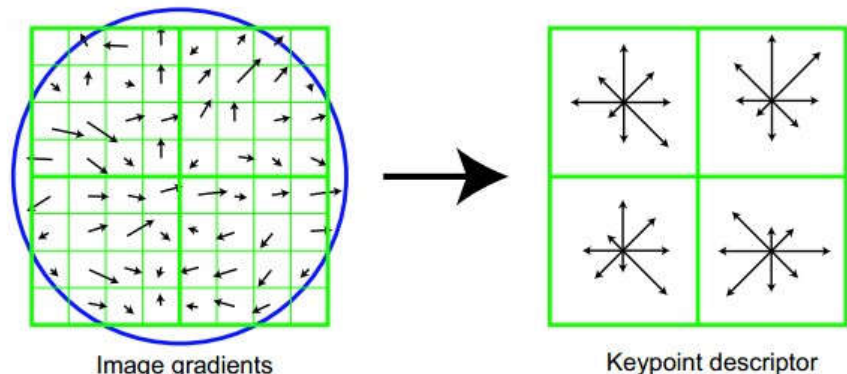
特征描述符的生成大致有三个步骤：

1. 校正旋转主方向，确保旋转不变性。
2. 生成描述子，最终形成一个128维的特征向量
3. 归一化处理，将特征向量长度进行归一化处理，进一步去除光照的影响。

为了保证特征矢量的旋转不变性，要以特征点为中心，在附近邻域内将坐标轴旋转 θ （特征点的主方向）角度，即将坐标轴旋转为特征点的主方向。旋转后邻域内像素的新坐标为：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

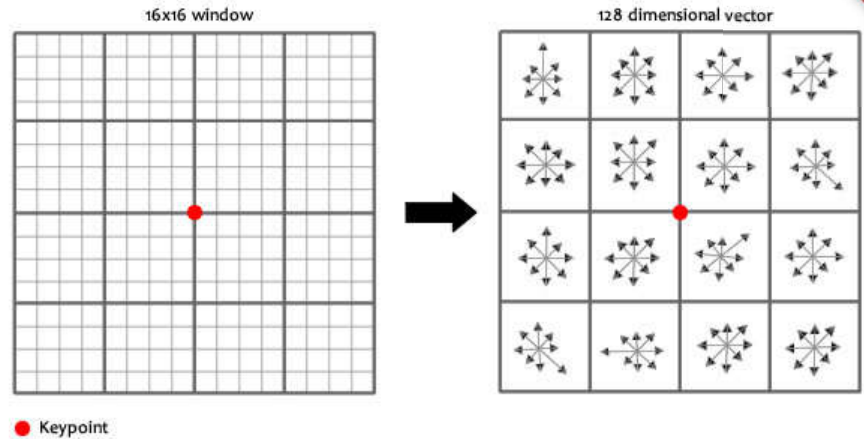
旋转后以主方向为中心取 8×8 的窗口。下图所示，左图的中央为当前关键点的位置，每个小格代表为关键点邻域所在尺度空间的一个像素，求取每个像素的梯度幅值与梯度方向，箭头方向代表该像素的梯度方向，长度代表梯度幅值，然后利用高斯窗口对其进行加权运算。最后在每个 4×4 的小块上绘制8个方向的梯度直方图，计算每个梯度方向的累加值，即可形成一个种子点，如右图所示。每个特征点由4个种子点组成，每个种子点有8个方向的向量信息。这种邻域方向性信息联合增强了算法的抗噪声能力，同时对于含有定位误差的特征匹配也提供了比较理性的容错性。



与求主方向不同，此时每个种子区域的梯度直方图在0-360之间划分为8个方向区间，每个区间为45度，即每个种子点有8个方向的梯度强度信息。

在实际的计算过程中，为了增强匹配的稳健性，Lowe建议

对每个关键点使用 4×4 共16个种子点来描述，这样一个关键点就可以产生128维的SIFT向量。



通过对特征点周围的像素进行分块，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。

7. 总结

SIFT特征以其对旋转、尺度缩放、亮度等保持不变性，是一种非常稳定的局部特征，在图像处理和计算机视觉领域有着很重要的作用，其本身也是非常复杂的，下面对其计算过程做一个粗略总结。

1. **DoG尺度空间的极值检测。** 首先是构造DoG尺度空间，在SIFT中使用不同参数的高斯模糊来表示不同的尺度空间。而构造尺度空间是为了检测在不同尺度下都存在的特征点，特征点的检测比较常用的方法是 $\Delta^2 G$ （高斯拉普拉斯LoG），但是LoG的运算量是比较大的，Marr和Hidreth曾指出，可以使用**DoG**（差分高斯）来近似计算LoG，所以在DoG的尺度空间下检测极值点。
2. **删除不稳定的极值点。** 主要删除两类：低对比度的极值点以及不稳定的边缘响应点。
3. **** 确定特征点的主方向**。** 以特征点的为中心、以 $3 \times 1.5\sigma$ 为半径的领域内计算各个像素点的梯度的幅角和幅值，然后使用直方图对梯度的幅角进行统计。直方图的横轴是梯度的方向，纵轴为梯度方向对应梯度幅值的累加值，直方图中最高峰所对应的方向即为特征点的方向。
4. **生成特征点的描述子。** 首先将坐标轴旋转为特征点的方向，以特征点为中心的 16×16 的窗口的像素的梯度幅值和方向，将窗口内的像素分成16块，每块是其像素内8个方向的直方图统计，共可形成128维的特征向量。

如果您觉得阅读本文对您有帮助，请点一下“推荐”按钮，您的“推荐”将是我最大的写作动力！欢迎各位转载，但是未经作者本人同意，转载文章之后必须在文章页面明显位置给出作者和原文连接，否则保留追究法律责任的权利。

分类：00-图像处理

好文要顶 关注我 收藏该文



Brook_icv
关注 - 10
粉丝 - 249

+加关注

« 上一篇：OpenCV，计算两幅图像的单应矩阵
» 下一篇：struct 大小计算

posted @ 2015-10-03 12:33 Brook_icv 阅读(30555) 评论(0) 编辑 收藏

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

**最新IT新闻:**

- 到自己的网络安全启蒙地 周鸿祎再谈大安全理念
 - 外媒：小米高管和承销商在IPO中出现失误 留下教训
 - 蔚来汽车创始人李斌：我们不是PPT造车 已开始交付
 - 这个原因 让无人车研发企业纷纷创办出租车公司
 - 苹果市值冲关万亿美元遭失败 分析称市值被低估
- » 更多新闻...

**最新知识库文章:**

- 如何提升你的能力？给年轻程序员的几条建议
 - 程序员的那些反模式
 - 程序员的宇宙时间线
 - 突破程序员思维
 - 云、雾和霭计算如何一起工作
- » 更多知识库文章...