

基于 SIFT 图像特征提取与 FLANN 匹配算法的研究

王金龙, 周志峰

(上海工程技术大学, 上海 201600)

摘要: 针对传统的图像匹配算法存在误匹配率高, 特征提取信息也比较少的问题; 文章提出了 SIFT 与 FLANN 匹配算法结合起来的一种图像匹配方法; 首先, 将尺度不变特征变换算法 SIFT 应用到图像的特征点提取与匹配中; 其次, SIFT 算法主要就是在在构建成功的尺度空间中寻找出极值点, 找出稳定的特征描述, 特征描述必须具备以下特征, 对尺度、光照以及图像的变形都要具有很强的适应能力; 最终, 利用高维数据的快速最近邻算法 FLANN 进行特征匹配, 找到图像中准确的匹配点对, 研究了 SIFT 这种应用广泛的特征点提取算法, 通过实验比较了这种组合算法的特征点提取在不同光照, 平移, 旋转下的结果, 匹配精度满足一定的要求。

关键词: SIFT; 特征提取; FLANN 匹配算法; 特征匹配

Image Matching Algorithm Based on SIFT Feature Extraction and FLANN Search

Wang Jinlong, Zhou Zhifeng

(Shanghai University of Engineering Science, Shanghai 201600, China)

Abstract: The traditional algorithm of image matching exist the problem of high rate false match and little feature information. An image matching algorithm is presented based on SIFT feature extraction and FLANN. Firstly, the scale invariant feature transformation algorithm SIFT is applied to the feature point extraction and matching of images. Secondly, the SIFT algorithm is mainly used to find out the extreme points and stable feature descriptions in the building of the successful scale space. Feature description must have a strong ability to adapt to scale, illumination and image distortion. Finally, we use the fast and nearest neighbor algorithm FLANN of high dimensional data to find the exact matching point in the image. In this paper, we study the widely used feature extraction algorithm of SIFT. The experimental results are obtained by comparing the characteristics of the algorithm with different illumination, translation and rotation and the result meets certain requirements.

Keywords: SIFT; feature extraction; FLANN matching algorithm; feature matching

0 引言

特征点的检测和匹配是计算机视觉中非常重要的技术之一, 在物体检测、视觉跟踪、三维重建等领域都有非常广泛的应用。

目前特征提取方法可以分为 3 种: 一是基于所制定模板的特征检测, 二是基于图形边缘的特征检测, 三是基于亮度变化特征检测^[1]。第一种由于设计的模板会随着检测图像的变化而变化, 所以遇到复杂的图像时就显得不适用, 第二种必须先进行边缘化处理之后才可以进行特征检测, 第三种就是目前的研究热点, Harris^[2]、SIFT、SURF 均属于这种类型。目前国内外学者已经在这方面做了大量的研究。图像的特征匹配技术主要分为两类, 一是基于灰度值的图像匹配, 二是基于特征的图像匹配方法, 其中基于灰度值的匹配方法, 主要是利用空间中一维或者二维的滑动模板实现图像的匹配, 这样做的优点在于匹配率高, 然后计算量太大, 所以匹配时间会比较长。而基于

特征的图像匹配的这种方法, 主要是通过提取出图像的一些显而易见的稳定的特征, 将不同图像中的一些相同性质关联起来, 由于不需要穷举匹配, 所以匹配速度较快。本文将 SIFT 特征提取与 FLANN 匹配算法结合在一起, 实现了对两幅图像的特征匹配, 并通过 VS2015 与 Opencv 库结合, 用 C++ 语言进行特征提取与匹配算法的实现, 并验证了在旋转, 亮度变化的情况下仍然能实现较精确的匹配结果。

1 特征点提取算法 (SIFT)

SIFT (Scale-invariant feature transform) 是一种检测局部特征的算法, 该算法通过求一幅图中的特征点及其有关的 scale 和 orientation 的描述子得到特征并进行图像特征点匹配, SIFT 特征不仅仅具有尺度不变性, 即使改变旋转角度, 图像的亮度等条件, 也能实现很好的检测效果。文章会针对 SIFT 特征做相应的理论分析, 并验证这一结论, 并与 FLANN 匹配算法结合, 实现快速准确的匹配。大致分为以下几个步骤: 构建尺度空间, LOG 近似 DOG 找到关键点及检测 DOG 尺度空间的极值点, 精确定位特征点, 确定特征点的方向, 最后生成 SIFT 特征描述子。

1.1 构建尺度空间

尺度空间理论就是利用高斯核函数对图像进行尺度变换来模拟图像数据的多尺度特征。获得图像在尺度空间下的多尺度

收稿日期: 2017-06-22; 修回日期: 2017-07-21。

作者简介: 王金龙(1993-), 男, 江苏泰州人, 硕士研究生, 主要从事机器视觉, 图像处理方向的研究。

周志峰(1976-), 男, 江苏常州人, 教授, 硕士研究生导师, 主要从事卫星北斗导航定位方向的研究。

序列表示。高斯卷积核是实现尺度变换的唯一线性核，如式 1 所示，一幅二维图像的尺度空间可以表示为^[3]：

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

其中： $G(x, y, \sigma)$ 是尺度可变高斯函数，随着尺度因子 σ 不同，将会产生不同尺度下的一组图像 $L(x, y, \sigma)$ ，称为高斯尺度空间， σ 大小决定图像的平滑程度，大尺度主要展示了图像的大致的外貌特征，小尺度主要展示图像的细节部分^[4]。大的 σ 值表示是图像比较粗糙的尺度（低分辨率），反之，对应精细的尺度（高分辨率），图 1 显示了 Gaussian 尺度空间中随着 σ 的变化图像变的越来越模糊。为了有效地在尺度空间检测到稳定的关键点，提出了高斯差分尺度空间 (DOG scale-space)。如式 2 所示，采用了不同尺度的高斯差分核与图像卷积生成，如图 1 所示。

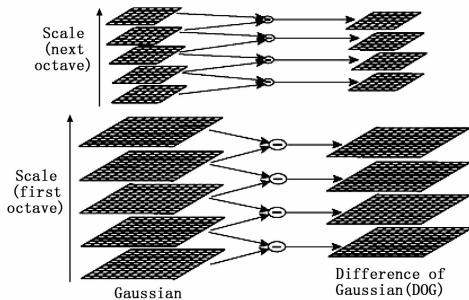


图 1 DOG 产生的原理图

$$D(x, y, \sigma) = G(x, y, \sigma) - G(x, y, \sigma) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (2)$$

对于尺度空间而言，在 Lowe 的论文中，他将第 0 层的初始尺度定义为 1.6，也就是说最模糊的，图片的初始尺度定义为 0.5（最清晰），在检验极值点之前，Lowe 建议在建立尺度空间之前，需要对原图进行长宽的扩展，以保留更多的图片信息，增加特征点的数量。

对于图像金字塔的建立问题，对于一幅图像，建立其在不同尺度 (scale) 的图像，也成为子八度，这是为了保证其尺度不变性 (scale-invariant)，也就是在任何尺度都能够有对应的特征点，第一个子八度的 scale 为原图大小（金字塔的最底端），后面每个 octave 为上一个 octave 降采样的结果，即原图的 1/4（长宽分别减半），构成下一个子八度（高一层金字塔）。每上一层是对下一层做 Laplacian 变换。

1.2 检测 DOG 空间的极值点

为了检测出高斯差分尺度空间中的存在的极值点，所选中的每一个采样点均要和周围的 26 个邻域点比较，即同尺度中相邻的 8 个像素点和上下相邻尺度的各 9 个像素点，总共 26 个像素点相比较，当采样点比这 26 个邻域点大或者小时，则将此点看作是候选的关键点^[5]，如图 2 所示。

在进行极值比较的过程中，每一组图像中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，对每组图像的顶层使用高斯模糊生成 3 幅图像，高斯金字塔每一组有 S+3 层图像，DOG 金字塔有 S+2 层。

1.3 精确定位特征点

如图 3 所示，展示了二维函数在离散空间里面所求出来的极值点与连续空间中的极值点区别。

通过拟合三维二次函数以准确的确定关键点的位置和尺度

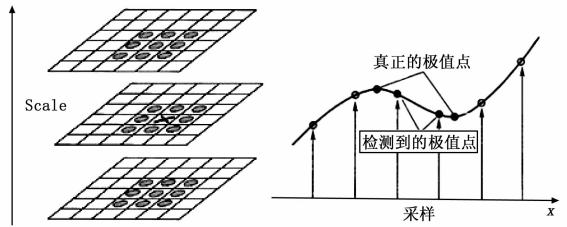


图 2 极值点查找 图 3 离散空间极值与连续极值

(达到亚像素要求)，同时去除一些不稳定的边缘特征点，提高匹配的准确度和稳定性，主要分为以下几个步骤：

1) 使用子像素插值的方法，通过对离散的空间点不断的插值可以求出连续的空间中的极值点，对尺度空间 DOG 函数进行子像素插值也就是数学上的曲线拟合^[6]，运用 DOG 函数在尺度空间里面的泰勒级数展开式，如式 (3) 所示：

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X \quad (3)$$

对式 (3) 求导，然后让这个导数等于 0，即可解出相对极值的偏移量，这样可以得到相应极值点：

$$A = (x, y, \sigma)^T = - \frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \quad (4)$$

得到极值点 x_{max} 的亚像素的位置。如果偏移值大于 0.5 这个条件成立，这就说明靠近另一侧的像素点，这时候让另一侧的像素选为候选的特征点，循环重复上面的计算，这样就可以获得新的亚像素的位置，之后在用该亚像素精度的位置取代所有尺度之前的候选的特征点位置^[7]。

2) 在已经检测出的所有的特征点中，需要去去除一些无关的响应点，比如一些低对比度的特征点和一些不稳定的边缘响应点。把公式 (4) 代入公式 (3)，即在 DOG Space 的极值点处取值，只取前两项可得，如式 (5) 所示：

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (5)$$

若 $|D(\hat{x})| \geq 0.03$ ，此特征点就可以保留下来相反则舍去，这样就可以提高匹配的可靠性和稳定性。本质上就是去除一些不好的特征点（去除 DOG 局部曲率不对称的像素）。

1.4 确定特征点的方向

关键点领域像素的梯度方向是不同的，根据他们分布特性的不同为每一个关键点指定一个确定的方向，使其可以具备旋转不变性。这也是判断特征子优越性的一个重要因素。

针对于窗口的每一个采样点 $L(x, y)$ ，其梯度方向的幅值和方向分别可以用 $m(x, y)$ 和 $\theta(x, y)$ 公式表示，分别如式 (6) 和式 (7)：

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (6)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (7)$$

每一个关键点需要 3 个信息：位置，尺度，方向，这样可以确定一个 SIFT 特征区域。

做一个包含所有梯度方向的分布直方图，取值范围是一个圆周 $0 \sim 360^\circ$ ，划分每 10° 为一个 bin，这样可以分为 36 个 bin。每个采样点根据其梯度方向 $\theta(x, y)$ 加权统计到分布直方图中，取幅度 $m(x, y)$ 与贡献因子的乘积为规定的权值。贡献因子定义为采样点到关键点即窗口中心的距离长度，距离的量度遵循

以下原则: 如果距离越大, 那么贡献因子就会越小反之则会越大, 选择分布直方图的最大值为所选关键点在此邻域梯度方向中的主要方向^[8], 如图 4 所示。

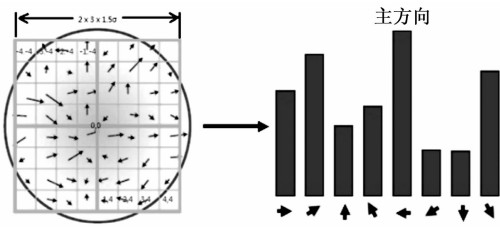


图 4 特征点方向的确示意图

1.5 生成 SIFT 特征描述子

SIFT 描述子是关键点领域高斯图像统计结果的一种表示, 特征描述子意味着特征点的一切信息包括梯度方向、幅值等等, 为了能够提高稳定性, 优秀的特征描述子应当包括此特征点的位置和灰度信息, 除此之外, 还需要反映这个特征点的一些局部的灰度变化信息。SIFT 特征描述子就是一个高维向量, 它包含着特征点的领域的所有信息, 生成特征描述子之前, 首先应该确定特征点邻域内像素的主方向, 我们可以选择 0 度作为主方向, 这样就可以消除旋转变换所带来的影响, 其次在每个 4×4 的 16 个区域中统计每个领域中的 8 个方向的梯度方向分布直方图。图 5 中, 选取了 16×16 的邻域, 要统计 16 个分布直方图, 所选择的每个直方图均代表了该领域内 8 个方向的信息, 这样就构成了 128 维的特征点描述子。

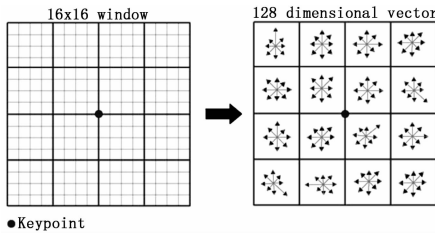


图 5 16X16 的特征点描述子

特征描述子需要具有光照不变性, 我们可以将特征向量通过式 (8) 归一化为单位长度, 下文的实验表现出很好的匹配效果。

$$n_i = \frac{n_i}{\left(\sum_{i=1}^{128} n_i^2\right)^{\frac{1}{2}}}, \quad i = 1, 2, 3, \dots, 128 \quad (8)$$

2 特征匹配最近邻算法 FLANN

在现代化的机器学习中, 训练一个高维特征数据, 然后找到训练数据中的最近邻计算是需要花费很高的代价的。对于一个高维特征, 目前来说最有效的方法是 The randomized k-d forest 和 The priority search k-means tree, 而对于二值特征的匹配 multiple hierarchical clustering trees 则比 LSH 方法更加有效^[9]。图 6 显示了特征匹配的一般步骤, 目前来说, fast library for approximate nearest neighbors (FLANN) 可以很好地解决这些问题, Muja 和 Lowe 于 2009 年提出 FLANN 算法, FLANN 算法模型的特征空间一般是 n 维实数向量空间, 该算法的核心是通过使用欧式距离来寻找与实例点的最邻近的点, 欧式距离的定义如式 (9) 所示。

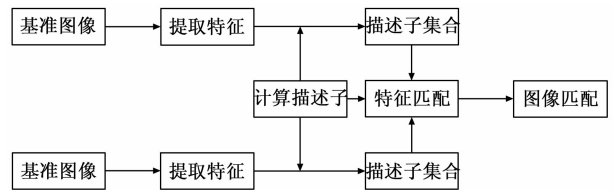


图 6 特征匹配的一般步骤

$$D(x, y) = \|X, Y\| = \sqrt{\sum_{i=1}^d (X_i - Y_i)^2} \quad (9)$$

如果 D 值越小, 这就表明了这些特征点对之间的距离越“近”, 也就是说它们相似程度越高。

2.1 快速近似 NN 匹配

2.1.1 随机 K-d 树算法

1) Classic k-d tree 求取出数据中方差最高的那个维度, 然后利用这个维度的数值将数据划分成 2 个部分, 接着对每个子集重复上述的相同的计算步骤。

2) Randomized k-d tree 通过创建许多颗随机树, 然后从那些具有最高方差的 $N-d$ 维中随机选取一些维度, 并用这些维度来对数据进行划分。另外在对随机 K-d 森林进行搜索时, 所有 K-d 均属于同一个优先级。从理论上说, 如果增加树的数量, 就能提高搜索速度^[10], 提高效率, 但由于硬件方面的种种限制, 树的数量需要控制在一定的范围内, 如果超出了速度不会增加甚至会变慢, 实现原理如图 7 所示。

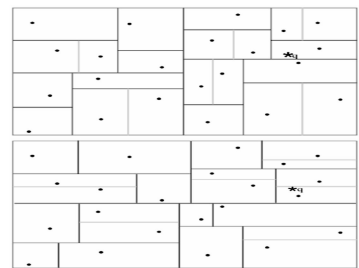


图 7 随机 K-d 森林的实现原理

2.1.2 层次聚类树

层次聚类树采用的是 K-medoids 的聚类方法, 而不是 K-means, 在本算法中, 并没有像在 K-medoids 聚类算法中直接求最小化方差求聚类中心, 而是在输入数据中随机选取聚类中心, 这种建立方式显得更加简单, 也可以保持各个树之间的独立性, 在建立多棵树的同时, 这个方法的搜索性能就提高了很多, 这主要是因为随机选取的聚类中心, 而不需要多次迭代计算聚类中心。建立多颗随机数的方法在 K-d tree 中比较有效, 在 K-means tree 中却不适用。

2.1.3 优先搜索 K-means 树算法

随机 k-d 森林适用范围比较广, 在很多情况下均有不错的搜索效果, 然后如果对精度要求比较高, 这样 k-means 树效果会更加好一点。K-means tree 充分挖掘了数据本身所固有的一些机构特征, 原原则是将数据的所有维度进行聚类处理, 与之前的随机 k-d tree 只使用了一次维度划分^[11]。本文采用的是 K-means 树的搜索原理, 算法描述如下:

- 1) 建立层次化的 K-means 树;
- 2) 树的节点就选层次化的聚类中心;
- 3) 如果某个 duster 内的点的数量小于 K 时, 在这样的前

提下就选择这些数据节点为叶子节点；

4) 从根节点 N 开始检索；

5) 如果 N 是叶子节点，则将处于相同层次的叶子节点添加到搜索结果中去，此时 $count + = |N|$ ；

6) 相反，如果 N 不是叶子节点，则将它的子节点与 query Q 比较，找出最近的那个节点 C_q ，并将同层次的其他节点加入到我们所考虑的优先队列中；

7) 对 C_q 节点进行递归搜索；

8) 如果优先队列不为空和 $count < L$ 条件成立，则将优先队列的第一个元素取出来并赋值给 N ，然后循环到步骤 4)

在匹配的过程中难免会出现错误的匹配对，通过 K -mean 算法处理之后，匹配的精度达到很高，速度也比较快。

3 实验结果及分析

本次实验中，通过对两幅图像分别进行旋转，缩放以及改变光照条件，检测该组合算法的抗干扰性以及匹配的成功率，检查匹配的准确率和速度。

3.1 光照变化

通过改变匹配时某个图片的光照情况（图片的亮度），检测特征描述子对光照变化的适应情况，实验结果如图 8 所示。

3.2 缩放变化

通过对匹配的物体进行缩放，检测缩放前后特征点匹配情况，实验结果如图 9 所示。



图 8 不同光照情况的结果图 图 9 物体缩放前后匹配对比图

3.3 旋转变化

同一场景下，对物体进行不同角度的旋转，通过对比，得出文章中组合算法对旋转变化的适应能力，结果如图 10 所示。

3.4 SIFT 与 SURF 对比

本文采用 SIFT 特征提取与 FLANN 匹配算法结合的方法，对匹配图像进行了综合性的实验，并与 SURF 特征提取与匹配进行了对比，对比结果图 11 所示。



图 10 不同旋转角度图像间匹配图 图 11 SIFT 与 SURF 提取匹配对比

表 1 SIFT 和 SURF 特征匹配对比

特征提取的方法	特征点个数	匹配点	运行时间/s	成功率/%
SIFT	120	90	1.3	75
SURF	115	89	0.9	77

从实验结果中可以看出 SIFT 算法的匹配准确度比 SURF 高很多，但是由于 SIFT 算法的复杂性，在特征点提取的过程中，时间较长。通过实验 SIFT 和 SURF 特征匹配的数据如表 1 所示。

4 结论

特征点提取是图像处理领域重要的一个环节，是接下来的图像匹配的前提，本文采用 SIFT 算法提取的图像的特征点，并与 SURF 特征点进行了简单的对比，SURF 算法的准确性较 SIFT 高很多，SIFT 对特征细节的表达也比 SURF 高很多，通过本次实验分析，可以得出 SIFT 特征算子对缩放、旋转、亮度变化的适应能力较强。虽然存在一些误差，但整体准确度比较高，满足一定的匹配要求。本文所采用的 FLANN 匹配算法的 K -means tree 匹配的准确率高，应用场景较广。本文的算法组合缺点也很明显，问题主要是在 SIFT 特征提取的时间比较长不能很好的应用到实时性处理中，但可以将 SIFT 与 LBP 特征结合^[12]，可以提高效率，改善算法。实验结果表明，本文的组合算法对图像的亮度，旋转，缩放等各个方面都有较强的适应性。可以应用与图像识别，三维重建等热门领域，在后续的研究中，提高 SIFT 算法的效率是关键，可以通过改进 SIFT 特征提取的步骤来提高提取速度。

参考文献：

[1] 谭博怡. 图像特征提取与匹配 [D]. 北京：中国科学院研究生院，2008.

[2] Harris C, Stephens M. A combined corner and edge detector [A]. Manchester: Proceedings of the 4th Alvey Vision Conference [C]. Manchester, UK, 1988: 147 - 151.

[3] Lowe D G. Object Recognition from Local Scale-Invariant Keypoints [J]. International Journal of Computer Vision, 2004, 60 (2): 91 - 110.

[4] Schmid C, Mohr R, Bouckhage C. Evaluation of interest point detectors [J]. International Journal of Computer Vision, 2000, 37 (2): 151 - 172.

[5] Bay H, Tuytelaars T, Van Gool L. SURF: speeded up robust features [A]. Proceedings of European Conference on Computer Vision [C]. Graz, Austria, 2006: 404 - 407.

[6] Marius Muja, Lowe D G. Scalable Nearest Neighbor Algorithms for High Dimensional Data [J].

[7] Brown M, Lowe D. Invariant features from interest point groups [A]. British Machine Vision Conference [C]. Cardiff, Wales, 2002: 656 - 665.

[8] Moller T, Haines E, Akenine - Moller T. 实时极端及图形学 [M]. 曹建涛, 译. 北京：北京大学出版社.

[9] Muja M, Lowe D G. Fast approximate nearest neighbors with automatic algorithm configuration [A]. Proceedings of IEEE Conference on Computer Vision Theory and Applications [C]. Lisbon, Portugal; IEEE Computer Society, 2009: 331 - 340

[10] 刘树勇, 杨超庆, 位秀雷, 等. 邻近点快速搜索方法在混沌识别中的应用 [J]. 华中科技大学学报, 2012, 40 (11): 89 - 92.

[11] 崔江涛, 刘卫光. 一种多分辨率高维图像特征匹配算法 [J]. 光子学报, 2005, 34 (1): 138 - 141.

[12] Zhao G, Pietikainen M. Dynamic texture recognition using local binary patterns with an application for facial expressions [A]. Asia Conference on Computer Vision (ACCV) [C]. 2011 (3): 281 - 292.