**摘要：**

最近几年在计算机视觉研究的各个领域可以看到对不变量特征点方法的大量使用。不变量特征点的引入，使尺度不变特征变换（sift）成为其中最有效和最广为使用的的方法之一，并且作为该方法流行的主要刺激因素。本文展示了开源sift库，以C实现，并且在http://eecs.oregonstate.edu/~hess/sift.html上免费提供，并与传统的David Lowe的提供的Sift执行效率作个简要的比较。

**1、引言：**

局部不变图像特征在计算机视觉领域扮演了一个基本的角色，帮助计算图像在每个点和块级别上的匹配。由于近年对稳健局部特征的检测和描述上的进步，局部特征的使用变得流行起来，在几乎每个计算机视觉研究领域，从3D视觉[12,5]，到目标识别[6,9]、机器人定位和地图创建[14,11]、目标跟踪[3,13] ，以及各个交叉领域、

尺度不变特征变换，或者SIFT算法[7,8]，是当今最广为人知和使用的局部不变特征方法，因为它是第一个通过一个稳健的描述子将不变量与旋转、尺度、和大范围的仿射变换以及光照改变联系起来，并且该描述子在能在大量数据中进行有效地匹配。SIFT算法本身扮演了一个主要的角色，在过去十年里对于局部不变图像特征方法的使用中。不幸的是，不管SIFT的大量使用，David Lowe，SIFT的创始人，只提供了该算法的二进制可执行版本，开发者通过使用计算机视觉软件可以生成SIFT的链接库。作为我自己的计算机视觉研究的一部分，我实现了SIFT算法的C版本，如实地根据Lowe04年发布的论文[8]，使用了通用的开源计算机视觉库OpenCV[10]。考虑到它在通用计算机视觉领域的潜在用途，我发布了我的SIFT实现代码在2006年，作为一个开源的库。在它发布的时候，它是SIFT算法的第一个开源可用版本，自从发布后，有相当多的使用。

在本文中，我简短地描述下SIFT算法以及我的开源SIFT库的实现，并且与原始版本做个比较。

**2、SIFT算法：**

SIFT算法主要通过4步来检测和描述一幅图像中的局部特征，或特征点：

1、在尺度空间上检测极值点

2、亚像素定位和特征点过滤

3、分配特征点的典型方向

4、计算特征点的描述子

**尺度空间极值检测：**SIFT算法首先，确定候选特征点的位置，查找作为图像尺度空间的二阶导数的近似的高斯差分金字塔中局部最大值和最小值。有兴趣的读者可以查看[8]。

**特征点定位和过滤：**在确定候选特征点后，它们在尺度空间中的位置已经插值到亚像素精度。并且插值点如果是低对比度或者高边缘效应-----计算基于主曲率的比率，那么由于潜在的不稳定将过滤掉该点。

**方向分配：**在过滤后剩下来的那些点，将被赋予一个或多个主方向，根据局部尺度空间梯度的主方向。在方向分配后，每个特征点的描述子就能依据特征点的位置、尺度和方向来进行计算，针对这些变换提高不变性。

**描述子计算：**最终，每个特征点的描述符将被计算，通过将该特征点周围尺度空间区域划分为网格，计算每个网格内局部梯度方向的直方图，并将这些直方图连接成一个向量。为了提供对光照变化的不变性，每个描述向量进行归一化，设定阈值来减少大梯度值的影响，再重新归一化。

**3、开源SIFT库**

该库是C版本的，并且对于Linux和Windows都是可用的，使用了OpenCV[10]。特别地，SIFT库的函数API使用OpenCV的数据类型来表示图像、矩阵等，为了使它能够容易地和现有的OpenCV版本结合。除此之外，所有内部操作也是用OpenCV函数实现的。

该库本身包含了4个主要部分，每个以一个不同的头文件表示。接下来会单独介绍。之后，我介绍3个简单的包含SIFT库的例子应用。

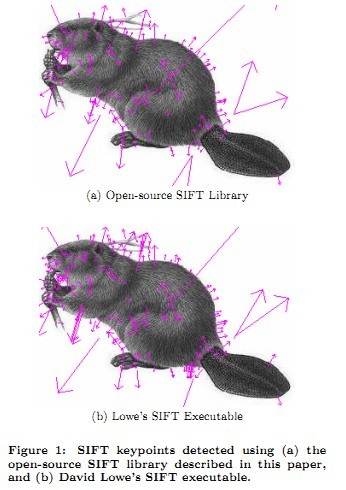
**3.1、SIFT库包含部分：**

**SIFT特征点检测：**本库的主要部分是一系列检测特征点的函数。特别地，本库包含了两种SIFT特征点的检测方法（在 sift.h头文件中），一个是直接使用Lowe在论文中建议的默认的参数设置[8]，另一个允许用户自行设定参数。这些函数很容易被调用，特别地，他们不需要初始化并且同时接受灰度和RGB图像类型（RGB图像类型在内部转换为灰度图）。特别地，下面的代码片段就是计算从本地读取的彩色图像的SIFT特征点所需的全部代码。

**[cpp]** [view plaincopy](http://blog.csdn.net/selinahuiling/article/details/9021929)

1. <span style="font-size:14px;">IplImage\* img;      /\* OpenCV image type \*/
2. **struct** feature\* keypoints;       /\* SIFT library keypoint type \*/
3. **int** n;     /\* feature count \*/
4. /\* load image using OpenCV and detect keypoints \*/
5. img = cvLoadImage( "/path/to/image.png", 1 );
6. n = sift\_features( img, &keypoints);</span>

图 1 描画了使用SIFT库所检测的特征点。作为比较，使用Lowe的课执行SIFT软件特征点检测也放在图1中。



**Kd-tree 特征点数据库的形成：**对一幅图上的SIFT特征点与另一幅图上的特征点或者大量的特征点数据库进行有效地匹配是基本的功能。在文章[1]中，Beis和Lowe提出一种方法来促进有效地特征点匹配，通过使用kd-tree和最近邻搜索的近似（拥有非常高的正确率）。本连接库也包含了该结构和函数（在kdtree.h头文件中）来实现该方法，和在[5]中提出的匹配方法一样。

**Ransac 变换计算：**SIFT特征点和其他局部图像特征点经常被用来计算变换---例如图像间的基本矩阵或者二维单应矩阵。尤其，一旦图像特征在图像间匹配上了，这种匹配关系的形成可以用来分析计算所需的变换。Ransac算法[2]被广泛应用于有部分外特征点匹配存在的情况下变换的计算。在SIFT库中（在xform.h头文件中）包含了一系列函数通过使用Ransac从图像匹配中计算图像变换。这些函数都很灵活。特别地，这个变换函数本身队库的Ransac函数来说是一个论据。因此，开发者可以自由实现任何他希望的函数根据2维点对应关机计算变换。该实现必须遵守在库里定义的函数原型。作为一个例子，库包含了函数可以用来与Ransac相结合来计算图像间的二维单应性矩阵。

图 2 描画了两幅图像间SIFT特征点匹配，通过使用库里的kd-tree函数和Ransac函数根据匹配点计算变换。



**不变图像特征处理：**最后，SIFT库也包含了一系列结果和方法来处理不变图像特征数据，包括由其他软件产生的数据。尤其，库的这部分包含了一个数据结构来表示图像特征数据以及提供了方法来导入和导出使用库自己的SIFT方法所计算的特征点，也可以导入Lowe的算法所计算的特征点以及牛津大学视觉几何小组的软件所计算的仿射协变特征点。利用这种功能性（在imgfeatures.h头文件里），kd-tree和Ransac算法可以应用于以上各个特征点类型。

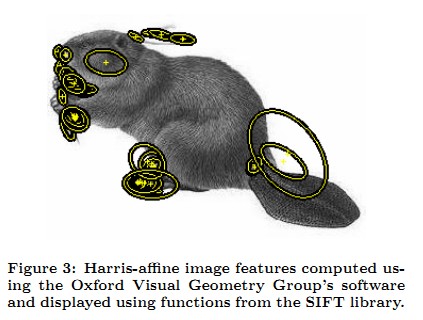
**3.2、例子应用：**

SIFT库也包含了3个非常简单的例子，展示了库的功能性。

1、siftfeat.c：这个例子只简单计算了一幅图里的SIFT特征点，并且输出到文件里。如图1（a）所示。

2、match.c：这个例子计算两幅图的SIFT特征点的匹配，利用库的kd-tree函数，并且依据这些匹配可选择地利用ransac计算一个变换。如图2所示。

3、dspfeat.c：这个例子导入和展示其他任何兼容的软件所生成的图像特征。图1显示了本库计算的sift特征点和Lowe的SIFT可执行结果，并导入。图3描画了Harris-affine特征点，利用牛津大学小组的软件计算所得。

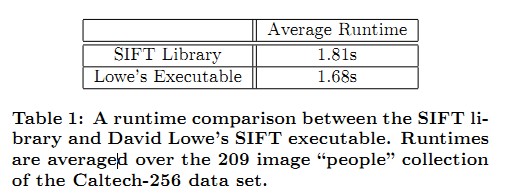


**4、性能**

下面对SIFT库的性能和Lowe的就运行时间和匹配以及变换的精度作对比。

**4.1、运行时间**

表 1 比较了两者的运行时间。实验图像数据来源于[4]。图像的平均检测面积是285350平方像素。两者的运行时间是类似的。



**4.2、匹配和变换的精确性**

表 2 比较了两者在特征点匹配和变换计算的精确性。这些结果是通过对10幅随机选择的图像作随机的透视变换，计算特征点匹配在原图和变换图之间，之后利用ransac计算透视变换。变换的精确性是通过计算原始特征点通过计算得到的变换矩阵和实际的透视变换计算所得到的点位置之间的MSE（均方误差）。同样两者的性能是类似的。

**5、参考文献**

[1] J. S. Beis and D. G. Lowe. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces. In CVPR, 2003.

[2] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model tting with applications to image analysis and automated cartography. Communications of the ACM, 24(6),1981.

[3] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In CVPR, 2010.

[4] G. Grin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[5] R. Hess and A. Fern. Improved video registration using non-distinctive local image features. In CVPR, 2007.

[6] B. Leibe, A. Leonardis, and S. Bernt. Robust object detection with interleaved categorization and segmentation. IJCV, 77(1{3), 2008.

[7] D. G. Lowe. Object recognition from local scale-invariant features. In ICCV, 1999.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. Intl. Journal of Computer Vision, 60(2):91{110, 2004.

[9] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. IEEE TPAMI, 28(3), 2006.

[10] OpenCV. http://opencv.willowgarage.com/.

[11] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. IEEE T-RO, 21(3), 2005.

[12] N. Snavely, R. Garg, S. M. Seitz, and R. Szeliski. Finding paths through the world's photos. ACM TOG (Proceedings of SIGGRAPH 2008), 27(3), 2008.

[13] S. Tran and L. Davis. Robust object tracking with regional ane invariant features. In ICCV, 2007.

[14] B. Williams, G. Klein, and I. Reid. Real-time SLAM relocalization. In ICCV, 2007.