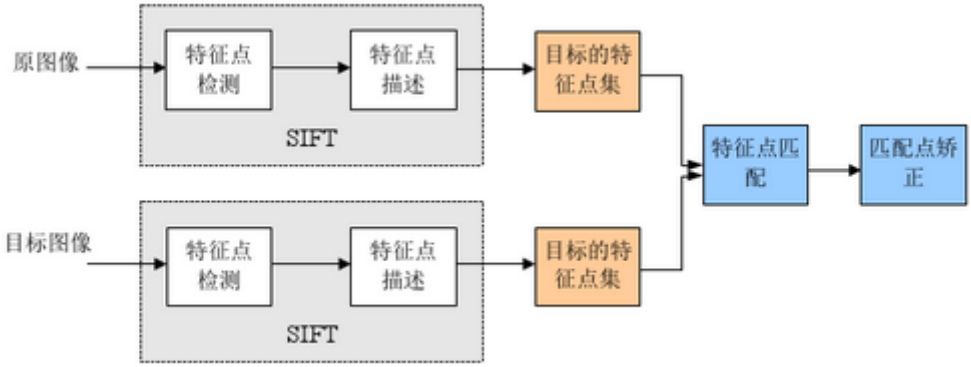
SIFT 尺度不变特征变换

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的图像局部特征描述算子。
sift 算法是指可以归为在不同尺度空间上查找特征点（关键点）的问题。图一为实现的步骤。

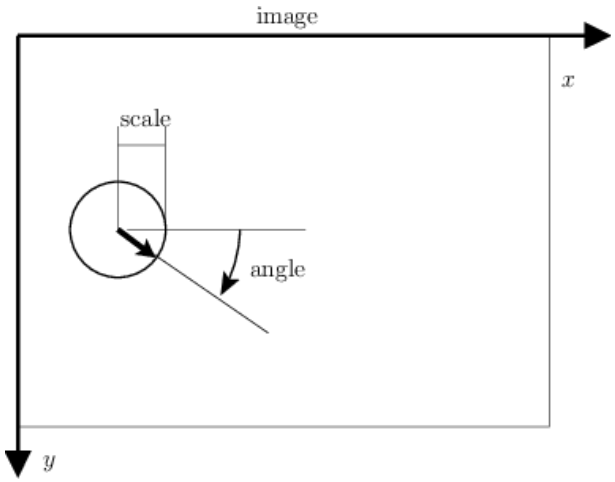


图一

SIFT detector

第一步就是特征点（关键点）的检测，所谓特征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。所谓的尺度空间的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的征提取等。尺度越大图像越模糊。

一个特征点可看做是一个有方向的圆形区域，有四个参数：圆心、坐标轴 xy、半径、方向（弧度）。如图二所示。



图二

通过构建“高斯尺度空间”来获得多尺度的关键点。尺度空间的构造受以下参数的影响，是通过 vl_sift_new 创建 SIFT 过滤器对象时来进行设置：

number of octave
first octave index
number of levels per octave
peak threshold　它是通过使用 vl_sift_set_peak_thresh（）配置 SIFT 过滤器对象来设置的。
edge threshold　它是通过使用 vl_sift_set_edge_thresh（）配置 SIFT 过滤器对象来设置的。

SIFT descriptor

局部区域（关键点）的 sift 描述符是图像梯度的 3-D 空间直方图。SIFT 描述符可以通过调用 vl_sift_calc_keypoint_descriptor 或 vl_sift_calc_raw_descriptor 来计算。以下参数影响描述符计算：
magnification factor 放大系数 它由 vl_sift_set_magnif 设置。
Gaussian window size.高斯窗口大小 由 vl_sift_set_window_size 设置
消除低对比度的描述符：用 vl_sift_set_ norm_thresh()

Using the SIFT filter object

◉ 使用 vl_sift_new（）初始化 SIFT 过滤器对象。滤波器可以重用于相同大小的多个图像（例如，对于整个视频序列）。

vl_sift_new 的调用：

VlSiftFilt* vl_sift_new (int width,int height,int noctaves,int nlevels,int o_min)

width 图像的宽度

height 图像的高度

noctaves octave 数

nlevels octave 的层次数

o_min 第一个 octave 的指数

该功能为指定的图像和比例尺空间几何分配并返回一个新的 SIFT 过滤器

◉对于尺度空间中的 octave：

- 使用 vl_sift_process_first_octave（）或 vl_sift_process_next_octave（）计算 DOG 标度空间的下一个 octave。（如果返回 VL_ERR_EOF，则停止处理）。

  vl_sift_process_first_octave（）：

  int vl_sift_process_first_octave (VlSiftFilt * f, vl_sift_pix const * im)

  f sift 过滤器

  im 图像数据

  该函数通过计算低八度的高斯尺度空间开始处理新图像。它还清空内部关键点缓冲区。错误代码。

  如果没有更多的 octave 要处理，该函数返回 VL_ERR_EOF（文件结尾或序列发生错误）。

- 使用 vl_sift_detect（）运行 SIFT 检测器以获取关键点。

  void vl_sift_detect (VlSiftFilt * f)

  功能检测当前倍频程中的关键点，填充内部关键点缓冲区。

  关键点可以通过 vl_sift_get_keypoints（）检索。

  调用语句 VlSiftKeypoint const * vl_sift_get_keypoints (VlSiftFilt const *f)返回一个指向关键点列表的指针。

- 对于每个关键点

  1.使用 vl_sift_calc_keypoint_orientations（）获取关键点方向。

  调用：int vl_sift_calc_keypoint_orientations (VlSiftFilt * f,double angles[4],VlSiftKeypoint const * k)

  f sift 过滤器

  angles 方向输出

  keypoint 关键点

  该函数计算关键点 k 的方向。该函数返回找到的方向数（最多四个）。方向被写入的是向量的角度。

  2. 对于每个方向

  使用 vl_sift_calc_keypoint_descriptor（）获取关键点描述符。

  调用 void vl_sift_calc_keypoint_descriptor (VlSiftFilt * f,vl_sift_pix * descr,VlSiftKeypoint const * k,double angle0)

  descr SIFT 描述符（输出）

k 关键点

angle0 关键点方向

该函数计算取向角度为 0 的关键点 k 的 SIFT 描述符。该函数填充必须足够大以容纳描述符的缓冲区。

- 通过 vl_sift_delete（）删除 SIFT 过滤器。

void vl_sift_detect (VlSiftFilt * f)

功能检测当前倍频程中的关键点，填充内部关键点缓冲区。关键点可以通过 vl_sift_get_keypoints()

总的来说，就是，提取关键点，对关键点附加详细的信息（局部特征）也就是所谓的描述器，关键点匹配，消除错配点。

感觉还是因为编程基础不够的原因，具体要怎么做要能会用，得边学边试着编。