

# 一种快速多尺度特征点匹配算法

邵巍<sup>1),2)</sup> 朱圣英<sup>1)</sup> 陈灵芝<sup>2)</sup>

<sup>1)</sup>(哈尔滨工业大学深空探测基础研究中心,哈尔滨 150080) <sup>2)</sup>(青岛科技大学自动化与电子工程学院,青岛 266042)

**摘要** 为了快速稳定地进行特征点的跟踪,提出了一种快速多尺度特征点的提取算法。该算法首先利用快速局部窗口极值搜索算法提取出不同尺度空间特征点的局部极值,减少了局部极值搜索的冗余性,然后再利用最近邻算法对特征点进行匹配。实验结果表明,该算法的计算速度快于 SIFT 算法和 MOPS 算法,稳定性强于传统的 Harris 算法,可以用于实时图像配准及目标跟踪。

**关键词** 特征点提取 特征点匹配 多尺度变换 MOPS 尺度不变特征变换 Harris 角点

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2009)12-2572-05

## A Fast Multi-scale Feature Matching Algorithm

SHAO Wei<sup>1),2)</sup>, ZHU Sheng-ying<sup>1)</sup>, CHEN Ling-zhi<sup>2)</sup>

<sup>1)</sup>(Deep Space Exploration Research Center, Harbin Institute of Technology, Harbin 150008)

<sup>2)</sup>(College of Automation and Electronic Engineering, Qingdao University of Science & Technology, Qingdao 266042)

**Abstract** This paper presents a Multi-scale feature extraction algorithm, which computes the maximum of the features in moving windows using fast algorithm reduces the search redundancy and obtains the matching features using the nearest neighbor matching algorithm. The experimental results show that this algorithm is faster than the SIFT and MOPS, and has more stability than Harris algorithm. The algorithm can be used in image registration and target tracking.

**Keywords** feature extraction, feature matching, multi-scale transform, multi-scale oriented patches (MOPS), scale-invariant feature transform(SIFT), Harris corner

## 1 引言

基于尺度空间的特征点提取算法是利用图像的特征不变描述符对不同尺度空间的特征点进行描述。Schmid 和 Mohr 最早利用高斯微分算子对传统的 Harris 算法<sup>[1]</sup>进行了改进,形成了旋转不变的特征描述<sup>[2]</sup>。Lowe 又对此算法进行了改进,在不同尺度空间进行特征点的提取,形成了 SIFT (scale-invariant feature transform) 算法<sup>[3]</sup>。SIFT 算法对图像的旋转、缩放及光照影响都具有一定的鲁棒性。Brown 等人提出了 MOPS (multi-scale oriented

patches) 算法<sup>[4]</sup>, 用于进行不同尺度上 Harris 特征点的提取, 并利用窗口搜索算法对特征点的局部极值进行提取。由于 SIFT 算法和 MOPS 算法对特征点的提取一般都采用穷举算法进行搜索, 计算量较大, 因而在图像实时处理中的应用受到限制。本文利用快速局部窗口搜索算法进行多尺度特征点局部极值的提取, 从而提高了特征点提取的速度。

## 2 多尺度 Harris 特征点提取

进行多尺度 Harris 特征点提取时, 要对灰度图像  $I(x, y)$  先利用高斯平滑函数进行卷积计算, 同时通过

基金项目: 国家自然科学基金项目(60874094)

收稿日期: 2008-02-29; 改回日期: 2008-12-12

第一作者简介: 邵巍(1980 ~ ), 男, 2009 年获哈尔滨工业大学博士学位, 现为青岛科技大学教师。主要研究工作是基于图像信息的深空探测器自主导航。E-mail: greatshao@126.com

间隔采样形成图像金字塔。金字塔的最底层可表示为  $P_0(x, y) = I(x, y)$ , 更高层的金字塔可以表示为

$$\tilde{P}_l(x, y) = P_l(x, y) * g_{\sigma_p}(x, y) \quad (1)$$

$$P_{l+1}(x, y) = \tilde{P}_l(sx, sy) \quad (2)$$

其中,  $l$  表示金字塔的层数,  $g_{\sigma}(x, y)$  表示标准差为  $\sigma$  的高斯平滑窗口,  $s$  为采样间隔, 一般取 2。第  $l$  层坐标  $(x, y)$  处的 Harris 特征点检测矩阵可以表示为

$$H_l(x, y) = \nabla_{\sigma_d} P_l(x, y) (\nabla_{\sigma_d} P_l(x, y))^T * g_{\sigma_i}(x, y) \quad (3)$$

其中,  $\nabla_{\sigma}$  表示在尺度  $\sigma$  上的梯度, 即

$$\nabla_{\sigma} f(x, y) \triangleq \nabla f(x, y) * g_{\sigma}(x, y) \quad (4)$$

文献[4]将积分尺度  $\sigma_i$  和微分尺度  $\sigma_d$  的值分别取为 1.5 和 1.0, 并利用矩阵  $H$  特征值  $(\lambda_1, \lambda_2)$  的调和平均检测函数  $f_{HM}$  来检测特征点, 即

$$f_{HM}(x, y) = \frac{\det H_l(x, y)}{\text{tr} H_l(x, y)} = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} \quad (5)$$

当  $f_{HM}$  大于某个阈值(一般取为 10)时, 则该点即为特征点。

### 3 局部区域快速特征点选取算法

由于特征点的匹配速度与特征点的个数直接相关, 同时由于底层金字塔提取的 Harris 特征点主要集中在物体的边缘和角点处, 分布很不均匀, 若采取只将整幅图像中  $f_{HM}$  值最大的多个特征点提取出来, 以减少特征点个数的方法, 则可能使得这些特征点仅局限在某个局部区域, 而不利于重合区域较小的图像间的匹配, 因此, 本文在不同尺度金字塔图像的一定半径范围内, 通过选取  $f_{HM}$  的局部极值, 并通过控制采样窗口的大小来控制特征点的提取个数。其中半径  $r$  可用以下公式选取:

$$r = -7 + 2 \times \text{lb}(\max(W_l, H_l)/2) \quad (6)$$

式中,  $W_l, H_l$  分别表示金字塔第  $l$  层图像的宽度和高度。

#### 3.1 特征点局部极值的快速算法

在传统的 Harris 算法中, 若某个像素对应的检测函数  $f_{HM}$  值在周围  $3 \times 3$  的邻域中是最大值, 则该像素即为特征点。这里将邻域范围扩大到  $w \times w$  窗口范围内来选取特征点, 这就将特征点的选取转化为在  $W_l \times H_l$  大小的图像中, 对大小为  $w \times w$  的窗口进行局部极值搜索的问题, 若中心元素为极值, 则判定该点为特征点。一般采用穷举法来进行搜索, 根据条件概

率计算, 平均要进行  $(W_l - w + 1) \times (H_l - w + 1) \times k$  ( $k = 1 + \frac{1}{2} + \dots + \frac{1}{w^2 - 1}$ ) 次比较, 计算量较大。本文利用形态学滤波中的极大值滤波的思想<sup>[5-7]</sup>, 采用改进的 HGW (Herik-Gil-Werman) 算法, 并将极大值滤波的求取过程转化为 2 维窗口内局部极值的求取, 以加快运算速度。具体步骤如下:

(1) 设图像的像素矩阵为  $X$ , 可先将图像扩充到  $\tilde{X} = [X \quad A]$ , 其中  $A$  为  $H \times (w - \text{mod}(W, w))$  大小的全零矩阵(mod 表示取余运算), 这样就可以将  $\tilde{X}$  均分为宽度为  $w$  的多个子矩阵。

(2) 将 2 维窗口内极值的计算分解为以下水平、垂直 2 次 1 维窗口极值的计算:

$$\max_{-w \leq l \leq w} \{f(x+k, y+l)\} = \max_{-w \leq l \leq w} \left\{ \max_{-w \leq k \leq w} \{f(x+k, y+l)\} \right\} \quad (7)$$

记  $\tilde{X}$  某一行的元素为  $x_0, \dots, x_{n-1}$ , 考虑 1 维数组的局部最大值, 即计算  $y_i = \max_{0 \leq j < p} x_{i+j}$  ( $i = 0, \dots, n-w$ ), 则可用序列  $x_{w-1}, x_{2w-1}, x_{3w-1}, \dots$  将该行元素分为多个小数组, 数组中间元素的下标记为  $t$ 。对于包括  $x_t$  的每个小数组, 定义如下序列元素  $R_k$  和  $Q_k$  ( $k = 0, \dots, w-1$ ):

$$R_k = R_k(t) = \max(x_t, x_{t-1}, \dots, x_{t-k}) = \max(R_{k-1}, x_{t-k})$$

$$Q_k = Q_k(t) = \max(x_t, x_{t+1}, \dots, x_{t+k}) = \max(Q_{k-1}, x_{t+k}) \quad (8)$$

该步骤对每个小数组需要进行  $2(w-1)$  次比较。本文采用如下的改进算法来提高  $R_k$  和  $Q_k$  的计算速度:

记

$$q = \left\lfloor \frac{w+1}{2} \right\rfloor = \frac{w}{2} + \frac{w \bmod 2}{2} \quad (9)$$

先计算  $Q_k$  ( $k = 0, \dots, q-1$ ) 和  $R_k(t+1)$  ( $k = q, \dots, w$ ), 需要进行  $q-1 + w - q = w-1$  次比较; 然后比较  $Q_{q-1}$  和  $R_q(t+1)$ , 由于  $Q_k$  是非递减数列,  $R_q(t+1)$  是非递增数列, 若  $R_q \geq Q_{q-1}$ , 则  $R_{q-1} = R_{q-2} = \dots = R_1 = R_q$ ; 最后计算  $Q_q, \dots, Q_{w-1}$ , 需要比较  $w-q$  次。同理, 若  $R_q \leq Q_{q-1}$ , 则无需计算  $Q_q, \dots, Q_{w-1}$ , 只要比较  $q-1$  次, 即可计算  $R_1, \dots, R_{q-1}$ 。因此计算  $R_k$  和  $Q_k$  平均共需要进行  $(w-1) + 1 + \max(w-q, q-1) = 1.5w - (w \bmod 2)/2$  次比较。

(3) 合并该行元素的  $R_k$  和  $Q_k$ , 即可得到以下窗口最大值的函数:

$$t_k = \max(x_{j-k}, \dots, x_j, \dots, x_{j+w-k-1}) = \max(R_k, Q_{w-k-1}) \quad (10)$$

考虑到  $R_{w-2} \geq R_{w-1} \geq \dots \geq R_1$  及  $Q_{w-2} \geq Q_{w-1} \geq \dots \geq Q_1$ , 若  $R_i \geq Q_{w-i-1}$ , 则对所有  $k > i$  的情况有  $R_k \geq R_i \geq S_{w-i-1} \geq S_{w-k-1}$ , 因此不需比较  $k > i$  的情况; 同理, 当  $R_i \leq S_{w-i-1}$  时, 则不需比较  $k < i$  的情况。根据二叉树搜索的原理, 该步骤对每个元素进行搜索的次数为  $\lceil \ln(w-1) \rceil / w$ 。这样就可将  $\tilde{Y}$  的每个元素用对应 1 维窗口的最大值  $t_k$  代替, 即得到矩阵  $Y$ 。综合步骤运算平均到每个元素的比较次数为

$$1.5 + \frac{\lceil \ln(w-1) \rceil}{w} - \frac{(w \bmod 2)}{2w} \quad (11)$$

若对矩阵  $Y$  进行列元素的 1 维局部极值求取, 则重复步骤(1)、步骤(2)即可得到 2 维窗口局部极值的分布矩阵  $\tilde{Y}$ , 即  $\tilde{Y}$  表示在某元素半径范围内的最大值的分布情况。原始图像及对底层金字塔进行局部最大值求取的结果如图 1 所示。

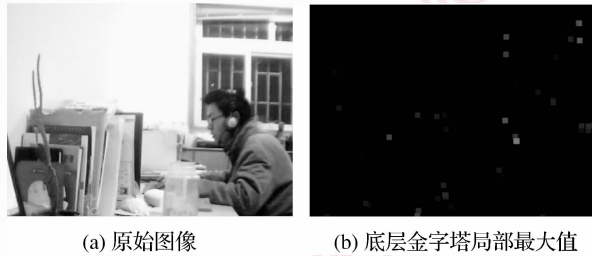


图 1 原始图像及局部最大值分布图

Fig. 1 Original image and local maximum distributing

将  $\tilde{Y}$  中的扩充矩阵去掉, 并将其与矩阵  $X$  的对应元素相减, 若某元素的对应值为 0, 则表示该值为  $w \times w$  窗口的局部最大值, 即在其周围  $w \times w$  窗口的元素中没有比它更大的元素。由于分别进行了一次水平方向和垂直方向的局部极值的求取, 所以总共的比较次数为

$$3 + \frac{2\lceil \ln(w-1) \rceil}{w} - \frac{(w \bmod 2)}{w} \quad (12)$$

常规算法、HGW 算法及改进的 HGW 算法每元素的比较次数如图 2 所示。由图 2 可以看出, 随着窗口的增大, 改进的 HGW 算法的单个元素的比较次数趋向于 3, 比较次数少于另外两种算法。窗口越大, 该算法的优势越明显。

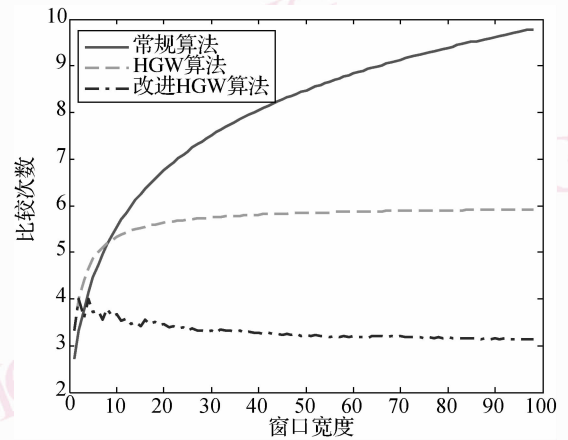


图 2 几种算法平均计算次数的比较

Fig. 2 Average number of comparisons of different algorithms

### 3.2 特征点方向确定

由于传统的 Harris 特征点检测方法提取的特征点不带有方向信息, 因此对于图像的旋转容易产生误匹配。为此本文采用 SIFT 算法的思想, 利用特征点邻域像素的梯度方向分布特征为每个特征点指定方向。对  $L$  层上的特征点邻域内的坐标  $(x, y)$  处的方向计算如下:

$$\theta(x, y) = \text{atan} \frac{(L(x, y+1) - L(x, y-1))}{(L(x+1, y) - L(x-1, y))} \quad (13)$$

若用直方图统计邻域像素加权梯度方向的值, 则其峰值即为该特征点的方向。

### 3.3 特征点描述符及特征点匹配

结合 MOPS 算法, 用特征点周围的  $8 \times 8$  大小的像素的标准化灰度值来对特征点进行描述。再进行 Haar 小波变换, 即形成 64 维的特征描述向量。

特征点描述符形成后, 再采用次近邻算法 (2-NN)<sup>[8]</sup> 对由不同图像提取的特征点进行匹配, 具体步骤可参考 MOPS 算法中的匹配过程。

## 4 实验结果及不同算法性能比较

为比较不同算法的特征点提取及匹配效果, 对本文算法、SIFT 算法及 Harris 算法进行了对比实验。运行环境为 PD820 (2.8 GHz), 编程环境为 Vc.net 结合 Opencv, 采集图像的分辨率为  $640 \times 480$ 。图中圆的半径大小代表特征点所在的尺度, 圆圈内部短线表示特征点的方向。从特征点提取的结果 (图 3) 看, Harris 算法提取的特征点更集中在边缘和角点处, 而用本文提出的快速 MOPS 算法提取的特

征点分布则比 SIFT 和 Harris 的特征点分布更为均匀。由于快速 MOPS 算法是在运算速度上对

MOPS 算法进行了改进,其特征点的结果是一致的,在此不再分开讨论。

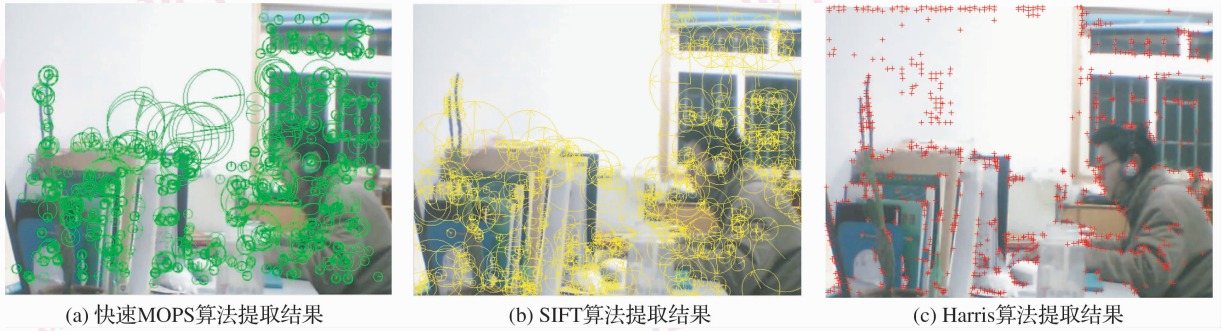


图 3 快速 MOPS, SIFT, Harris 提取特征点分布图  
Fig. 3 Features extracted by fast MOPS, SIFT and Harris

通过表 1 的比较可以看出, Harris 算法运行速度最快, 但该算法对于图像的旋转、缩放的稳定性不高, 误匹配率较高, 不适用于旋转和缩放变化较大的图像间的匹配。SIFT 算法用 128 维描述符对特征点进行描述, 稳定性更好, 但运算最慢, 不利于实时计算。而 MOPS 算法是两种算法的折中, 利用快速 MOPS 算法可以进一步提高 MOPS 算法的运行速度, 基本上可以满足实时处理的要求。

是 Harris 角点算法, 对于大角度的旋转情况, 其准确匹配的特征点个数要少于 SIFT 算法。因此, 可以根据相机、物体运动的不同性质和特征跟踪对稳定性和实时性的要求来选择不同的特征匹配算法。另外, 可以根据不同的需要, 通过变化式(6)改变半径大小来控制特征点提取的个数及分散程度, 以进一步提高运算速度。

表 1 几种算法的比较  
Tab. 1 Compare of these algorithms

算法	平均运行时间(s)	正确匹配的特征点个数	错误匹配的特征点个数
SIFT	6.62	233	17
MOPS	1.71	196	10
快速 MOPS	1.36	196	10
Harris	0.92	211	116

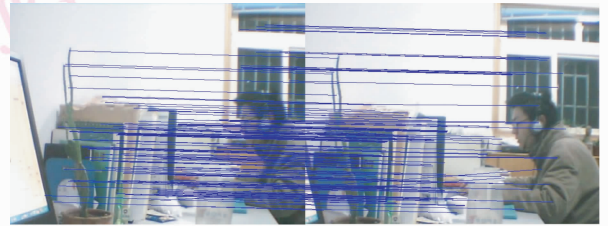


图 4 利用快速 MOPS 进行匹配的结果  
Fig. 4 Features matching using fast MOPS

图 4 是利用快速 MOPS 算法对两幅图像进行特征点匹配的结果。从图中可以看出, 虽然两幅图像存在缩放与旋转变换, 但利用快速 MOPS 算法依然可以使得正确匹配特征点的比率达到 90% 左右。通过多次实验测试, 对于旋转缩放情况不是很大的情况, 利用快速 MOPS 算法是稳定可靠的, 大多数情况都可以使得正确匹配特征点的比率大于 50%, 这样就可以通过 RANSAC(random sample consensus)<sup>[9]</sup> 等算法去除误匹配的特征点。同时, 采用该算法的特征点分布比较均匀, 对于特征跟踪来说, 减小了由于运动物体某一部分受遮挡而不能连续跟踪所造成的影响。由于对每层金字塔上的特征点提取, 采用的

### 5 结 论

本文提出了一种利用局部极值的快速求法来提高特征点提取速度的算法, 同时通过比较实验分析了几种特征点提取算法的优缺点, 并验证了该算法的有效性。今后的研究主要集中在以下几个方面:

(1) 该算法运行时所占的内存要大于传统算法, 因此进一步提高算法的快速性、减少内存消耗是下一步研究的重点。同时也需研究窗口半径对匹配效果的影响, 并通过选取合适的窗口半径, 以使其适应于多种图像间平移、旋转及缩放变换。

(2) 将该算法灵活运用到 SIFT, SURF (speeded up robust features) 等多种特征点局部极值的求取过

程,以拓展该算法的应用范围。

(3)考虑将随机采样策略应用到局部极值的求取问题中,虽然不能保证全部局部极值的求取,但可进一步提高该算法的快速性。

### 参考文献 (References)

- 1 Harris C G, Stephens M J. A combined corner and edge detector [A]. In: Proceedings Fourth Alvey Vision Conference [C], Manchester, UK, 1988:147-151.
- 2 Schmid C, Mohr R. Local grayvalue invariants for image retrieval [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, **19**(5):530-535.
- 3 Lowe D. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, **60**(2):91-110.
- 4 Brown M, Szeliski R, Winder S. Multi-image matching using multi-scale oriented patches [A]. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern[C], San Diego, CA, USA, 2005:510-517.
- 5 Van Herk M. A fast algorithm for local minimum and maximum filters on rectangular and octagonal kernels [J]. Pattern Recognition Letters, 1992, **13**(7):517-521.
- 6 Joseph G, Michael W. Efficient dilation, erosion, opening, and closing algorithms[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993, **15**(5):504-507.
- 7 Gil J Y, Kimmel R. Efficient dilation, erosion, opening, and closing algorithms[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, **24**(12):1606-1617.
- 8 Arya S, Mount D M, Netanyahu N S, *et al.* An optimal algorithm for approximate nearest neighbor searching[J]. Journal of ACM, 1998, **45**(6):891-923.
- 9 Fischler M A, Bolles R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography [J]. Communications of the ACM, 1981, **24**(6):381-395.