面向移动设备的快速特征点匹配方法研究*

罗亚威[1]，魏本昌[1]，管　涛[1]，严　卡[1]，张　岩[2]
(1. 华中科技大学 数字媒体处理与检索实验室，武汉 430074；2. 防灾科技学院，河北 三河 065200)

摘　要：针对增强现实在移动设备应用中特征点匹配速度缓慢的情况，提出了一种特征点匹配优化算法。该算法基于移动设备自带的重力感应系统，用重力方向代替原有的描述符方向，以此来降低传统 ORB(oriented FAST and rotated BRIEF)算法中特征点方向角计算的时间复杂度。在实际移动设备上的测试结果表明，改进后的算法同时提高了匹配速度和匹配准确率，具有较好的效果。

关键词：特征点匹配；重力感应；ORB；增强现实；移动设备
中图分类号：TP391　　**文献标志码**：A　　**文章编号**：1001-3695(2013)02-0591-04
doi:10.3969/j.issn.1001-3695.2013.02.076

Research on fast feature match method for mobile devices

LUO Ya-wei[1], WEI Ben-chang[1], GUAN Tao[1], YAN Ka[1], ZHANG Yan[2]
(1. *Laboratory of Digital Media Processing & Retrieval*, *Huazhong University of Science & Technology*, *Wuhan* 430074, *China*; 2. *Institute of Disaster Prevention*, *Sanhe Hebei* 065200, *China*)

Abstract: Aiming at the problem that feature match on mobile devices was slow, this paper presented a new optimized algorithm which was based on gravity sensor. The algorithm replaced the original descriptor direction with the direction of gravity, so as to improve ORB(oriented FAST and rotated BRIEF). The experiments on mobile devices show that the method is faster and more precise on feature matching.

Key words: feature matching; gravity sensing; ORB; augmented reality(AR); mobile devices

0　引言

增强现实(augmented reality, AR)[1]是计算机视觉领域最近十几年来兴起的一门新的技术，目前在计算机图像处理方面变得越来越热门。随着移动设备计算能力的日益提高，在手机上实现增强现实的应用已成为可能。然而，针对移动设备自身的计算和存储能力远远不及计算机等缺点，要将增强现实技术完全应用于移动设备，当前面对的首要问题就是改进传统意义上的增强现实技术中的两大主要算法：特征点匹配算法和计算描述符算法。这两个算法在增强现实技术中缺一不可，但时间复杂度较高。因此改进此类算法的时间复杂度是本文研究的核心。

传统的特征点匹配方法包括 SIFT(scale-invariant feature transform)[2]、SURF(speeded up robust features)[3]、BRIEF(binary robust independent elementary features)[4]、ORB(oriented FAST and rotated BRIEF)[5]等。SIFT 描述符虽然可区分度高，但是每个描述符占用的空间比较大，需要消耗较大的内存，这对于移动设备上的 AR 应用来讲是非常不利的。SURF 描述符的计算速度虽然比较快，然而其描述符特征向量中每一维都是浮点数，所以它与 SIFT 一样，内存消耗比较大。与 SIFT、SURF 相比，BRIEF 算法有很大的优势，它不仅在生成描述符和匹配阶段速度更胜一筹，而且能够比前两者得到更高的识别表现，同时，BRIEF 描述符占用的内存较少。但该算法的不足之处在于它对方向和噪声比较敏感，不具备旋转不变性，因此对于旋转比较频繁的移动设备而言准确性不高。

BRIEF 的改进算法 ORB 为每个描述符增加了一个方向角，有效地弥补了 BRIEF 算法对方向敏感的不足；然而，添加方向角的步骤又引入了大量的计算，增加了算法的时间复杂度。因此，寻找一种具备 ORB 算法的优点，同时时间复杂度较低的特征点匹配算法非常重要。

本文利用现代化移动设备上比较普及的重力感应系统，提出了一种以重力信息作为参量的快速特征点匹配算法思想，并以此改进了传统 ORB 算法。改进后的算法能同时提高特征点匹配的速度以及准确率。

1　传统的 ORB 算法

1.1　ORB 算法基本原理

ORB(快速定位和旋转的 BRIEF)是一种非常快速的二进制描述符，它基于 BRIEF 健壮并且独立的二进制基本特征)。ORB 采用 FAST[6]提取特征点，然后以特征点为中心选取 48 × 48 的邻域(长宽都为 48 的像素区域)。在该像素区域随机选取 n_d 个坐标点对，然后比较每个点对的大小，比较方法如下：

$$r(p;x,y):=\begin{cases}1 & \text{if } p(x)<p(y)\\ 0 & \text{otherwise}\end{cases} \tag{1}$$

收稿日期：2012-06-20；**修回日期**：2012-07-28　　**基金项目**：国家自然科学基金青年基金资助项目(60903095)

作者简介：罗亚威(1991-)，男，湖北武汉人，主要研究方向为计算机视觉、增强现实等(ywluo123@gmail.com)；魏本昌(1975-)，男，湖北荆州人，博士研究生，主要研究方向为计算机视觉、增强现实；管涛(1978-)，男，山东青岛人，副教授，硕导，博士，主要研究方向为虚拟现实与增强现实；严卡(1988-)，男，湖北天门人，硕士研究生，主要研究方向为计算机视觉、增强现实等；张岩(1982-)，男，河北三河人，博士研究生，主要研究方向为虚拟现实与增强现实.

式(1)中 $p(x)$ 是对像素区域经过平滑之后位于 $x=(u,v)$ 的像素的灰度强度值。用 n_d 个点对进行比较,然后生成一个特征点的描述符。特征点描述符的计算公式如下:

$$f_{n_d}(p):=\sum_{1\leqslant i\leqslant n_d}2^{i-1}r(p;x_i,y_i) \tag{2}$$

$$S=\begin{pmatrix}x_1,\cdots,x_n\\ y_1,\cdots,y_n\end{pmatrix} \tag{3}$$

$$S_\theta=R_\theta S \tag{4}$$

$$g_n(p,\theta):=f_{n_d}(p)\mid(x_i,y_i)\in S_\theta \tag{5}$$

其中:n_d 为描述符的维度,S 为特征点 (x_i,y_i) 邻域的 n 个随机的点对矩阵,R_θ 为旋转矩阵。

与传统的 FAST 提取特征点不同的是,ORB 在提取到特征点之后给每个特征点都添加了一个方向角,然后基于 BRIEF 生成描述符,并在生成描述符之前针对每个特征点的方向角进行了旋转。ORB 在计算特征点角度时用到了强度质心[8]这个概念。强度质心假设一个角点的灰度由它的中心补偿,并且定义一个向量,O 为角点的中心,P 为角点的质心,如图 1 所示。

向量 OP 与 X 轴的夹角 θ 将作为该角点的方向角。夹角 θ 的计算公式如下:

$$m_{pq}=\sum_{x,y}x^py^qI(x,y) \tag{6}$$

$$P=\left(\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}}\right) \tag{7}$$

$$\theta=\arctan\left(\frac{m_{01}}{m_{10}}\right) \tag{8}$$

式(6)中 $I(x,y)$ 为图像矩阵中点 (x,y) 的像素值。P 为强度质心坐标,θ 即为所求角。由于特征点邻域的随机点对的相关性比较大,用此求角方法计算出的描述符可区分度较低。为了降低特征点邻域随机点对的相关性,ORB 从 Pascal 2006[2]点集中训练一组点对作为参照。具体的方法如下:选取特征点邻域长宽均为 31 的像素区域,每一个测试点为一个 5×5 的像素区域,可以得到 $N=(31-5)^2$ 个 5×5 的像素区域,这样一共可以得到 $\binom{N}{2}$ 个二值测试,去掉重复值,最后可以得到 205 590 个测试值。具体的计算过程如下:

a)在特征点 31×31 邻域中对每一个 5×5 的子窗口进行二值测试。

b)对每一个测试结果,根据其与 0.5 的距离进行排序形成一个向量 T。

c)对测试值进行贪心法搜索。首先将步骤 b)中的第一个测试结果放入向量 R 中,同时将其从 T 中去掉,再从 T 中取下一个测试结果,然后将其与 R 中的所有元素进行比较,如果它的绝对相关性大于某一个阈值,就将其从 T 中移除,否则,将其添加到 R。重复前面的两个步骤直到 R 中存在 256 个元素,如果 R 中依然不足 256 个元素,则将阈值提高并继续贪心搜索。

1.2 ORB 算法的不足

由 1.1 节中对传统 ORB 算法的介绍可以看出,计算特征点角度的过程需要经过复杂计算。由式(6)~(8)可知,传统 ORB 算法针对每个特征点都需要计算一次正弦和余弦,并且计算特征点描述符中需要进行五次乘法。这些计算在特征点较多的情况下会花费大量的时间,特别是在手机等计算能力较弱的移动设备上,采用该算法的增强现实应用的实时性较差。针对传统 ORB 算法的这些不足,本文采用了重力信息作为参考用来辅助计算特征点的方向角,改进了 ORB 算法中对方向角复杂的计算方式,提高了算法的运行效率。

2 重力感应加速特征点匹配

2.1 利用重力计算手机旋转角

为了将重力方向加入到图像之中,需要知道拍摄图像时手机的姿态。因此,计算手机旋转角是本算法的关键步骤之一。本文将利用重力感应系统来完成计算手机旋转角的任务。

首先,需要将世界坐标系转换为相机坐标系。假定 $(X_c,Y_c,Z_c)^{\mathrm{T}}$ 为相机坐标系,$(X_w,Y_w,Z_w)^{\mathrm{T}}$ 为世界坐标系,则两者之间的转换关系(由齐次坐标表示)为

$$\begin{pmatrix}X_c\\Y_c\\Z_c\\1\end{pmatrix}=\begin{pmatrix}r_{11}&r_{12}&r_{13}&t_1\\r_{21}&r_{22}&r_{23}&t_2\\r_{31}&r_{32}&r_{33}&t_3\\0&0&0&1\end{pmatrix}\begin{pmatrix}X_w\\Y_w\\Z_w\\1\end{pmatrix} \tag{9}$$

其中 $R=\begin{pmatrix}r_{11}&r_{12}&r_{13}\\r_{21}&r_{22}&r_{23}\\r_{31}&r_{32}&r_{33}\end{pmatrix}$ 为旋转、比例、错切变换矩阵,$T=\begin{pmatrix}t_1\\t_2\\t_3\end{pmatrix}$ 为投影矩阵。

手机的旋转一般是指手机相对于自身相机坐标系的旋转。本文将以图 2 的相机坐标系作为参考坐标系。

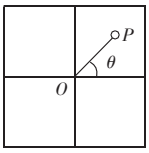


图 1 强度质心模型

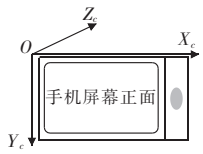


图 2 手机相机坐标系

a)手机绕 X_c 轴旋转时,相机图像平面上的点的坐标值在旋转前后的对应关系为

$$(x',y',z',1)=(x,y,z,1)R_x=(x,y,z,1)\begin{pmatrix}1&0&0&0\\0&\cos\theta&\sin\theta&0\\0&-\sin\theta&\cos\theta&0\\0&0&0&1\end{pmatrix} \tag{10}$$

b)手机绕 Y_c 轴旋转时,相机图像平面上的点的坐标值在旋转前后的对应关系为

$$(x',y',z',1)=(x,y,z,1)R_y=(x,y,z,1)\begin{pmatrix}\cos\theta&0&-\sin\theta&0\\0&1&0&0\\\sin\theta&0&\cos\theta&0\\0&0&0&1\end{pmatrix} \tag{11}$$

c)手机绕 Z_c 轴旋转时,相机图像平面上的点的坐标值在旋转前后的对应关系为

$$(x',y',z',1)=(x,y,z,1)R_z=(x,y,z,1)\begin{pmatrix}\cos\theta&\sin\theta&0&0\\-\sin\theta&\cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix} \tag{12}$$

d)设相机沿 Y_c 轴旋转 θ_1,沿 X_c 轴旋转 θ_2,沿 Z_c 轴旋转

θ_3,则经复合旋转后的坐标值为

$$(x',y',z',1)=(x,y,z,1)R_y(\theta=\theta_1)R_x(\theta=\theta_2)R_z(\theta=\theta_3) \quad (13)$$

以上公式中,(x,y,z) 为旋转之前的点,(x',y',z') 为旋转之后的点。

在已知复合旋转角时,可以通过上述映射关系求出旋转前后每个点的坐标。然而重力传感器的返回值为重力沿手机的三个轴向的分量,并非旋转角。因此,还需要进一步将重力分量转换为旋转角。当手机发生旋转时,重力在相应坐标轴上的分量也会发生改变,具体分为以下三种情况:

a)手机绕 X_c 轴旋转。当手机绕 X_c 轴旋转时,旋转的角度反映在重力分量 Z_g 值的变化。

b)手机绕 Y_c 轴旋转。当手机绕 Y_c 旋转的时候,重力分量始终没有变化,说明手机绕 Y_c 旋转时,旋转角度通过重力分量无法测量,可根据手机内的方位传感器获取。

c)手机绕 Z_c 轴旋转。当手机绕 Z_c 轴旋转时,旋转的角度反映在重力分量 Y_g 值的变化。为便于计算,将重力沿各个轴向的分量 i、j、k 的大小范围限定为 $-9\sim9$,即分量的大小每变化 1 就相当于手机绕对应的轴旋转了。以图 3 为例,(i,j,k) 即为重力感应装置的返回值。

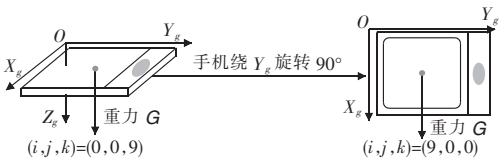


图 3　手机绕 Y_g 旋转 90°

可见利用重力传感器的返回值可以很快地计算出当前手机的复合旋转角,并进一步得到手机旋转前后每点坐标的变换。

2.2　重力感应对齐特征点主方向

特征点主方向是特征点邻域像素的梯度累加和。主方向的计算过程往往要经过很多复杂的运算,因此考虑将重力方向作为描述符的主方向代替传统的计算方法,以降低算法复杂度。

如图 4 所示,中间窗格的四个角点具有完全相同或者近似完全相同的特征,它们沿四个不同的方位分布。在左方手机的左边,显示的是四个不同的角点在其归一化区域中生成的描述符,很明显,这四个不同的角点产生的描述符完全相同。而在图 4 右边手机的旁边,显示的是以重力作为特征点描述符的主方向之后生成的描述符,肉眼就可以很清楚地分辨出它们分别对应于窗子的哪个角。可见,引入重力信息后可增加描述符的可区分性。

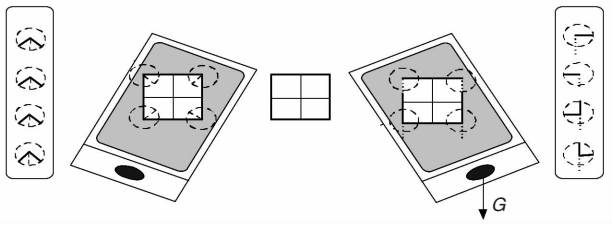


图 4　传统方法描述符(左)与重力对齐描述符(右)

用手机对场景进行拍摄时,由于人为因素,手机屏幕平面不可能完全平行于被拍照的物体,会产生不同程度的倾斜。该倾斜的程度可以由重力感应器获取,然后转换成沿不同方向的倾斜角。转换关系在 2.1 节中已经介绍。重力对齐描述符的具体思想如下:

a)由重力感应器记下拍摄时的手机的重力向量 G_1,即拍照时的 X_g、Y_g、Z_g 的值;

b)旋转手机后再对该场景进行拍摄,重力感应器再记下一个重力向量 G_2;

c)将 G_2 与 G_1 相减,得到一个新的重力向量 $G=(X',Y',Z')$,根据 X'、Y'、Z' 换算出手机相对于第一次拍摄时的旋转角度;

d)最后将步骤 b)中图像特征点描述符按照 c)中的角度进行旋转,与步骤 a)中图像特征点描述符对齐。

这四个步骤如图 5(a)~(d)所示。

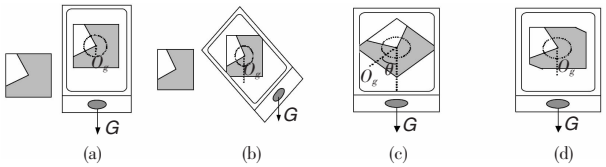


图 5　重力对齐主方向

2.3　基于重力感应的改进 ORB 算法

第 1.1 节中介绍的 ORB 算法的工作原理中已指出,ORB 在检测特征点时,对每个检测到的特征点都进行了方向角的计算,这样每一个特征点就多了一个角度属性。在计算特征点描述符时利用此角度对从 Pascal 2006[2] 选取的训练点对进行旋转。传统的 ORB 算法在计算特征点描述符的过程中,需要计算该角度的正弦和余弦值,在梯度的计算中还要执行五次乘法运算。尽管计算一次乘法的时间开销较小,但当次数相当大时,时间开销还是非常大的。

基于重力感应的 ORB 算法对其的改进措施步骤如下:

a)去掉 ORB 在检测特征点时计算方向角的步骤;

b)将重力感应对应的分量值转换为角度后传递给 ORB 计算描述符;

c)根据 b)中得到的角度,将 ORB 在计算描述符中用到的训练点对提前旋转,由于每个特征点的主方向都定为重力方向,故只需旋转一次;

d)将 ORB 中计算描述符的具体运算进行简化,减少其乘法计算的次数。

本文将改进后的算法命名为 GAFD_ORB(gravity aligning feature descriptor ORB)。在采用 GAFD_ORB 算法之后,可省略原 ORB 算法中对方向角的计算过程,对提取特征点的速度有一定的提升。重力感应对 ORB 匹配特征点时间节省贡献最大的是在生成描述符这个阶段。

应用重力对齐特征点描述符思想之后,GAFD_ORB 算法相比于传统的 ORB 计算描述符方法主要有以下不同:a)传统 ORB 算法在检测特征点时需要经过复杂计算来得到特征点角度,而 GAFD_ORB 算法只需要通过重力和方位传感器就可以很轻松地获取角度信息;b)传统 ORB 算法针对每个特征点都需要计算一次正弦和余弦,而 GAFD_ORB 算法中只需计算一次;c)传统 ORB 算法中计算特征点描述符需要进行五次乘法,而 GAFD_ORB 算法不需计算强度质心的夹角,只需要进行一次乘法。

经过这些改进之后,在实验过程中发现,GAFD_ORB 不仅匹配效果要优于传统 ORB,而且匹配速度比传统 ORB 要提高

40% 左右。可见,GAFD_ORB 算法相比于传统的 ORB 算法不仅提高了特征点的匹配率,同时提高了匹配的速度,具有更优的性能。

3 实验结果与分析

为评估 GAFD_ORB 的性能,本文在实际移动设备上对该算法的检测特征点时间、生成描述符时间、正确匹配率这三个方面进行了实验,并与传统 ORB 算法进行了对比。

3.1 实验环境

软件环境为操作系统 Windows XP SP3,开发工具为 Eclipse(SDK 3.7)、Linux(Ubuntu 10.0)、OpencvTrunk、Android 开发(installer_r12)。

硬件环境为 HTC G7 手机(Android 2.1 系统)、Intel® Core™ Quad CPU Q8200 @2.33 GHz,4 GB 内存。

实验所用场景如图 6 所示。

图 6 实验场景图

3.2 检测特征点时间对比

对实验场景提取 10 组不同的特征点数目进行实验比较,实验结果如表 1 所示。

表 1 GAFD_ORB 算法与传统 ORB 算法提点时间对比

特征点	GAFD_ORB 提点时间/ms	ORB 提点时间/ms	特征点	GAFD_ORB 提点时间/ms	ORB 提点时间/ms
75	8.362	10.132	100	10.620	11.780
80	9.857	9.796	105	9.125	10.254
85	9.186	8.423	110	7.996	10.468
90	8.698	8.484	115	9.918	12.695
95	10.040	11.688	120	11.169	13.719

在提取的特征点数目比较少时,GAFD_ORB 与 ORB 的提点时间几乎相同,但当提取的特征点数目增大时,GAFD_ORB 的速度优势就变得十分明显,这也印证了在去掉 ORB 提点机制中计算特征点方向步骤之后带来的速度提升。

3.3 生成描述符时间对比

采用图 6 的场景来对两者在生成描述符的时间进行测试对比。选取 10 组实验数据进行比较,结果如表 2 所示。

表 2 GAFD_ORB 算法与传统 ORB 算法计算描述符时间对比

描述符	GAFD_ORB 计算时间/ms	ORB 计算时间/ms	描述符	GAFD_ORB 计算时间/ms	ORB 计算时间/ms
75	18.341	25.787	100	24.872	31.372
80	18.585	29.785	105	23.468	33.783
85	21.759	30.362	110	23.376	35.248
90	21.759	32.898	115	25.482	36.133
95	20.355	31.799	120	25.177	60.425

GAFD_ORB 算法在计算描述符的时间开销至少要比传统 ORB 算法节省 10 ms 左右的时间,而且随着提取的特征点数目的增加,时间优势更加明显,当提取 120 个特征点时,可以节省 20 ms 以上的时间,这对于移动设备来说速度提升非常可观。

3.4 正确匹配率对比

在垂直平面内(绕 z 轴)分别将手机旋转 0° 和 30° 进行实验,每组选择 10 组数据。实验结果如图 7 所示。

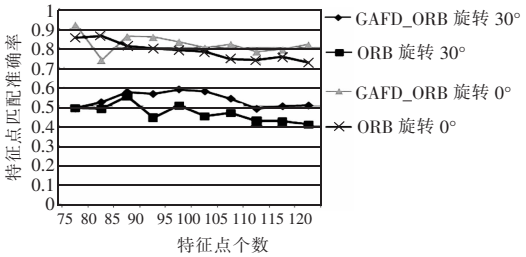


图 7 GAFD_ORB 与传统 ORB 匹配点正确率对比(绕 z 轴旋转)

分别绕 x 轴(前滚)和 y 轴(侧旋)将手机旋转 30° 进行实验,每组选择 10 组数据。实验结果如图 8 所示。

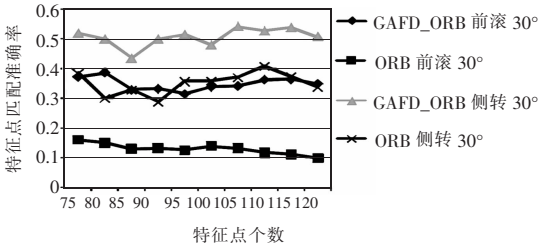
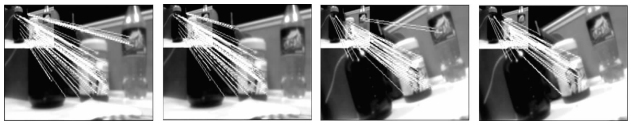


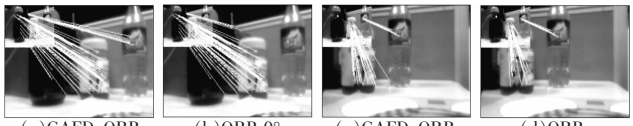
图 8 GAFD_ORB 与传统 ORB 匹配点正确率对比(绕 x 轴和 y 轴旋转)

在手机没有旋转时,GAFD_ORB 算法的正确匹配率与 ORB 算法大致相当,基本维持在 80% 左右。而当手机在平面内分别绕三个坐标轴旋转时,GAFD_ORB 算法的正确匹配率明显要高于 ORB 算法,这主要得益于重力感应器测得的三个精确的重力角度分量。引入重力信息后可增加描述符的可区分性,使得计算特征点描述符的训练点对位置可以很好地得到纠正,从而保证较高的匹配效果。由此可见,引入重力信息作为已知量后,对特征点匹配的准确率和速度都有较大的提升。



(a)GAFD_ORB 0°效果图 (b)ORB 0°效果图 (c)GAFD_ORB 30°效果图 (d)ORB 30°效果图

图 9 实验结果对比(绕 z 轴)



(a)GAFD_ORB 0°效果图 (b)ORB 0° 效果图 (c)GAFD_ORB 30°效果图 (d)ORB 30°效果图

图 10 实验结果对比(绕 x 轴)

(a)GAFD_ORB 0°效果图 (b)ORB 0°效果图 (c)GAFD_ORB 30°效果图 (d)ORB 30°效果图

图 11 实验结果对比(绕 y 轴)

4 结束语

ORB 算法是目前移动设备上比较常用的特征点匹配算法。本文针对 ORB 算法中计算特征点方向角时间复杂度高的不足,提出了基于重力感应系统的快速特征点匹配方法(GAFD_ORB)。通过引入重力信息,将重力方向作为特征点的主方向进行计算,简化了原始算法中方向角的计算过程。实验证明,GAFD_ORB 算法不仅提高了特征点匹配率,而且减小了运算量,具有比传统 ORB 算法更优的性能。(下转第 604 页)

高,对于 Weizmann 数据库可得 100% 的正确识别率,对于 KTH 数据库可得高达 95% 的识别率。

从图 7(a1)(a2)可以看出,识别结果中有很多的错误识别,如(a1)中 skip 误判为 run,jump 误判为 pjump 等;完全正确识别的动作只有三个,分别为 bend、jack、walk。其主要原因是拍摄视频的距离相对较远、分辨率较低,造成提取人体剪影轮廓图时四肢模糊不易分辨。从图 7(b1)(b2)可以看出,光流特征的识别要比图 7(a1)(a2)的结果好很多,Weizmann 数据库只有动作 skip 被误判为 run,而 KTH 数据库只有一种动作 run 被误判,说明光流特征可以对相对远距离、分辨率低的视频进行很好的识别。将光流特征和轮廓向量特征相结合,得到图 7(c1)(c2)的识别结果,可以明显地看出识别结果非常好,充分验证了提出特征的有效性。

本文方法与近期的相关方法基于 Weizmann 数据库识别性能比较如表 2 所示。从表 2 可以看出,虽然选择的特征相近,但本文提出的特征及其算法的识别性能要优于其他算法。此外,所提出的特征易于提取和表征、具有较高的可靠性,避免了基于人体模型的特征提取等方法的复杂运算。

表 2 不同特征结合对应的识别率

方法	所用特征	识别率/%
Ahmad 等人[10]	光流 + 形状流	88.29
Sawant 等人[11]	剪影 + 局部光流	97.8
Tran 等人[8]	局部剪影 + 局部光流	96.7
本文方法	整体轮廓 + 局部光流	100

4 结束语

本文提出一种基于混合特征的人体动作识别的算法,将轮廓质心到边缘点的轮廓向量距离特征和光流特征进行结合,组成混合特征进行动作识别。从第 3 章的识别结果可以看出,这种混合特征在 Weizmann 及 KTH 数据库上得到了 95% 以上的正确识别率,充分证明了该算法的有效性及可行性。下一步工作将尝试对特征进行降维处理,进一步提高算法的计算效率,以期实现算法的实时在线应用。

参考文献:

[1] ZHU Guang-yu, XU Chang-sheng, HUANG Qing-ming. Action recognition in broadcast tennis video [C]//Proc of the 18th International Conference on Pattern Recognition. [S. l.]: IEEE Press, 2006: 251-254.

[2] WANG Liang, SUTER D. Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model [C]//Proc of IEEE Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Press, 2007: 1-8.

[3] DEDEOLU Y. A silhouette-based method for object classification and human action recognition in video[C]//Proc of European Conference on Computer Vision. [S. l.]: IEEE Press, 2006: 64-77.

[4] GRITAI A, SHEIKH Y, SHAH M. On the use of anthropometry in the invariant analysis of human actions[C]//Proc of the 17th International IEEE Conference on Pattern Recognition. [S. l.]: IEEE Press, 2004: 923-926.

[5] LAPTEV I. On space-time interest points[J]. International Journal of Computer Vision, 2005, 64(2-3): 107-123.

[6] DOLLAR P, RABAUDV, COTTRELL G. Behavior recognition via sparse spatio-temporal features[C]//Proc of the 2nd Joint IEEE International Workshop on VS-PETS. [S. l.]: IEEE Press, 2005: 65-72.

[7] RONALD P. A survey on vision-based human action recognition [J]. Image and Vision Computing, 2010, 28(6): 976-990.

[8] TRAN D, SOROKIN A. Activity recognition with metric learning [C]//Proc of European Conference on Compute Vision. [S. l.]: IEEE Press, 2008: 61-66.

[9] WANG Liang, GENG Xin, LECKIE C, *et al*. Moving shape dynamics: a signal processing perspective[C]//Proc of IEEE International Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Press, 2008: 1649-1656.

[10] AHMAD M, LEE S. Human action recognition using shape and CLG-motion flow from multi-view image sequences[J]. Pattern Recognition, 2008, 41(7): 2237-2252.

[11] SAWANT N, BISWAS K. Human action recognition based on spatio-temporal features [C]//Proc of the 3rd International Conference on Pattern Recognition and Machine Intelligence. Berlin: Springer-Verlag, 2009: 357-362.

(上接第 594 页)

参考文献:

[1] AZUMA R. Recent advances in augmented reality[J]. IEEE Computer Graphics and Application, 2001, 21(6): 340-347.

[2] DAVID G L. Distinctive image features from scale-invariant key-points [J]. International Journal of Computer Vision, 2004, 20(2): 91-110.

[3] BAY H, TUYTELAARS T, GOOL L V. Speeded-up robust features [J]. Computer Vision and Image Understanding, 2008, 110(3): 346-359.

[4] CALONDER M, LEPETIT V, STRECHA C. BRIEF: binary robust independent elementary features[C]//Proc of the 11th European Conference on Computer Vision. Berlin: Springer-Verlag, 2010: 778-792.

[5] ETHAN R, VINCENT R, KURT K, *et al*. ORB: an efficient alternative to SIFT or SURF[C]//Proc of International Conference on Computer Vision. 2011: 2564-2571.

[6] ROSTEN E, DRUMMOND T. Fusing points and lines for high performance tracking[C]//Proc of the 10th International Conference on Computer Vision. 2005: 1508-1515.

[7] KLEIN G, MURRARY D. Improving the agility of keyframe-based SLAM[C]//Proc of the 10th European Conference on Computer Vision. 2008: 802-815.

[8] ASSAD M, CARMICHAEL D J, CUTTING D. AR phone: accessible augmented reality in the intelligent environment[C]//Proc of OZCHI. 2003: 232-237.

[9] MIKA H, CHARLES W, MARK B. Augmented assembly using a mobile phone[C]//Proc of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. 2008: 167-168.

[10] ROSTEN E, PORTER R, DRUMMOND T. Faster and better: a machine learning approach to corner detection[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2010, 32(1): 105-119.

[11] MICHAEL R, BEAT G. Using camera-equipped mobile phones for interacting with real-world objects[C]//Advances in Pervasive Computing. [S. l.]: Austrian Computer Society, 2004: 265-271.

[12] ANDERS H, MARK B, MARK O. Face to face collaborative AR on mobile phones[C]//Proc of the 4th IEEE International Symposium on Mixed and Augmented Reality. 2005: 80-89.

[13] ARTH C, WAGNER D, KLOPSCHITZ M, *et al*. Wide area localization on mobile phones[C]//Proc of the 8th IEEE International Symposium on Mixed and Augmented Reality. 2009: 73-82.

[14] WAGNER D, MULLONI A, LANGLOTZ T. Real-time panoramic mapping and tracking on mobile phones [C]//Proc of IEEE Virtual Reality Conference. 2010: 211-218.