

基于加速鲁棒特征的非特定目标跟踪算法研究

王招娣, 邓记才

(郑州大学 信息工程学院, 河南 郑州 450001)

摘要: 提出了将加速鲁棒特征 SURF 与混合匹配法相结合实现对非特定目标的跟踪, 跟踪目标既可以是特定目标, 亦可在跟踪过程中进行目标切换。首先提取目标图像和待匹配视频的 SURF 特征点并生成特征向量, 然后采用欧氏距离和 Hessian 矩阵迹相结合确定特征点匹配对, 实现目标的定位与跟踪。选用 VS 2008 平台进行仿真实验, 结果表明, 该算法既保持了跟踪目标的尺度不变性, 又实现了跟踪过程的可控性。

关键词: 加速鲁棒特征; 特征匹配; 非特定目标; VS 2008

中图分类号: TP391

文献标识码: A

文章编号: 1674-7720(2014)01-0046-04

Research of unspecific target tracking algorithm based on speeded up robust features

Wang Zhaodi, Deng Jicai

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: Through the extraction target image and video sequence motion object SURF features, and by using the euclidean distance calculation feature matching similarity, and ultimately to determine the position of the moving target. This paper used for target matching SURF feature is the local characteristics of the image, with scale invariance, in view of the target happen translation, rotation, scaling, brightness change, part of the barrier and perspective transformation can get better matching effect. Finally, this paper adopts VS 2008 platform to simulate, the simulation experiment results show that the method is a kind of concise and effective movement target tracking method. Speeded up robust features is the local features of image, and has the scale-invariance.

Key words: SURF; feature matching; unspecific target; VS 2008

运动目标的识别与跟踪^[1]是机器视觉中研究的热点问题之一, 目标匹配是目标跟踪的关键技术, 目前广泛采用的图像特征匹配法主要有诸如边缘特征、颜色特征^[2]、骨架特征、区域特征和特征点等。其中, 特征点检测法作为图像的局部特征提取法, 其特征匹配算法计算量小, 鲁棒性好, 且对图像旋转平移、光照及分辨率等变化有较好的适应性, 逐渐成为特征匹配领域的研究热点。常用的特征点检测方法有 Harris 角点检测^[3]、SUSAN、Hessian-Laplace 等, 但这些特征点在图像尺度空间发生变化时, 易导致匹配失败。

2004 年, LOWE D G 等人提出 SIFT (Scale-Invariant Feature Transform) 特征检测法, 解决了尺度变化的问题, 但该算法由于检测的特征点太多, 使得计算时间开销大。随后 BAY H 等人于 2006 年对 SIFT 算法作了改进, 提出了

加速鲁棒特征 SURF (Speeded UP Roubust Features)^[4]算法。该算法不仅保持了 SIFT 算法对图像旋转、尺度缩放、亮度变化的不变性, 同时由于引入了积分图、箱式滤波器和近似 Hessian 矩阵等概念, 使算法的执行效率得到大大提高。目前, 该算法主要用于人脸识别^[5]、掌纹识别、医学遥感影像配准和全景拼接^[6]等领域。因此, 本文选用 SURF 特征匹配的方法来实现视频图像中运动目标的跟踪。

1 SURF 算法

SURF 算法作为图像局部特征的提取算法, 它对图像尺度变换、视角变化、仿射变换及噪声保持一定程度的稳定性, 可完成两幅图像特征点的匹配。该算法的实现过程分为特征点检测和特征向量的生成。

1.1 特征点检测

尺度空间的建立是图像 SURF 特征点检测的首要步

骤,将一幅二维图像 $I(x, y)$ 与高斯核函数 $G(x, y, \sigma)$ 作卷积运算,即得到不同的尺度空间^[7]:

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

再采用 Hessian 矩阵在尺度空间的每一层上采集图像的极值点,对二维图像 $I(x, y)$ 上的任意一点 $X(x, y)$,在不同空间尺度 σ 上, Hessian 矩阵的定义为:

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (2)$$

其中, $L_{xx}(X, \sigma)$ 、 $L_{yy}(X, \sigma)$ 和 $L_{xy}(X, \sigma)$ 分别表示图像上的 $X(x, y)$ 点与二阶偏导在 x 、 y 和 xy 方向的卷积。

为了减少计算的复杂度, SURF 算法采用箱式滤波器模型中矩形波来代替二阶高斯滤波,同时引入积分图像的概念加速了卷积的运算过程。因此,用图像 $I(x, y)$ 上的任一点 $X(x, y)$ 分别与箱式滤波模版的二阶偏导作卷积的结果 D_{xx} 、 D_{yy} 和 D_{xy} 来代替 $L_{xx}(X, \sigma)$ 、 $L_{yy}(X, \sigma)$ 和 $L_{xy}(X, \sigma)$ 。为了保持计算的精度,选取矩形核函数与高斯核函数的近似比例因子 ω ,一般记作 0.9,故 Hessian 的行列式可近似记作:

$$\det(\text{Hessian}) = D_{xx} \times D_{yy} - (\omega D_{xy})^2 \quad (3)$$

根据 Hessian 矩阵求出各个尺度空间上的极值之后,采用非极大值抑制 NMS (Non-Maxima Suppression) 将这些极值点与它立体邻域内的点进行比较,以 3×3 的滤波器为例,如图 1 所示,将该极值点与同尺度层中其余 8 个点及上下两个尺度层 9 个点进行比较,只有当该极值点都大于或都小于其余 26 个相邻的点,才能将该极值点作为候选区的 SURF 特征点。

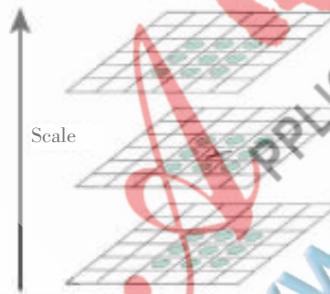


图 1 不同尺度空间的特征点检测

1.2 特征向量的生成

SURF 特征描述子的生成分为主方向的确定和描述子向量的生成两个步骤。首先,在以特征点为中心、半径为 6σ (σ 为特征点所在的尺度值) 的圆形域邻域内,计算像素点在 x 、 y 方向上的 Haar^[31] 小波响应值并对它们进行加权统计,再从 x 轴开始将圆形区域 60° 范围内的 Haar 小波响应值叠加,从而得到一个新的矢量记作矢量 L_1 ,如图 2 所示。以同样的方式每隔 5° 遍历整个圆域,计算 $\frac{360}{5} = 72$ 个不同矢量 L_1, L_2, \dots ,再从这 72 个矢量中选择最长的矢量方向作为该特征点的主方向。



图 2 特征点主方向确定

然后,以特征点为中心将坐标轴旋转到主方向,取边长为 20σ 的正方形区域,将该区域划分为 4×4 的子区域,如图 3 所示,统计每个子区域内像素点在 x 轴方向和 y 轴方向的 Haar 小波响应值,并分别记作 d_x 和 d_y ,再对这些值进行加权、求和操作,即得到 $\sum d_x$ 、 $\sum |d_x|$ 、 $\sum d_y$ 和 $\sum |d_y|$,于是在子区域中就生成特征点描述符的特征向量 V ,如式(4)所示。若将 16 个区域的向量都加入特征向量中,就变成 $16 \times 4 = 64$ 维特征向量。

$$V = (\sum d_x, \sum |d_x|, \sum d_y, \sum |d_y|) \quad (4)$$

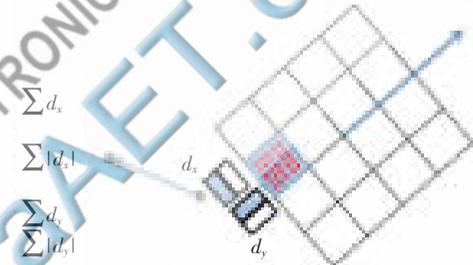


图 3 特征点描述子的生成

2 基于 SURF 特征的目标匹配算法

本文选用 SURF 算法对视频序列中非特定运动目标进行跟踪研究,具体步骤如下。

(1) 特征点提取。指定目标模板作为视频图像的跟踪对象,提取目标模板和视频图像 SURF 特征点并生成特征向量。

(2) 特征匹配与跟踪。将目标图像与视频图像之间的 SURF 特征点进行相似度匹配,确定目标在视频序列中的位置并进行跟踪。

(3) 目标模板更新。在视频图像发生改变或运动目标消失时,可随机切换跟踪目标实时更新目标模板,做到实时有效的跟踪效果。

2.1 特征点匹配方法

根据快速最近邻逼近搜索法 FLANN (Fast Approximate Nearest Neighbor Search) 获得目标图像与视频序列图像中特征点匹配对,并选用欧氏距离和 Hessian 矩阵的迹相结合的相似度计算方法找到相应的特征点匹配对。

首先根据 Hessian 矩阵的迹实现特征点初步匹配。因为图像对比度相同的特征点对应的 Hessian 矩阵的迹

(即矩阵对角元素之和)互为同号,而图像对比度不同的特征点对应的 Hessian 矩阵迹互为异号,因此可将目标图像与视频图像特征点进行比较,选择出 Hessian 矩阵迹互为同号的特征对,放弃互为异号的特征对。然后,在上一步初匹配的基础上,再采用欧氏距离匹配法进行相似度判断,最终找到两幅图像中特征点的一一对应关系。欧氏距离计算公式为:

$$D_{ij} = \left[\sum_{m=0}^{n-1} (X_{im} - X_{jm})^2 \right]^{1/2} \quad (5)$$

其中, X_{im} 表示目标图中第 i 个特征描述子的第 m 个元素, X_{jk} 表示源图像中第 j 个特征描述子的第 k 个元素, n 表示特征向量的维数 64。通过计算得出最小欧氏距离 D_{a1} 、次最小欧氏距离 D_{a2} 及两者的比值 $\frac{D_{a1}}{D_{a2}}$, 将该比值与设定阈值 0.7 作比较。若小于阈值则证明目标图像上的该特征点与源图像上最小欧氏距离对应点是匹配的; 否则, 不匹配。

本文算法的流程图如图 4 所示。

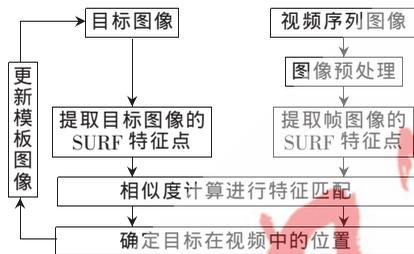


图 4 非特定目标跟踪算法的流程图

2.2 算法的具体实现步骤

本文算法的具体实现步骤如下。

(1) 读取待检测视频图像, 采用中值滤波的方式对视频进行预处理操作, 滤除视频中的噪声点和干扰点, 并将 RGB 彩色视频进行灰度化处理转换成灰度视频以备用。

(2) 既可读取固定的目标图像作为跟踪对象, 亦可在视频序列中随意指定要跟踪的运动目标, 并将该目标从视频序列中分割出来单独作为跟踪对象。

(3) 利用 Hessian 矩阵分别检测并提取目标模板和视频图像每一帧图像的 SURF 特征点。

(4) 将目标模板的特征点与视频帧的特征点进行相似度匹配, 保留匹配正确的特征对, 剔除不能匹配的特征点。

(5) 对匹配成功的特征对, 计算其特征点在视频图像中的分布情况并计算其重心位置, 从而确定目标图像在视频图像中出现的 4 个顶点坐标。

(6) 根据顶点坐标画出目标图像在视频序列中的位置, 并实时更新跟踪结果。

(7) 在视频序列中若发生运动目标改变或跟踪目标丢失的情况, 可随机切换跟踪目标实现目标模板的实时

更新, 从而做到实时有效的跟踪效果。

3 实验结果及分析

为验证 SURF 算法对视频序列中的运动目标动态跟踪的有效性, 本文选用 VS 2008 作为实验平台, 结合 OpenCV 2.3 视觉函数库^[8]进行仿真实验。实验中的目标模板为 JPEG 格式的图像, 视频序列为 352 像素×240 像素, 帧速率为 25 f/s, XVID 压缩编码的 24 位 AVI 格式的 video1 和 video2 视频源, 其中 video1 为单个运动目标视频, video2 为两个运动目标的视频。

3.1 特定目标跟踪过程

以特定的目标图像为匹配模板进行仿真实验, 并提取该模板的 SURF 特征点如图 5 所示, 共提取 20 个特征点。

将该目标模板的 SURF 特征点分别与视频序列帧图像的特征点进行对比匹配, 找到确定的 SURF 特征匹配对从而确定目标出现的位置, 以视频第 50 帧、100 帧和 150 帧为例的仿真结果如图 6 所示。



图 5 固定目标模板特征点提取图



图 6 特定目标的跟踪效果图

将提取到的特征点数目及特征匹配对数进行统计, 结果如表 1 所示。其中目标模板的 SURF 特征点数为 20 个, 对应不同视频帧的 SURF 特征点数分别为 44、45 和 42 个。在特定目标跟踪的情况下特征点匹配率是比较低的。

表 1 特定目标跟踪过程的特征点统计表

目标特征点个数	20			24		
对应视频目标特征点个数	44	45	42	39	41	44
特征点匹配对/对	3	5	6	5	4	7
特征点匹配率	0.15	0.25	0.3	0.21	0.16	0.29

3.2 非特定目标跟踪过程

本文算法采用将 SURF 特征与跟踪目标的可控性原理相结合, 实现对非特定运动目标的跟踪, 即在跟踪过程中可进行目标实时切换。将目标模板特征点与视频序列帧图像的特征点进行对比匹配, 找到正确的 SURF 特征对, 从而确定目标出现的位置。以 video1 和 video2 分别为跟踪视频进行算法仿真, 图 7 所示为 video1 的非特定目标跟踪结果, 图 8 所示为 video2 的非特定目标跟踪结果。其中黑色线段表示匹配正确的 SURF 特征点对应关系, 白色方框即为标定的目标跟踪区域。

对于 video1 和 video2 视频源, 将提取到特征点数目及特征匹配对数进行统计, 如表 2 所示, 其中 video1 的

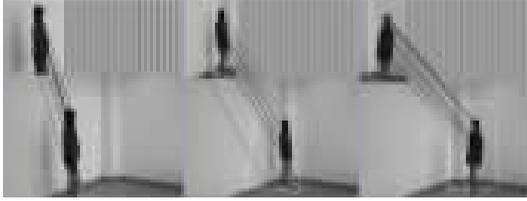


图7 video1中运动目标的跟踪效果图

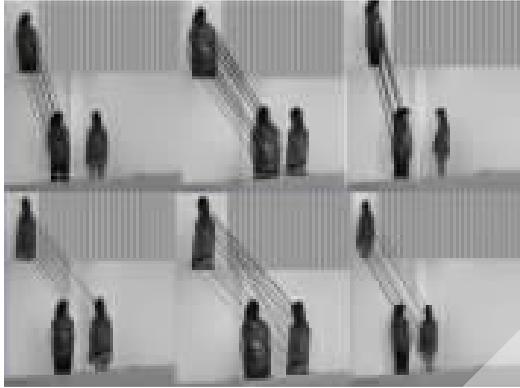


图8 video2中运动目标的跟踪效果图

特征点匹配率在 57.3%左右, video2 的特征点匹配率在 58%左右。

表2 非特定目标跟踪过程的特征点统计表

视频源	video1			video2		
	目标特征点数/个	30	27	37	51	40
视频目标特征点数/个	43	56	44	119	86	106
特征点匹配对/对	18	15	21	32	23	20
特征点匹配率	0.6	0.55	0.57	0.63	0.57	0.54

将本文所采用的 SURF 的非特定目标跟踪算法与一般特定目标跟踪算法进行比较, 特征点匹配率的对比情况如图 9 所示。

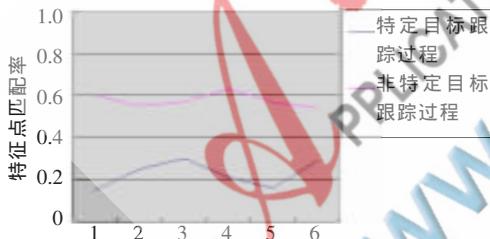


图9 两种算法特征点匹配率对比图

实验表明, 本文算法的特征点匹配率在 57%~58% 范围内波动, 而特定目标跟踪算法特征点匹配率在 15%~30% 范围内, 前者特征匹配率要比后者提高 30%~40%。

本文采用一种 SURF 算法对视频序列中非特定运动目标进行跟踪研究, 并通过实验对该算法进行仿真验证。结果表明, 该过程既可指定特定目标作为跟踪对象, 亦可在跟踪过程中实时进行目标切换, 体现跟踪过程的灵活性; 在运动目标发生尺度变化、旋转、缩放的情况下, 仍能实现目标的正确检测且不使目标跟踪丢失, 体现了算法的准确性和一定的环境适应性; 同时由于本文算法结合了目标的可控性切换原理, 使 SURF 特征点匹配率较特定目标跟踪算法提高了 30%~40%, 增强了特征点匹配的准确性。因此, 本文算法是一种有效且灵活的非特定目标跟踪法。如何进一步解决多运动目标的遮挡问题, 使它能够更好地适应较为复杂环境下的目标检测与跟踪, 是下一步工作中要研究的重点。

参考文献

- [1] Zhang Jianming, Zhang Lingzeng, Liu Zhiqiang. Approach to front vehicle detection and tracking based on multiple features[J]. Computer Engineering and Applications, 2011, 47(5):220-223.
- [2] Song Linlin, Wang Xianghai. Image retrieval algorithm using both color and texture features[J]. Computer Engineering and Applications, 2011, 47(34):203-207.
- [3] 张永, 纪东升. 一种改进的 Harris 特征点检测算法[J]. 计算机工程, 2011, 37(13): 196-198, 201.
- [4] BAY H, TUYTEPLAARS T, VAN GOOL L. SURF: Speeded Up Robust Features [C]. Processings of the European Conference on Computer Vision, 2006:404-417.
- [5] 明悦. 基于不变性特征的三维人脸识别研究[D]. 北京: 北京交通大学, 2013.
- [6] 宋延爽, 耿楠. 基于 SURF 算法的全景图拼接技术研究与应用[J]. 计算机工程与设计, 2012, 33(12): 4647-4651.
- [7] 林建群, 平西建, 马德宝. 基于二阶方向导数尺度空间的小目标检测[J]. 电子与信息学报, 2012, 34(12): 2992-2998.
- [8] 方玫. OpenCV 技术在数字图像处理中的应用[J]. 北京教育学院学报(自然科学版), 2011, 6(1): 7-11.

(收稿日期: 2013-10-02)

作者简介:

王招娣, 女, 1988 年生, 硕士研究生, 主要研究方向: 数字图像、视频处理。

邓记才, 男, 1964 年生, 博士, 教授, 主要研究方向: 射频信号识别, 嵌入式系统和数字视频处理。