

基于加速稳健特征算法的监控摄像头视频融合技术

曹恒峰¹, 彭来湖^{1,2}, 史伟民¹

(1.浙江理工大学浙江省现代纺织装备技术重点实验室, 浙江 杭州 310018;

2.杭州勤诚微电子科技有限公司, 浙江 杭州 310018)

✉17826856989@163.com; 43233212@qq.com; swm@zstu.edu.cn



摘 要: 视频监控摄像头在现代安防中起到了不可替代的作用, 为适应更多环境, 摄像头技术也一直在不断发展。本文提出了一种可实现多摄像头画面实时拼接的技术设计方案: 基于加速稳健特征算法, 实现对视频单帧画面间的特征点进行提取及拼接; 使用FFmpeg(Fast Forward Mpeg)媒体处理库进行视频摄像头媒体流文件的分解及最终融合视频的呈现; 使用Qt框架实现程序开发及效果展示。经测试表明, 该设计方案在保证不降低画面分辨率的前提下, 实现多摄像头的视频画面融合, 画面扭曲程度低, 为摄像头画面融合提供了一些参考。

关键词: 加速稳健特征算法; FFmpeg; 画面拼接; 视频融合

中图分类号: TP311.5 **文献标识码:** A

Surveillance Camera Video Fusion Technology based on Accelerated Robust Feature Algorithm

CAO Hengfeng¹, PENG Laihu^{1,2}, SHI Weimin¹

(1. Key Laboratory of Modern Textile Machinery & Technology of Zhejiang Province, Zhejiang Sci-tech University, Hangzhou 310018, China;

2. Hangzhou Qincheng Microelectronics Technology Co., Ltd., Hangzhou 310018, China)

✉17826856989@163.com; 43233212@qq.com; swm@zstu.edu.cn

Abstract: Video surveillance cameras have played an irreplaceable role in modern security. In order to adapt to more environments, camera technology has been constantly developing. This paper proposes a technical design solution that can realize real-time splicing of images from multiple cameras. Accelerated robust feature algorithm is applied to realize the feature point extraction and splicing between single video frames. FFmpeg (Fast Forward Mpeg) media processing library is used to decompose stream files of video camera media and present the final fusion video. Qt framework is used to develop program and display its effect. Tests show that the proposed solution can achieve multi-camera video picture fusion without reducing picture resolution, with low picture distortion. It provides some references for camera picture fusion.

Keywords: accelerated robust feature algorithm; FFmpeg; screen splicing; video fusion

1 引言(Introduction)

在数字化急速发展的今天, 视频监控设备已经得到大范围的普及, 在安防、军工领域及智慧交通等方面为人们提供了巨大的便利^[1]。随着科技的发展, 不同的应用场景对监控提出了新的要求, 如更高的清晰度、更广泛的应用场景、更低的画面延迟等, 其中对视野范围的要求呼声最高, 但高分辨率的广角摄像头成本昂贵, 使用场景受限, 而且其画面变形程度较高, 无法满足一些条件苛刻的场景, 而一种多摄像头的视频画面拼接技术, 为大范围广角视频监控提供了一种方案^[2]。

本文所研究的视频拼接技术可将多个高分辨率监控摄像头的视频画面进行拼接, 从而达到在不降低分辨率的同时实现广角全景监控的效果。监控摄像头视频拼接不仅仅是单纯地把画面堆砌在一起, 而是需要根据画面内容将重复部分的画面进行融合, 实现整体的效果。与图像拼接相比, 视频拼接是在其基础上的一个提升^[1], 不仅需要实现每一帧画面的拼接, 而且需要保证拼接速度, 否则在每秒25帧的视频呈现中, 程序会非常卡顿, 无法达到实际应用效果。

2 视频融合技术实现方案(Implementation plan of video fusion technology)

视频画面的拼接是在图像拼接的基础上实现的，但又是在单纯的图像拼接技术上的提升。图像拼接包含特征点提取、特征点匹配、图像矩阵变换和画面拼接等步骤，若是对视频中的每一帧都进行图像拼接的全过程操作，不仅会导致严重的视频延时，而且将造成极大的内存和CPU消耗，无法满足日常生活使用的需求。为此，通过研究图像拼接技术的原理，本文提出了一种适合视频监控摄像头的视频画面融合技术，其主要流程有四个步骤：

(1)获取摄像头基于实时流传输协议(Real Time Streaming Protocol, RTSP)的媒体流画面并进行解码，提取其同一时刻的画面帧。

(2)通过加速稳健特征(Speeded-Up Robust Features, SURF)算法^[2]对两幅画面进行特征点集匹配。

(3)根据匹配后的特征点集进行变换矩阵计算，并将摄像头2的图像进行矩阵变换。

(4)将摄像头1的图像与摄像头2的图像按坐标进行拼接并输出，实现图像拼接。

(5)重复提取视频画面帧，跳过特征点集匹配步骤，直接按照上一步所计算的变换矩阵进行矩阵变换，并将图像画面输出，实现拼接视频输出。

本次视频拼接程序技术方案流程如图1所示。

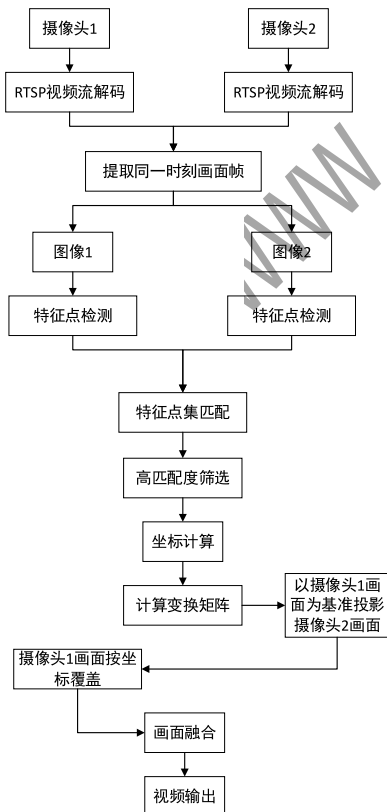


图1 技术方案流程

Fig.1 Technical solution process

3 系统实现(System implementation)

3.1 视频流读取

目前主流摄像头的视频画面均支持RTSP媒体流输出，在计算机中导入FFmpeg媒体开发库。FFmpeg是一套开源的计算机程序，主要用于视频、音频文件的转化、输出和保存，其最主要的功能是可以将音视频文件转化为流，实现媒体文件在互联网上的传输^[3]。在计算机中配置好环境变量后，即可在Qt程序中使用FFmpeg相关的音视频处理函数，将摄像头中所提取的RTSP视频流信息导入程序中，遍历与视频流所匹配的解码器类型，即可将摄像头画面解码后的YUV数据转换成RGB 32^[4]，再将其逐帧转化为图像格式文件，等待下一步处理。

3.2 SURF特征匹配

特征点是指在一幅图像中纹理、灰度变化较大的点，通常为图像中物体的边界点、阴影分界点、拐点等关键信息点^[5-6]。一幅图像可以通过特征点来进行描述，它反映了图像的本质特征，不同图像之间也可以通过特征点进行区分，在图像融合技术中特征点可以进行两幅图像间的匹配。

特征点检测的核心为找到不受图像的模糊程度及分辨率影响而变化的点。SURF算法特征点检测过程主要可以概括为尺度空间的极值检测、特征点方向赋值以及特征点描述符与匹配三个步骤。

3.2.1 尺度空间的极值检测

局部极值检测涉及Hessian矩阵。Hessian矩阵用于检测图像中的突变点^[7]，由一个多元函数的二阶偏导数构成，对于图像中任意一点 (x, y) 的Hessian矩阵可表示为：

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (1)$$

在构建Hessian矩阵前需要对图像进行高斯滤波，经过滤波后的Hessian矩阵表达式为：

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

式(2)中， $L(x, y, \sigma)$ 为图像的高斯尺度空间， σ 为尺度因子，由其各自对应的二阶高斯偏导与图像卷积所得。根据式(2)可得Hessian矩阵的判别式为：

$$\text{Det}(H) = L_{xx} \times L_{yy} - L_{xy}^2 \quad (3)$$

为提高运算速度，SURF算法使用方块滤波器代替高斯滤波器，并加入0.9的权值系数，消除其近似误差，因此Hessian矩阵的判别式可表示为：

$$\text{Det}(H) = D_{xx} \times D_{yy} - (0.9 \times D_{xy})^2 \quad (4)$$

将经过Hessian矩阵处理的像素点同与其空间尺度相邻的

26个像素点进行比较(其中与其同层的像素点八个,上下两层像素点各九个,如图2所示),即可根据极值得到图像特征点。

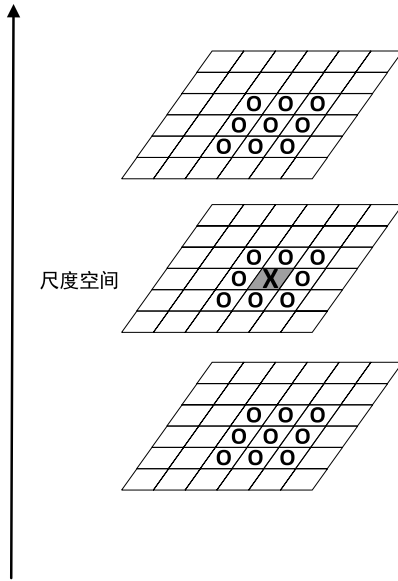


图2 像素尺度空间分布

Fig.2 Pixel-scale spatial distribution

3.2.2 特征点方向赋值

SURF算法通过统计特征点圆形领域内Haar小波特征来确定特征点主方向。以特征点为中心,统计在其半径为 6σ 的圆形区域内, 60° 扇形区域中特征点的水平和垂直Haar小波特征总和,再将扇形以每 5° 角度对整个圆形区域进行遍历,得到小波特征最大的扇形区域方向即为该特征点主方向,如图3所示。

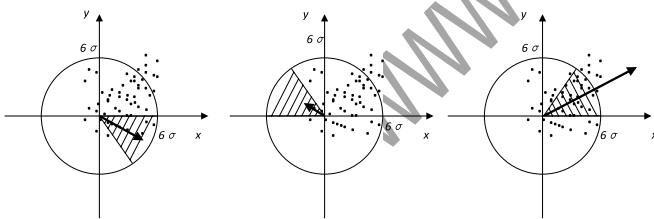


图3 特征点方向选择

Fig.3 Feature point direction selection

3.2.3 特征点描述符与匹配

以所选特征点为中心,构造一个与特征点方向相同的 $20\sigma \times 20\sigma$ 方形区域。再将该区域划分为 4×4 的小区域,统计其中每个小区域内等间隔的 5×5 采样点在水平和垂直方向上的Haar小波特征,并记作 dx 和 dy ,对其做求和运算,即 $\sum dx$ 、 $\sum |dx|$ 、 $\sum dy$ 、 $\sum |dy|$,得到每个子区域的四维向量。最后计算两幅图像中每个特征描述向量间欧式距离,所得最小值即为最佳匹配点^[8-9],如图4所示。

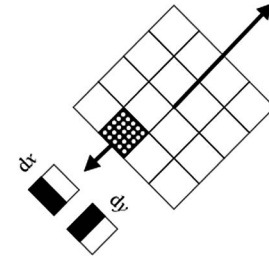


图4 特征点描述符计算

Fig.4 Feature point descriptor calculation

3.3 变换矩阵

在实际应用中,不同摄像头之间由于摆放位置不正等因素,会导致所得画面之间存在角度、视野范围等偏差,因此在进行视频图像拼接前,不仅仅需要进行坐标变换,还需要将摄像头2的画面进行矩阵投影变化,使得两个画面间有更好的重合度。

通常经过SURF算法匹配的特征点数量较多,误匹配的特征点数量也会包含在其中,如果直接对该特征点集进行变换矩阵计算,会导致变换后的图像拼接效果差,甚至会造成拼接失败,因此在计算变换矩阵前,需要对特征点集进行筛选。首先将特征点集按照匹配程度排序,需取出匹配程度最高的50组特征点集。在矩阵变换中,要保证变换矩阵有唯一解只需要八对匹配点,因此还需对初筛后的特征点集进行下一步筛选。本次设计使用了随机一致性采样(Random Sample Consensus, RANSAC)的鲁棒算法进行筛选,RANSAC算法在外点多的环境中筛选效率极高,最高能处理50%外点的情况^[10]。RANSAC算法的具体实现有如下几个步骤:

(1)随机选择一组点集作为假设的内点集,再根据这部分假设的内点集得到计算模型。

(2)将其他的点集代入步骤1计算所得的模型中,若有足够多的点集符合该模型,则认为该模型是合理的。

(3)由于步骤1所得的模型仅仅由初始所选的随机点集计算所得,模型不严谨,用步骤2符合该模型的所有内点集重新计算模型,得到最终模型。

(4)选择不同的初始随机点集,重复步骤1—3,选择出包含点集最多的模型。

经过RANSAC算法计算的模型过滤了匹配程度异常的点集,再次筛选了符合要求的特征点集合,减少了点集数量,还能提高变换矩阵的计算速度及准确性,根据所得的优质点集即可计算出图像2投影至图像1所需的变换矩阵。

3.4 融合视频输出

在计算出变换矩阵后,即可实现最终的视频输出。将经过矩阵变换的图像2放置于Qt中的QPainter画布中,再将图像1覆盖至图像2上,实现两个图像画面的拼接。设置FFmpeg输出

画面为25帧, 视频摄像头中上传的RTSP视频流中每一帧的数据依次经过FFmpeg解码、变换、拼接这些输出的步骤后^[3,11], 即可实现融合视频画面输出的效果。

4 运行调试分析(Running debug analysis)

4.1 实验环境

本次实验所使用的摄像头为海康威视DS-2CD3325F-I网络摄像机, 开发工具为Qt 5.13.0 MinGW64位版本, 开发环境为Intel(R)Core(TM)i5-7300HQ CPU@2.50 GHz, 8 GB内存, Windows 10操作系统。

4.2 调试分析

由于摄像头位置固定, 因此只需在第一次读取画面时计算变换矩阵, 后续画面均可直接使用该变换矩阵, 可有效减少程序运行时的负载, 达到流畅的视频呈现效果。实验结果显示, 本文所设计的软件在视频拼接过程中延时为0.9 s, 内存占用161.8 MB, CPU占用28.8%, 符合视频拼接软件实际应用需求, 如图5至图7所示。



(a)视频图像1

(b)视频图像2

图5 原始视频图像

Fig.5 Original video image

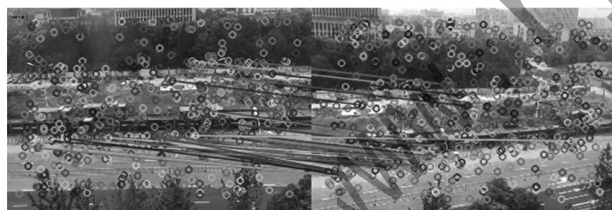


图6 匹配特征点

Fig.6 Matching feature points



图7 拼接画面输出

Fig.7 Splicing screen output

5 结论(Conclusion)

本文设计了一种基于SURF算法的摄像头监控视频拼接方案, 解决摄像头监控画面范围不足的问题。借助Qt开发框架的高拓展性, 载入FFmpeg媒体开发库, 实现摄像头RTSP视频流提取, 使用SURF算法提取图像间的特征点集, 经过

RANSAC算法筛选后计算图像间的变换矩阵, 实现视频画面的拼接融合; 利用固定摄像头画面角度不变的特点, 只计算一次变化矩阵, 达到程序流畅运行的效果。经实验证明, 此次所设计的程序符合实际应用的要求, 为视频处理提供了一种新的升级思路。

参考文献(References)

- [1] 蒋辉, 杨娟. 基于高清图像处理的全景拼接技术研究[J]. 电子世界, 2020(14):17-18.
- [2] 李慧慧. 基于Harris-SURF描述符的图像配准方法[J]. 科学技术创新, 2020(20):108-109.
- [3] 郭晶晶, 刘光尧, 汪磊, 等. FFmpeg在视频图像处理中的应用[J]. 刑事技术, 2020, 45(03):234-237.
- [4] 廖银萍. 支持HTML5的RTSP流媒体网络摄像头设计与实现[J]. 信息通信, 2020(03):106-109.
- [5] 赵潇洒, 陈西江, 班亚, 等. 融合改进SURF和Cell加速的幂函数加权图像拼接方法[J/OL]. 激光与光电子学进展:1-19[2020-08-15]. <http://kns.cnki.net/kcms/detail/31.1690.TN.20200713.1810.040.html>.
- [6] SEUNG H C, IL K E, SEOK W H, et al. An enhanced SURF algorithm based on new interest point detection procedure and fast computation technique[J]. Journal of Real-Time Image Processing, 2019, 16(4):1177-1187.
- [7] 杨蕊, 李雪. 基于显著特性的红外全景图像拼接[J]. 激光杂志, 2020, 41(06):98-101.
- [8] BAY H, ESS A, TUYTELAARS T, et al. Speeded-up robust features[J]. Computer Vision and Image Understanding, 2008, 110(3):346-359.
- [9] 夏磊, 胡欣宇, 岳亚伟, 等. 基于改进SURF算法的红外图像拼接[J]. 物联网技术, 2020, 10(06):48-51.
- [10] ZAHRA H N, MEHDI N. An adaptive image registration method based on SIFT features and RANSAC transform[J]. Computers and Electrical Engineering, 2017, 62(5):524-537.
- [11] 阿不都克里木·玉素甫, 王亮亮. 基于自主可控平台的FFMPEG在线视频转换系统[J]. 计算机与现代化, 2020(01):81-84;116.

作者简介:

曹恒峰(1996-), 男, 硕士生. 研究领域: 控制工程。

彭来湖(1980-), 男, 博士, 讲师. 研究领域: 智能装备与嵌入式控制技术, 智能传感器与仪器仪表, 智能制造与车间信息化技术, 工业互联网通信。

史伟氏(1965-), 男, 博士, 教授. 研究领域: 纺织机械自动控制, 轻工机械, 邮政装备。