

# 远程全景实时监控系统设计实现

杨 涛, 罗健欣, 金凤林, 张小峰

(陆军工程大学 指挥控制工程学院, 江苏 南京 210007)

**摘 要:**针对现有实时视频通信系统因视频编解码模块和传输协议之间协作不紧密影响视频通信的传输性能和视觉质量,以及常用的远程实时监控系统的监控范围和角度有限,多画面独立显示不够直观的问题,采用新型紧耦合式的视频通信架构,通过逐帧联合控制视频的编解码和传输,优化了每一帧的编码长度,提高了实时监控视频传输的端到端时延性能和视频图像质量;设计了多路远程视频源间的帧同步算法,并基于 OpenCV 的图像拼接模块定制了图像拼接流程;通过在特征点提取和匹配、图像变形和图像融合阶段使用 GPU 加速,并在逐帧拼接时设置合适的拼接缝更新间隔,实现了监控视频的近实时拼接。实验证明,系统能够获得良好的监控视频质量和更广的监控范围,一定程度上解决了视频监控中的盲区问题,同时具有较好的实时性和视觉效果。

**关键词:**实时监控;紧耦合;视频编解码;视频通信;视频拼接;全景

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2021)04-0118-07

doi:10.3969/j.issn.1673-629X.2021.04.020

## Design and Implementation of Remote Real-time Panoramic Monitoring System

YANG Tao, LUO Jian-xin, JIN Feng-lin, ZHANG Xiao-feng

(School of Command & Control Engineering, Army Engineering University of PLA, Nanjing 210007, China)

**Abstract:** Aiming at the problem that the collaboration between the video codec and the transmission protocol in existing real-time video communication systems is not close, which affects the transmission performance and visual quality of video communication, the commonly used remote monitoring system has a limited monitoring range and angle, and the video is not intuitive enough with multi videos displayed independently, a new kind of tightly coupled video communication architecture is adopted to jointly control the video encoding, decoding and transmission frame by frame, which optimizes the encoding length of each frame and improves the end-to-end delay performance and image quality of real-time monitoring video transmission. A frame synchronization algorithm between multiple channels of remote video is designed, and an image stitching process is customized based on OpenCV image stitching module. By using GPU acceleration at the stage of feature extraction and matching, image deformation and blending, setting appropriate stitching seam update interval, near real-time stitching is achieved. Experiment shows that this system can obtain better monitoring picture quality and wider monitoring range, which solves the problem of blind spots in video monitoring to some extent, with well real-time capability and visual effects at the same time.

**Key words:** real-time monitoring; tightly coupled; video codec; video communication; video stitching; panoramic

## 0 引言

随着网络的普及和带宽的飞速增长,实时视频通信由于其独特的优势,逐渐在网络流量中占据重要的一部分。实时监控作为实时视频通信的一种重要形式,已经深入到生产生活的方方面面,尤其是在一些涉及安全问题的场合<sup>[1]</sup>,更是发挥着不可替代的作用。实时视频通信一般包括视频的采集、压缩编码、网络传

输和可视化等技术环节。在传统的视频通信架构中,视频编解码模块和传输协议模块各自工作在相对独立的控制回路,两者之间的协作不够紧密,由此导致编码器的输出比特率与当前的网络容量不匹配,有可能引起拥塞或者性能的损失<sup>[2]</sup>。传统的视频监控通常采用B/S模式,由服务器将被监控端摄像头采集的视频流推送到监控端的浏览器。这类监控系统通常使用单个

收稿日期:2020-06-08

修回日期:2020-10-12

基金项目:国防科技基金(3602027);江苏省自然科学基金(BK20150722)

作者简介:杨 涛(1990-),男,硕士研究生,研究方向为多媒体通信、计算机网络;罗健欣,CCF会员(13223M),讲师,研究方向为计算机视觉;金凤林,副教授,研究方向为计算机网络。

或者多个摄像头对被监控场景进行视频采集,在监控端独立显示各个监控画面。通过这种方式获取的实时监控由于各摄像头相互独立,画面通常不够直观,甚至存在一定的盲区。针对以上两个问题,设计了一种采用紧耦合式视频通信架构的远程全景实时监控系统。一方面使得视频编解码模块和传输协议之间的协作更为紧密,优化了每一帧的编码长度,提升了视频通信的时延性能和视频质量<sup>[2]</sup>;另一方面全景监控使得监控角度更大,画面更加直观,可以在一定程度上解决视频监控盲区问题。

## 1 相关关键技术

### 1.1 紧耦合式视频编解码和传输

在现有的常规视频通信系统中,如 Skype、WebRTC<sup>[3]</sup>、和 Facetime 等,视频编解码模块和传输协议通常工作在各自的控制回路之中。传输协议通过处理网络中的确认、拥塞等信号,对当前的网络平均速率进行估计并将此通告给编码模块。以 WebRTC 为例,通告网络速率估计值的时间间隔大约为 1 秒。编码模块根据此估计值设置编码输出比特率,生成压缩的比特流,其平均比特率接近估计的网络速率,这样的速率

控制相对来说是比较粗糙的。而在紧耦合式的新型视频架构中,编解码器和传输协议被整合进一个控制回路,相互之间协作更为紧密,如图 1 所示。在这样的架构中,编码器并行编码同一帧的不同质量版本。传输协议通过估算当前网络容量,计算当前所能发送的最大帧长度,从而选择编码输出大小最接近但不超过当前网络容量的质量版本进行发送,实现粒度更细的逐帧控制,从而更有效地避免拥塞或者网络性能浪费。能够将编解码模块和传输协议紧耦合到一起,主要依赖于两点:

(1) 基于现有的编解码器进行改造,退变为纯函数式编码器,不再控制比特率而只提供编解码的接口;在编解码器中引入“状态(state)”的概念,每一个状态值与每一次编码一帧时所用的参数相关,在收发双方的内存中维护了状态队列,用类似于 TCP 滑动窗口的模式来管理队列中的状态并实现收发双方状态的同步。

(2) 通过指数加权移动平均法(exponentially weighted moving average, EWMA) 计算平均包到达时延,并据此计算当前网络容量,实现帧的编码长度和瞬时网络容量的最佳适配。

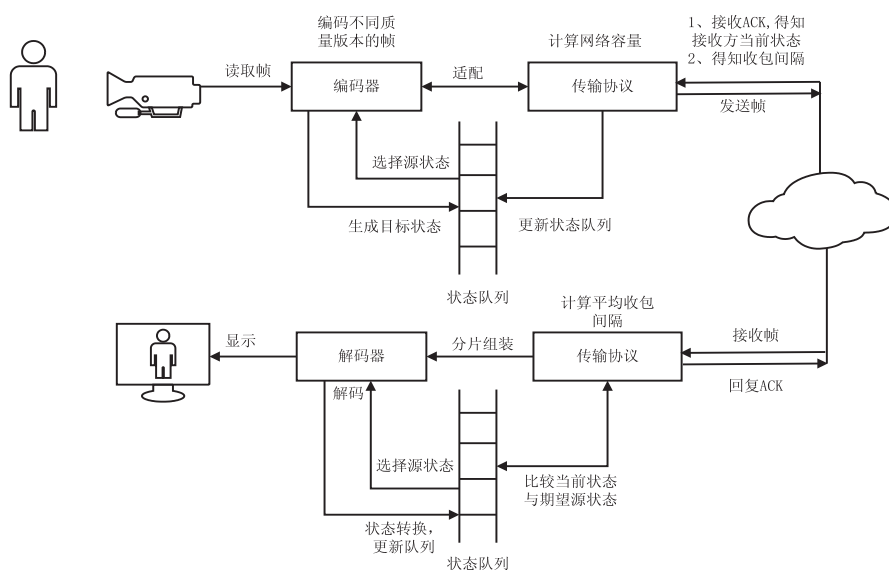


图1 紧耦合式视频编解码和传输

### 1.2 视频拼接技术

视频拼接一直以来都是一项非常具有挑战性的课题,尤其是对运动摄像头拍摄的视频进行实时拼接,由于特征点的检测匹配和相机运动路径的估计都需要大量的计算,目前仍然没有拼接效果和实时性俱佳的解决方案。Yoon 等<sup>[4]</sup>将帧划分成均匀的网格,通过计算相邻帧间对应网格点的像素差值大小来代替传统特征点的提取,而后通过光流法匹配估计相机运动,大幅度简化了相机运动估计,最后将所有的帧都重投影到第一帧的坐标系中。1 280 \* 720 分辨率的视频测试帧率

达到了 13 fps。但显然这种粗糙的运动估计和配准方式牺牲了较多的拼接质量。

对于运动摄像头拍摄好的视频进行离线拼接,目前已经有一些比较好的解决方案。其中 Lin 等人<sup>[5]</sup>提出的先稳定后拼接的方案,需要利用三维重建算法计算出相机的原始运动路径;Guo 等人<sup>[6]</sup>提出通过优化相机的运动路径,同时完成视频稳定和拼接的任务,不需要三维重建,是目前较为完整的 2D 方案,拼接效果也比较好。

对于摄像头固定的实时监控的场景,王文学<sup>[7]</sup>采

用了改进特征点的提取和匹配来加速拼接,其中陶荷梦<sup>[8]</sup>使用了 CUDA 框架进行了 GPU 加速,但均未真正达到实时要求。LIU 等<sup>[9]</sup>在首次计算好模板帧的拼接矩阵后,所有后续帧均按照同样的模板拼接,虽然实时性较好,但后续拼接效果显然不够理想。相比之下刘有科等<sup>[10]</sup>用固定间隔插帧和 GPU 加速,达到了 13 fps 的帧率,也不失为一种比较有效的近似实时的方式。He<sup>[11]</sup>受 DHW<sup>[12]</sup>算法的启发,将特征点分成不同的层次并预先对准,在此基础上构成背景并寻找拼接缝,而后通过检测拼接缝附近的变化来更新拼接缝。该方法的优点在于对视差比较鲁棒,但不管有无前景

变化,该算法在拼接每一帧时都要计算接缝处每个像素的梯度变化,使得单帧拼接时间增加较为明显,帧率在 12 fps 左右。

## 2 系统架构

设计并实现的系统主要功能模块有:

- (1) 多路视频的采集;
- (2) 紧耦合式的多路视频编解码和传输;
- (3) 多路视频的同步和实时拼接。

系统的整体架构如图 2 所示。

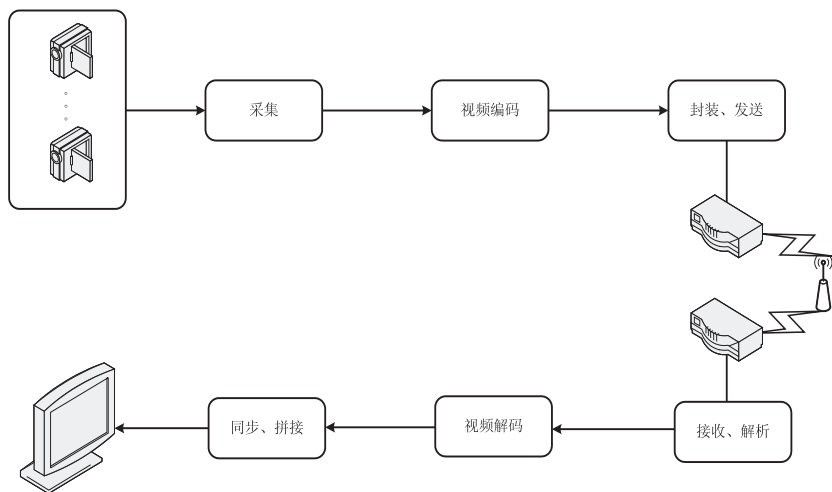


图 2 系统架构

## 3 系统实现

### 3.1 发送端

#### 3.1.1 视频采集

系统采用 V4L2 (Video For Linux 2) 框架进行视频采集。V4L2 框架是 Linux 内核提供给应用程序访问视频设备的统一接口,在嵌入式系统中应用非常广泛。在 Linux 中,摄像头被看作文件,位于/dev 目录下,用唯一的文件描述符来标识。

#### 3.1.2 视频编码

采用 FOULADI 团队<sup>[2]</sup>基于 libvpx 库进行改造实现的纯函数式编解码器,视频格式采用 VP8<sup>[13]</sup>。其中编码过程的各个环节,包括:变换、预测、自适应量化、环路滤波和和熵编码均按照 VP8 的官方文档<sup>[14]</sup>实现。唯一的区别是将编码器内部的“状态”暴露出来供外部使用,其编解码接口如下:

```
encode (state, image, quality) → frame
decode (state, frame) → (state', image)
```

其中 state 表示编码器的内部状态,在实现中是前一帧解码时使用的参考帧和熵编码概率表的哈希值。由于采用预测编码的编码器需要进行帧的重构,编码

器实际上包含了解码器的完整实现<sup>[15]</sup>。因此收发双方可以实现状态转换的同步。quality 参数用来设置同一帧的不同编码质量版本,在发送前选择与瞬时网络容量最适配的一个版本。

### 3.2 传输协议

监控视频数据的收发采用 UDP 协议,压缩后的每一帧被分成 MTU 大小(在实验中设为 1 400 字节)的分片,在 UDP 数据报的数据部分头部增加字段,设计应用层协议。设计的协议字段如表 1 所示。

表 1 视频分片数据包协议字段

字段	类型	作用
connection_id	uint16_t	标识不同的链接
source_state	uint32_t	源状态值
target_state	uint32_t	目标状态值
frame_no	uint32_t	帧编号
fragment_no	uint16_t	分片编号(帧内)
cap_time	uint64_t	采集时间
Fragments_in_this_frame	uint16_t	帧内分片数
payload	string	视频数据

接收端收到每一个分片后都要通过 ACK 包和发送方进行确认,并向发送方通告一些信息。ACK 包的

字段如表 2 所示。

表 2 ACK 包协议字段

字段	类型	作用
connection_id	uint16_t	标识不同的链接
frame_no	uint32_t	帧编号
fragment_no	uint16_t	分片编号(帧内)
avg_delay	uint32_t	包平均到达时延
current_state	int32_t	当前解码器状态
complete_states	deque<uint32_t>	接收方已完成解码状态列表

3.3 接收端

包,根据包头中的帧号、分片号等信息,处理流程如图 3 所示。

3.3.1 包处理流程

对于收到的来自于发送方的每一个视频分片数据

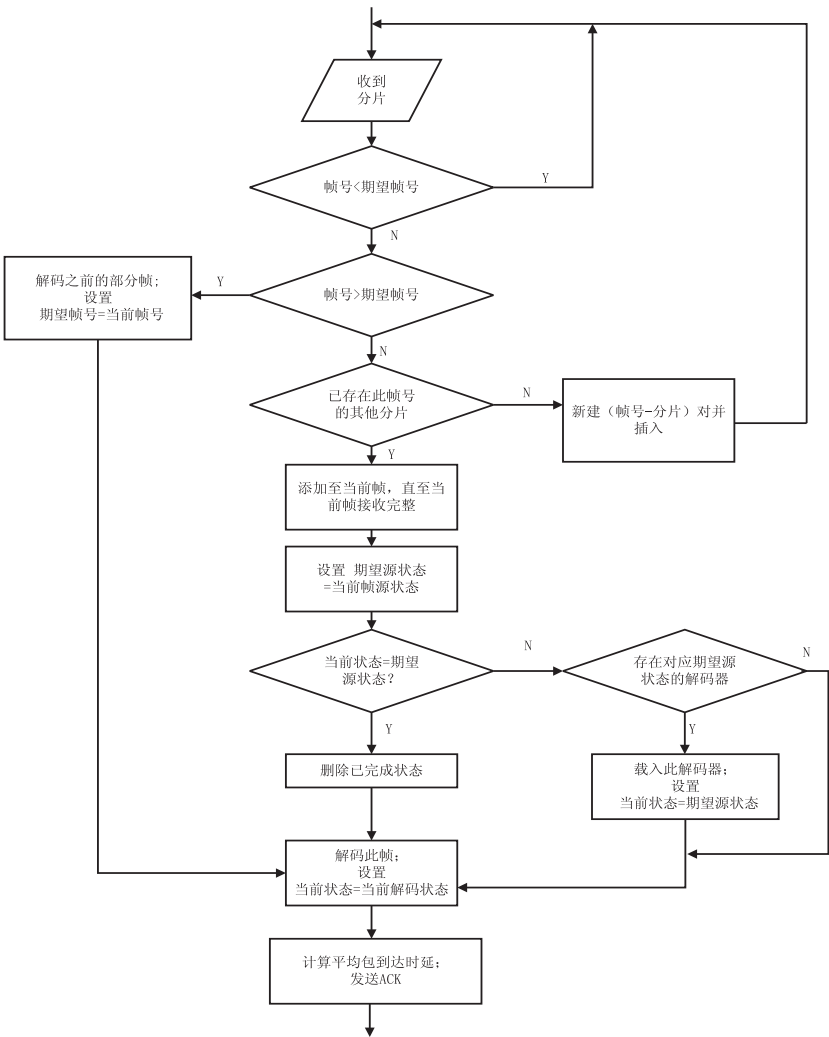


图 3 接收方包处理流程

3.3.2 多线程接收解码

系统为多发单收模型,需要在接收端同时接收来自不同摄像头的的数据。系统在接收端采用 OMP 并行框架,开启多个线程在不同的端口上接收来自不同摄像头的的数据,解码后存放到各自对应的接收缓存队列中。与接收线程并行的还有多路视频的同步线程和视频拼接线程,整体框架如算法 1 所示。

算法 1:接收端整体框架。

```
#pragma omp parallel num_threads(n)
{
#pragma omp sections
{
#pragma omp section/* 接收,解码,入队 */
{
receive_and_decode(port1, queue1);
}
#pragma omp section/* 接收,解码,入队 */
{
receive_and_decode(port2, queue2);
}
}
}
```

```

...../* 根据摄像头个数 */
#pragma omp section
{prepare1();}/* 出队至拼接区、同步 */
#pragma omp section
{prepare2();}/* 出队至拼接区、同步 */
.....
#pragma omp section
{stitch();}/* 拼接 */
}
}

```

### 3.4 多路视频同步

由于要对来自不同摄像头的不同视频进行拼接,必须保证待拼接的若干张帧相互之间的采集时间差很小,通常不能超过几十毫秒,否则可能匹配失败或者出现明显的拼接错位甚至出现鬼影。因此,在采集的时候为每一帧打上时间戳。此外,每个摄像头在同一时刻获取的系统时间可能本身就存在差值,这个差值一般可以认为是固定的,为此在每个摄像头开启之后,主动与接收端进行校时。最后由于各个摄像头开启的时间不一定相同,设置只有在所有的待拼接帧都已经到达接收端,并且相互之间的时间差不超过 50 ms 时才能进行拼接。由于要对共享数据进行操作,需配合使用锁 (unique lock)、互斥量 (mutex) 和条件变量

(condition variable) 来实现并发控制。算法步骤如下:

(1) 各摄像头打开后与接收端校时,接收端计算并保存时间差 time\_diff\_1, time\_diff\_2...;

(2) 根据摄像头个数设置标志位 pic\_n\_ok, 标识来自摄像头  $n$  的数据是否已经准备好;

(3) 接收来自各个摄像头的帧,在各自的同步线程等待数据没有准备好的信号,然后将接收队列头部的帧放入待拼接区域,并设置对应的标志位为 TRUE;

(4) 拼接线程等待所有的标志位均为 TRUE 后,判断各帧之间的采集时间差,若不满足要求则将时间过早的帧的标志位设置为 FALSE,以便取其下一帧。若满足条件则将所有标志位设置为 FALSE,通知其他线程准备下一轮拼接的帧,而后开始拼接。

### 3.5 实时视频拼接

一般来说,监控视频因为摄像机的位置是固定的,其拼接可以看作是对每一帧的图像拼接。经过 3.4 中的同步过程后,各待拼接帧的采集时间已经非常接近,当某两帧满足存在一定的重叠特征时就可以进行拼接。考虑到实时性的要求,没有采用一些拼接效果较好,但计算复杂度相对较高,拼接速度较慢的图像拼接算法如 APAP<sup>[16]</sup>等。系统基于 OpenCV 提供的图像处理和拼接的接口,定制了如图 4 所示的图像拼接流程。

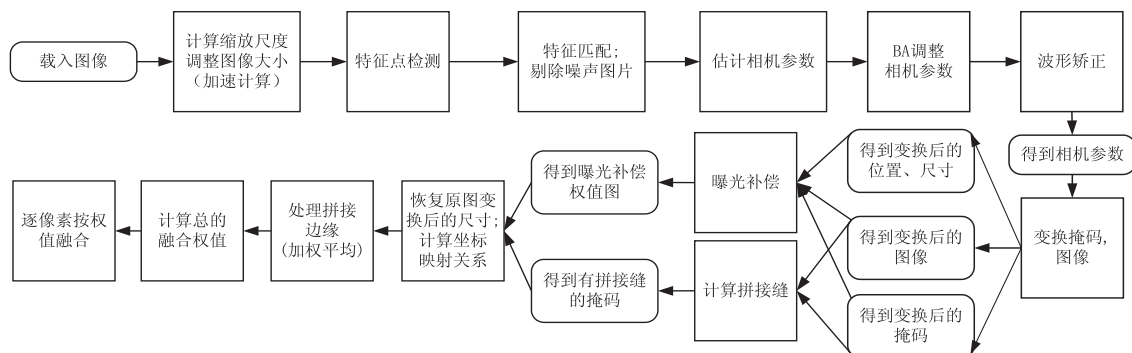


图 4 图像拼接流程

为了提升拼接速度,在特征点检测、特征点匹配、图像投影变换阶段均使用经过 GPU 加速的 OpenCV 库函数。在图像融合阶段,基于 CUDA 框架对整个拼接区域按像素并行加速填充。考虑到视频监控的特殊性质,有很大一部分情况下监控场景并不会发生剧烈的变化,尤其是在拼接缝处没有发生变化的情况,并不需要重新计算拼接参数。因此采用简单的固定间隔更新拼接参数的方法,设置合适的更新间隔,使得系统既能保证较高的帧率,也能在拼接缝处发生变化时,在短时间内修复,保持了良好的视觉效果。

## 4 系统测试

### 4.1 实验环境

系统在 Ubuntu 下用 C++ 开发,两台主机接入同一

个局域网。监控发送端开启多个摄像头,采集视频并编码后经局域网传输到监控端后实时拼接显示。实验环境如表 3 所示。

表 3 监控接收端实验环境

	发送端	接收端
处理器	Core i7-6820HQ	Core i7-8550U
内存	16 GB	8 GB
显卡	Quadro M2000M	集成显卡
操作系统	Ubuntu 16.04	Ubuntu 16.04
IDE	Clion 2019	Clion 2019
	Cmake3.5.1	Cmake3.5.1
其他	OpenCV2.4.13	OpenCV2.4.13
	CUDA 9.0.176	USB 摄像头 * 3

4.2 性能指标

为了验证系统采用的紧耦合式视频通信框架的有效性,在局域网环境下使用该系统点对点进行视频监控,并测量帧率;通过收发双方分别在采集帧和放入显

示队列之前记录时间点,计算对应同一帧从采集到显示的时延,取 100 帧的平均值;通过计算收发双方对应同一帧的 SSIM<sup>[17]</sup> 值来衡量视频质量,值越接近 1 质量越高,同样取 100 帧的平均值,结果如表 4 所示。

表 4 传输性能

分辨率	帧率/fps	时延/ms	SSIM
1 280 * 720	20.2	257.54	0.984 8
640 * 480	22.6	140.50	0.993 6

通过实验可以发现,系统能够获得较小的端到端时延和较高的视频质量。并且根据 FOULADI 团队的实验<sup>[2]</sup>,紧耦合式的视频架构对比 WebRTC 有较大的性能优势。

在视频拼接部分,该系统采用固定间隔更新拼接参数的方法,更新参数的帧称为关键帧。关键帧的拼接时间相对比较长,但是在计算好的参数和拼接缝上直接融合,在 GPU 加速下速度非常快。若设置较大的

间隔帧数,实时性能较好,但拼接质量较差;若设置较小的间隔帧数,则拼接质量较好,但视频不够流畅。又由于受到视频传输端接收帧率的限制,综合考虑实时性能和拼接质量,通过设置合适的间隔帧数,使得平均帧率略高于接收帧率。由于实验用摄像头的限制,分别测试了拼接 2 路 640 \* 480 的视频和 3 路 640 \* 480 的视频,拼接了 500 帧。分别计算平均拼接时间和总体平均拼接时间,如表 5 所示。

表 5 拼接性能

视频数量	关键帧拼接时间/ms	非关键帧拼接时间/ms	更新间隔帧数	平均拼接时间/ms
2	128.37	1.32	3	33.08
3	435.72	2.87	12	36.63

4.3 实验效果

在室内环境下分别对 2 路和 3 路分辨率为 640×

48 的视频进行了拼接,效果分别如图 5 和图 6 所示。

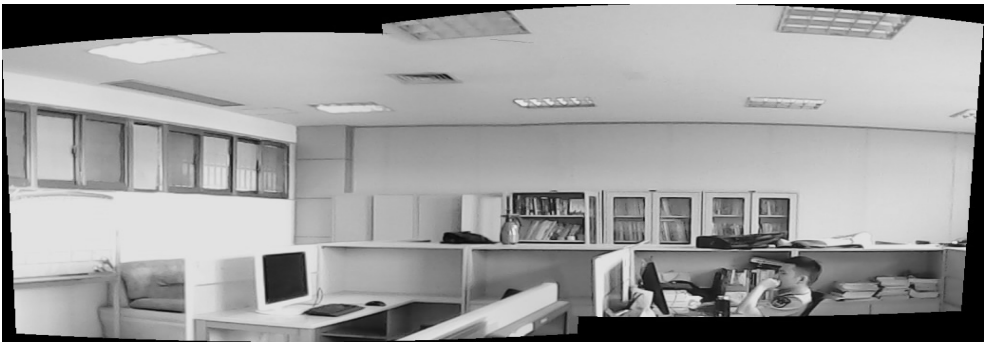


图 5 2 路视频拼接效果



图 6 3 路视频拼接效果

5 结束语

针对传统视频通信系统和视频监控中存在的问题,基于新型紧耦合式的视频通信架构,通过设计多路

视频间的传输协议和同步算法,并在接收端基于 OpenCV 的图像拼接算法定制多路视频实时拼接流程,实现了一种多摄像头远程全景实时监控系统。实验证明,通过设置适当的更新间隔帧数,该系统能够在

近实时的情况下获得较好的视频监控质量和较宽的监控视野。同样的框架也能应用于远程灾情防控指挥、战场态势感知等应用场景,具有一定的实际应用价值。

#### 参考文献:

- [1] 宋磊,黄祥林,沈兰荪. 视频监控系统概述[J]. 测控技术,2003,22(5):33-35.
  - [2] FOULADI S, EMMONS J, ORBAY E, et al. Salsify: low-latency network video through tighter integration between a video codec and a transport protocol [C]//15th USENIX symposium on networked systems design and implementation. Renton, USA: USENIX Association, 2018:267-282.
  - [3] JOHNSTON A B, BURNETT D C. WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web [M]//Digital codex LLC. Lilburn, GA, USA: [s. n.], 2012.
  - [4] YOON J, LEE D. Real-time video stitching using camera path estimation and homography refinement [J]. Symmetry, 2017,10(1):4-24.
  - [5] LIN K, LIU S, CHEONG L F, et al. Seamless video stitching from hand-held camera inputs [J]. Computer Graphics Form, 2016,35(2):479-487.
  - [6] GUO H, LIU S, HE T, et al. Joint video stitching and stabilization from moving cameras [J]. IEEE Transactions on Image Processing, 2016,25(11):5491-5503.
  - [7] 王文学. 实时视频拼接技术研究 [D]. 北京:北京工业大学, 2014.
  - [8] 陶荷梦. 基于多摄像头的实时视频拼接技术的研究与实现 [D]. 北京:北京工业大学, 2016.
  - [9] LIU H, TANG C, WU S, et al. Real-time video surveillance for large scenes [C]//Proceedings of the international conference on wireless communications and signal processing (WCSP). Nanjing, China: IEEE, 2011:1-4.
  - [10] 刘有科,高珏,谭松,等. 一种基于 CUDA 的快速宽视视频拼接的方法 [J]. 计算机技术与发展, 2015,25(1):15-18.
  - [11] HE Botao, YU Shaohua. Parallax-robust surveillance video stitching [J]. Sensors, 2015,16(1):7-18.
  - [12] GAO J, KIM S J, BROWN M S. Constructing image panoramas using dual homography warping [C]//Computer vision and pattern recognition 2011. Colorado, CO, USA, 2011:49-56.
  - [13] BANKOSKI J, WILKINS P, XU Y. Technical overview of VP8, an open source video codec for the web [C]//Proceedings of the 2011 IEEE international conference on multimedia and expo. Barcelona, Spain: IEEE, 2011:1-6.
  - [14] GLOVER M. VP8 data format and decoding guide [R]. San Francisco, USA: IETF, 2011.
  - [15] 蔡安妮. 多媒体通信技术基础 [M]. 北京:电子工业出版社, 2012.
  - [16] ZARAGOZA J, CHIN T, TRAN Q H, et al. As-projective-as-possible image stitching with moving DLT [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014,36(7):1285-1298.
  - [17] HORÉ A, ZIOU D. Image quality metrics: PSNR vs. SSIM [C]//20th international conference on pattern recognition. Istanbul, Turkey: IEEE, 2010:2366-2369.
- +++++
- (上接第 33 页)
- [16] FURTADO A, OLIVEIRA N, ANDRADE N. A case study of contributor behavior in Q&A site and tags; the importance of prominent profiles in community productivity [J]. Journal of the Brazilian Computer Society, 2014,20(1):1-16.
  - [17] YANG M, LEE D, PARK S, et al. Knowledge-based question answering using the semantic embedding space [J]. Expert Systems with Applications, 2015,42(23):9086-9104.
  - [18] CHEN L, ZHANG D, MARK L, et al. Understanding user intent in community question answering [C]//WWW 2012: 21st world wide web conference 2012. Lyon, France: Association for Computing Machinery, 2012:823-828.
  - [19] LIU Y, BIAN J, AGICHTEIN E, et al. Predicting information seeker satisfaction in community question answering [C]//International acm sigir conference on research and development in information retrieval. [s. l.]: Association for Computing Machinery, 2008:483-490.
  - [20] XIANG S, RONG W, SHEN Y, et al. Multidimensional scaling based knowledge provision for new questions in community question answering systems [C]//International joint conference on neural network. Vancouver, BC, Canada: IEEE, 2016:115-122.
  - [21] ZHANG M, CHEN Y. Link prediction based on graph neural networks [C]//Neural information processing systems. Montreal, Canada: Curran Associates Inc., 2018:5171-5181.
  - [22] ZHANG M, CUI Z, NEUMANN M, et al. An end-to-end deep learning architecture for graph classification [C]//AAAI conference on artificial intelligence. [s. l.]: AAAI, 2018:4438-4445.