

文章编号: 2095-2163(2019)04-0065-06

中图分类号: TP311.52

文献标识码: A

# 基于无人机的实时图传系统

严羽<sup>1</sup>, 王永众<sup>2</sup>, 杨来邦<sup>1</sup>

(1 浙江省林业智能监测与信息技术研究重点实验室(浙江农林大学), 杭州 311300;

2 杭州感知科技有限公司, 杭州 311300)

**摘要:** 民用领域上无人机的航拍系统, 不仅给专业工作者带来方便, 还给普通生活带来了乐趣。本文以无人机本身自带的图传系统为基础通过视频处理技术将视频流处理后通过 Android 手机 4G 网络实时上传到搭建好的服务器上, 同时也将无人机飞行参数一并上传, 然后通过 PC 端获取服务器的视频流数据以及飞行参数, PC 端通过软件能实时显示前方无人机拍摄的画面以及当前无人机的飞行状态。

**关键词:** 无人机; 实时图传; 视频传输

## Real-time image transmission system based on UAV

YAN Yu<sup>1</sup>, WANG Yongzhong<sup>2</sup>, YANG Laibang<sup>1</sup>

(1 Zhejiang Provincial Key Laboratory of Forestry Intelligent Monitoring and Information Technology

(Zhejiang A &amp; F University), Hangzhou 311300, China; 2 Hangzhou Ganzhi Software Company, Hangzhou 311300, China)

**[Abstract]** With the development of Unmanned Aerial Vehicle (UAV) technology, it shines brilliantly in various fields. Aerial photography system of UAV in civil field not only brings convenience to professional workers, but also brings fun to ordinary life. Based on the image transmission system of the UAV itself, this paper uses video processing technology to process the video stream and upload it to the established server in real time through the 4G network of mobile phone. Meanwhile, the flight parameters of the UAV are also uploaded. After that, the video streaming data and flight parameters of the server are acquired through the PC terminal. The PC terminal can display the pictures taken by the UAV in the front and the current flight status of UAV in real time through the software.

**[Key words]** Unmanned Aerial Vehicle; real-time image transmission; video transmission

## 0 引言

无人机出现在 1917 年, 早期的无人驾驶飞行器的研制和应用主要用作靶机, 应用范围主要是在军事上, 后来逐渐用于作战、侦察及民用遥感飞行平台。20 世纪 80 年代以来, 随着计算机技术、通讯技术的迅速发展以及各种数字化、重量轻、体积小、探测精度高的新型传感器的不断面世, 无人机的性能不断提高, 应用范围和应用领域迅速拓展。世界范围内的各种用途、各种性能指标的无人机的类型已达数百种之多。续航时间从 1 h 延长到几十个 h, 任务载荷从几 kg 到几百 kg, 这为长时间、大范围的遥感监测提供了保障, 也为搭载多种传感器和执行多种任务创造了有利条件。

近年来, 传统的卫星遥感系统, 已经不能够满足于高分辨率影像的需求, 所以在这样的情形之下, 无人机<sup>[1]</sup>因其自身所独有的优势被广泛应用, 成为了卫星遥感系统检测的补充。

无人机遥感技术可快速地对地质环境信息和过时的 GIS 数据库进行更新、修正和升级。为政府和相关部门的行政管理、土地、地质环境治理, 提供及时的技术保证。

随着国家改革开放的逐步深入, 经济建设迅猛发展, 各地区的地貌发生巨大变迁。现有的航空遥感技术手段已无法适应经济发展的需要。新的遥感技术为日益发展的经济建设和文化事业服务。以无人驾驶飞机为空中遥感平台的技术, 是为适应这一需要而发展起来的一项新型应用性技术, 能够较好地满足现阶段国家对航空遥感业务的需求, 对陈旧的地理资料进行更新。

目前低空无人机遥感技术在航拍、航测、国土、农业、环保、应急救援和科学研究等领域应用广泛<sup>[2-3]</sup>, 无人机遥感航空技术以低速无人驾驶飞机为空中遥感平台, 用彩色、黑白、红外、摄像技术拍摄空中影像数据; 并用计算机对图像信息加工处理。该系统在设计 and 最优化组合方面具有突出的特点,

**作者简介:** 严羽(1993-), 男, 硕士研究生, 主要研究方向: 计算机在农林业上的应用; 王永众(1985-), 男, 硕士, 工程师, 主要研究方向: 林业信息系统; 杨来邦(1983-), 男, 硕士, 工程师, 主要研究方向: 计算机在农林业上的应用。

收稿日期: 2018-12-18

哈尔滨工业大学主办 ◆ 学术研究与应用

是集成了遥感、遥控、遥测技术与计算机技术的新型应用技术。

## 1 无人机飞行参数获取与图传系统

### 1.1 软硬件平台简介

- (1) 大疆系列无人机一台;
- (2) 搭载 Android 系统手机一台(要带有 4 G 流量卡);
- (3) X86 系列多核处理服务器一台,该服务器要固定 IP 同时要有大容量带宽;
- (4) 通用计算机一台作为接收终端使用。

本文无人机采用大疆精灵系列无人机。该无人机搭载了 6 个视觉传感器、主相机、2 组红外传感器、1 组超声波传感器、GPS/GLONASS 双模卫星定位系统、IMU 和指南针双冗余传感器。通过无人机的无线传输设备连接遥控器,遥控器一端与手机相连。无人机将拍摄到的图像画面以及飞行参数通过无线传回到遥控器,手机从遥控器获取数据并通过 4G 网络传递到已经搭建好的服务器平台,PC 端通过网络可以实时获取无人机拍摄的影像以及各飞行参数。

### 1.2 H264 视频流简介

H264 是新一代的编码标准,以高压缩、高质量和支持多种网络的流媒体传输著称<sup>[4]</sup>。在 H264 协议里定义了 3 种帧:完整编码的帧叫 I 帧;参考之前的 I 帧生成的只包含差异部分编码的帧叫 P 帧;参考前后的帧编码的帧叫 B 帧。H264 采用的核心算法是帧内压缩和帧间压缩,帧内压缩是生成 I 帧的算法,帧间压缩是生成 B 帧和 P 帧的算法。3 种帧的说明:

#### 1.2.1 I 帧

I 帧:帧内编码帧。为关键帧,是一帧画面的完整保留;解码时只需要 I 帧就可以完成。

I 帧特点:

- (1) 解码时仅用 I 帧的数据就可重构完整图像;
- (2) I 帧是 P 帧和 B 帧的参考帧;
- (3) I 帧是基础帧即第一帧,在一组中只有一个 I 帧;
- (4) 帧所占数据的信息量比较大。

#### 1.2.2 P 帧

P 帧:前向预测编码帧。P 帧表示的是这一帧跟之前的一个关键帧(或 P 帧)的差别,解码时需要用之前缓存的画面叠加上本帧定义的差别,生成最终画面。

P 帧特点:

- (1) P 帧是 I 帧后面相隔 1~2 帧的编码帧;
- (2) P 帧采用运动补偿的方法传送其与前面的

I 或 P 帧的差值及运动矢量(预测误差);

(3) 解码时必须将 I 帧中的预测值与预测误差求和后才能重构完整的 P 帧图像;

(4) P 帧属于前向预测的帧间编码。其只参考前面最靠近其的 I 帧或 P 帧;

(5) 由于 P 帧是参考帧,可能造成解码错误的扩散。

#### 1.2.3 B 帧

B 帧:双向预测内插编码帧。是双向差别帧,也就是 B 帧记录的是本帧与前后帧的差别。要解码 B 帧,不仅要取得之前的缓存画面,还要解码之后的画面,通过前后画面与本帧数据的叠加取得最终的画面。

B 帧特点

(1) B 帧是由前面的 I 或 P 帧和后面的 P 帧来进行预测的;

(2) B 帧传送的是其与前面的 I 或 P 帧和后面的 P 帧之间的预测误差及运动矢量;

(3) B 帧压缩比最高,因为其只反映参考帧间运动主体的变化情况,预测比较准确;

H264 功能分为 2 层:视频编码层面(VCL)和网络抽象层面(NAL)。其中,前者负责有效表示视频数据的内容,而后者负责格式化数据并提供头信息,以保证数据适合各种信道和存储介质上的传输<sup>[5]</sup>。因此每帧数据就是一个 NAL 单元(SPS 与 PPS 除外)。在实际的 H264 数据帧中,往往帧前面带有 00 00 00 01 或 00 00 01 分隔符,一般来说编码器编出的首帧数据为 PPS 与 SPS,接着为 I 帧。

通过 NALU 类型人们可以判断帧类型见表 1。

表 1 帧类型

Tab. 1 Frame type

nal_unit_type	NAL 类型	C
0	未使用	
1	不分区、非 IDR 图像中的片	2,3,4
2	片分区 A	2
3	片分区 B	3
4	片分区 C	4
5	IDR 图像中的片	2,3
6	补充增强信息单元(SEI)	5
7	序列参数集	0
8	图像参数集	1
9	分界符	6
10	序列结果	7
11	码流结束	8
12	填充	9
13..23	保留	
24..31	未使用	

以 00 00 00 01 分割之后的下一个字节就是 NALU 类型,将其转为二进制数据后:

- (1)第 1 位禁止位,值为 1 表示语法出错;
- (2)第 2~3 位为参考级别;
- (3)第 4~8 位为 nal 单元类型。

图 1 所示是从无人机视频流中截取的视频文件,以 16 进制的格式打开的视频流数据。

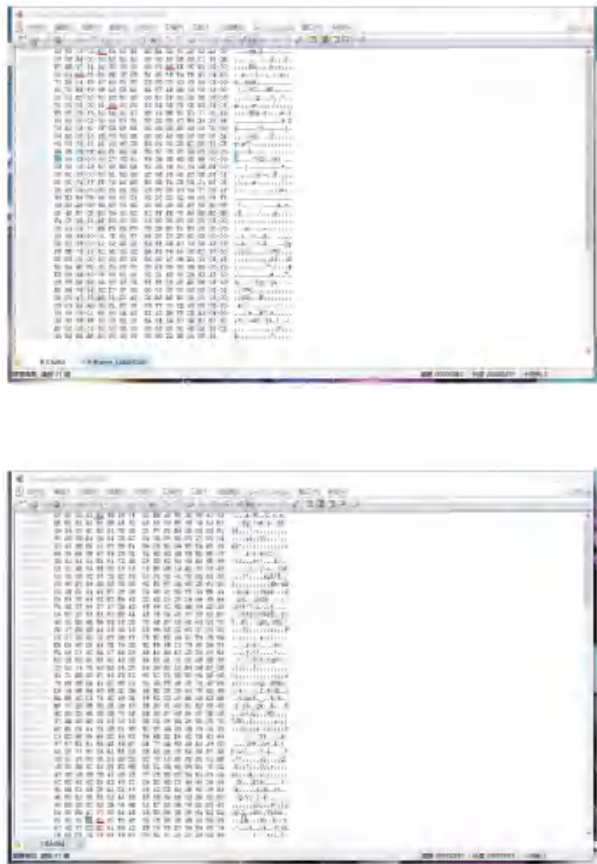


图 1 视频流文件

Fig. 1 Streaming video files

从划红线处可以看出以 00 00 00 01 分割之后有 67 68 65 61

其中,0x67 的二进制码为:0110 0111

4-8 位 00111,转为十进制 7,参考表 1,7 对应序列参数集 SPS。

其中,0x68 的二进制码为:0110 1000

4-8 位 01000,转为十进制 8,参考表 1,8 对应图像参数集 PPS。

其中,0x65 的二进制码为:0110 0101

4-8 位 00101,转为十进制 5,参考表 1,5 对应 IDR 图像中的片(I 帧)。

其中,0x61 的二进制码为:01100001

4-8 位 00001,转为十进制 1,参考表 1,1 对应

非 IDR 图像的片即非 I 帧,因为该帧在 I 帧后面则根据 P 帧特点可以判断出是 P 帧而不可能是 B 帧。

H264 的 SPS 和 PPS 串,包含了初始化 H264 解码器所需要的信息参数,包括编码所用的 profile、level、图像的宽和高,deblock 滤波器等。SPS 中的信息至关重要。如果其中的数据丢失或出现错误,那么解码过程很可能会失败,分离 H264 码流时,需要首先写入 SPS 和 PPS,否则会导致分离出来的数据没有 SPS、PPS 而无法播放。

### 1.3 H264 视频流处理

手机端实时从遥控器获取的 H264 视频流是一段非常简短且由多个或一个帧组成的流文件。如图 1 所示,所做的工作就是把每一帧给分离出来。处理的方法是组建一个通道,该通道的作用是一端一直获取传输过来的 H264 视频流且将该视频流的每一帧分离出来,通过管道传输到另一端。而另一端封装成 flv 的格式通过网络发送到服务器,为保证视频质量以及实时性,上述过程都是并行化执行。因为实时图传对运行效率有较高的要求,因此上述涉及到大量运算的过程都是在 c 环境中运行的。

### 1.4 RTMP 协议

本地视频流数据需要通过流媒体协议(如 RTMP、HTTP、UDP、TCP、RTP 等)推送至服务器,而本文以 RTMP 协议推送为主。

RTMP 是 Real Time Messaging Protocol(实时消息传输协议)的首字母缩写。该协议基于 TCP,是一个协议族,包括 RTMP 基本协议及 RTMPT/RTMPS/RTMPE 等多种变种<sup>[6]</sup>。RTMP 是一种设计用来进行实时数据通信的网络协议,主要用来在 Flash/AIR 平台和支持 RTMP 协议的流媒体/交互服务器之间进行音视频和数据通信。支持该协议的软件包括 Adobe Media Server/Ultrant Media Server/red5 等。

通过从官网下载的开源函数库编译后的 librtmp 提供的 API 来进行推流,其中的 RTMP 协议已经封装在相关函数中。librtmp 提供的 API 中只需要常见的几个 API 就可以将数据流推送到服务器,所需要的 API 如下:

RTMP\_Init()//初始化结构体

RTMP\_Free()

RTMP\_Alloc()

RTMP\_SetupURL()//设置 rtmp server 地址

RTMP\_EnableWrite()//打开可写选项,设定为

## 推流状态

RTMP\_Connect()//建立 NetConnection  
 RTMP\_Close()//关闭连接  
 RTMP\_ConnectStream()//建立 NetStream  
 RTMP\_DeleteStream()//删除 NetStream  
 RTMP\_SendPacket()//发送数据  
 推流函数流程如图 2 所示。

### 1.5 基于 RTMP 协议的 H264 推流器

通过 RTMP 协议将 H264 格式的视频流发送到服务器中实现推流效果具体程序流程如图 3 所示。

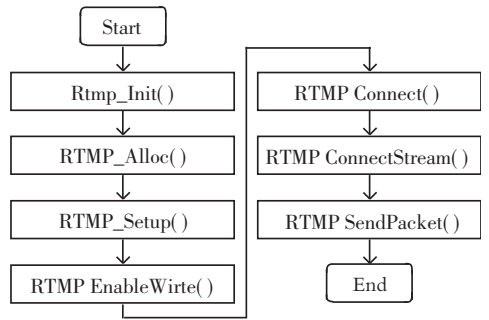


图 2 推流流程图

Fig. 2 The flow chart of push flow

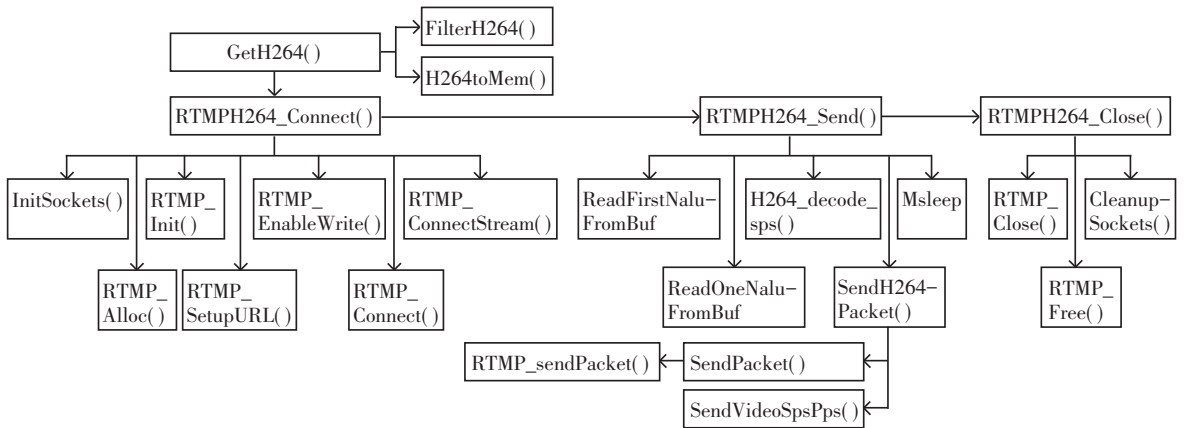


图 3 基于 RTMP 协议的 H264 推流器

Fig. 3 H264 pusher based on RTMP protocol

整个程序框图包含 4 个接口函数：

GetH264() : 获取 H264 视频流数据。

RTMPH264\_Connect() : 建立 RTMP 连接。

RTMPH264\_Send() : 发送数据。

RTMPH264\_Close() : 关闭 RTMP 连接。

按照先后顺序依次调用上述函数可以推流 H264 格式视频流到服务器。上述 4 个接口函数中又包含以下函数,这些函数主要功能如下:

GetH264() 中包含以下函数:

FilterH264() : 对原始视频流过滤。

H264toMem() : 对接收到的 H264 视频流以队列的形式放入内存缓存中。

RTMPH264\_Connect() 中包含以下函数:

InitSockets() : 初始化 Socket。

RTMP\_Alloc() : 为结构体“RTMP”分配内存。

RTMP\_Init() : 初始化结构体“RTMP”中的成员变量。

RTMP\_SetupURL() : 设置输入的 RTMP 连接的 URL。

RTMP\_EnableWrite() : 发布流的时候必须要使用。如果不使用则代表接收流。

RTMP\_Connect() : 建立 RTMP 连接,创建一个 RTMP 协议规范中的 NetConnection。

RTMP\_ConnectStream() : 创建一个 RTMP 协议规范中的 NetStream。

RTMPH264\_Send() 中包含以下函数:

ReadFirstNaluFromBuf() : 从内存中读取出一个 NAL 单元。

ReadOneNaluFromBuf() : 从内存中读取出一个 NAL 单元。

H264\_decode\_sps() : 解码 SPS, 获取视频的宽、高、帧率信息。

SendH264Packet() : 发送一个 NAL 单元。

SendH264Packet() 中包含以下函数:

SendVideoSpsPps() : 如果是关键帧,则在发送该帧之前先发送 SPS 和 PPS。

SendPacket() : 组装一个 RTMPPacket, 调用 RTMP\_SendPacket() 发送出去。

RTMP\_SendPacket(): 发送一个 RTMP 数据 RTMPPacket。

RTMPH264\_Close()中包含以下函数:

RTMP\_Close(): 关闭 RTMP 连接。

RTMP\_Free(): 释放结构体“RTMP”。

CleanupSockets(): 关闭 Socket。

### 1.6 服务器技术

由于视频的实时性对服务器提出较高的要求,数据传输时要尽量避免延迟所以选用较为成熟的 nginx web 服务器来接收发送过来的视频流。

Nginx("engine x")是一款由俄罗斯程序设计师 Igor Sysoev 所开发高性能的 Web 和反向代理服务器,也是一个 IMAP/POP3/SMTP 代理服务器<sup>[7]</sup>。该服务器具有以下优势:

(1) Nginx 使用基于事件驱动架构,使得其可以支持数以百万级别的 TCP 连接。

(2) 高度的模块化和自由软件许可证使得第三方模块层出不穷。

(3) Nginx 是一个跨平台服务器,可以运行在 Linux、FreeBSD、Solaris、AIX、Mac OS、Windows 等操作系统上。

最重要的一点是该服务器完全开源,可以通过自己的需要专门定制自己的需求同时该服务器具有的模块化更易于开发和维护,这些优秀的设计带来的极大的稳定性能,使得大量从事软件开发的人员使用和维护之。

Nginx 带有的 RTMP 模块使得配置该服务器变得非常简单且高效,因此配置时只要修改 nginx.conf 配置文件就行,在配置文件中主要加入以下程序:

```
rtmp {
server {
listen 1935;
application rtmplive {
live on;
record off;
}
}
```

RTMP 表示的是网络传输 RTMP 协议而非 HTTP 协议,其中 1935 是监听端口,服务器一直监听该端口若有数据进来则从该端口拉取数据,获取的数据可以存放在硬盘或内存中,本文涉及到延迟问题则直接存放在内存中。

### 1.7 程序流程图及主要代码

具体的程序流程如图 4 所示。

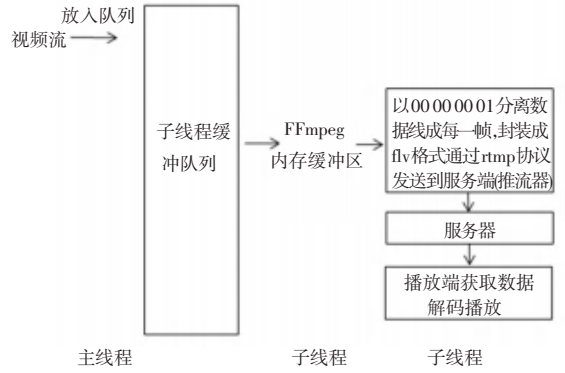


图 4 程序主要过程

Fig. 4 Main process

上述流程是无人机视频流部分,下面是关于无人机相关参数获取过程,其中包括:高度、速度、经纬度、飞行参数等。

Aircraft aircraft = ( Aircraft ) DJISDKManager.getInstance().getProduct();

if ( aircraft! = null ) { aircraft. getFlight

Controller().setStateCallback( new

FlightControllerState.Callback() }

Public void onUpdate( FlightController State flightControllerState ) {

if ( flightControllerState. getGPSSignal Level() ! = null ) {

GPSSignalLevel gpsSignalLevel = flight ControllerState.

getGPSSignalLevel();

.... }

}

}); }

函数 onUpdate() 是获取无人机飞行时的相关参数包括 GPS 信号强度、当前飞行的高度、速度、经纬度、飞行时的姿态角等。

public String SendData ( String url, Wurenji express, String contentType) {

try {

vehicle = new JSONStringer().object().key(" Wurenji").object()

.key(" guid").value( express.guid)

.key(" Site").value( express.Site)

.key(" Pitch").value( express.Pitch)

.key(" Yaw").value( express.Yaw)

.key(" Roll").value( express.Roll)

....

```

.endObject();
StringEntity entity = new StringEntity( vehicle.
toString(), encode);
request.setEntity(entity);
DefaultHttpClient httpClient = new DefaultHttpClient
();
HttpResponse response = httpClient.execute
(request);
// 判断是否成功
if (response.getStatusLine().getStatusCode() =
= 200) {
strResp = EntityUtils.toString(response.getEntity
()), encode);
}
} catch (Exception e) {e.printStackTrace();} return
strResp;}

```

该函数的主要作用是将无人机的相关参数发送到服务器,其中采用 post 的方式将这些参数以 json 的格式封装在请求头上,通过 http 协议发送到服务器后端并从后端返回一个响应码,若是 200 则表示发送成功,服务器后端已经接收到数据了。

获取无人机实时视频流和相关参数后,可以把数据回传给后端服务器处理,系统流程如图 5 所示。

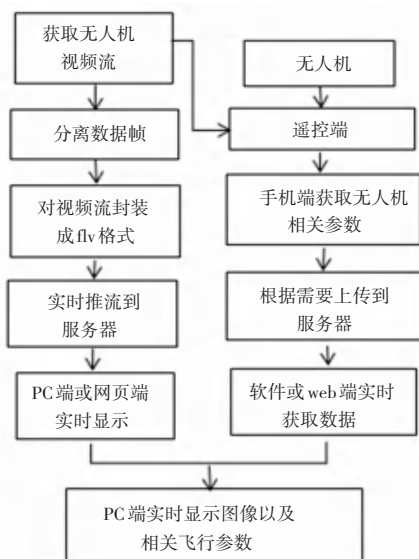


图 5 系统主要流程

Fig. 5 Main process of the system

图 5 所示为整个无人机结合服务器从前端即林区获取数据实时传到服务器,PC 端软件从服务器实时取数的过程。图 6 所示是 PC 端无人机视频流及参数获取界面图。



图 6 图传界面

Fig. 6 The system interface

## 2 结束语

本文以大疆无人机为平台,研究了图像的实时图传技术。但有些不足之处:因本文是在实验室环境中进行的,测试时采用的是 720p 分辨率的码率。该码率在 4G 网络下有较好的效果,若在 4G 信号差的环境下,并不能满足该码率传输。此时则会造成黑屏、模糊等现象。因此在网络环境差的情况下如何降低视频码率来保证视频的传输问题有待进一步研究解决。

## 参考文献

- [1] 王文飞. 无人机技术发展趋势[J]. 国际航空,1994(6):23-25.
- [2] 雷添杰,宫阿都,李长春,等. 无人机遥感系统在低温雨雪冰冻灾害监测中的应用[J]. 安徽农业科学,2011,39(4):2417-2419,2423.
- [3] BRECKENRIDGE R P, DAKINS M, BUNTING S, et al. Using unmanned helicopters to assess vegetation cover in sagebrush steppe ecosystems[J]. Rangeland Ecology & Management, 2012, 65(4):362-370.
- [4] 温兴宇. H.264 视频压缩算法应用研究[J]. 北京:中国科学院大学(中国科学院长春光学精密机械与物理研究所),2011.
- [5] 林海东,余强. 基于 H264 的网络视频监控系统设计与实现[J]. 西华大学学报(自然科学版),2014,33(2):22-26.
- [6] 雷霄骅,姜秀华,王彩虹. 基于 RTMP 协议的流媒体技术的原理与应用[J]. 中国传媒大学学报(自然科学版),2013,20(6):59-64.
- [7] 陆亮. 基于 Nginx 技术的直播平台[J]. 视听界(广播电视技术),2018(3):22-30.