

基于 DataSocket 技术传输图像的速率研究

白金鹏

(上海理工大学光电信息与计算机工程学院 上海 200093)

摘要: DataSocket 是 LabVIEW 中的一种实时传输技术,具有传输速率快、编程简单、支持多客户端等优点,能够用于图像传输。对 DataSocket 传输不同图像的速率进行研究,设计图像传输程序,分别运行于服务器端 PC、客户端 PC,由于客户端接收一帧图像时间长于服务器端发送一帧图像的时间,服务器端在发送图像后需要延时若干 ms 等耐客户端响应,延时越短则传输速率越快,每秒可传输的帧数越多,测得发送时间和最短延时后可经过换算得到不同图像在该系统中的传输速率。研究表明,不同格式图像使用 DataSocket 传输的速率相差较大,bmp 格式经压缩后在 300M 局域网内最高可达 31MB/s;同种格式但不同大小的图像传输速率相近,且较小图像传输速率略快。

关键词: DataSocket;图像传输;速率;LabVIEW

中图分类号: TP919.8 **文献标识码:** A **国家标准学科分类代码:** 510.4030

The speed research of transferring images through DataSocket

Bai Jinpeng

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: DataSocket is a kind of real-time transfer technology in LabVIEW, which has some advantages such as fast speed, easy-to-program, supporting multi-client, and can be used to transfer images. A set of LabVIEW programs including DataSocket were designed to research the speed of transferring images through DataSocket from server PC to client PC. Since receiving one image took more time than sending one, server must delay several milliseconds after sending out one image which meant that the shorter delay time, the faster transfer speed was. Once the shortest delay was recorded, the highest transfer speed could be figured out. According to the result, transfer speed difference is related to image format, the fastest one can reach 31 MB/s after compressing within 300 M bandwidth local area network; Images with the same format but different sizes have similar transfer speed and smaller ones' is a little faster than big ones'.

Keywords: DataSocket; image transfer; speed; LabVIEW

1 引言

尽管有多种通讯技术可以实现不同的应用程序之间以及网络上不同的设备之间传输数据,如 TCP/IP 等,但大部分编程复杂,不支持多客户端,甚至不能实现实时传输。DataSocket(以下简称 DS)技术的出现,使用户只需要知道数据源和数据宿及需要交换的数据就可以直接进行高层应用程序的开发,实现高速数据传输,而不必关心底层的实现细节,从而简化通信程序的编写过程、提高编程效率^[1]。已有众多开发人员将 DS 技术用于图像的传输,但对其传输速率的研究较少,本篇将对此速率进行研究。设计 DataSocket 图像传输程序,将现有的图像使用 DS 技术从服务器传输至客户端,重复传输 100 次,测量不同图像在该

传输系统中的传输速率。根据 DS 传输不同图像的速率,开发人员在设计图像传输系统时,可根据自己的需求决定是否采用 DS 技术。

2 DataSocket 技术传输图像的原理

DataSocket(以下简称 DS)是 National Instrument(以下简称 NI)公司独有的实时传输技术,它对 TCP/IP 协议进行了高度封装,专门面向测量和自动化应用领域,用于共享和发布实时数据^[2]。虽然 TCP/IP 等技术支持不同应用程序间的数据共享,但不能用来传输动态数据,DS 专门用来传输和发布动态数据^[3],DS 能有效地支持本地计算机上不同应用程序对特定数据的实时共享,以及网络内不同计算机的多个应用程序之间的数据交互^[4]。它是一种用于

动态数据传输的网络编程与通信工具,它为底层通信协议提供统一的编程接口^[5]。

DS传输系统包括服务器端、客户端、DS Server三部分,服务器端将数据发布到DS Server中的指定区域,客户端通过URL(统一资源定位符)与DS Server建立连接,从DS Server读取数据,完成传输过程^[6]。URL由数据传输协议、存储数据资源的主机IP地址、资源文件名组成^[7]。DS技术也包括三个部分:1)DS API:DS API提供了多语言访问、多种数据类型的单一接口^[8]。2)DS Server:服务器端将数据发送到DS Server中,客户端远程读取DS Server中数据完成传输过程。3)DS Server Manager:管理DS Server,如设置缓冲区大小、最大客户端数、用户组权限等。它可以和测控应用程序安装在同一台计算机上,也可以分装在不同的计算机上^[9]。

在LabVIEW中,使用Flatten to String.vi(LabVIEW中的函数称为vi)对图像进行扁平化,得到二进制字符串,然后通过写入DS.vi将二进制字符串写入DS Server中;在接收端,使用读取DS.vi从DS Server中接收字符串,然后将接收到的字符串输入Unflatten from String.vi进行去扁平化,将其还原成图像,如图1所示:

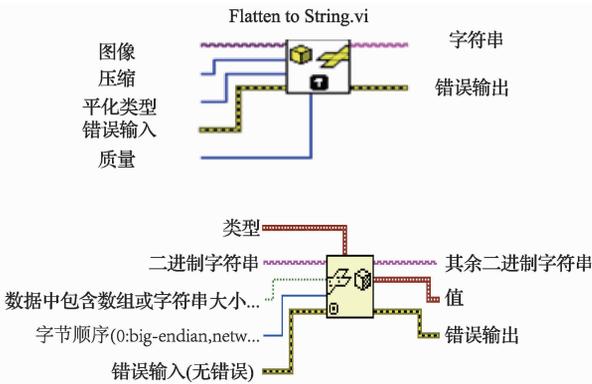


图1 Flatten to String.vi和Unflatten from String.vi

3 研究方案

图像来自图像采集设备,实现图像的实时传输,有必要对图像的来源进行研究。图像采集设备分两大类:1)通过图像采集卡接收来自视频输入端的模拟视频信号,对该信号进行采集、量化成数字信号,传输到计算机;2)相机本身带有数字化部件可以直接将数字图像通过计算机端口传到计算机。数字图像信号传输到计算机后,可以保存在本地,也可以将数字信号实时传输到另一台设备,保存在远程设备,这就是通过服务器/客户端模式,通过两台计算机、路由器构建的局域网环境,将本地图像数据实时传输到远程客户端。传输的数字信号其实是二进制的字符串,无论在

本地还是远程客户端,都需要将二进制字符串还原成图像,并保存下来。

本文研究的是DS传输图像的速率问题,若采用相机的实时图像,每次采集的图像都是不同的,测试的结果相差也比较大。因此,在本文中将有图像扁平化成二进制字符串,既可以完成传输,又可以模拟真实相机的数据流,传输给远程客户端。DS是在TCP的基础上封装而成,其性能随网络带宽、流量等不同而有所不同,但主要取决于网络的带宽,本系统采用300 M带宽路由器组成局域网,两台PC的主频均为2.6 GHz,传输系统原理图如图2所示。

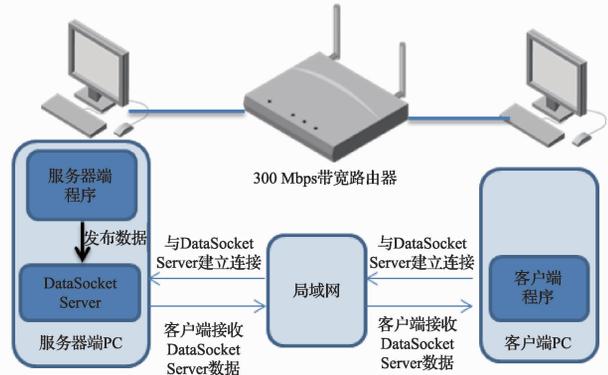


图2 DataSocket图像传输系统原理图

研究方案:1)客户端请求与DS Server建立连接,成功后,服务器端读取一幅图像发布到DS Server中,客户端与DS Server建立连接后,客户端读取DS Server中数据完成一次接收。2)有时网络拥塞会使数据传送周期变长,可以在程序的循环中设置等待时间(ms),通过LabVIEW中的等待.vi的设置,不仅保证传输的可靠性也可以模拟真实相机的帧率(例如:设置等待时间为20 ms,则在服务器端,每秒最多写入50帧图像,模拟真实相机每秒传输的50帧图像),等待.vi是一种定时函数,程序执行的时间不到设定的等待时间,则以等待时间为准;若程序执行的时间超过等待时间,以程序实际执行时间为准,等待.vi无效,等待时间可以理解为程序的最短执行时间。3)将同一幅图像进行重复操作100次,不断地调整等待时间,找到能够保证客户端收到全部100张图像的最短等待时间,根据此最低等待时间,可计算出不同格式、大小的图像在本系统内传输速率。

4 程序设计

使用LabVIEW程序框图面板中基本的编程模块(包括数组、结构、比较等)、前面板中基本的Express模块(包括数值输入、文本输入、数值显示等)以及前面板中的Vision模块和程序框图面板中的视觉与运动模块进行编程^[10]。DS传输系统的程序设计如图3、4所示:

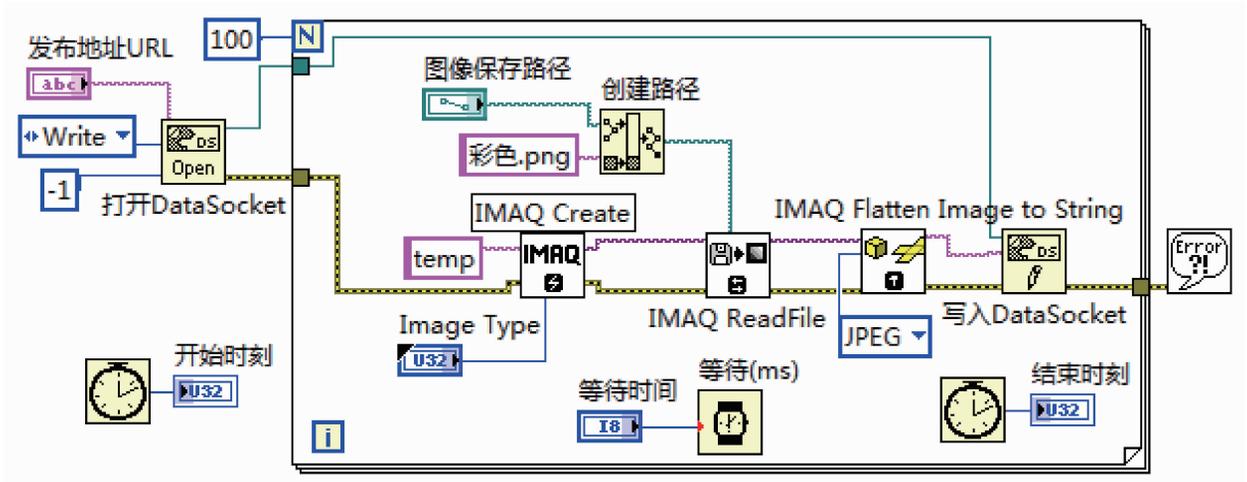


图3 服务器端程序

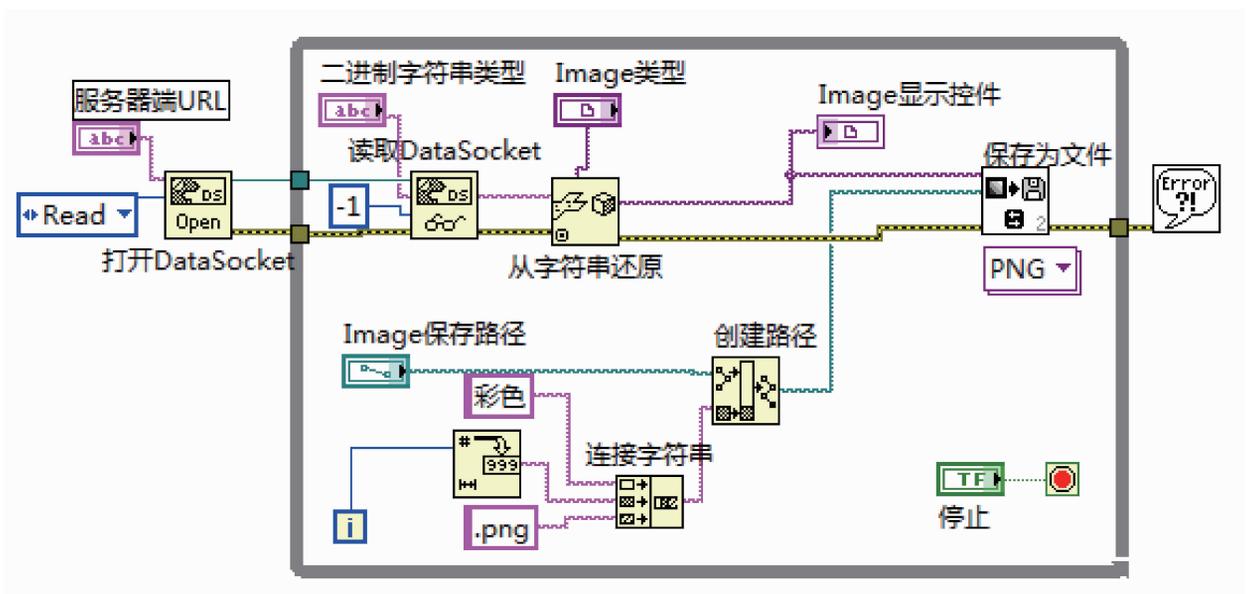


图4 客户端程序

程序说明:在图3中:1)“发布地址URL”指定DS Server所在设备的IP地址、数据缓冲区名称,如dstp://192.168.16.100/image中dstp(DataSocket Transfer Protocol)表示采用dstp传输协议,192.168.16.100是DS Server所在设备的IP地址,image为数据缓冲区变量名,通过此URL和打开DataSocket.vi创建一个数据缓冲区;2)“IMAQ Create.vi”创建一帧图像的副本文件,系统实际上是将此副本发送出去,不是直接发送原图像。副本的意义在于可以将原图像进行大小、格式、色彩等转换,将转换后的图像发送,不改变原图像;3)“IMAQ Flatten Image to String”将图像扁平化处理,转为二进制字符串,在客户端将二进制字符串还原成图像;将图像压缩成JPEG格式,虽然压缩图像占用一定的时间,但可以节省较多的传输时间;

4)“等待.vi”服务器端写入DS Server的速率要大于客户端从DS Server中读取的速率,用“等待.vi”来降低服务器写入DS Server的速率,消除两边的速率差异,否则,DS Server中缓冲区没有被及时读取造成数据丢失;5)结束时刻和开始时刻之差就是服务器写入100帧图像的时间,它包括读取、压缩、写入DS Server的所有时间。

在图4中:1)读取DataSocket.vi读取二进制字符串,在输入端指定它的输出类型是“二进制字符串”;2)“从字符串还原.vi”将字符串还原成图像,它的输入端也是通过“Image类型”控件指定它的输出为Image类型;3)“保存为文件.vi”将图像保存在本地磁盘;4)左侧方框内的i是While循环执行的次数,它用于文件命名。

程序运行过程如下:打开DS Server—>先运行DS客

户端再运行 DS 服务器端—>服务器端从本机读取图像并创建一个副本文件,命名为 temp—>将 temp 压缩成 jpg 格式然后平化成字符串—>字符串写入 DS Server,等待 N 毫秒开始下一次 for 循环写入下一帧图像—>客户端从 DS Server 接收二进制字符串—>去平化处理,将字符串

还原成图像—>若客户端收到新的数据则开始下一次 While 循环,直到第 100 帧图像接收完成。5 研究数据记录及分析同一帧图像连续传输 100 次,保存在同一目录下,分别命名为“接收的图片 1, 2, 接收的图片 100”,各种规格的图片传输过程的数据记录见表 1:

表 1 使用 DataSocket 传输各种图像的时间、速率

图像信息	RGB (U32)	RGB(U32)	RGB(U32)	RGB (U32)	Gray (U8)	Gray (U8)	Gray (U8)
类型、格式	Jpg,485×467	Jpg,3328×1 872	Png,296×260	Bmp,367×301	Jpg,829×632	Png,901×651	Bmp,570×513
像素大小/kB	57	480	181	325	41	106	288
发送时间/ms	1 358	12 280	1 129	830	1 246	2 786	859
每帧时间/ms	13.58	122.80	11.29	8.30	12.46	27.86	8.59
等待时间/ms	14	122.80	24	10	12.46	27.86	10
延时/ms	0.42	0	12.71	1.7	0	0	1.41
传输帧率/fps	71.4	8.1	41.6	100	80.2	35.8	100
传输速率/MB·s ⁻¹	3.97	3.81	7.35	31.74	3.21	3.71	28.13

注: 1)RGB (U32)格式彩色图,每个像素占用 32 bit; Gray(U8) 黑白图,每个像素占用 8 bit;2)发送时间:服务器向 DS Server 中连续写入 100 帧图像的最短时间;3)等待时间:能够保证 100 帧图像被客户端完整接收的最短等待时间,结果精确到 2 ms(例如等待 6 ms 无法完成传输 100 帧,则尝试 8 ms 是否能完成传输,若可以,则记录等待时间为 8 ms)

在表 1 中:1)以第二列为例,写入时间为 1358 ms,则每秒钟最多写入 $100/1.358 = 73.6$ 帧;等待时间为 14 ms 才能保证图像不丢失,既客户端每秒最多读取大约 $1000/14 = 71.4$ 帧;综合起来,传输帧率应该为 71.4 fps,传输速率 = $57\text{kB} \times 71.4\text{fps}/1024 = 3.97 \text{ MB/s}$;2)等待时间等于每帧发送时间,说明客户端接收速率快于服务器端发送速率,无需延时。

分析:1)不同格式的图像传输速率存在较大差异,系统将各种图像统一压缩成 jpg 格式图像进行传输,在客户端进行还原,虽然压缩耗费一定的时间,但能够节省传输数据量,进而节省传输时间;2)表 1 的第二列和第三列的图像均为 jpg 格式,虽然大小不同,但传输速率相近;3)bmp 格式图像传输速率远大于其他格式,这是等效计算后的结果,并非真实传输的数据量,将 bmp 格式压缩成 jpg 格式后减少了传输的数据量;4)传输黑白图像的延时平均为 1 ms 左右,可见,黑白图的传输效果较彩色图好。

6 应用实例

本系统的设计目的在于预估不同图像使用 DS 传输的速率,以决定是否在设计中使用 DS 技术。根据上述研究结论,使用本系统传输 jpg 格式 RGB(U32)图像,大小分别为 119KB、833KB,估测其传输速率均约为 3.9Mbps,实测数据如表 2 所示,客户端界面如图 5 所示:

表 2 应用实例的实测时间、速率

图像信息	1 024×768	2 448×3 264
	119 kB	833 kB
发送时间/ms	2 590	20 751
每帧发送时间/ms	25.90	207.51
最短等待时间/ms	28	218
延时/ms	0.10	2.49
传输速率/MB·s ⁻¹	4.15	3.73

根据表 2 所记录的数据,用上述分析的结果估测传输速率与实际测得的速率基本相符,本系统的设计达到预期的目的。



图 5 传输系统客户端界面

7 结 论

1)不同格式的图像使用 DS 传输的速率相差较大;2)相同格式的图像,传输速率相近,并且较小图像传输速率略快于较大图像;3)使用 DS 传输黑白图像的效果一般好

于彩色图像。

使用DS技术传输图像无需编写复杂的程序代码,开发周期短,实用性较好。总之,DataSocket作为一种实时传输技术,在图像传输领域具有广阔的应用前景。

参考文献

- [1] 王琳,郑正奇.基于DataSocket的雷达测距数据传输系统设计[J].电子测量技术,2011,34(1):99-101.
- [2] 马国胜,夏晓波.远程监测中多用户数据共享平台的设计与实现[J].国外电子测量技术,2014,33(10):83-87.
- [3] 张峰,王宇.在LabVIEW环境下实现基于C_S模式的远程数据采集系统[J].产业与科技论坛,2014,13(7):62-63.
- [4] 李婧,张永祥,石炳寅.基于LabVIEW远程校准系统的设计与实现[J].电子测量与仪器学报,2012(z):59-62.
- [5] 宋彦达.远程检测技术在故障诊断系统中的应用[J].信息技术,2014(3):179-181.

- [6] 周红标,张宇林.基于DataSocket的脉搏数据传输系统的设计[J].国外电子测量技术,2010,29(1):56-58.
- [7] 胡扬坡,何云峰,吴光文,等.基于虚拟仪器的数据采集及远程监测系统的设计[J].测试技术学报,2013,27(5):377-383.
- [8] 许爱强,李佳,陈育良,等.DataSocket在远程测试与诊断系统中的应用[J].电子设计工程,2011,19(24):130-132.
- [9] 田道坤,郑庆利,郭连领.基于虚拟仪器的数据传输系统的设计[J].科技信息,2013,(1):90-91.
- [10] 李丽娜,马俊,王嘉璐,等.基于虚拟仪器的视频图像采集系统设计与应用[J].辽宁大学学报(自然科学版),2014,41(3):222-229.

作者简介

白金鹏,1989年出生,硕士研究生,主要研究方向为电子测量技术及应用、嵌入式系统。

E-mail: 2741043827@qq.com

(上接第48页)

- [2] HU H T, HSU L Y. Robust, transparent and high-capacity audio watermarking in DCT domain [J]. Signal Processing, 2015, 109: 226-235.
- [3] 施惠娟,吴岳.一种基于FFMPEG系统的视频水印系统[J].电子设计工程,2011,22(2):161-164.
- [4] LEI B, SOON Y, TAN E L. Robust SVD-based audio watermarking scheme with differential evolution optimization [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2013, 21(11): 2368-2378.
- [5] PATEL M J, BORISAGAR K R. A Review on Audio Watermarking Techniques[J]. Journal of Open Source Developments, 2014, 1(2): 1-5.
- [6] YANG Y, HUANG R, XU M T. A novel audio watermarking algorithm for copyright protection based on DCT domain [C]//Second International Symposium on Electronic Commerce and Security, 2009. ISECS09. IEEE, 2009, 1: 184-188.
- [7] LALITHA N V, SURESH G, TELAGARAPU P. Audio authentication using Arnold and Discrete cosine transform[C]//2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET). IEEE, 2012: 530-532.
- [8] YIN C, YUAN S. A novel algorithm for embedding watermarks into audio signal based on dct [C]//Proceedings of the International Conference on

Information Engineering and Applications (IEA) 2012. Springer London, 2013: 683-688.

- [9] KHAN M I, SARKER M I H, DEB K, et al. A new audio watermarking method based on discrete cosine transform with a gray image [J]. International Journal of Computer Science & Information Technology (IJCSIT), 2012, 4(4): 119-128.
- [10] ZENG G R, LIU J M, JIANG A W. Robustness analysis of QIM watermarking against additional noise[C]//Advanced Materials Research. 2011, 225: 601-604.
- [11] GAO J, PENG W. An audio watermarking algorithm based on minimum energy harmonic phase deflection [C]//2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC). IEEE, 2013: 208-212.

作者简介

潘洋绪,1990年出生,硕士研究生,主要研究方向为信号信息处理、视频水印。

E-mail: pyangxuhhu@163.com

刘猛,1990年出生,硕士研究生,主要研究方向为语音信号处理。

E-mail: liumeng1320@sina.com