Postprint of the Image Acquisition System Built with USB3 and GigE Cameras

Authors: Wang Yao, Binhua Li, Yang Shuaiming, Zhang Yigong, Cheng Xiangming

Date: 2019-03-25T00:00:00+00:00

Abstract

Constructing a unified control image acquisition system using high-speed cameras from different manufacturers with varying interface types significantly increases development complexity. This paper presents the hardware architecture and software development solution for an image acquisition system comprising USB3.0 interface CCD cameras and GigE interface CCD cameras. On the hardware aspect, under the control of an external synchronization trigger signal, an image workstation simultaneously acquires images from both cameras. On the software aspect, secondary development is conducted using the respective development kits of the different cameras. Upon system startup, control and acquisition interfaces for both cameras are launched, enabling camera configuration through human-computer interaction. The two cameras can operate independently or perform synchronous acquisition. Experimental results from the image acquisition system demonstrate that the system functions properly, though the average time difference for completing single-frame image acquisition from the two cameras at the workstation is approximately 15 milliseconds.

Full Text

Image Acquisition System Built by USB3 Camera and GigE Camera

Wang Yao[1], Li Binhua[1], Yang Shuaiming[1], Zhang Yigong[2], Cheng Xiangming[2]

[1](Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan 650600, China)

[2](Yunnan Observatories, Chinese Academy of Sciences, Kunming, Yunnan 650011, China)

Abstract

Constructing a unified control image acquisition system using high-speed cameras from different manufacturers with different interface types significantly increases development complexity. This paper presents the hardware architecture and software development scheme for an image acquisition system composed of a USB3.0 interface CCD camera and a GigE interface CCD camera. On the hardware side, under the action of an external synchronization trigger signal, an image workstation simultaneously acquires images from both cameras. On the software side, secondary development is performed using the development kits provided by different camera manufacturers. After system startup, control and acquisition interfaces for both cameras are launched, enabling camera configuration through human-computer interaction. The two cameras can operate independently or synchronously. Experimental results from the image acquisition system demonstrate normal system operation, with an average time difference of approximately 15 milliseconds for single-frame image acquisition between the two cameras at the workstation.

Keywords: image acquisition; USB3; GigE; CCD camera; synchronization

In multi-target monitoring image acquisition systems, it is common practice to use identical cameras from the same manufacturer whenever possible, such as in multi-camera image stitching for security surveillance, production process control and monitoring, 3D image-based target recognition or motion detection, and ground-based wide-field camera arrays [1-3]. However, in certain special applications requiring observation or monitoring of different types of targets, different camera types must be configured according to the specific characteristics of each target. Furthermore, the image acquisition timing requirements vary among different cameras—some require synchronous acquisition while others require acquisition at specified intervals, depending on the application requirements of each system. For example, the first multifunctional astronomical theodolite developed by Yunnan Observatories several years ago was equipped with three CCD cameras: one digital camera with a Camera Link interface for capturing stellar and artificial star images, and two analog cameras with coaxial cable interfaces for measuring level difference and axis collimation error, with strict requirements on the timing sequence and moments of acquisition for each camera [4-5]. Similar situations exist in other industrial monitoring or measurement systems [2].

In recent years, with the rapid development of modern electronic technology, various high-speed interface cameras have emerged continuously, such as GigE, CXP, USB3, IIDC2, etc. [6-9], providing greater development space for image acquisition systems. This not only offers more options for building new systems but also presents additional technical challenges for developers.

This paper describes an image acquisition system composed of one USB3.0 interface CCD camera and one GigE interface CCD camera, focusing on the image acquisition process, control methods, system hardware architecture, and software development scheme under a unified external trigger signal. The key challenges in developing cameras with USB3.0 and GigE interfaces from different manufacturers involve resolving platform inconsistencies and mutual incompatibility issues between the two cameras, as well as achieving synchronous image acquisition in external trigger mode.

1.1 Hardware Architecture of the Image Acquisition System

The image acquisition system described in this paper serves as a preliminary research platform for an ongoing project at Yunnan Observatories. The system hardware consists of one workstation (PC) and two CCD cameras. The two cameras include one Point Gray GS3 series USB 3.0 interface CCD camera [10] and one Adimec OPAL series GigE interface CCD camera [11]. These two cameras are required to synchronously acquire images according to the telescope control system specifications, with the exposure start time difference being less than or equal to 5 ms. Therefore, both cameras must feature external synchronization trigger functionality and be controlled by a synchronization signal output from the telescope control system.

The system hardware architecture block diagram is shown in Figure 1 [Figure 1: see original paper]. The image acquisition control signal from the telescope control system is split into two channels via an opto-isolation circuit, outputting external trigger signal 1 and external trigger signal 2, which are respectively connected to the external trigger inputs of the USB3 camera and the GigE camera. The image acquisition workstation' s USB3 interface connects to the USB3 camera via a USB cable, while its Gigabit Ethernet interface connects to the GigE camera via a network cable.

The system workflow is as follows: After powering on all system components, the image acquisition workstation sends imaging control commands to both cameras through the USB3.0 and GigE interfaces to configure the cameras' operating modes and imaging parameters (such as trigger mode, exposure time, gain, acquisition frame count, and frame rate, etc.). When the cameras are in non-external trigger mode, the workstation sends acquisition commands to control image capture and transmission, as well as to display and store images. When in external hardware signal edge trigger mode, the image acquisition control signals from the telescope control system trigger the USB3 camera and GigE camera respectively. The cameras generate their images and automatically transmit the image data to the workstation via USB cable or network cable, where the workstation displays the images and stores them as required.

1.2 Software Development Scheme for Secondary Development

The image acquisition software for this dual-camera high-speed image acquisition system was developed on the Windows 10 operating system using Visual

Studio 2015 as the development platform. The USB camera utilizes Point Gray' s FlyCapture2 SDK, while the GigE camera employs Adimec' s Bobcat development kit. During secondary development of the two cameras, we discovered that the Bobcat development kit for GigE cameras is incompatible with the FlyCapture2 SDK for USB 3.0 cameras. Specifically, the pre-compiled lib and DLL files encapsulated in the two camera libraries are incompatible, and the Bobcat development kit is based on a 32-bit solution platform, whereas the FlyCapture2 SDK for USB 3.0 cameras is based on a 64-bit solution platform. Changing the platform for either camera would render the library files unusable. Additionally, the GenICam standard [12] adopted in the Bobcat development kit does not support the FlyCapture2 SDK for USB 3.0 cameras. To resolve this issue, we added the startup module program for the USB 3.0 camera interface to the GigE camera software, then independently designed the interface and background programs using their respective development kits.

During system operation, the two cameras have synchronous acquisition requirements. Therefore, a critical operating mode for the system is external hardware trigger mode. In this mode, image acquisition is controlled by external trigger signals; each incoming external trigger signal causes both cameras to acquire one frame of image. This external trigger mode is generally adopted for multi-camera synchronous control applications with timing measurement requirements or high real-time demands.

Additionally, considering system debugging needs, this paper introduces an alternative operating mode: non-external hardware trigger mode, also known as software trigger mode. In this mode, image acquisition by the two cameras proceeds independently, with no intersection between their imaging and image acquisition processes. The non-external trigger mode includes single-frame and multi-frame image acquisition. In this mode, the two cameras operate independently, with separate control and acquisition processes, and do not possess synchronization functionality.

2 System Software Design

Every camera manufacturer provides a simple image acquisition program, but these demonstration-type programs cannot meet users' specific requirements. Therefore, users must typically obtain the library files provided by manufacturers for secondary development of camera image acquisition software and independently develop system software within a suitable development environment.

2.1 System Program Design Flow

Based on the aforementioned system workflow and overall software development scheme, we established the basic framework for the system imaging control and image acquisition software. The program design flow is illustrated in Figure 2 [Figure 2: see original paper]. After system power-on and stable operation, clicking the program icon on the workstation desktop simultaneously launches the GigE camera software interface and USB camera interface, entering camera parameter setting mode. According to specific imaging control requirements, camera parameters are manually configured (such as trigger mode, exposure time, gain, acquisition frame count, and frame rate, etc.). Upon completion of configuration, the cameras enter a waiting-for-control-commands state. If the cameras are set to external hardware trigger mode, they begin image acquisition upon receiving trigger signals, with each trigger signal acquiring one frame of image. If in non-external trigger mode, the cameras can be set to either single-frame trigger or continuous trigger mode. In this state, the two cameras operate in non-synchronous acquisition mode, allowing independent acquisition settings, separate start, and separate stop operations.

2.2 USB3 Camera Software Design

The USB3 camera image acquisition software interface designed in this paper is divided into a menu bar and a client area. The menu bar contains a settings menu for configuring the number of images that can be saved in memory. The client area is subdivided into acquisition region, parameter setting region, and trigger acquisition region. These respectively include controls for single-frame and continuous acquisition, acquisition exposure and gain parameter controls, software and hardware trigger buttons, and functions for loading static images, storage settings, and format conversion.

Before acquisition begins, a preparation phase, also called a preliminary stage, is required. This includes: bus scanning, obtaining the camera GUID, initializing camera connection, and parameter setting. After completing the preparation phase, USB3 camera image acquisition primarily utilizes two camera functions from the FlyCapture2 SDK: StartCapture() and RetrieveBuffer(). The former controls the start of camera capture, while the latter controls image retrieval from the camera. The acquisition and storage functions of the USB3 camera software are implemented through three functions: the start function Start(), the acquisition thread GrabThread(), and the storage function SaveImage().

This system features two trigger modes. The USB3 camera acquisition program implements trigger mode setting by first obtaining the required trigger mode and then configuring the camera to the corresponding mode, using parameters from the TriggerMode structure and PTTriggerMode enumeration type in the FlyCapture2 library.

Additionally, the camera used is a general industrial camera whose development kit supports common image file formats such as BMP but does not support FITS format. Since this project is oriented toward astronomical applications requiring FITS image storage, image file format conversion is necessary. For this purpose, a format conversion dialog was designed to implement saving BMP images as FITS files for astronomical use. As the USB3 camera used in experiments is a color camera, the image must first be converted to a 16-bit grayscale image, then converted to a 16-bit FITS file using methods described in relevant literature

[13-14].

2.3 GigE Camera Software Design

The GigE camera image acquisition software interface designed in this paper is also divided into a menu bar and a client area. The menu bar includes four options: File, Tools, Convert, and Help. The File menu contains an exit sub-item for closing the interface. The Tools menu includes three tabs: Buffer Options (defaulting to 8), Save Image, and Save Current Image, which respectively provide sub-functions for setting buffer size, saving images, and saving the current image. The client area is divided into Connection group box, USB Connection group box, Acquition Control group box, Parameters and Controls group box, and Display group box. These respectively provide functions for camera connection, USB camera interface opening, acquisition control, parameter setting, and image display.

The GigE camera also requires a preliminary stage before acquisition: first obtaining the interface number, MAC address, subnet mask, and IP address, then connecting to the camera, and finally writing parameter values to camera registers to configure the camera. The GigE camera image acquisition process involves: first opening a data stream (or pipeline) and setting the buffer variable Buffer for the data stream object; next, enabling the data stream, at which point data continuously refreshes into the buffer; then retrieving image data from the buffer variable lBuffer through the RetrieveNextBuffer() function; and finally passing the image data to subsequent storage or display functions via the ProcessBuffer(lBuffer) function. During storage, raw data is obtained from the Buffer and the Store function is used to save the image.

Similar to the USB3 camera software, format conversion is required when saving images. The software and hardware trigger function implementation method for the GigE camera image acquisition system is: first using the PvGenParameter class to map parameters in the trigger control combo box, then using the Execute() function of the PvGenCommand class to execute the settings and achieve triggering.

Regarding memory control for the GigE image acquisition software, when directly defining PVImage class objects to store data from the Buffer, permission issues arise, preventing customization of PVImage class object sizes. Therefore, we ultimately chose to redefine the SnapImageInfo structure to define the image data format, then define an array object of this structure type. The data and parameters from the Buffer are retrieved and assigned to the SnapImageInfo object defined in memory, thereby composing and saving the image in memory.

3 System Test Results and Analysis

After completing the hardware construction and software design of the high-speed image acquisition system, comprehensive testing of the entire system is required. All tests were conducted on a 4-core laptop running Windows 10

64-bit operating system with a 2.30 GHz CPU. The testing experiments were divided into two parts: software testing and synchronization testing.

3.1 Software Testing

The software testing scheme involved setting the cameras to hardware trigger mode, configuring the image storage format as BMP images, and ultimately converting the generated BMP images to FITS image format. The memory control function stores image data in memory using custom structures or class objects in array format; during testing, this was verified by outputting and storing images from memory. Both the GigE camera acquisition software and USB3.0 camera acquisition software were tested separately, with records kept of the FITS file header format, acquisition parameters, and other information from camera acquisition and conversion.

Figure 3 [Figure 3: see original paper] shows the software operating in hardware trigger mode (with the GigE camera operation interface on the left and the USB camera operation interface on the right). Both camera software can normally complete functions including single-frame and multi-frame image acquisition, software and hardware triggering, memory control, storage, and format conversion.

3.2.1 Test Scheme

The project development requirements specify that the CCD image acquisition frequency of the image acquisition system should be three images every two seconds, with each image being 55 MB in size, corresponding to a data transmission rate of 82.5 MB/s. The USB3 camera used in the image acquisition system has a resolution of 1928×1448 pixels and a stored image depth of 24 bits. Image frame rate determines image smoothness—the higher the frame rate, the smoother the image appears. Therefore, the camera output frame rate should be guaranteed to be above 12 frames per second. The trigger signal frame rate was set to 21 frames per second, at which point the USB3 camera' s data transmission volume is 167.58 MB/s. The GigE camera has a resolution of 1024×1024 pixels and an image depth of 32 bits, with a data transmission speed of 84 MB/s at a trigger signal frequency of 21 Hz. Both meet the project requirements.

The synchronization test scheme and steps for the image acquisition system software are as follows:

(1) Two test scenarios: The first scenario involves the image acquisition workstation powering on and running without launching any other applications, only executing our developed image acquisition program—this is referred to as the light-load case. The second scenario involves the acquisition workstation powering on and launching multiple other applications (such as Office, NetEase Cloud Music, Thunder, etc.) in addition to our image acquisition program. However, during image acquisition, these other programs are not used and do not execute specific tasks—this is referred to

as the heavy-load case.

(2) In both the GigE camera acquisition software and USB 3.0 camera acquisition software, under external hardware trigger mode, the moment when a frame acquisition is completed into the buffer is recorded to a txt document. Each incoming trigger signal acquires one frame of image while simultaneously recording the time information. The external trigger signal is configured as 3000 consecutive pulses, meaning each camera performs 3000 consecutive image acquisitions.

(3) MATLAB software is used to read the time data from the txt documents collected by the GigE camera acquisition software and USB 3.0 camera acquisition software, calculate the differences, generate data distribution plots, and compute the mean and variance from the differential values.

3.2.2 Synchronization Test Results and Analysis

Two time information documents from the software operation were saved and analyzed, including: USB3 camera acquisition time record table and GigE camera acquisition time record table. The original time format was hour-minute-second-millisecond, totaling 9 digits. The recorded times were first converted to milliseconds (ms), then the GigE camera data was subtracted from the USB3 camera data to obtain the time difference data for each completed image acquisition. The time differences for image acquisition completion between the two cameras under both light-load and heavy-load test scenarios are shown in Figures 4 Figure 4: see original paper and (b). This test was conducted more than a dozen times, with minimal variation in experimental data. The results shown in Figures 4(a) and (b) represent just one test instance. In the figures, the horizontal axis represents the image acquisition sequence (or frame number), and the vertical axis represents the time difference for each completed image acquisition.

In Figure 4(a), the three points at approximately 20 ms and two points at approximately 4 ms are due to the image acquisition program running under the Windows operating system, which is a preemptive time-sharing multitasking operating system with a time slice of approximately 10-15 ms. During operation of the acquisition software, the operating system allocates certain processing time to other applications or devices, thereby causing errors in some acquisition record times. Additionally, the integer-level differences appearing in milliseconds in both figures result from the millisecond-level precision of the acquisition time during system testing.

The statistical results for both scenarios are: under light-load conditions, the time difference mean is 15.2907 ms with a variance of 0.7480 ms; under heavy-load conditions, the time difference mean is 15.4745 ms with a variance of 3.1328 ms.

In reality, for multi-camera image acquisition systems triggered by external synchronization signals, the exposure moments of each camera are synchronized at the millisecond level. Disregarding the influence of the Windows operating system, the theoretical time difference for image acquisition completion between the two cameras should be fixed. Calculated based on the previously set frame rate (21 frames/second) for both cameras, this fixed time difference should be 0.3626 ms. This indicates that at the millisecond level, image acquisition completion is also synchronized. The synchronization differences observed in experiments refer to the time differences for each camera to transmit images to the computer and for the computer to complete image acquisition. The statistical results of these time difference data verify that other programs under the Windows operating system introduce certain time differences to the dual-camera image acquisition system. This demonstrates that for dual-camera image acquisition systems built on the Windows operating system to achieve synchronization performance within 10 ms, software triggering cannot meet the synchronous exposure requirement of less than 5 ms. Cameras with external signal edge trigger functionality must be used to constitute multi-camera image acquisition systems. Furthermore, during system operation, other tasks (programs) should be minimized as much as possible.

The dual-camera synchronous control fabric quality detection system introduced by Jin Shaobo et al. exhibited image capture time differences of 100-600 ms under software triggering and 500-600 ms under hardware triggering [2]. In comparison with these results, the feasibility and superiority of the system and design method presented in this paper are demonstrated.

4 Conclusion

This paper has described the architecture of a synchronous high-speed image acquisition system composed of UBS3 interface cameras and GigE interface cameras, explored collaborative development methods for UBS3 cameras and GigE cameras under mutually incompatible development kits, and implemented convenient control. The control software for both cameras has realized functions including single-frame and multi-frame image acquisition, image display, software and hardware triggering, acquisition parameter setting, image storage, and format conversion. Software testing and synchronization testing were conducted on the image acquisition system, demonstrating stable system and software operation. The synchronization test results indicate that the system meets design requirements. Additionally, the paper has discussed the impact of the Windows operating system on image acquisition tasks.

References

[1] Wang Rongben, Li Linhui, Jin Lisheng. Research on obstacle detection technology for intelligent vehicles based on binocular vision[J]. Journal of Image and Graphics, 2007, 12: 2158-2163.

[2] Jin Shaobo, Zhao Fengxia, Li Jifeng, et al. Research on fabric quality detection technology based on multi-camera synchronous control[J]. Mechanical Research and Application, 2016, 29(4): 190-192, 195.

[3] Zhang Shenghua, Huang Lei, Wei Jianyan, et al. Preliminary study on the application of AC servo systems in astronomical telescopes[J]. Astronomical Research and Technology (National Astronomical Observatories of China), 2017, 14(3): 337-346.

[4] Li B., Zhang Y., Yang L., et al. Imaging acquisition system with three CCD Cameras, 2012, Proc. SPIE, 8451:84512Y.

[5] Zhang Yigong, Li Binhua, Yang Lei, et al. Image acquisition system for multifunctional astronomical theodolite[J]. Astronomical Research and Technology (National Astronomical Observatories of China), 2014, 11(1): 72-79.

[6] Feith W. Pro and con of using GenIcam based standard interfaces (GEV, U3V, CXP and CLHS) in a camera or image processing design, 2015, Proc. SPIE, 9405:940511.

[7] Li Dalun, Li Binhua, Yan Jia. Research on high-speed data transmission system for EMCCD camera based on USB3.0[J]. Astronomical Research and Technology (National Astronomical Observatories of China), 2014, 11(3): 255-263.

[8] JIIA. Coaxpress Standard v1.1.1, 2015, http://jiia.org/wp-content/themes/jiia/pdf/standard_dl/coaxpress/ 001-2015.pdf, 2019-02-26.

[9] JIIA. IIDC2 Digital Camera Control Specification v1.1.1, 2015, http://jiia.org/wp -content/themes/jiia/pdf/standard_dl/ngcp/CP-001-2015_TS2015001.pdf, 2019-02-26.

[10] FLIR. GS3-U3-Technical-Reference [DB/OL], 2017, https://www.ptgrey.com/support/downloads/10125/, 2019-02-26.

[11] Adimec. Technical specifications for OPAL series with 5.5 micron CCD [DB/OL], https://www.adimec.com/cameras/machine-vision-cameras/opal-series/opal-series-5-5-micron/.

[12] Emva. GenICam Standard v2.1.1, 2016, https://www.emva.org/standards-technology/genicam/.

[13] Wang Yanfang, Zhao Kexin, Lu Xiaomeng. Conversion between FITS and BMP formats using VC[J]. Microcomputer Information, 2006, 22(4): 121, 232-233.

[14] Ji Kaifan, Cao Wenda, Song Qian. FITS, BMP, and SCR image formats and their mutual conversion[J]. Publications of Yunnan Observatory, 1996(02): 60-64.

Note: Figure translations are in progress. See original paper for figures.