2013 AASRI Conference on Parallel and Distributed Computing and Systems

Design and Implementation of Multi-sensor Integrated 3D Laser Scanning Data Acquisition System

Yibo Sun[a], Xinqi Zheng[a],*, Zongren Jia[a], Herao Wang[a]

[a]School of Land Science and Technology, China University of Geosciences, Beijing, 100083, China

Abstract

A LiDAR system is combined by various kinds of sensor systems. And the real-time data acquisition of multi-sensor is the basis of it. According to the characteristics of data acquisition for multi-sensor integrated mobile LiDAR system, we use Qt as development platform and adopt serial communication programming technology and multi-thread technology to achieve real-time data collection and preservation of the mobile 3D Laser Scanning Measuring system. Compared with the existing system, we designed 3D Laser Scanning Data Acquisition System overcame the inconvenience that we need to start each sensor data collection program individually and difficult for the data synchronization. In addition, this system also has superb portability. It can also be applied to other operating system supported by Qt such as Linux, Windows CE, etc. Finally, the experiment results show that this 3D Laser Scanning Data Acquisition System operation is stable and reliable, it could collect and preserve different kind of sensor data simultaneously and meet the application needs.

1. Introduction

Multi-sensor integrated mobile LiDAR technology is a hot topic among the current development of mobile

---

* Corresponding author. Tel.: +86 10 82322138
E-mail address: zxqsd@126.com

mapping technology. A typical mobile LiDAR system usually consist of many kinds of sensors such as GPS positioning system used for obtaining the position of carrier, inertial navigation system(INS) used for obtaining the attitude of carrier, laser scanner used for getting the distance and angle of the scanning spot and digital camera used for capturing target texture [1-3]. As the foundation of the whole system, data acquisition is one of the most important parts in the mobile 3D laser scanning system. Current researches on mobile LiDAR system are focused on system integration and system application, but little research work on data acquisition. In practical work, data acquisition usually has been completed by programs which have been offered by the manufacturer of each sensor. This will cause some problems such as complex to operate, the time difference between starting time of each sensor is big, difficult to unify and so on. Meanwhile, it also reduces the integration between the performances of the system. Therefore, it is necessary to design the data acquisition software for the overall system.

Qt is a cross-platform application and UI development framework [4]. QThread in Qt provides a platform-independent way to manage threads. Through QThread we can easily develop portable multithreading Qt applications and take full advantage of multi-processor machines. Qt application has excellent portability, so the application using Qt development cannot be limited to the Windows operating platform.

Relying on data acquisition software offered by sensor manufacturers, 3D laser scanner acquisition has some problems such as the complicated operation, lower integration, and poor portability and so on. In order to solve these problems, this paper designed and implemented a multi-sensor integrated mobile 3D laser scanning data acquisition system using Qt multithreading mechanism and serial programming techniques to set up 3D laser scanner parameters and acquire and save multi-sensor data. This system also could be used on different platforms.

2. System design and implementation

2.1. The hardware system

The 3D laser scanner which is the important part of the system is composed of GPS, INS, laser scanner and digital camera. GPS is used for real-time transport the position information of the system, INS is used for transmitting the attitude information of the system and the laser scanner is used for transmitting the distance and angle information. In work process, all the sensors are connected to the data acquisition computer through the serial port. The data acquisition computer is responsible for setting the working parameters of each sensor and receiving and saving the data collected by sensors. The hardware system structure is shown in Fig. 1.

Using RS-232 to USB convertor to connect the GPS receiver to the data acquisition computer and using RS-485 to USB convertor to connect the INS to the data acquisition computer. The laser scanner is connected to the computer through USB. Besides, access the GPS synchronization signals to the INS external sync signal input terminal, meanwhile the GPS synchronization signal is used to trigger the digital camera automatically.
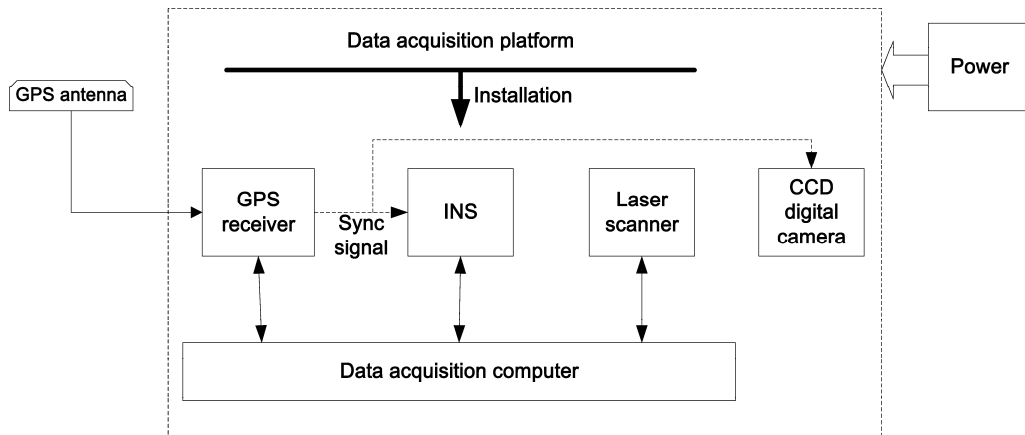
Fig. 1. System hardware structure

2.2. Software system design

This paper designed the data acquisition system includes two main function modules: the data acquisition module and the data acquisition monitoring module.

The data acquisition module completes the system data acquisition. It includes laser scanner data acquisition module, GPS data acquisition module and INS data acquisition module. The data acquisition monitoring module is used in the process of data collection to monitor the working status of each module. The structure design of this system is shown in Fig. 2. The main function of each module is described below:

• Laser scanner data acquisition module: set the scanning range, set the scan data saved path, complete the acquisition of laser scanning data.

• GPS data acquisition module: Set serial port parameters (baudrate, parity, data bits, stop bits), set the GPS data saved path, and accomplish the acquisition of GPS data.

• INS data acquisition module: set the INS data saved path, complete the acquisition of INS data.

• Data acquisition monitoring module: In the work process, feedback the GPS, INS and laser scanning data to the monitoring window. It is convenient for users to monitor data acquisition in the whole process.
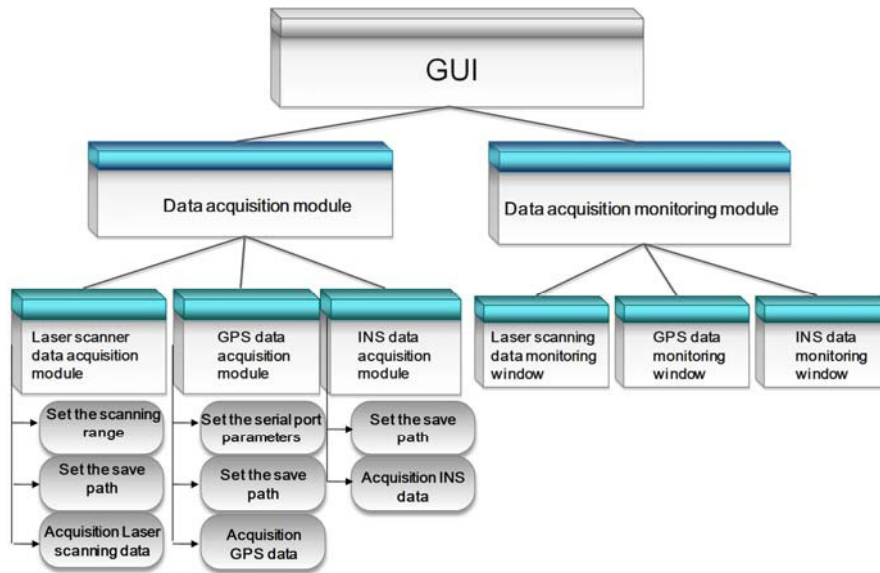
Fig. 2. Data acquisition system structure

In order to achieve the simultaneous acquisition of different sensor data, we adopt multi-thread program design way. In this paper, we designed GPS thread, INS thread and laser scanner thread. The data acquisition process is shown in Fig. 3. After starting the program, first initialize the serial ports which connect the sensors, then set the relevant parameter, including the laser scanning range, the serial parameter and the path for saving the data. When begin to collect the data, reading serial data usually has two types: event query mode and event driven mode. In the design of the program, the event query mode continuously needs to design a timer. The timer queries the serial buffer at a fixed time to perform the data capturing. The event driven mode completes the data reception when the data sending the serial port. In this paper, we adopt the event driven mode to accomplish serial data reading.

In order to make each of the sensor data hold time information for time synchronization afterwards, we write the begin receiving time of the system UTC time to the laser scanning and INS data. And then use mutex to protect the data in the memory buffer collected by each sensor to ensure that at any one time there can be only one thread to access the object. At last, save the data in the memory buffer to the local hard disk files of the specified path and determine whether the data collection is stopped, if stopped subsequently close the serial ports and finish the data collecting.
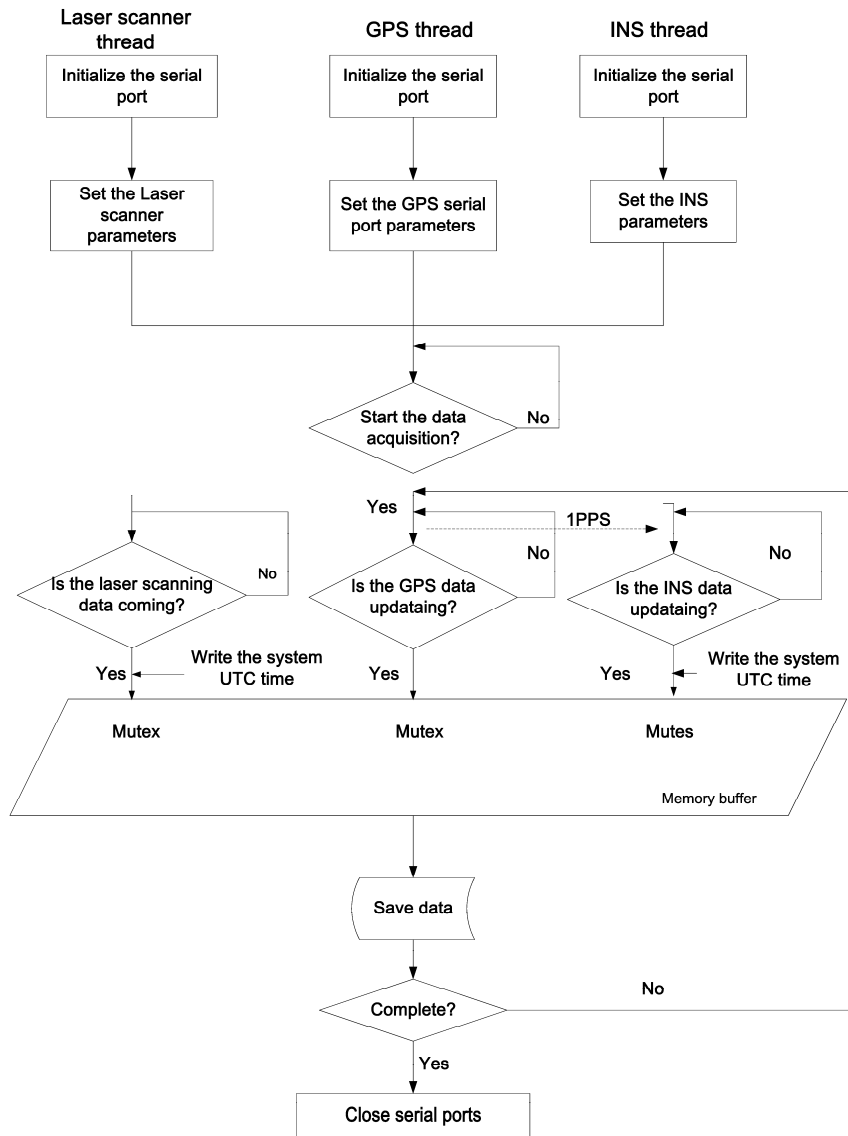
Fig. 3. Data acquisition process

2.3. Key technologies and implementation

In this paper, multi-threading technology achieves the concurrency function for multi-sensor data acquisition at the software level. Multithread technology not only ensure each of sensors to get the data independently, but also improve the efficiency of multi-core computer utility. Because that Qt does not have specific serial control class, this paper uses the third party QextSerialPort class to complete the control of the serial port. This paper uses Qt 4.7.2 version and Qt Creator 2.1.0 completed the development of the system

under the Windows operating system. The implementation methods of the key technologies involved in this paper are described below.

2.3.1. The implemented of multithreading class

We designed the threads of GPS, INS and laser scanner individually due to the reason that the frequencies of each sensor are different [5]. QThread in Qt provides a platform-independent way to manage threads. The QThread represents a thread that can be controlled independently in an application and shares data with other threads in the process, but execute independently [6]. Due to the reason that the creation of each data acquisition thread class is similar, we will put GPS data acquisition thread code as an example.

```
class GPS_Thread : public QThread
{
Q_OBJECT
public:
  GPS_Thread (QObject *parent=0);
signals:
……
Protected:
    ……
    void run();//The starting point for the thread
private:
    ……
QextSerialPort *GPS_port; //The serial port connect the GPS device
}
```

Begins execution of the QThread by calling run().Usually we should create an instance of subclass of QThread and reimplement the run() function. Then calling start() function to start execution the thread, the start() function will default call the run() function. Note here that in the laser scanner thread we use the API provided by the manufacturer of the laser scanner to achieve the setting of the scanning range and scanning parameters of the laser scanner. The thread synchronization can use mutex QMutex to ensure that at any time and at most there is only one thread can get the mutex. Qt provides a convenient class QMutex Locker used to lock and unlock the QMutex automatically.

2.3.2. Creating the serial port object and data reception

This paper relates to three data acquisition threads, so we should create the relevant serial port object to constructors of each thread and initialize the serial port. The QextSerialPort class provide the struct PortSetting to set the baudrate, parity, databits, stopbits, flowtype and timeout of the serial port. In Qt, reading serial port data is like with reading a file data, we can use ReadOnly mode to open the port. The serial object created as follows:

port = new QextSerialPort(const QString &name, const PortSettings& s, QueryMode mode = EventDriven, QObject *parent=0).

We use Qt signal-slot mechanism to receive the ports data. Signal-slot is the core mechanic in Qt [7]. It can be applied in the communication between objects. The following example is how to connect serial events and data reception function.

connect (port, SIGNAL (readyRead ( )), this, SLOT (receiveData ( ))).

When serial data is coming, the serial port object emit readyReady() signal. Then the current data acquisition thread receives the signal and executes the data reception function receiveData() and complete data reception and preservation.

To achieve the purpose that unified control of the data acquisition of each sensor, we designed a button to start up and shut down the data acquisition thread of each sensor at the same time. Calling the function start() to start the threads and calling the function stop() to close the threads.

3. System verification and discussion

At first, set the COM port for connecting sensors. If the system does not check the serial port, we need to click on the "rescan" button to re-examine. Then, click the "Connect" button after determining the sensor serial port correct and click the "Apply" button after setting the relevant parameters in the device setting. For completion of the above work, we completed the settings before data acquisition. The entire system is placed in the open air areas. Until the GPS receiver signal indication has been positioned, click on the "Start" button in the main interface to begin the collection and preservation of data. The results show that the data acquisition system can get the data of GPS, INS and laser scanner effectively and each thread is independent.

Through the research, there are two characteristics can be found in the multi-sensor data acquisition system:

• Multiport serial, multi-protocol data transmission: The sensor communicates with the computer through a serial port separately. Different sensor has different data protocol and the completion of the data analysis typically requires using the API provided by the sensor manufacturers.

• High-speed mass data transmission: In the data acquisition process, the maximum of GPS data acquisition frequency is 100Hz, and laser scanner is 25Hz, and INS is 100Hz, which proves that the speed of data transmission is fast and the quantity is large. Such very quick and large amount of data acquisition and preservation may cause thread deadlock. Therefore, use the mutex to protect the thread and to ensure the safety of the thread in the system.

4. Conclusion

In this paper, we adopt Qt multi-threading mechanism as well as serial port programming technology to achieve a multi-sensor integrated 3D laser scanning data acquisition system.This system can easily set up the 3D laser scanner and implement data acquisition and storage of GPS, INS and laser scanner simultaneously. The collected data are integrated and dependable and have time information, so it can meet the needs of late data processing. Besides, this system has good portability. It can be used into other platforms according to different application requirements. However, this research achieved the data acquisition system is a systematic prototype, there are various storages of system function and interface design and we will constantly improve this system in the future work.

References

[1] M. Nagai*, T. Chen, R. Shibasaki, et al.*, "UAV-borne 3-D mapping system by multisensor integration," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 47, 2009, pp. 701-708.
[2] XueXian Geng, SiDong Zhong, "A mobile system using lidar and photogrammetry for urban spatial objects extraction," *2009 International Conference on Information Engineering and Computer Science,* 2009.
[3] A. Kukko*, A. Jaakkola, M. Lehtomaki, et al.*, "Mobile mapping system and computing methods for modelling of road environment," *2009 Joint Urban Remote Sensing Event, 2009.*
[4] B. B. Akinkuolie*, C. F. Lin, S. M. Yuan*, "A cross-platform mobile learning system using QT SDK Framework," *5th International Conference on Genetic and Evolutionary Computing*, 2011, pp. 163-167.
[5] LI Jing*, SHI Zhixing, CUI Zhewei*, "The optimization about multi-parallel embedded server in farmland based on Qt Embedded," *Journal of Agricultural University of Hebei,* vol. 34, 2011, pp. 108-111.
[6] B. W. Barrett*, J. W. Berry, R. C. Murphy, et al.*, "Implementing a Portable Multi-threaded Graph Library: the MTGL on Qthreads," *2009 IEEE International Symposium on Parallel & Distributed Processing, Vols 1-5*, 2009, pp. 2123-2130.
[7] Q. X. Wu and J. Ou, "QT Programming Technology and Application with Linux," *Software Engineering and Knowledge Engineering: Theory and Practice, Vol 1*. vol. 114, 2012, pp. 573-578.