Application note

A Pocket PC based field information fast collection system

Hui Fang, Yong He*

College of Biosystems Engineering and Food Science, Zhejiang University, Hangzhou 310029, PR China

Abstract
To acquire and analyze field information quickly and effectively, a Pocket PC (PPC) based field information fast collection system was designed. Compared with a desktop computer, the operation system of this system (Windows CE) is greatly different in many aspects of system development and design. Key technologies investigated included geographic information system (GIS), spatial access methods, database technology on Windows CE and sampling control plan. Considering the uniqueness of farms, an object-oriented farm model was developed to describe farm objects properly, edit sampling plan conveniently, and perform sampling control easily. The system can be used with the global positioning system (GPS) and sensors to acquire and analyze field information instantly. Some examples are used to explain the use of the system. The development of the system supports fast collection and real-time processing of precision farming field information.

1. Introduction

Accurate and reliable spatial information is the base of precision farming field operations. There are two kinds of field information: (1) the field position information which includes the size and shape of the field, longitude and latitude of sampling point, the irrigation canals position in the field, etc. In precision farming, only after the position information has been confirmed and archived, can further data be collected. These data include soil properties such as water content, organic matter, nitrogen (N), phosphorus (P), potassium (K), yield information such as seed yield, biology production, etc. With a traditional field information collection method, soil or crop samples are collected in the field and submitted to a laboratory for analyses. Though precise laboratory results are obtained, the cost of these analyses is high. Getting field information as quickly and cheaply as possible was also a basic request of most practitioners of precision farming. With the recent development of sensing technologies, it’s now possible to solve this problem. For automatic field data collection, the data store, data collection control and real-time sampling plan are parts of the mission. This research focused on the software control part of these kinds of systems.

Because it is field information that is to be collected, a desktop computer is too bulky to be used as a platform for this system. Laptops are portable but have limitations which include higher price, short power supply time and are somewhat bulky. Recently, the performance of Pocket personal computers (PPCs) has been upgraded in all aspects, such as improved operating system and more powerful development environment. Today, it is possible to develop a PPC based field data collection and control system.

Up to now, there is some PPC software to solve agriculture problems, but in China, such research work is still not
common, and seldom is agriculture software developed for farmers based on PPC. The only example in China was the use of a PPC as a platform of an expert system (Chen et al., 2002). When talking about the information collection-based PPCs in worldwide. Some recent developments, such as the Trimble AgGPS 170 Field Computer, are integral software–hardware computer platforms designed especially for agriculture applications. The AgGPS 170 receives sensor information and GPS information in ASCII code through an RS232 interface to track agriculture machine position and the fertilized amount of variable rate application. Another good agriculture PPC resolve scheme of Trimble Co. is the software called AgGPS EZ-Map, which is based on Windows CE. The AgGPS EZ-Map software is designed to provide some mapping functions and simple sampling control. The sampling control software includes two kinds of sampling plans: manual input and bar code scanning. The function of such system is limited by the published research results of precision farming. Wang et al. (2004) have developed a computer-based timber cruising system. The system consists of three components: handheld data collection system, data transfer, and data analysis, all of which were developed as stand-alone objects. The handheld system is Windows CE based. This is a system designed especially for forest trees.

As a research organization focusing on PA research, we designed our own field information collection system. The purpose of this design is to establish a strong framework for these kinds of systems, which is flexible and can evolve with future PA research developments. In this article, the framework and function of the system is first described and then the key technologies are discussed.

2. System frame and functions

Three major parts: spatial data management, property data management and sampling control compose the final system. The functions of these parts are shown in Fig. 1. Among these functions, the serial port communication module acquires GPS and sensor data, and can input other kinds of data manually. This module is also responsible for sampling according to a sampling plan. Sampling point calculation acquires sampling point according to the irregular field border. Map-browser mainly includes some map operation such as zoom in, zoom out, moving. Spatial property query and analysis includes field area calculation, perimeter calculation, line segment length calculation and simple buffer analyses. Up to now, object property query is designed to list a layer object property. Object property edit can edit selected object property manually after sampling. The architecture of the PPC based field information system is shown in Fig. 2.
3. Research on the key technology

3.1. The integration of geographic information system

The system is actually a special GIS for field information collection. The unique requirements of precision farming require features that differ significantly from the current PPC GIS system (cadastration GIS, navigation GIS, for instance). The cadastration GIS currently accepts GPS data automatically and other input information manually, but in this system, besides position information, the field property data coming from sensors must be logged automatically. Other data, such as laboratory analyses, can be acquired after sampling. So, for the flexibility of the system, there must be many different data sampling plans which produced more complex sampling control flow in this system. Other GIS programs are mainly used for navigation. For navigation with GIS, most data is stored in advance. Data structures between the two special GIS cases are significantly different.

3.2. Object-oriented precision agriculture model

In this system, object-oriented methods were used to represent all objects. The C++ object-oriented language was used. The object-oriented approach is a software implementation of the methodology that man uses to recognize and understand the world. In an object-oriented system, each concept entity is modeled as an object. Using an object-oriented method, objects can be abstracted as points, lines or polygons just as with many old traditional GIS engines. For example, the initial model of this system was: since the program environment is a PPC, the hardware performance is not as good as that of a PC. An objective is to make the real-time dataset as small as possible. So the All-Point model was used to store different points in list form so that the same point won’t be stored more than once. Different points here means points in different longitude, latitude or altitude. The All-Point model saves longitude, latitude and altitude of every GPS position point. Other objects only possessed pointers to every used real point but not the real point; that is, other objects are composed by pointers to real points. The real coordinate of each pointer points is stored in the single point list. Point Object was used to describe activities which can be abstracted as a point, such as moving GPS signal source; Line Object was used to describe entities which can be represented by a line. To every line object, a continuous line constituted by a set of points was called "Part". Polygon Object was used to describe one or more closed surfaces. Every closed surface was a Part of a Polygon Object; Rectangle Object was used to describe entities which can be abstracted as a rectangle. A rectangle object is also a special Polygon Object.

During the design process, it was determined that this method paid too much attention to abstract things into point, line and polygon, and the spatial relationship among the objects in farms was ignored. It was difficult to describe unique farm features and add appropriate sampling plans, especially since farmers usually are not familiar with point, line and polygon but are familiar with crops, pools, roads, fields, etc.

To resolve this problem, design pattern methodology was adopted for this system. Design pattern is a reusable scheme intended to solve repeated problems. In software engineering, one design model may resolve a series of software design problems. Referring to kinds of design patterns (Gamma et al., 1995), a new design model was created: farm model. The model graph is showed in Fig. 3.

In this structure: CFarmObjects acts as a main interface class. It provides all outer interfaces for any of its subclasses. CFarmBorders, a subclass of CFarmObjects, is an abstract class of all border objects. Every object in a farm must have a border. A point can be seen as a special border object, so that the CFarmSamplePoint, which is designed only for Sample Point Objects, can be a subset of CFarmBorders. CFarmComposite, a composite class, consists of a pointer list which points to CFarmObjects objects. Therefore, a CFarmComposite object can be a composite of different kinds of border objects. Because special sampling plans of sample point, CFarmSample, is a subclass of CFarmBorder. CFarmFields, which delegates fields in a farm, is a sub class of CFarmComposite. Besides the functions of CFarmComposite, it can have instance of CFarmSamplePoint and CMapGridLayer, a class responsible for grid data in a field. This means that a field in a farm is composed of one sampling layer, one grid layer and one common bor-
under layer. In other words, two vector layers and one grid layer overlap and constitute a field object. As for water system in a farm, we designed a CFarmWaterSystem (not shown in Fig. 3), which is a subclass of CFarmComposite and is actually composed of a line layer and a polygon layer. Line layer represents narrow water paths and polygon layer represents pools in a farm. In this way, we simulated the common objects in a farm.

This farm model is actually a combination of composite and prototype methods. The composite method, composting objects into a tree structure, is a “part-whole” hierarchy. In this farm model, CFarmObject, CFarmBorder, CFarmComposite, CFarmField and CFarmSamplePoints constitute composite parts. The prototype method designates the type of object to be created and creates new objects by copying these prototypes. This method is mainly used when the class is confirmed at run time or the instance of a class is one of the limited combinations of several statuses. CMapLayer, CMapGeometry and CMapPoint, CMapPoints, CMapLine, CMapPolygon constitute the prototype part. The prototype model here helps CFarmBorder to generate different layers dynamically as different type of borders.

### 3.3. Spatial index structure

The performance of GIS spatial query is, to a great degree, determined by the performance of a spatial index structure. Up to this point in time, research on spatial index structure was done under conditions of abundant hardware resource, such as a desktop computer or file server. Most work was approached with the precondition that hardware resource was abundant with relatively limited memory and large external storage (Gu and Wu, 2001; Liu et al., 1996; Chen, 2001; Samet, 1995). The access speed of external storage was slower than internal memory. To speed up indexing, it is necessary to reduce external storage access times. Therefore, most research in this area was focused on external storage indexing. But compared with PC or file server, PPC exhibit greater different in the structure. The limit of PPCs for spatial indexing is the speed of the CPU, the storage size (internal memory and external storage), power, and the size of display.

As far as storage size is concerned, PPC memory is usually very small. For example, the iPAQ3700 only has 32 MB programming memory and 64 MB object memory. However, its CPU accesses data from them at the same speed. So, if no other storage facility is used, the main storage structure is recommended. The usual main storage structures used in PCs are: K-D tree, BSP tree, quadtree and Cell tree. As far as the power supply and CPU are concerned, the power consumed by the PPC is directly related to CPU load. Proper data structure and arithmetic require fewer computations and therefore consume less power. According to An et al. (2001), in the PPC, index structures (R-tree, quadtree, Buddy tree) are suited for different query methods (such as point query, extension query and closest distance query). As far as the display size is concerned, the PPC screen is usually very small. Layer structure is preferable if the screen is the only consideration. One example is a PPC vector data blocked data index structure proposed by Li et al. (2002). If data are inserted dynamically, there must be additional classification which remains unresolved. Therefore this structure can be used for static vector data but not for dynamic vector data.

Based on the analysis above, the spatial indexing plan of this system required memory mapping file or allocation of virtual memory larger than 2 MB. In such instance, the memory was managed by the operation system, so the program can use memory above 32 MB. For this system, when data requirements were small, the usual main memory index was used. If the data storage requirements were large, a memory index was used, but the data were stored in object memory RAM and the majority of data were stored in memory mapping file. If the storage requirements are very large, then all data files can be saved on CF cards and memory mapping must be used to establish spatial indexing. This spatial index method is similar to the disk index method.

### 3.4. Discuss of databases

Considering the PPC with the Windows CE operating system, there are three database development schemes in Windows CE 3.0. The first uses a set of unique database application program interfaces (API), the second uses ActiveX Data Objects (ADO) interfaces, and the third uses Object Linking and Embedding Database CE (OLEDB CE) interfaces. Among these, ADO is a technology based on OLEDB. This scheme makes client programs use the uniform data accessing methods provided by OLEDB to access and manipulate data in the database server.

When using a Windows CE database API, the only Choice is CEDB. But CEDB lacks windows, reports, queries, and relationship between tables, and only provides table methods for storage and accessing. If ADOCE or OLEDB CE is used, more databases including the CDEB database, SQL Server CE, and other third party provided databases (Sybase SQL Anywhere or Oracle Lite) can be accessed. Compared with OLE DB, the main virtue of ADO is higher efficiency, but with less flexibility. The choice between ADOCE and OLEDBCE is also influenced by the selected compiler.

In VBCE (Fig. 4), OLEDB CE cannot be used directly. It must be called through ADOCE, VC++EC can access OLEDB CE directly. And directly using OLED CE increases flexibility. For VC++CE, support classes of OLEDB CE do not exist, and must be developed by the programmer or third-party providers. The end result is that development cost is greater than choosing VBCE and ADOCE.

Because the object-oriented data model was adopted in this system, C++ becomes the preferred choice. And since C++ was chosen, the OLEDB CE database interface was selected for the SQL Server CE.

### 3.5. Design of sampling plan

According to the characteristic requirement of precision farming, and the peculiarity and variety of field data, five sampling flows were designed in this system: border points sampling, manual post-sampling, manually sampling at a point, all auto sampling, and auto sampling at a point. The definitions of these sampling plans are:
Border points sampling. This sampling plan is used to get the border of objects in a farm, e.g., buildings, fields, pools, raceways, etc. In this kind of sampling, only GPS data is acquired from the communications port. There are two instances in this sampling plan: one is moving along the border and recording data. The other is only recording data at certain turning points of the border and at each turning point, and calculating the mean value as a valid point position.

Manually after-sampling. If the information can’t be acquired through a sensor, output samples must be obtained at a specific point in the field, and then submitted to a laboratory for analysis, just like traditional field data collection. In this situation, the main operation is to digitize and label the sample site, and store these parameters in a database.

Manually sampling at point. In this situation, we must navigate to a location with GPS, make a measurement or observation, and record the result.

All auto sampling. This sampling is used in agricultural field machinery. Equipment performance parameters are sent to the PPC. GPS and sensor data are acquired, registered and stored into a database.

Auto sampling at point. GPS is used to navigate to a predetermined field location and a sensor collects a reading. This process is a little like all auto sampling, except that it may include calculation of the mean of a series of sensor readings.

Besides the five sampling plans outlined above, a more specific situation is observing the moving of GPS only, which is, monitoring GPS signal quality.

All control processes were completed by two threads: the main thread and the serial thread. The serial thread was responsible for reading data arriving at the serial port. If the data are from the GPS, the integrity of the signal is checked and the appropriate data are written to a buffer and a notification sent to the main thread. When the main thread is informed of GPS data written to the buffer, it transferred a signal to the tracking layer to update tracking data. A signal is also transferred to the current active layer to activate some kinds of sampling plans.

4. Realization and experiment of the system

We first finished the PPC based field information fast collection and real-time processing system in 2003. The operation system utilized was PocketPC 2002. All programming development was completed under Microsoft embedded Visual C++3.0. The hardware platform was a Compaq H3700. The system user interface is shown in Fig. 5.

4.1. GPS Data acquired

Fig. 5 shows two field borders acquired by this system. Data are from a Trimble AgGPS 132. This integrated GPS receiver accesses the OmniStar radio beacon so that Racal Land Star differential correction signals can be utilized. Since the differential position precision of the AgGPS 132 is submeter, usually it is used for agriculture purposes. The big polygon with green color (Fig. 5) is a pool, and the small polygon with blue color is a field, illustrating that we can set a different display property for each different kind of farm object.
From the different GPS receiver data test, we found two different NMEA string formats in use in different brands of GPS receivers. The system can also be adapted for both of these formats.

4.2. **Layer management**

In this example, we suppose there are some pools (polygon layer) and irrigation canals (line layer) which composed of water system. There is also some pathway (line layer) and some buildings (polygon) in a farm. The most important is the fields layer, which is composed of the sample points layer (points layer) and farmland (polygon layer). These different kinds of layers make up a farm, and through the layer management interface, we can control the “eyeable” and “hide” status. Also, we can select one layer as the current edit layer, so the new added GPS information and new property information can be related into that current editable layer (Fig. 6).

4.3. **GPS data and sensor data acquired simulation**

This system is designed to acquire both GPS data and sensor data. But currently, we haven’t any sensors which can output ASCII by control. To test this part of functions, an AgGPS 132 receiver is used to set up a small test platform.

The Trimble AgGPS 132 has submeter precision when utilizing differential correction and it is located close to a base station. More importantly here, the AgGPS 132 supports a kind of PTNLAG001 format NMEA string. With the PTNLAG001 string, the sensor data in ASCII format can be combined with the GPS signal and then transferred into the PPC. The exact PTNLAG001 string format is shown below:

\[ \text{a} \text{b} \text{c} \]

(a) Clause identifier.
(b) At most, 66 ASCII characters can be read from port A or port B of the GPS receiver, based on its initial setting.
(c) Check sum.

So information from sensor in ASCII format can be combined with GPS data, and GPS data and sensor data can be output from the same communication port. Then the combined data are transferred to the PPC through the serial port. Fig. 7 is the connection sketch map.
With this connection configuration, we can set the Port B input parameter as “TEXTA”, and the output parameter as “None”. The input parameter of Port A can be “None”, and its output parameter can be NMEA.

Then at port B, we use another computer to output ASCII code to simulate the sensor signal. In the PPC simulator, the sentence beginning with “$PTNLAG001” is the information from the sensor. So we can use a different sample plan to deal with the “$PTNLAG001” data (Fig. 8).

For instance of sensor data acquirement: at first, signals arrived at main thread. Then the main thread sends a signal to CMapTrackingLayer to update tracking data, and also sends a signal to the current active layer. If the sample plan of the current active layer is “All auto sampling” then only the sample point layer of field object can have such sample plan. As we know, GPS data are usually output at a frequency of 1 Hz. We can also set the sensor data output frequency. Thus, the system can acquire a serial of valid GPS positions and valid sensor data according to the data frequency. When data from a sensor is acquired, the system saves the sensor data and along with the most recent GPS data as a data set, and dismisses all other GPS data without sensor data.

If the plan “Manually after-sampling” is used (and after acquisition of some GPS data within a field), the interface below (Fig. 9) is used to input some extra information.

5. Conclusion

A field information collection, control and management system was developed and the utility of this system was demonstrated for use in PA. The main technologies used in this system include GIS, data models, spatial analysis method, Windows CE database technology and sampling control. Considering the uniqueness of farms and production practices, an object-oriented farm model was developed to describe farm objects properly, edit sampling plans conveniently, and to perform sampling control. The research work and development of this system will serve as a foundation for development of field data collection equipment.

Acknowledgements

This study was supported by National Science and Technology Support Program (2006BAD10A09), the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE, P.R.C., Natural Science Foundation of China (project no. 30671213), Science and Technology Department of Zhejiang Province (project no. 2005C12029). Natural Science Foundation of Zhejiang (project no. Y506152).

References


