IDR visor map: a simple and friendly geographical information system
J.D. García-Consuegra, J.A. Gallud & F. Abad
IDR-S. Teledetección y GIS, Departamento de Informática, Universidad de Castilla-La Mancha, Spain
e-mail: {jgallud,jdgarcia}@info-ab.uclm.es

Abstract
Nowadays, the construction of a Geographical Information System on a commercial package takes up a great deal of time and resources. However, the distribution of such an application can be more expensive than the construction because of the need to distribute the commercial package with the application. Usually, commercial packages cover this lack by providing reduced versions from their powerful applications. With the IDR Visor Map we have developed a low-cost and at the same time powerful GIS which allows us to distribute GIS applications on a friendly workbench (Windows NT and 95). The IDR Visor Map has been developed using an object oriented paradigm (OO). This paper shows our experience in the design and implementation of a new GIS called IDR Visor Map, as well as some of its capabilities. The addition of some classes to the powerful Microsoft Foundation Classes (MFC) has developed this application. Such classes provide the layout to integrate and to monitor raster, polygon and vector images, a key aspect in the process of managing spatial information. This information is used together with the ODBC interface to obtain additional data from a georeferenced database. The main advantages of the IDR Visor Map are scalability and flexibility, since our tool could be improved to offer the same capabilities that can be found in commercial GIS. On the other hand, the distribution of the IDR Visor Map tool has a minimal cost; furthermore, ad hoc applications can be easily implemented using our libraries. Our experience in this area says that a GIS, as the IDR Visor Map, must be able to answer all requests and demands from users, what means that features like flexibility and scalability must be taken into account in the construction of a GIS application.
1 **IDR Visor Map's Features and Capabilities**

The origin of this project called *IDR Visor Map* was in the need to provide sophisticated data for our real clients. The Regional Developing Institute's Remote Sensing and GIS Section receives a great number of requests about partial studies of a geographical zone. Such studies should be constructed to generate not only paper-based results but also to generate them in a visual and manageable way. We needed to get a Geographical Information System (GIS) to solve such problems [6,7].

Why have we designed a GIS? The main requirements of our clients are simple, like monitoring, printing, and other operations related to satellite imagery. We must offer not only a powerful and flexible tool but also, most importantly, a low-cost GIS. Thus, a first aim of such a project was to develop a low-cost Geographical Information System, which covers the needs of our clients[10,11,12].

Moreover, a number of libraries and workbenches have recently appeared, that might be thought good enough for our purpose [1]. This is, unfortunately, not the case: these tools do not provide the capabilities we would want to endow the GIS with. Such capacity and flexibility are more easily obtained by modelling the system using the object oriented paradigm.

Recently, there has been a noticeable increase in object-oriented approaches [4]. The organisation of spatial data in computer systems has received considerable attention in the field of GIS. In a simplified form, we are interested in knowing what is where and when, using an object-based concept of the "real world". There are several proposals of OO-GIS like GEO++ system [8] and some others that look like solving the implementation problem [2]. Such prototypes have a disadvantage: they are usually closed systems, so it is more difficult to adapt them to a particular case.

We have designed and included our classes in the Microsoft Foundation Class. Our purpose is to obtain a personal computer based-GIS; this criterion obliged us to undertake the study of this known class in order to incorporate several subclasses. This method has allowed us to design a powerful, flexible and low-cost GIS. Thus, we can scale our GIS as much as the standard libraries like MFC, OpenGL, ActiveX, OCX, DLL do.
GIS Technologies and their Environmental Applications 295

The IDR Visor Map GIS has a number of procedures grouped into three categories: visualization functions, database functions and GUI functions. The user can manage segment data, vector map data, raster image and polygon images, which can be related to alphanumeric information [3], as well as mix and filter these data. The IDR Visor Map manages such information by several layers, one for each data type. We have defined a set of classes to represent them. In this sense, the IDR Visor Map is much more than a simple integration approach [5], since in Visor Map we have established the grounds of a complete GIS.

The rest of the paper is organized in three sections: in section two we explain the class designed and its integration in the Microsoft Foundation Class; in section three we give an example of the IDR Visor Map possibilities. Finally we discuss the conclusions and future work.

2 Inside of IDR Visor Map

The implementation of the IDR Visor Map is explained in two stages. First, we describe the set of functions that are offered by our GIS. Then, we will explain how these functions have been implemented by adding some classes to the Microsoft Foundation Classes.

2.1 IDR Visor Map functionality

The functionality of GIS follows a known schema based on ten major categories [9]: capture, transfer, validate and edit, store and structure, restructure, generalize, transform, query, analysis and present. The IDR Visor Map offers a number of functions but not all of them, which will be implemented in a future release. The functions offered by our application are the following: validate and edit attributes, all the capabilities included in the store and structure category, query and information presentation.

The description given of the list of procedures and functions shows the functionality of our GIS which a user can employ in order to manage some spatial information together with one or more georeferenced databases, which provide additional information in alphanumeric format. The following enumeration of the procedures and functions is included in the validate & edit, data presentation, and store & structure categories:
GIS Technologies and their Environmental Applications

a) Functions to manage image and text files: A GIS must offer a number of operations which allow us to manage different data files, which are raster, polygon, segment and text. A dynamic link library has been developed in order to provide some procedures to process remotely sensed imagery.

b) Image visualization operations as zoom operation, multiple windows, etc.

c) Data operations: to add a layer, to eliminate a layer, to change data properties, etc.

d) Edition functions: these allow us to modify a file, to get a bitmap file with window information, etc.

e) Database capabilities: to acquire relevant information about the showed image (pixel info).

f) Database management: this includes the procedures necessary to maintain the databases.

In the next schema, we can see the visualization functions present in the IDR Visor Map. A discontinuous box means the function is not present.
Query and analysis are the main functions of GIS. So, in addition to the above list of functions, we are going to describe our database management system which form the main level of the IDR Visor Map. We have defined three databases to store three different data types, as follows:

a) Information about the database structure (a query generator uses this kind of information), together with auxiliary tables to store attribute values like colour, pen attributes, brush attributes and so on.

b) A database to manage data files with information about the maps generated and its structure.

c) A module with the procedures to visualizate all kind of images. In this database we have the tables needed to store all alfanumeric information related to each problem.

The IDR Visor Map offers a graphical user interface to access to all information types. Users open or create a MAP file when they init our application. At this point, they can add, view, hide, and consult a layer. IDR Visor Map is able to manage five layer types: raster, segment, polygon, text and punctual, represented by icons. The number of layers open in a MAP file is not bounded. Each layer can be related to alphanumeric data through the key field.

In the figure 1, we can see the IDR Visor Map showing a Landsat-TM satellite image in raster format from Castilla-La Mancha Region in Spain.

2.2 IDR Visor Map visualization module implementation

The most complex module of our application is the visualization code. In this section we are going to focus on describing the implementation details of visualization. The IDR Visor Map uses Microsoft Foundation Class to derive the required classes. MFC defines a root class named CObject and each Windows application includes four basic classes derived from CObject. These four classes are CDocument, CFrameWnd, CView and CWinApp. Functions implemented in these four classes allow us to manage a generic Window program. Nowadays, this visualization module has been become an OCX (OLE Control).
In the design and implementation of our GIS, we had to define the method to manage and represent five data types: raster data, polygon data, point type, vector data and alphanumeric information. It is not possible to manipulate adequately these five data types with only the functions defined in MFC. Our solution to this problem is to add a set of new classes to the MFC. Up to the present day, we have derived thirteen classes that can be divided into two groups: A first group of classes related to windows management and a second group to the classes that manage data images.

Let us show the main classes within the first group:

1. *CmapView*. This class defines all functions related to selection of the data types we desire to manage. CMapView is derived from CView. Also we define the model of data representation: it should be noted that we want to show four types of information. The most important function here is OnDraw which we will explain later.

2. *CmapDoc*. This class is derived from CDocument. In CMapDoc we have the data and functions that allow us to manage information sources, usually by means of files. An important function in
CMapDoc is AddLayer: this function allows the user to add a new layer onto an image.

3. CdialogOpen. CDialogOpen is a class derived from CFileDialog. This class manages the window dialogs related to file treatments.

4. CmapText. This class is derived from COObject and manages the alphanumerical information.

In the second group we can see the classes that provide the functions for problem solving:

1. **Vector type.** We have two specific classes supporting the vector data:
   1. CLayer. CLayer is derived from COObject. This class takes charge of supporting vector images in order to its representation and visualization.
   2. Csegment. This class is also derived from COObject and takes charge of showing the segments of any vector image.

2. **Raster type.** We have one only class for raster type: CLayerMPD. This class is derived from COObject and manages and represents the raster data type.

3. **Polygon type.** We have defined two classes associated to polygon images:
   1. CPolygon. We have derived this class from COObject. This class manages polygon data to provide image visualization and representation.
   2. CLayerPOL. CLayerPOL is a class derived from CLayer and takes charge of managing each segment and polygon to compose a polygon image. This is possible since CLayerPOL includes two objects of the MFC COBlist class with all segments and polygons.

4. **Icon and Text types.** Another two classes are defined to support icon and text information: CLayerIco and CLayerTxt.

We cannot explain all functions defined in each class, as it would take too long. However we show to explain the behaviour of the system by describing an example.
300 GIS Technologies and their Environmental Applications

3 The IDR Visor Map: An Example

The last section has showed how our tool is implemented. Now we are going to explain how the IDR Visor Map works from a user’s point of view. The user does not need to know anything about OOP (Object Oriented Programming) of the GIS: it is a design criterion.

A GIS user usually searches for geographical data of a zone by exploring the features of different images. As we can see in the figure 2, where a Landsat TM image (in raster format) of the Albacete province is shown.

Some related information can be obtained from this image (pixel info). Thus, selecting a determined pixel we can obtain its geographical coordinates as well as its digital value. When we are working with a classified image, information about the class to which it belongs (centroide, statistics, etc.) can be retrieved. The IDR Visor Map allows us to get a new image by overlapping a vector data on the raster image. For example, if we want to see which rivers cross the region and where they are, then we have to add a vector image (rivers).

Together with pixel info, this GIS offers the possibility of associating and consulting alphanumeric information of each data type by a previously made selection.

The IDR Visor Map also works in the opposite direction. The user can observe how an attribute value is spatially distributed. Also, the result (spatial objects or attributes of the spatial objects) of a query can be displayed in the form of a table and a graphic, as well as being render on the screen as a new layer or be highlighted.

It can be noted that in all this process we assume there are a set of files containing all kinds of data: raster, text, vector and polygon information. This means that a the IDR Visor Map user has to scan the images, introduce information and obtain the satellite data, so that GIS can work over the desired region.

This example has a number of limitations. It only shows a partial execution of the IDR Visor Map to give a global point of view. The IDR Visor Map offers all GIS facilities and depends on the user's knowledge about GIS and Remote Sensing.
4 Conclusions and Future Work

In this paper we have showed the implementation of a low-cost GIS called the IDR Visor Map. The design of our application using a number of standard libraries, allow us to add capabilities in a easier way. As an example, the newer releases allow polygon to be filled in with bitmap files of 32x32 pixels or more in Windows 95 and NT, respectively, instead of 8x8 bitmap files of Windows 3.x.

The IDR Visor Map represents an example in the effort to extend the use of Geographical Information Systems by using hardware platforms based on PC’s or workstations and by designing a low-cost and GUI application. Our GIS offers a variety of functions or procedures to manage and at the same time process raster images, in order to consult a geographical area. The IDR Visor Map together with the DLL libraries have been shown to be good enough to cover our clients’ needs.

Object oriented programming has permitted an adequate structurization of the geographical information that is to be analyzed. In addition, OO modelling has provided the new class generation from an initial class by using the heredity feature. In particular we have defined a number of new classes to manage special data, of these, CMapText, CSegment, CLayerMPD and CPolygon are the most important. The IDR Visor Map hides class hierarchy from the user, so that he can focus on obtaining information from a set of images.

At present, we are working on extending the IDR Visor Map capabilities such as manipulating segments as well as polygons, setting the location of points, text, introducing “extended query”, providing an intranet version based on ActiveX and introducing the 3D representation using the OpenGL standard for Windows 95 and NT.

References

302 GIS Technologies and their Environmental Applications


