MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

A Kernel System for 3D GIS Applications

Ying Ming LIN † Yutaka OHSAWA ‡ Masao SAKAUCHI †

Institute of Industrial Science †
University of Tokyo
7-22-1, Roppongi, Minato-ku
Tokyo, 106 Japan

Information and Computer Sciences ‡
Saitama University
255, Simo-okubo, Urawa
Saitama, 338 Japan

Abstract

A kernel system which can support 3-dimensional applications of GIS is proposed. Internally, spatial data are managed by a 2-dimensional data structure, since most of the practical GIS data are 2-dimensional. Information about the Z-axis, i.e. direction of height, is separated from coordinates describing objects, and treated as one of the attributes attached to objects. This is just the same as the so called 2.5D description which is applied extensively to describe 3-dimensional objects in GIS, except that we extended the 2.5D description to be able to hold multiple height information. We call the extended 2.5D description the 2D-h description to distinguish it from the traditional 2.5D one.

The proposed kernel contains a set of C++ class libraries defining some basic spatial objects. For different applications, special objects can be derived easily from the basic libraries by using the object-oriented technic. We show the effectiveness of 3-dimensional supporting ability of the proposed kernel system by applying it to implement a new expression technic for walkthrough, which display objects in different details according to their distances from the view point.

1 Introduction

There are two approachs for development of 3-dimensional GIS (Geographic Information System), one from the data managers' view and the other from the application developers' view. The former asserts that GIS consists of 3-dimensional data, data model and data structure is 3-dimensional. Whereas the latter asserts that regardless of the internal management, GIS providing 3-dimensional interface for applications is 3-dimensional. In fact, the first approach is still at experimental stage, and there exist very few such 3D GIS in practice[3][6][1]. The main problem of this approach is that at present time it is hardly to obtain the fully 3-dimensional GIS data for economic and technical reasons [2]. Thus, the second approach can be thought as the transitional approach for 3D GIS development in the near future. Before the 3D data becoming practical, existing 2D data and height information can be used instead of complete 3-dimensional data.

In this paper we describe a kernel system for 3D GIS applications, which is developed in an object-oriented fashion by following the second approach stated above. For spatial objects in GIS applications, geometric properties and related operations are encapsulated to define the abstract data types[4]. Internally, spatial data are represented by the 2D-h (2-dimensional and height information) representation which will be described below, and managed implicitly by a 2-dimensional spatial data structure, the GBD tree[5]. Moreover, in order to support 3-dimensional processes, we extend the 2D GBD tree by attaching height information to it. And the effectiveness of such 3D supporting abilities is examined by a walkthrough example. This example also proposes an expression technique for 3D animation which can be applied to simulate the real view in the 3-dimensional space.

2 System Overview

The kernel system is composed of the main body and the interface parts. The main body consists of a sets of C++ class libraries and the spatial data management structure. And the interface part consists of environment-dependent operations (Figure 1).

2.1 Class Libraries

There are two C++ class libraries in our kernel system. The basic one is a group of general-purpose class definitions for 2D geometric objects like points, lines, polygons, circles, arcs etc.. A special grouping class is used to represent irregular shapes which are composed of simple geometric pieces. In addition, containers like lists, sets, tables of these basic classes are also supported. These classes can be applied by not only GIS but also other graphic applications.

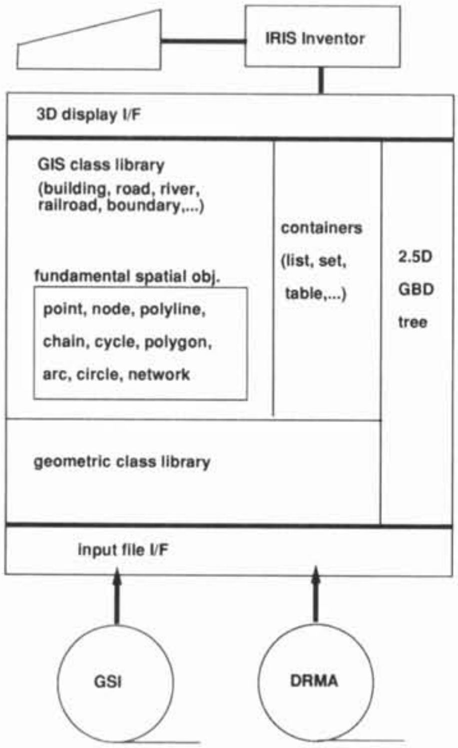

Figure 1: Block diagram of the proposed kernel system

From basic classes we derived a set of GIS-oriented classes, such as roads, buildings, contours, subways etc.. To each class definition, a special multi-valued attribute called height is attached. The height attribute provides information of object's shape description in Z-axis. By using this information, the proposed system thus can support 3D operations. This will be described in detail in the next section. Instances of the same class are grouped in one layer, for example layer of roads is the collection of all roads in a map. And a map may contain multiple layers, one for each category[7].

2.2 External Interfaces

For portable consideration, we separate the environment dependent operations from the system. Mainly, the interface part is composed of operations between *I/O* devices and the main body of the system. The display interface is implemented with the IRIS 3D inventor library. However, this can be replaced by any other graphic library, e.g. the X-window system, efficiently by only redefining a new interface for it. Similarly, *I/O* for files or database with different formats can be carried out by simply building corresponding interfaces.

3 Spatial Data Management

When Considering the data space of GIS, although it is 3-dimensional, its extent of the XY-plane is far beyond that of the Z-axis (the height direction). And in most of the cases, we are more interested in information of the X and Y axes than that of the Z-axis. In other words, unlike the 3D CAD systems which deal with the information of the 3 axes equally, in GIS, information about the Z-axis is usually ignored or simplified. As a matter of fact, as we can find in the existing GIS data, the information of Z-axis is not contained as part of coordinates in object descriptions. Instead, it is treated as one of aspatial attributes of objects. For example, buildings and contour lines are usually described by the 2.5D representations which consist of a set of XY-coordinates and a single height information describing shapes in the XY-plane and the height in Z-axis direction respectively.

3.1 The 2D-h Representation

Obviously, the 2.5D representation is too restricted, since only one height information is allowed for each object. For example, a sloping road may need multiple height information to describe its slops. Clearly, description and management of such objects cause no problem if we use the 3D data model and data structure. However as we stated above, this approach is inefficient, since most of the practical data are 2-dimensional at this stage. To fill the gap between 3D data and 2D data model, we propose the 2D-h representation (2D data plus multiple height information) by simply extending the height attribute to become multi-valued. And let information about Z-axis (i.e. Z-coordinate, from the 3D's view) be components of the height attribute. Since the 2D-h representation can hold 0 to multiple height information, all 2D, 2.5D and 3D description can be represented in a unified way. Indeed, this is conceptually the same as the 3D representation, with components of height equal to Z-coordinate. But physically, they are quite different. The Z-coordinate is part of the 3D coordinates, while the height is a attribute of objects. This difference will become much clearer, when we considering the data structure, as shown in the following.

3.2 The Extended GBD Tree

The GBD tree is a point-oriented spatial data structure. Point data are arranged in the B tree-like tree structure by their Z-order which are generated by interleaving coordinates of axes in binary expression. In every node of the tree structure, a 2D MBR (Minimum Bounding Rectangle)is attached, which indicates the region enclosing all its descends (subtrees for non-leaf nodes, objects for leaf nodes). Since no any information about the Z-axis are kept in the tree structure, it can not support queries about the the height, the Z-axis direction. Therefore, we extend the original 2D MBR to become 3D MBR by attaching the extremes of height informations, ie. the Z_{min} and the Z_{max}, of descends to every node. Although the extended GBD tree is similar to a 3D R tree, it differs from any 3D data structure by the fact that overflow splits and underflow merges never occur in the Z-axis, because there is no Z-axis in the 2D GBD tree. And this distinguishes the difference between the 3D and the 2D-h representations. By using the 3D MBR's information, the extended GBD tree can thus supports 3D applications as efficiently as the 3D R tree can. In our kernel system, all spatial data represented by the 2D-h representation are managed implicitly by the extended GBD tree. However, data management is not restricted to GBD tree, any spatial data structure providing MBR's information for its nodes works as well.

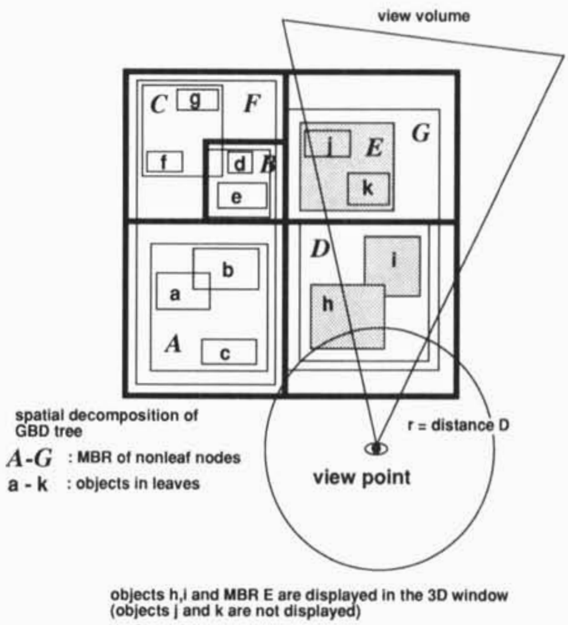

Figure 2: Selecting objects and MBRs to be displayed in the walkthrough implementation

4 The Walkthrough Example

The walkthrough is a typical 3-dimensional application of GIS, which gives users a simulated 3D view of an urban area. In order to get the animation effect, response time to user's operation is required strictly, typically about 1/6 second for one frame. This limits the number of objects being processed to be about 500 simple ones[2]. But in practice, it is quite common that the number of objects being handled exceeds ten thousands. And the time to generate one single frame may take a few seconds. As a result, the walkthrough then becomes meaningless; it just looks like a series of static pictures but not an animation.

Even though we can improve the efficiency of the rendering processes, there still exists an upper bound for the quantity of objects which can be processed during the required time, i.e. in 1/6 second. Hence the only way is to reduce the amount of objects being displaied. Here we propose an expressing technology based on the idea that for distant objects it is unnecessary to display them in detail. Consider human viewing, when objects are far from our eyes, we can not see their detail and can only distinguish roughly their outlines. This suggests that for distant objects we may display their representing objects, the 3D MBR enclosing them, instead of their details. That is, for distant objects, they are grouped into one single box to be displayed. The objects being displayed thus can be suppressed greatly. And for the grouping policy, information of 3D MBR in the extended GBD tree can be applied immediately.

Figure 3 is an example of continuous snapshots of the proposed walkthrough expression which is carried out by applying the foregoing kernel system as follows (Figure 2).

Step.1 Let the the view volume be V.

Step.2 In GBD tree, traverse nodes overlapping with V from the root to leaves.

Step.3 Calculate the distance d between the view point and the MBR of the node encountered.

Step.4 If d is less than a predefined distance D, descend to a son node and goto to step 3. Stop when all sons had been traversed.

Step.5 If $d > D$ or a leaf node is encountered, send the MBR of node or objects in the leaf to the window system to display.

The value of D can be defined dynamically according to the number of objects being displayed. Large values of D reduce objects being displayed, and thus decrease rendering time. Whereas the smaller D can get a much precise expression for objects. And resolutions of objects are adjusted automaticly. Objects look clearer and clearer as we get closer to them. That is MBRs' of nodes are presented level by level from the root to leaf as we move toward them.

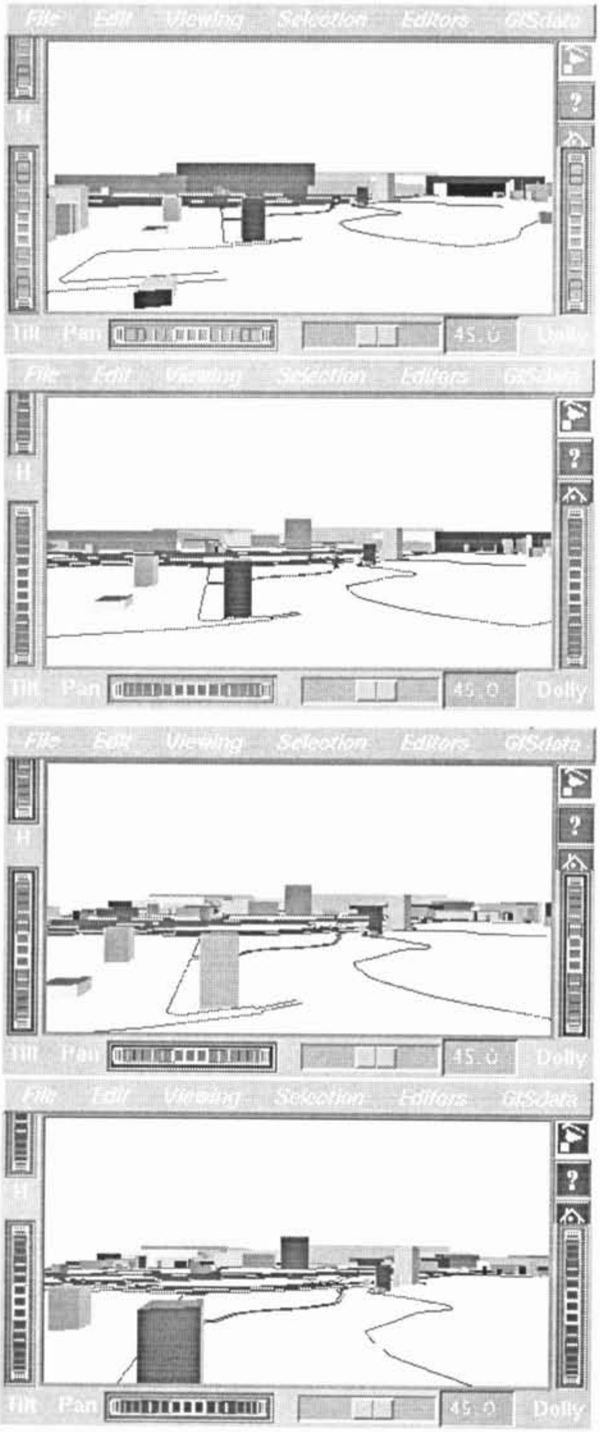

Figure 3: Continuous snapshots from proposed walk-through

5 Conclusion

A kernel system for 3D GIS applications is presented. A unified representation for 2D, 2.5D and 3D descriptions is proposed. The 2D-h representation and its implementation can be used as a transitional approach for the development of 3D GIS in the near future. By using the proposed approach, existing 2D GISs can be extended easily to be able to support 3D applications. Besides, this paper also proposes a new expression technology of 3D animation for view simulation in GIS space, and shows the effectiveness of the 3D supporting ability of the kernel system proposed.

References

[1] Takeshi Kawaguchi, "A study on the classification of map visualization in GIS", The 4th Symposium of Functional Graphic Information System, pp.79-83, April 1993.

[2] Yasuaki Nakamura, "Three Dimensional Map Data Management and High Speed Walkthrough : Data Management and Spatial Query",Image Lab., pp.68-71, Feb.1994.

[3] Osamu Imai, "A Review of Recent Three-Dimensional GIS Research", The 3rd Symposium of Functional Graphic Information System, pp.55-58, April 1992.

[4] D.J.Abel, "SIRO-DBMS : a database tool-kit for geographical information systems," *Int. J. Geographical Information Systems*, Vol. 3, No. 2, pp . 103-116, 1989.

[5] Y.Ohsawa, Masao Sakauchi, "A New Type Data Structure with Homogeneous Nodes Suitable for a Very Large Spatial Database," *Proc. of 6-th International Conference on Data Engineering*, pp.296-303, Feb.1990.

[6] Kaneko K.,Kuroki S.,Makinouchi A., "Design of 3D CG Data Model of 'MOVE' Animation Database System," *Proc. of 2nd Far-East Workshop on Future Database Systems*, pp.364-372, 1992.

[7] J.Star, J.Estes, *Geographic Information Systems : an introduction*, Prentice-Hall,Inc., 1990.