# Implementation of a virtual environment system based on geographical information system and environmental models

by

Chen-Hsiang Yeang

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1996

Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Electrical Engineering and Computer Science
May 1st, 1996

Certified by . . . . . . . . . . . . . . . .
David Zeltzer
Principal Research Scientist
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . .
-----ric R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

# Implementation of a virtual environment system based on geographical information system and environmental models

by

Chen-Hsiang Yeang

Submitted to the Department of Electrical Engineering and Computer Science
on May 1st, 1996, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

## Abstract

In this thesis, virtual environment, GIS, 3D graphics rendering, and environmental model simulation, are integrated and implemented within a single system. This system, which I call GISVE, emulates an environment of a geographic place based on the 2D GIS database and 3D objects generated from the 2D database. It allows users to navigate through and modify this environment by adding or deleting elements of GIS database. Furthermore, it applies water runoff, soil erosion carried pollution, and fire diffusion models to the modified environment to simulate environmental impacts resulted from users' modification. The application domain of GISVE system is focused on landscape architecture. Domain experts are consulted during the task analysis stage to determine the specifications and features of the system; after the prototype is finished it is evaluated by a domain expert to determine whether it achieves these specifications; the development stage ends with a project test by comparing two landscape development proposals using GISVE to determine which one is better in terms of environmental concern. This testifies the achievement of the original goals.

Thesis Supervisor: David Zeltzer
Title: Principal Research Scientist

# Acknowledgments

I would like to thank the people who gave me precious help in research and writing works. Many thanks to my thesis supervisor, Dr. David Zeltzer; I have learned about how to do research work gradually from our discussions. Thanks to Mr. Show-Chung Ho and Mrs. Sing-Ju Ho of Hsin-Yi Foundation for partially sponsoring my research. Thanks to Mr. Nathaniel Durlach and Dr. Jeng-Feng Lee of MIT's Research Laboratory of Electronics in indicating technical resources and in helping me to define the general objectives of this thesis research. Thanks to Professor Kristina Hill of Department of Urban Studies and Planning of MIT in providing information about GIS and landscape architecture and in introducing landscape architects to participate the task analysis and evaluation sessions. Thanks to Professors Carl Steinitz, Stephen Ervin, and Michael Binford of Garduate School of Design, Harvard University; their opinions in the task analysis session helped me to form the system specifications. Thanks to Ms. Shirley Kressel in giving evaluations of the entire system. Thanks to Ms. Dorrie Hall of Research Laboratory of Electronics in technical support and consultancy of software. Finally, I would like to express my gratitude to my parents, Mr. Chia-Chu Yeang and Mrs. Lan-Chun Hsu, and other family members. I can not finish this thesis without their strongest support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Virtual environment (VE) is a technology that uses computers to simulate an environment. Through the multi-modal human-machine interfaces, users are immersed in and interacting with this artificial environment. For the past few years, the fast development of the semiconductor industry has caused the hardware prices to drop quickly, which has reduced the cost of building interactive and high quality user interfaces and has spreaded out the application domains and users of VE technology. Today VE is widely used in training, education, medicine, entertainment, scientific visualization, and art.

Geographic Information System (GIS) first came out in the 1960's. Basically, it is a digital map which uses the concept of layers representing different geographic data. By manipulating the layers, different combinations of geographic data are displayed on the map (computer screen). This makes the map more flexible, and facilitates the understanding of spatial correlation and variation of geographic data. GIS is widely used in geography, ecology, landscape architecture, and natural resource management.

3D Computer Aided Design (CAD) is another widely used computer technology. Using the graphical display, the user can design his/her work on computer screen and see different perspectives of objects. Because keyboard and mouse manipulations substitute for the laborious hand drawing work, the design work is greatly facilitated.

This tool is widely used in industrial products, architecture, and graphic art design. Environment model simulation is thriving these years due to the growth of computational power and the reduction of computer prices. This covers many fields of science, including hydrology, geology, atmosphere science, ecology. Today using computers to do model simulation becomes an essential step in all fields of scientific research.

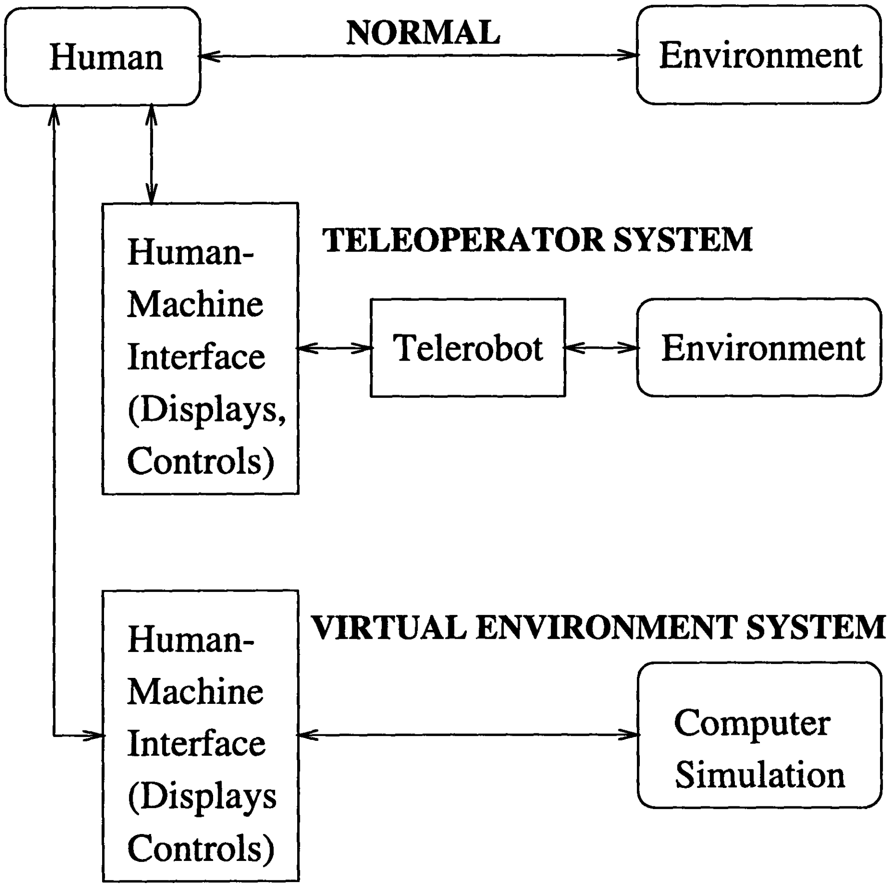
All of the four applications (VE, GIS, 3D CAD, model simulation) are computer based. However, they are so diversified that to integrate them together in one system seems difficult. It is really difficult, but will be very useful. If we can build up a virtual environment system that contains a GIS database, has CAD functions to allow users to modify the environment, and can do environment model simulations to let users know what will happen if they make such changes to environment, it will be very convenient in working on landscape design, site selection of important constructions (for example, power plants, dams), environmental policy decision making, natural resource management and conservation.

The goal of this research project is to create such system: to implement a virtual environment based on a geographical information system and environmental models, that can aid the design process of landscape architecture.

## 1.2 Virtual Environment

Virtual reality, virtual environment and telepresence: these three terms are similar but not totally identical. They all use computers to create a human- machine interface. Through this multi-modal human-machine interface, the user has the impression that he/she is immersed in this environment. The difference between virtual environment and telepresence is the existence of real environment. A virtual environment system consists of a human operator, a human-machine interface, and a computer generated environment. The computer generated environment contains "virtual objects", they have positions, shapes, sizes, and other features as do real objects. Users or operators can feel these objects through the human-machine interface; because of

Figure 1-1: Comparison between a teleoperator system, a virtual environment system, and a regular system


Human
NORMAL
Environment
Human-Machine Interface (Displays, Controls)
TELEOPERATOR SYSTEM
Telerobot
Environment
Human-Machine Interface (Displays Controls)
VIRTUAL ENVIRONMENT SYSTEM
Computer Simulation

the quality of sensory inputs/outputs, the user will have the feeling he/she is in this environment. A telepresence system consists of a human operator, a human-machine interface, and a telerobot. The telerobot is exposed in a real environment which is either remote from the human operator (for example, a spacecraft) or in a scale which the human operator cannot detect (for example, microsurgery). Environmental signals are sensed by telerobots, transmitted to the human-machine interface, and presented to the human by means of display devices in the interface. To sum up, virtual environment is to immerse a human agent in a computer generated environment, telepresence is to immerse a human agent in a real environment [DM94, pages 17–19].

The human-machine interface can contain several sensory input/output devices.

We briefly introduce each modal device as follows.

The most common and important sensory output is visual display. There are several alternatives: computer monitor, flat panel display, head-mount display (HMD), stereo glasses or goggles. The choice of different displays depends on features of specific applications. For example, for some high-end military flight training systems, head-mount displays are put on the helmets for displaying the scene interactively responding to the head movement. For visualization of scientific data, the quality of graphics such as colors and boundary lines are more important, so a high definition monitor will fit this need.

Audio output is important in some applications. In flight or vehicle training and entertainment, the stereo sound offers users another cue to locate 3D positions of objects; background audio adds the feeling of an immersive environment; the human voice generated by computer synthesizers is often used in guiding instrument manipulations or indicating what to do next.

Haptic devices form another intricate sensory input-output part. Most haptic devices have both sensors and motors on them. They can detect the direction and intensity of forces superimposed on different parts of them, and generate different forces on different parts, forming a "biofeedback loop". Haptic devices have several forms: data gloves [PT92, pages 47–49], wired clothing [PT92, pages 265–266], and PhanToM [Pio95, page 17]. PhanToM is a haptic robot which has sensors to detect the direction and magnitude of the force superimposed on it and motors to insert forces to the device. Its shape is like the tip of a pencil or a scalpel; the user holds its handle and move it in the 3D space to simulate the movement of writing, operating the scalpel, or shifting a stick. Haptic senses are crucial if the application domains are related to body movements and manipulations (especially hand manipulations), such as surgical training and telemedicine, instrument operation, athletic training.

Traditional sensory inputs for computers are the keyboard and the mouse. Keyboards are useful in textual communication between the human operator and the computer, but they are unable to represent movements. The mouse is appropriate for 2D movement but has difficulty in representing 3D movement. However these input devices are the most common and will cause the least confusion to the user.

Speech input is a substitute for keyboard input. The speech recognizer on the computer can recognize the words spoken by the operator, so that computers will respond to the user's commands by doing adequate actions. So far natural language processing still has a long way to go, but some VE systems have already implemented speech recognizers that users can employ to direct the computers [Pio95].

There are several methods for tracking body movements. The head tracker is a small sensor put on the top of a helmet, which can track the rotation of the head. The sensors can be optical, electromagnetic, or ultrasonic. The head tracker is usually integrated with a head mount display to form the interactive visual display device. The head tracker keeps track of the 3D position and orientation of the viewer in the virtual environment, the computer uses the position and orientation data to change the 3D juxtapositions of objects in the environment, and updates the 3D scene on the HMD [PT92, pages 137–142]. Data gloves and other haptic devices can be used to track the 3D positions and orientations of hands, legs, or other parts of the body.

The forms of sensory input/output used in a specific VE system depend on the application of this system. Different tasks have different preferences in sensory modes. For example, in a flight simulator VE system, head-mount display and force feedback system simulating stick movement are essential components, while keyboard inputs and stereo sound environment may not be so important; in an entertainment VE system, a combination of 3D graphical display and stereo sound can generate tremendous effects, while the haptic device that senses slight finger movements is not necessary in this system.

Application domains of virtual environment technology are spreading quickly and have become more diversified. In some fields VE applications are already popular, such as entertainment; in some other fields, the studies and research works of applying VE in training or aiding work are germinating, such as medicine. Some of current or potential application fields are as follows:

1. *Military training* The earliest application of virtual environment was for mili-

tary training. VE is a good substitute for expensive aircraft training systems. One of the first flight simulators to use computers, built by General Electric Company's Electronics Laboratory for the Navy in 1972, was called ADM (Advanced Development Model) [PT92, page 35]. The head-mount display and polhemus head tracking devices were put on the pilot's helmet to obtain a 3D interactive display. Other military training applications include submarine officer mission training [ZL93], and Distributed Interactive Simulation (DIS) to reconstruct the Persian Gulf war.

2. *Medicine.* Medical application is one of concentrations of VE research works. High quality 2D and 3D images can describe all parts of the human body precisely, so it is very useful in anatomic education and research. A famous project is the Visible Human Project by the National Library of Medicine; it uses MRI (Magnetic Resonance Imaging), CT (Computer Tomography), and cross section pictures to construct a complete 3D representation of a human body. (The URL address, http://www.nlm.nih.gov/extramural_research.dir/visible_human.html) Another application domain is surgical training, for which the high quality visual display of microscopic images and force feedback devices are used to simulate laser scalpels, so that the surgeon can receive training in high precision microsurgery [HD$^{+}$93]. There are many other research projects going on several fields such as telemedicine or psychotherapy [DM94, pages 395–405].

3. *Scientific visualization.* The 3D graphics offers a good way to visualize abstract scientific data such as fluid flow and material property. Some systems use visual displays concomitantly with data gloves to generate forces simulating interaction forces of drug molecules [PT92, pages 42–44].

4. *Entertainment.* The applications in entertainment may be the most attractive directions for VE or VR research in economic terms. Many products are already commercialized, including the VR rooms in Disneyland, Universal Studio, and many 3D games. The integration of virtual reality and computer animation is the tendency of film industries and game software markets.

## 1.3 GIS, environmental model simulation, CAD, and landscape architecture

In this section, the basic elements constituting the GISVE system are introduced: GIS, environmental model simulation, computer aided design. The domain of its application - landscape architecture, is also introduced. In later chapters the ideas about and methods of combining these elements within a system will be discussed.

### 1.3.1 Geographical Information System

Geographical information system (GIS) was first used in 1960's in US and Canada. It is used for two purposes: to represent collected geographic data in an appropriate form that can be easily understood, and to facilitate the analysis of the spatial correlation of the data. Currently it is widely used by governments, universities and businesses in diversified application domains; it is also embedded with increasing amount of information from different sources, such as remote sensing, satellite pictures, GPS (global positioning system), and environmental models. These addings have made better able to satisfy the demand for different uses.

The fundamental issue in GIS research is how to find out the appropriate forms for representing geographic data. Most geographic data is continuous and intermingled with other types of data. In order to transform it into a form which is both easily handled by computers and recognized by people, we have to discretize the data and parametrize its properties. People use several models in geographic data representation; which model is used depends on its specific application and the properties of the data collected [Goo93].

The most commonly used model is the layer model: each data is treated as a layer, and if we want to put several types of data together, we can overlay them on a single map. The layer model was adopted before the invention of computers. It has a long tradition in cartography to overlay different layers on one paper to facilitate cartographic modeling such as calculating areas or distances [Tom90, pages 6–12]. The layer model is widely used for the convenience of data processing.

Another model similar to the layer model is the field model. In this model, spatially distributed data can be represented by a field if it has values at each point in the space. The topography (elevation) of a place, for example, is a field over the 2D plane; in mathematical language, elevation $z = f(x,y)$, x and y are plane coordinates. The complexity increases when the dimensions of domain and range increase, and usually only scalar or vector fields are used in GIS. Since the field is continuous, it has an infinite number of values; several methods are used to sample the values: irregular or regular point sampling, contours, polygons, cell grids, and triangular nets. The field model, like the layer model, is easily adapted to data processing, and it is especially suitable for applying distributed model simulation. It is, however, difficult for users to understand and recognize.

The object model often serves as a complement for field and layer models. Geographic data is represented by an "object," rather than a continuous-value field over the space. For example, "Lake Ontario" will be represented as one single object, rather than a collection of cell grids with the value "lake." Object model representation is influenced by an object oriented programming concept of computer science. Although this allows the user to comprehend and manipulate data easily, it is a little more difficult to use in data processing and model simulation. Despite of the problems that exist, however, current trend of GIS development is toward the object model.

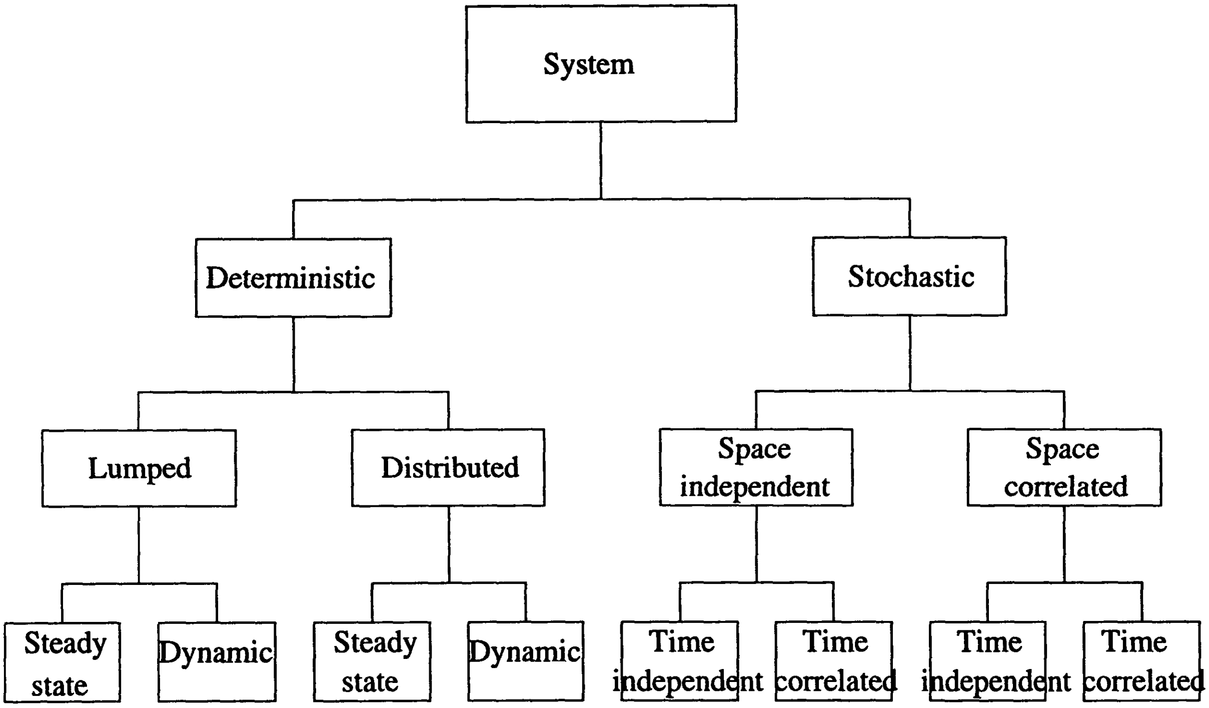
### 1.3.2 Environmental Model Simulation

Environmental model simulation is a very diversified field that covers many disciplines of natural sciences. The earliest mathematical model used to analyze ecological phenomena can be traced back to Lotka [Lot24, page 460], who used ordinary differential equations to simulate intra-species interaction. With the progress of computer technology, computations of complicated quantitative models become feasible. Today scientists of all fields use computers to simulate their models.

Generally there are three important parameters in building a model for computer simulation: randomness, space and time. A natural or social phenomenon can be described either as a deterministic event or a stochastic process: for a deterministic system, all equations describing system behavior are known, so we can write them down and solve these equations either analytically or by numerical methods; for a stochastic system, there are processes unknown that we can only depict as random variables and observe their statistical properties. Space is another important concern in determining which model is being used in simulation; some models are used for obtaining the aggregated quantities of the entire system, for example, the model of estimating the biomass of a forest, or the energy influx into and outflux from an ecosystem; some models are used for obtaining spatial distributions of some parameters, for example, the distribution of industrial pollutants over a town. Distributed models benefit more from GIS because they can obtain the distributed input parameters from GIS. Time is another important concern. Some models calculate state variables at equilibrium, and dynamic processes are neglected; some models, on the contrary, trace back or predict the system behavior in the dynamic processes of system evolution.
According to these three parameters, environmental models can be classified into various categories: deterministic versus stochastic, lumped versus distributed, and steady state versus dynamic.

Today using quantitative models to estimate environmental impacts due to global changes bring out important issues. For example, destruction of ozone layer, accumulation of carbon dioxide, and desertification of sbutropical region. Environmental scientists understand that they have to build a model to deal with the entire biosphere, which includes many global and local processes, such as climate, bio-geochemical cycles, ocean dynamics, and many other processes [Ste93]. This causes the collaboration of scientists from different nations and disciplines.

Figure 1-2: Model classification



### 1.3.3 Computer Aided Design

Computer aided design (CAD) is widely used in many different industries and professions, from electronic circuits to motor vehicles, from costumes to architecture. The CAD tool provides a graphical environment to display the designs or drafts on the screen; it also provides a simulation tool to test if the designs or drafts satisfy the desirable specifications. These features save the users a lot of time and effort. The users can modify their designs without repeating all the processes again; also they don't have to build many real or miniature models to test if they work as the users thought. CAD tools are highly developed and widely used in many industrial products, such as VLSI circuits, vehicles, and aircrafts. Most of these tools allow users to create or change the designs on computers, see and manipulate objects in 3D space with different perspectives (circuit design tool is an exception, because currently the layouts of most VLSI circuits are still planar), and test the designs by using simulation models to see if they work correctly. In architecture, however, most CAD tools are used only for displaying the designs in 2D or 3D view; in some tools users can create or modify their designs on 2D templates, but direct manipulations in the 3D space or quantitative simulations rarely appear in architecture CAD tools.

There are many research projects going on which try to make CAD systems smarter [SO92]. Some try to endow geometric objects with semantic representations to allow users to manipulate objects more easily; some try to incorporate the cognitive and psychological issues in implementing the human-machine interface; and some try to build a knowledge based expert system that not only supports the design, but also has the ability to design automatically. These efforts reflect the potential directions CAD tools may go in providing more powerful and convenient help to the design of products.

### 1.3.4 Landscape Architecture

Compared to the history of virtual environment, GIS, CAD, or model simulation, the history of landscape architecture is much longer. Frederick Law Olmsted, designer of

New York City's Central Park, coined the term landscape architecture in 1858 [Lau86]. In early years the phrase "landscape architecture" was vague and easily confused with other terms like environmental planning, urban design, and site planning. Generally speaking, landscape architecture is taken as an extension of architecture (by some schools of thought), so that architecture covers the design of buildings, while landscape architecture covers everything else in the environment; it can be the landscape of a natural environment like a national park, preservation land, or the landscape of an artificial environment like a garden, road, public space.

Traditionally embellishing functions played important roles in landscape architecture. Most public parks, gardens, and open land built in the 19th or early 20th centuries were for the city residents' enjoyment of nature [Mar91]. Public health was another important concern, since plagues and epidemics were prevalent from medieval period and they occurred occasionally even after the industrial revolution. The public health related landscape work includes sewers, septics, and garbage collection system. To complete these infrastructures was the major mission of landscape architecture up to the early 20th century. Recently environmental crises and ecological concerns become the main issues of landscape architecture; although the aesthetic part is still important, more scientific parts are involved in dealing with environments; all of these evolve landscape architecture into a comprehensive, multi-disciplinary process.

Practically landscape architecture is divided into four categories in terms of the geographic scales they deal with.

1. The first one is landscape evaluation and planning. It is concerned with the systematic study of large areas of land and has a strong ecological and natural science base in addition to a concern for visual quality. Many experts including soil scientists, geologists, economists, etc., take part in this process. The product is a land use plan or policy recommending development proposal.

2. The second one is site planning. Site planning is the process in which the characteristics of the site and the requirements of the program for its use are

brought together in creative synthesis. Elements and facilities are located in functional and aesthetic relationships.

3. The third one is detailed landscape design. It involves the selection of components, materials, and plants and their combination in the three dimensions as solutions to limited and well-defined problems such as entrance, terrace, amphitheater, parking area, and so on.

4. The fourth one is urban design. It involves the location of buildings, the organization of the space between them for circulation and public use, construction of infrastructures such as roads, sewers, electricity, and pollutant processing plants. Currently urban design is taken as a separate field from landscape architecture, most architecture or design schools have separate departments of these two disciplines. However, the distinction is not so inflexible and the interrelationship is pretty close.

## 1.4 General Objectives

The general objectives of this thesis can be discussed via two aspects: integration and application. In the integration aspect, the author intends to build a system integrating several computer application tools described above, it worths pursuing and studying because so far there are very few (if any) such systems. In the application aspect, the author intends to explore the possibilty of implementing this system for the application of environmental impact assessment in landscape architecture; it is also a pioneering work because so far the extension of VE in the application of this field is still at the initial stage.

In previous sections, the basic concepts and application domains of geographical information system, computer aided design tool, and environmental model simulation have been introduced. All of them are very useful in some specific fields. However, there are also intrinsic restrictions for these three computer tools.

The main advantage of GIS is that it displays spatial correlations of different data layers clearly. Like a map, GIS offers a channel for users to understand the spatial variation of collected data; better than a map, GIS allows users to manipulate data, either to display different layers or do some calculations on data. The drawback of GIS is the lack of representing temporal variation and 3D spatial variation. Most GIS systems can only display static data; in order to show the temporal variation, a set of data layers corresponding to the data at different time slots will be taken as consecutive "frames" and played sequentially, just like computer animations. These frame data are either collected at different times, for example, the satellite pictures taking at different seasons, or generated by using dynamic models, for example, the distribution of predators and preys by Lockta-Votera equations [BHT90, pages 247–250]. In most cases because of the diversity of application fields, GIS per se doesn't provide such models to generate animation. Another drawback of GIS is its restriction in two dimensional display. The topographic variations can only be represented by contour lines, and 3D structures such as buildings and trees will be replaced by abstract symbols on two-dimensional maps. Abstract representations will be fine if users care about only quantitative properties of scalar data (like sociologists studying demographic data); however, if the users want to grab the feeling of real environment, 3D display becomes crucial. For example, in flight simulation, the pilot should see the graphic environment that simulates the runways and fields of an airport; in landscape design, visual quality is important in determining the specific form of the design, pictures of different places are as important as quantitative data and descriptive documents.

3D CAD tools cooperate with GIS in complimentary means. Most computer aided design tools offer different 3D views of an object. The user can rotate, move, zoom the object, see different faces at different view points, and even go inside it to walk around (if the object is a house). The main advantage of 3D CAD tool is its 3D displaying ability which two-dimensional GIS lacks. Its drawbacks are the weakness in representing temporal variation and quantitative values. Like GIS, CAD tools usually lack ability to generate themselves a sequence of frame data to represent time evolution. Most CAD tools only display the 3D views of static objects, in other words, the objects themselves rarely change unless the user change them. It is unsatisfactory if we want to simulate the dynamics of the system. Another drawback of 3D CAD tools is the weakness of displaying quantitative properties. In contrast to GIS, a 3D object is not easy to manipulate on the values corresponding to different cubic blocks (on 2D GIS, manipulation is on grids or planes, which is much easier to handle) and do calculations on values on these cubic blocks. There are some GIS dealing with 3D structures, such as crude oil distribution on different sites as well as different depths; in general, however, 3D "manifold" is harder than 2D's to handle.

Environmental models, on the contrary, are specialized in representing the time evolution. In models based either on continuous differential equations or on discrete state transitions, time plays the most important role. Giving initial conditions and time evolution rules (differential equations or state transition rules), the models run automatically and generate time series results of specific parameters. Within the range of appropriate applications, the model simulation provides a good tool to understand time variation of parameters. The weakness of model simulation is on its ability in representing spatial variation. Some environmental models only offer time evolution of an aggregated parameter in a specific geographic area, (aggregate models); for example, the time evolution of total water runoff in a specific area after a heavy rain (hydrograph) [Bed92]. Other environmental models contain spatial variation parts, which predict the time evolution of specific parameters at different sites (distributed model); for example, to predict the fire distribution after one house is burned. Simulation results of distributed models are difficult to decipher if they are in numeric or textual forms. They are better understood if they are put on a series of 2D frames to animate time evolution. In other words, distributed environmental model simulations rely on other tools to incarnate their results to understandable forms; they can be displayed either on GIS map or CAD 3D environment.

The best way to extract the advantages and compensate the drawbacks of GIS, 3D CAD, and environmental model simulation is to integrate them in one system; this is the main goal of GISVE, Geographical Information System Virtual Environment.

- Users can work on GIS data on 2D maps, which are easier to deal with than in textual or quantitative representations. They can display, remove or add data layers on display lists, modify a specific data layer by adding or deleting elements of it, do arithmetic manipulations, and analyze correlations of different layers.

- Users can work on 3D CAD graphics, which is more realistic in representing objects. Users can walk through the 3D environment and see objects from different perspectives. Abstract data can be converted into a more understandable form that helps users to know what's going on in the environment.

- Users can work on environmental model simulation tools, which supply another dimension of information - time. Users can see the time evolution result by applying environmental models and can observe dynamic processes in this environment.

- By integrating these three tools, users obtain three types of information within one system instead of running three systems separately. It saves time and effort greatly.

Such system is very useful for many people who work on or care about environmental issues. For environmental scientists, this system can help them to test and to verify their models and to see the prediction in a graphic form. For environmental decision makers, this system can help them to have a sense about what will happen in the future if such decision has been made. For public, this system can help them to know what will happen to the environment they live and what parts of the environment they care about will be influenced. For landscape designers, this system can help them to understand and analyze various information together and test the possible results of their prototype ideas.

In this paper the application domain is focused on landscape design process. There are several reasons to choose this field. At first, the resources of information necessary for landscape design closely match the three computer tools mentioned; the data about topography, geology, hydrology and ecology is acquired from GIS, the landscape of the target area is sketched by pictures of that area, environmental models are required in assessing the impact of their design works. Secondly, the immersive and interactive properties of virtual environment closely match the demand in landscape design process. Like architecture, visual effect plays a very important role in landscape architecture, the match between the artificial design work and the surrounding environment becomes the major criterion for judging the quality of a design; so a system which can immerse the designer in surrounding environment is critical in aiding design process. Furthermore, the designer has to have the ability to change the environment and understand what happens after he/she makes such changes. The interactive property that computer updates the scene or environment quickly satisfies such demand. Thirdly, environmental impact assessment becomes a critical issue in contemporary landscape design. How to see the impact of artificial construction in a natural environment, and furthermore, how to determine the site and form of artificial construction to alleviate environment impact, are very important in current landscape design process. Since there are few if any such tools which have all these features described above; it is time to develop such system.

## 1.5 Related Works

There are huge number of research works for each element of GISVE: GIS, VE, 3D CAD tools, and environmental model simulation. However, the integration of them is rarely found if not at all.

Integration of GIS and environmental model simulation has attracted many efforts and interests recently and is spreaded in many fields of environmental sciences, including hydrology [Mai93], non-point source pollution [VK93], and ecology [NB93]. A research work by Theobald et al combining GIS, system dynamics, and cellular automata in one system delivers similar concept about integration to this thesis research [TG94]. However, it only has dots in 2D grids to represent simple data types such as fire cells and clusters of population; it doesn't describe how the user can change parameters or manipulate data, it also lacks functions to allow the users to change or move through the 3D environment. Integration of GIS and 3D graphics rendering is mainly used for modeling vertical structures of underground geographical data, such as water or mineral distribution. POLYTRIM is a software that can transform GIS data into surface polygons and allow the user see the scene in different perspectives, but the change of perspectives is done by keying in special command, rather than real time interaction [M$^+$90]. A study similar to this thesis research is done by Ervin [Erv92], he used POLYTRIM to convert 2D GIS data into 3D polygons, applied models to the target place according to some criteria, assigned each grid an appropriate value (for example, 1 represents the highest negative impact), then attached the values at 3D scene using color coding. However, constrained from the rendering tool, the system he described lacks the capability of real time updating the 3D scene responding to the user's movement in the space. Besides that, there are no environmental models embedded in the system; the values at cells depend upon the user's judgement from the GIS data displayed, this may be appropriate for qualitative measures, but not enough if we want to be more quantitative or concrete.

Virtual environment and 3D graphics rendering are already widely used in architecture design. It is even advocated that the progress of virtual reality and network communication will not only change the paradigms of design processes, but also change the form of physical spaces [Mit95]. The application of VE in modeling landscape is still not as popular as the case in architecture. SGI's demonstration of Performer town is an example. It displays a small town that allows the user to navigate through it; its delicate graphics shows a potential use in the application in landscape design.

## 1.6 Framework of this thesis

The framework of this thesis is based on developmental stages of GISVE system. Chapter 1 is an introduction which gives the motivations and backgrounds of GISVE system.

Chapter 2 describes how the specifications and features of GISVE are formed. It covers task analysis, determination of specific goals, system feature requirements and function requirements.

Chapter 3 talks about the implementation of GISVE system. It first sketches the system architecture, then describes all modules separately, including GIS data modules, 2D to 3D data modules, navigate modules, model modules, graphical user interface (GUI) modules, and application main loop. It will describe some problems encountered when we build these modules and how to overcome these difficulties.

Chapter 4 covers verification, evaluation and modification of the system. After the system has been built, at first it should pass the validation, which means the system will be checked if it is consistent with original objectives and specifications. Then domain experts are invited to evaluate this system, from the evaluation result, we know whether the original goals are achieved or not. The system will then be modified based on domain experts' suggestions and opinions. Finally a project test will be applied by using GISVE, it will be shown how GISVE system helps the design process.

Chapter 6 is conclusion. It will wrap up the results and discussions of previous chapters to justify that the general and specific goals have been achieved. It will also discuss about future works.

# Chapter 2

# System Specifications

The determination of system specifications is perhaps the most important stage in the software development process. It is especially important in modualized system design. If the input/output specifications of each module are determined, then different modules with the same specifications can substitute for each other, which results in great flexibility and versatility of the system. In this thesis, the specifications are determined to satisfy system objectives; by defining what the system can do and how to do clearly, objectives can be fulfilled step by step.

## 2.1 Task Analysis

### 2.1.1 General concepts about task analysis

According to the definition of McCormick [$D^{+}87$], "Task analysis is a process of identifying and describing units of work, and analyzing the resources necessary for successful work performance. Resources in this context are both those brought by the worker (e.g. skills, knowledge, physical capabilities) and those which may be provided in the work environment (e.g., controls, displays, tools, procedures/aids)." The object of task analysis is human being, and the main goal of task analysis is to identify performance demands of jobs and job tasks. The result of task analysis can be used in design and improvement of many aspects of human-machine integrated system to fulfill a specific task: training procedures, operation procedures, work environments, and equipments.

Usually task analysis is used for analyzing works which have well defined step-by-step operations. The most often used methods in task analysis include interviewing crews of the task, observing their behaviors at work, reading documentation about the working process, and keeping notes on all the equipments, knowledges and skills necessary for the job. A task can be taken as actions to fulfill a specific objective (system output goal), because step-by-step operations are easier to analyze, they are more often used. Early applications of task analysis were focused on procedural works such as pipeline operations in factory and pilot training. Recently, with the improvement of computers the task analysis of a human-machine system becomes more important in designing new computer products. The more information can be accessed from the computer, the more complicated parameters the user need to control, and the more difficult the user can handle the working environment. In this situation, to define and analyze human's performance in a mission and to use this knowledge in design or improvement of working environment become critical issues in modern human-machine system development, for example, spacecraft or airplane development.

Some jobs, however, are more related to cognitive part, rather than behavioral part, of human activities. These jobs may not have well-defined operational procedures. A good example is landscape architecture, the application domain of this thesis research. The design process can still be represented by flow charts, (it will be described later), but each procedure in the flow chart lacks well-defined operational representation. In this thesis research, the basic concept of task analysis will be followed with minor modifications; the overall task is divided into several stages, we are focusing on building a system for aiding the work process of one stage, so essential information and specific functions needed when working at this stage will be scrutinized. The flow chart will be used to facilitate understanding the overall design process, but we don't divide each stage into detailed operations and analyze these operations separately.

### 2.1.2 Plan for task analysis

As mentioned in the previous section, it is hard to define landscape design process in operational steps because of its highly diversity and flexibility. In this thesis, a less formal method of task analysis, which is called scenario method, will be applied. It is based on informal conversations between the domain experts and the interviewer. In our case the interview contains two parts, in the first part the software developer obtained general ideas about landscape design, general and specific requirements about a software system that can help their work, and opinions about possible forms and features this software system would have, through conversations with these experts; the second part in the interview session is the demonstration, in order to give domain experts a quick sense of how a 3D interactive virtual environment could be, a 3D scenery based on GIS data elements had been built previously. During the interview session the 3D scene was demonstrated to domain experts and their responses were recorded.

Four experts in landscape architecture from MIT DUSP (Department of Urban Studies and Planning) and Harvard University GSD (Graduate School of Design) landscape architecture department were invited into the interview: they are Carl Steinitz, Steven Ervin, Michael Binford, and Kristina Hill. There are two interview sessions. The purpose of the first session was to obtain a broad idea of landscape design process and potential application domain. The purpose of the second session was to ask domain experts' opinions and requirements about system specifications, features, and functions. Both sessions have demonstrations before conversations.

### 2.1.3 First interview

The first interview was held on July 10th, 1995, at room 36-705, VETT (Virtual Environment Training Technologies) lab of RLE (Research Laboratory of Electronics), MIT. A domain expert was invited to participate the interview. The first part of the session is a demonstration. A 3D scene whose objects were generated from the

GIS data was demonstrated. It was a pre-prototype which was built for testing and demonstrating in the interview session: the user could fly through the 3D environment by manipulating the mouse; the objects themselves were static that could not be changed by the user. After the demonstration, her quick response and first impression were polled. In spite of the low graphic quality and manipulation constraints of this pre-prototype kit, she gave a highly positive opinion about the potential use of such system in landscape design process.

She mentioned two directions to go in terms of application domains. One is for the use of professionals and students in landscape architecture, it could give them a quick sense about the landscape of the target place. It is especially advantageous to let the user fly through the environment. Landscape design, like architecture, is on three dimensional space; to manipulate on 3D space is always better than on 2D plane. Another advantage she mentioned in aiding professional design is visualization of abstract data type, including natural and social factors; for example, the pollutant content at different places can be represented by using different colors to denote different levels, the land values obtained from cost-benefit analysis could be color coded to represent the values of the cost or benefit. This system is a very good vehicle to convert the numeric data into more concrete and understandable forms. By using this system, it could demonstrate natural or anthropogenic processes, like the flood runoff, the spread of epidemics, fire, diffusion of pollutants. The designers or decision makers can explain to the public (or clients) why they make such design or decision (because if I don't do that the flood will attack us or the epidemics will spread...). On the other hand, the public can also use this tool to convince the designers or decision makers to do or not to do something (if you put nuclear waste near this fault, it is highly probable that the waste will leak out then all inhabitants within 500km radius will...). In other words, this system could be a very useful tool for communication and negotiation between designers and clients or between public and government.

Then went to the second part of the session. In responding to software developer's requests, she described and explained the 5 steps in the design process.

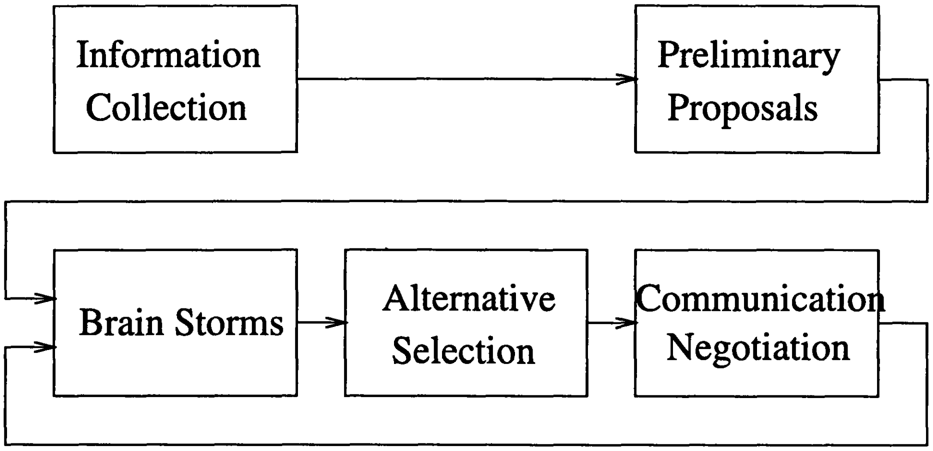
1. At first information from different sources are gathered, including 2D maps (with different scales and different data types, for example, some indicate topography, some indicate property boundaries, some indicate zoning), GIS (also from different sources and scales and with different applications), photographs (to show different perspectives of the target place and give a relatively realistic impression to that environment), interviews (with clients, inhabitants of that area, experts studying the environment of that area, etc). All the information will be used in later stages as references to determine the final come-out.

2. The second step is generation of preliminary proposals. The preliminary proposal is usually based on clients' need for specific changes, for example, to build 100 houses in an area. All information gathered in step 1 is used in determining the preliminary proposal, but they have in-breadth, rather then in-depth influences. The aim of preliminary design is to filter out choices which are definitely unsatisfactory, for example, the project which raises costs or causes environmental problems. However, to determine these choices is not trivial, it still needs many information sources to judge if this choice should be discarded or not, in spite of the relatively rough scale and precision.

3. The third step is so called brain storm. Usually the landscape design is fulfilled by a group of people. In this stage members of this group meet together in studio. They create many ideas from their "mental catalogs" and scratch these ideas on paper (or computer). They discuss and argue these ideas, which make these plans more complete and comprehensive. Information gathered in step 1 is used to sieve out ideas which are unsuitable and to keep ideas which are consistent to achieve objectives. The bigger the project is, the more persons join this process, and the more information should be collected to use in the project.

4. The fourth step is to select an alternative from those proposed projects. Proposals are ordered based on some criteria, the project which satisfies the highest priority of the criteria will be chosen as the final project. The priorities are

Figure 2-1: Landscape design processes



based on goals of this project, sometimes economic profit has the highest priority, sometimes environmental quality and public health are more important.

5. The fifth step is the communication with clients or other persons involved. The final plan determined by landscape architects is never the final plan to be implemented, it still needs to communicate and negotiate with clients or the public. Usually this step will last for several years long; the larger the project, the more persons are involved in it, more complicated it is to handle this project, the longer it takes to finish. This step is a political and communication activity, rather than a scientific or art work.

From the result of first interview, the potential application of GISVE system was indicated, and the general working process of landscape design was depicted. It is clear that building a system to satisfy the needs of all these stages described above was very difficult, so the objective of GISVE system would focus on one specific step.

### 2.1.4 Second interview

There were two more interviews subsequently on July 17th and August 10th, 1995. Domain experts came to see the demo and gave opinions, both of them were held at the same place as the previous one. We discuss these two interviews together without distinguishing them because their aims and procedures are similar.

Initially there was a demonstration, then domain experts gave responses and software developer asked questions. The main focus of questions and discussions was different from the first one; in this session, we focused on possible application domains and how the system specifications would be to satisfy the needs of these application domains. To sum up their opinions and comments, there are several points which are really important in determining the configuration of GISVE system.

1. They all expressed interest in the 3D rendering capability of the system, however, they thought that was not enough. One reason is there are already many 3D rendering tools such as AutoCAD and POLYTRIM, it will be less significant to add another one in terms of research concerns. Another reason is the visual quality and navigate ability in the pre-prototype system were not very well, this impeded their visions about the potential use.

2. They all concerned about the visual quality the system could provide. Being landscape architects, they always asked themselves a question: did the tools I used represent the "real" landscape of the target place? Besides using maps, GIS, and pictures, they would go to visit that place as well to gain the experience in the real landscape. For this reason, during the demonstration, they cared about how well the computer graphics could simulate the real environment; for example, they expressed the discontent toward the low density of tree profiles on computer graphics, compared to real situation of the target place being displayed. They found in some perspectives the view was close to real scene, such as looked around at the top of a tree, but in other perspectives such as a bird eye's view, the visual effect was not good enough.

3. They also challenged the only navigate function the pre-prototype system possessed - fly. They found fly function was not quite easy to manipulate. Sometimes they didn't want to change elevation by moving the mouse forward and backward, sometimes they wanted to fix at a position and see all perspectives by turning around, sometimes they just wanted to drive automatically along a designated path without moving the mouse. The requirement of navigate functions varies according to different applications, it was very diversified.

4. Sometimes they were confused while flying in the 3D environment. The most often asked question was: where am I? Although they were all familiar with the environment of target place, they still could not tell their positions without some remarkable references. Disorientation problem also occurs in other virtual environment applications (for example, the intelligent camera in a virtual environment of museum browsing [DZ94]). In order to avoid the confusion, information about the context of the 3D environment should be provided, this information includes where the viewer is and what the system is doing currently. For a 3D environment, a 2D map is the best channel to show these contexts.

5. They also concerned about how they could change the environment. As mentioned, landscape design is to change the natural environment and process. The system should equip the function to incarnate the design works in the virtual environment. The changes they wanted to do included: topography (for example, to dig a hole or pile up a hill), hydrology (for example, to build a dam, to fill a lake, or to dig a canal), vegetation (for example, to cut trees to plant grass, or to change the tree species, to change crop types), artificial constructions (for example, to build houses, shopping malls, roads, sewers, and other infrastructures). These are essential elements for landscape design.

6. They hoped that this system could provide models to simulate dynamic processes of the environment. Different applications need different models, some of them were widely used : hydrological models, pollution distribution models, fire diffusion models, and social change models. Based on these models, they

wanted to see which site was least influenced by natural or anthropogenic impacts, for example, where to build houses which were not flooded when storm came.

7. Furthermore, they wanted to assess the impact to environment after they made such changes. The environmental impact assessment has importance in both environmental and economic considerations. If changes made by the designer caused destructive effects, for example, the leak of pollutant, the expansion of flooded area, or the desertification of neighbor countries, he had to not only take the moral responsibility, but also financial loss. So if environmental models could be implemented in this system to simulate the result after changes, it would be very useful in decision making of landscape design.

8. Finally, they expressed highly interest in a system which integrated both spatial navigation and dynamic process functions. They required large degree of freedom while navigating through space-time: sometimes they could fix time and navigate through the 3D environment; sometimes they could fix space and run a model simulation to see the time evolution result; most of the time they hoped to simulate spatial and temporal variations together: while navigating in the 3D space, the model was still running. For example, when the storm came, some places were flooded, if the viewer was looking down from a helicopter flying across the flooded area, what would he/she see ?

These two interview sessions are very interesting; although they are less formal, it doesn't mean less informative. The result of these interviews provide information in determining the specific goals, features, and functions of GISVE system, these will be mentioned in subsequent sections.

## 2.2 Specific objectives of GISVE system

After finishing the two interviews of task analysis, the specific objectives the GISVE system are outlined. The first thing to be determined is in which part of design process is the GISVE system going to help. From the conversation of the first interview it is known that there are 5 stages in landscape design process. We found that a system integrating with virtual environment, GIS, 3D CAD, and environmental model simulation would match the demand of the second stage of landscape design process. At this stage, the designer acquires a quick sense about the site according to various information sources gathered at the first stage. He/she digests the information, then generates some mental pictures of what this place should be, these mental pictures become preliminary proposals. Then he/she will use more detailed and complete information of that place to test if these proposals satisfy specific criteria of this project. It is difficult to build a computer system to tell people how to form mental images, since we are not yet clear about the mechanism of thinking. However, we can build a computer system that helps people to form the mental images. GISVE helps designers to form their mental pictures via two ways: it encapsulates various data and information and represents them in more understandable forms such as 2D and 3D images, it is also embedded with environmental models such that designers can test their proposals by simulating these models. By using this tool, landscape designers can eliminate proposals which are against economical, political, or environmental criteria. For these reasons the specific objective of GISVE system was decided to focus on the preliminary proposal forming stage of landscape design process. There are two specific properties of the GISVE system corresponding to this objective.

- Since it is focused on the preliminary design stage, the system should integrate as much information as possible, the information will be relatively simpler because its aim is to provide the user a quick sense of the place, it is in-breadth, rather than in-depth.

- Based on the same reason, the environmental models been implemented don't have to be very complicated. The models are used for sieving some inappropriate proposals in a coarse scale; more complete analysis based on more accurate environmental models will be done in later stages.

Another thing to be determined in forming the specific objective is what kind of problems in landscape design will be addressed to solve on the purpose of this system: art, engineering, economical, and environmental. This thesis research is concentrated on the environmental part. To be more specific, this system provides environmental impact assessment functions and represents the impact result in graphic forms. In order to do that, the system should (1) allow the user to modify the objects in the virtual environment. (2) apply environmental models to simulate the modified data and to display the result. For example, the landscape designer can build 100 new houses on the site and use flood models to see the influence of these new houses to flood water runoff.

The assessment of environmental impacts is used in determining many parts of landscape design work: the site where some construction works should avoid, how large the area of change should be, what kind of architectural form should be chosen, which species of trees should plant, what kind of building technology will be used, etc. In the preliminary stage, detailed problems will not be considered yet, the most probable concern is : where the change will be put and how large the area of change will be.

To sum up, the specific objective of GISVE system is focused on the preliminary stage of landscape design process, in assessing and representing environmental impact for the decision of appropriate sites.

## 2.3 GISVE system features

Based on the specific goals and the opinions of the landscape architects in the task analysis stage, the features of GISVE system are determined.

### 2.3.1 Objects to to displayed

Several data types are important in determining the landscape of a place. Elevation is the most important because it determines the topography of a place. Traditional methods of sketching the topography is by looking at the contour map; GIS also provides elevation (topography) files in different formats such as DEM (Digital Elevation Map). Most of elevation data of GIS is represented by different colors on 2D map. In the GISVE system, elevation profile of a place is represented by displaying its topography in the 3D scene, the user can see the rise and fall of the terrain at different places rather than using color coding or contour lines.

Roads are also very important structures. They are represented by strips with texture mapping. Road data representations in 2D and 3D scenes are similar, the only difference is that the road strips in 3D scene should follow terrain of that place; in other words, their heights will vary according to the elevation values at different places.

Water bodies include linear structures (rivers, streams) and patch structures (ponds, lakes). They also follow the terrain of that place, and they will have the color and texture of water in the 3D scene. It is very difficult to visualize the flow of water, which needs sophisticated computer graphics and animation techniques, and it's not crucial in the preliminary landscape design process, so in this system, water body will be represented by a static, rather than flowing object. The diffusion and movement of water can be visualized by updating the area covered by water.

Houses are the main artificial structures of the landscape. The representation of houses in a 3D environment could be very complicated if the main concern is the architectural form. For example, if a virtual environment is built to simulate a city, it should demonstrate the multitude of architectural styles of buildings. However, in the application domain of GISVE, the structures of landscape and the natural environment are more important than individual houses. In order to reduce the unnecessary information been displayed and to reduce the effort and time of creating each house object in the virtual environment one by one, all houses are represented with the same form and size.

Trees are another important elements that the landscape architect can use in the design. The requirement for tree display is that it can show individual trees as well as the whole forest. The tree data representation in 3D scene is a difficult problem in computer graphics because of the fractal geometry and scale dependence of its shape. Use the texture mapping method and level of detail (LOD), this requirement will be attained in GISVE system. More discussions will be in Chapter 3.

Other data includes the simulation results of environmental models. The data to be displayed depends on models been used, it can be an abstract or concrete data type. In GISVE system, the simulation results include flood, soil erosion, and fire.

### 2.3.2 Visual quality

As mentioned in previous sections, all domain experts in the interview stressed the importance of visual quality. In GISVE, visual quality is attained by various computer graphics technologies such as texture mapping and level of detail. The requirement of visual quality also depends on the specific applications, this requirement is different for different objects displayed. In landscape design, terrain, trees, water bodies, roads are more important than house forms. Another thing to be noted is that from system consideration, it is a computational intensive work to build complicated and realistic graphics, and it will compete with other functions of the system, such as the short response time for interactivity, for computer resources. In this system, some objects are texture mapped but some others are not, it depends upon the consideration for both visual quality and space-time efficiency of the system.

### 2.3.3 2D and 3D displays

It is necessary to have both 2D and 3D displays in the GISVE system because they provide different functions and these functions cannot be substituted with each other. 2D display is for easiness of data manipulation, clear understanding of abstract data type, and indication of the current position in the environment; 3D display is for immersive and realistic feeling of environment, and different perspectives. In GISVE, they are placed at separate windows. 3D display is the main window which shows the virtual environment of the place. 2D display is an auxiliary window which shows the GIS data layers and function buttons (widgets) of the system; a position icon indicating the current position (x and y coordinates) of the viewer in the virtual environment will be marked on the 2D map. The user can freely switch between the two windows, all the changes he/she made or all the simulation results will be updated on both windows.

### 2.3.4 Time response

Time response is very critical in virtual environment system because it is involved with the interactive capabilities. If the system's response is very slow, the environment will not update instantaneously in responding to user's input, the user will lose the immersive sense and be distracted by the waiting time. Usually the real time response needed to create interactivity is a tradeoff of high quality graphics : the more complicated objects in the 3D scene are, the more time it needs to detect which objects should be displayed in this view angle and position (cull) and to calculate the projection and mapping of objects onto the projected plane (draw), thus the slower the response is. In GISVE this problem becomes more serious because another time-consuming job - model simulation, is added into the system. The time response of this system should be fast enough that will not block the navigate functions, in other words, the user should not experience idle when he/she flies or drives through the 3D scene.

### 2.3.5 User interface modes

Most VE systems are supposed to use multi-modal sensory inputs/outputs as user interfaces. The determination of using which modes depends on the specific application this system is focused on. In this project, it is decided only to use graphical display mode because it is the most essential and important mode in landscape design process. Landscape architects concern about the visual impact, most of them are accustomed to transform these information in visual forms and think in visual forms. Other sensory modal devices may provide further aid for design process, for example, sound output can be used in estimating the noise generated on a traffic intensive road, but not essential. In this project there are no important information related to hearing or haptic senses, so mount visual mode only is sufficient.

There are many visual devices, such as head-mount display, TV projector, or computer monitor. Here computer monitor is considered more suitable for this project. Although interactivity is also important, it is not necessary to track slight body movement because landscape design is a static work in terms of body movement. In figuring out preliminary proposals, the user needs a display that he/she can see the change conveniently. Large screen display is appropriate in later stages which many persons participate and share the same information, such as the brain storm stage or the communication stage.

## 2.4 GISVE system functions

The GISVE system has several functions which the user can choose to fulfill the tasks and objectives described above. These functions are divided into three categories: functions for GIS data manipulation, functions for navigation, and functions for model simulation.

### 2.4.1 Data manipulation functions

All manipulation functions on GIS data are done on 2D data format, the system automatically converts the manipulated 2D data into 3D objects. Data manipulations contain two parts: load data and change data. A load data function can load a specific type of GIS data into the system, or unload a GIS data layer from the system; the data been loaded includes terrain (elevation), roads, water bodies, houses, trees, model simulation result (flood, erosion, fire). The loaded data is displayed on both 2D and 3D windows. Change data functions include add and delete data on different data types. For raster data (houses, fire and soil erosion spots), the user can modify it cell by cell, for example, add or delete one house at a specific position; for linear data (streams, roads), the user can add a new branch or delete an old branch from the original data, for example, build a new road; for patch data (ponds, forest), the user can add or delete the data inside the polygon set by the user.

In order to do the change functions, another class of accessory functions - choose functions, are needed. They allow the user to choose the region that he/she wants to modify on 2D map. For raster data the region is a collection of individual cells, for linear data the region is a path, for patch data the region is a polygon.

Figure 2-2: Raster data manipulation

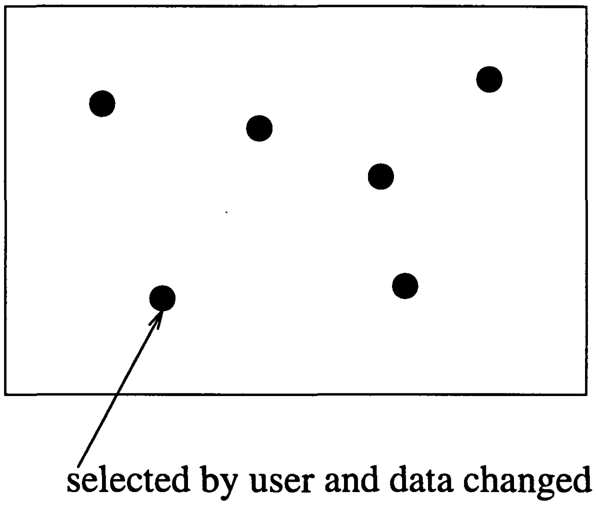


Figure 2-3: Linear data manipulation

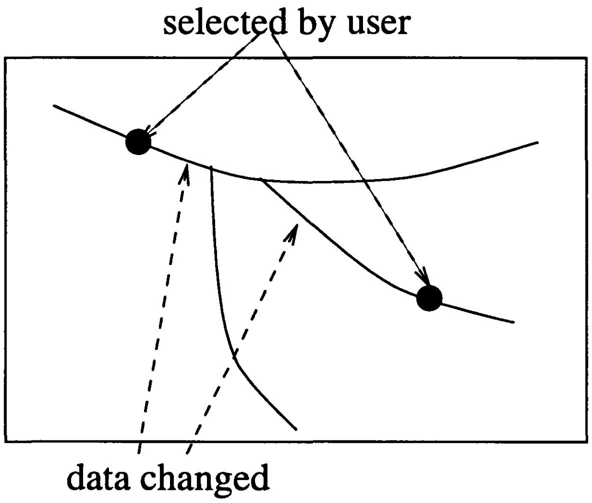
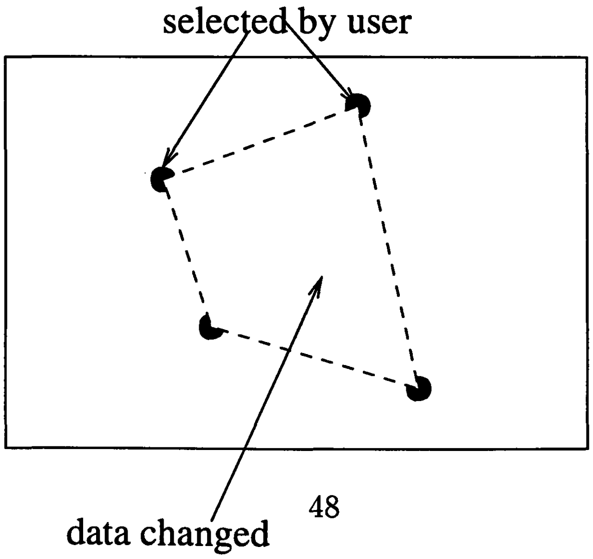


Figure 2-4: Patch data manipulation



### 2.4.2 Navigate functions

Navigate functions are crucial in GISVE system because they offer different modes that the user can move in the environment. The more diversified modes the user can choose to observe the environment, the more information will be obtained. The GISVE system provides five modes of navigation: fly, drive, automatic drive, fix position, and turn around perspectives.

The fly function simulates the operation of an aircraft. The user uses the mouse to move in the environment. When the mouse is dragged and moved left, the position in the environment will shift left, when the mouse is dragged and moved right, it will shift right, when the mouse is dragged and moved forward, the position will turn down, when the mouse is dragged and moved backward, it will turn up; the viewing angle is toward the direction of movement. In other words, the mouse is used as input interface to track the 3D position in the environment. This navigation mode gives the user the maximum degree of freedom that he/she can observe the environment at any position in any perspective.

The drive function, on the other hand, simulates the operation of driving. The user can still use the mouse to change viewing angles and 2D positions, however the altitude maintains at a fixed value above ground. The drive function is used in complementing the inconvenience of fly function. When the user only wants to drive along the road and doesn't want to take care of the unnecessary control variable - altitude, the drive function is adopted.

The automatic drive function provides the user more convenience, although less freedom, in the movement. Sometimes the user even doesn't want to use the mouse to drive in the environment. If he/she selects a path (a segment of road, for example), and wants to see the landscape when driving along the path, it is better to have another function - automatic drive function, to take care of it. After a specific path on the 2D map is selected, he/she can choose the automatic drive function. The viewer's position in the environment will automatically move according to the designated path, altitude maintains at a fixed value above surface.

Fix position function is simple but necessary. The user chooses a position on a 2D map (use set point function) then applies the fix position function, he/she will go to this position and keep at this position without movement. This function is used for reset position in avoiding disorientation. Sometimes improper manipulations of fly functions will result in strange scenes, e.g., upside down, the user loses the sense of current position and orientation (no wonder because the fly function simulates the aircraft fly, and pilots usually have such problems). In this case, the user needs a function to set him/her at a known position. Another condition the fix position will be used is when the user is interested in a specific position, then he/she can jump to that position directly without taking time to move there.

Turning around perspectives function also sets the user at a fixed position, but it varies the viewing angles by turning around (varies the yaw angle, spin around the z axis). In landscape design, the "view" at a position plays an important role in site decision, in this case, the designer can use turning around perspective function to see the panorama views at that position, which facilitates in choosing sites.

### 2.4.3 Model functions

There are many environmental models which are useful for landscape design and can be put in the GISVE system. However, it is impossible to cover many processes of the natural environment in GISVE. Three environmental models are chosen to be implemented in GISVE, the choices are based on the importance, complexity and response time of these models.

The first model is the flood runoff model, it should show the flood levels after a rain with designated intensity and duration. The model should be a distributed rather than an aggregate model, and it should have the ability to show flood levels of all cells at that place.

The second model is the soil erosion carried pollution model, it should show the soil loss and sedimentation of each cell after the designated rain. If there are pollutants in soil, it will also show the accumulation of pollutants due to soil sedimentation. This model should be distributed, too.

The third model is the fire model, it should show the spread of fire after an initial fire spot is set. This model should also be distributed.

Another requirement is that the input parameters of these models are either set up by the user or determined by existed GIS data layers. The model which needs a lot of input parameters beyond what GIS can support, although may generate more accurate result, is not appropriate to put in this system. For GISVE system, it requires all input parameters should be related to terrain, roads, houses, water bodies, and trees.

The input parameters the user can load are restricted only to some initial conditions, too many parameter inputs will just annoy the user. For flood model, the user can input the duration and overall precipitation of rain; for soil erosion carried pollution model, the user can input rain data and set locations and concentrations of pollutant sources; for fire model, the user can indicate cells which are on fire initially.

The simulation results of model are displayed on both 2D and 3D windows. In most cases the model simulation results also affect GIS data, for example, flood will expand water body, fire will destroy houses and trees; all these GIS data will also be modified and updated on 2D and 3D display.

There are two modes of operations in running the model. One is so called the dynamic simulation. When the user applies the model, it will run without blocking other functions of GISVE; the user can still navigate through the environment, change the data, or run other models; and the system will periodically update the 2D and 3D windows based on simulation results. The other operational mode is the fix duration simulation, the user sets up the duration of simulation, and the system will concentrate on running this model until the stop time. When a simulation is running, other functions of the system are blocked; this mode is used when a relatively long term prediction is needed. For example, if the user wants to know the flood condition after one day, he/she will not sit before the computer one day to wait for the result. On the other hand, dynamic simulation is useful when the user wants to see the temporal change as well as the spatial change.

# Chapter 3

# System Implementation

The implementation of GISVE system, which means the programming of all modules and subroutines, takes the longest time and effort in the development process. It is involved in various problems: computer graphics, user interface design, model simulation, and coordination of modules. When we are implementing the system, we should make sure that each module has functions and features to meet some of the specifications and required functions described in chapter 2; furthermore, we should learn to balance the tradeoffs among different functions of the system to attain the optimization in terms of space and time efficiency.

## 3.1 System Architecture

### 3.1.1 Hardware Architecture

The hardware of GISVE is already built in. It is implemented in a two-processor SGI Onyx connecting to a SGI Reality Engine for graphics processing, mounted at the computer room of VETT lab (Virtual Reality Technology Training), Research Laboratory of Electronics, MIT. As mentioned, the only relevant sensory I/O modes are mouse movement and visual display, so the only sensory I/O devices are the mouse and the monitor, which are already mounted on the computer. No other sensory devices or computers are needed to attach to the system.

### 3.1.2 Software Architecture

The GISVE uses C as its programming language; besides the hard coding part, it also refers to two C based libraries: Performer [$F^+94$] and Motif [JR93]. Performer is used for creating a 3D scene and rendering through this scene, and Motif is used for graphical user interface design.

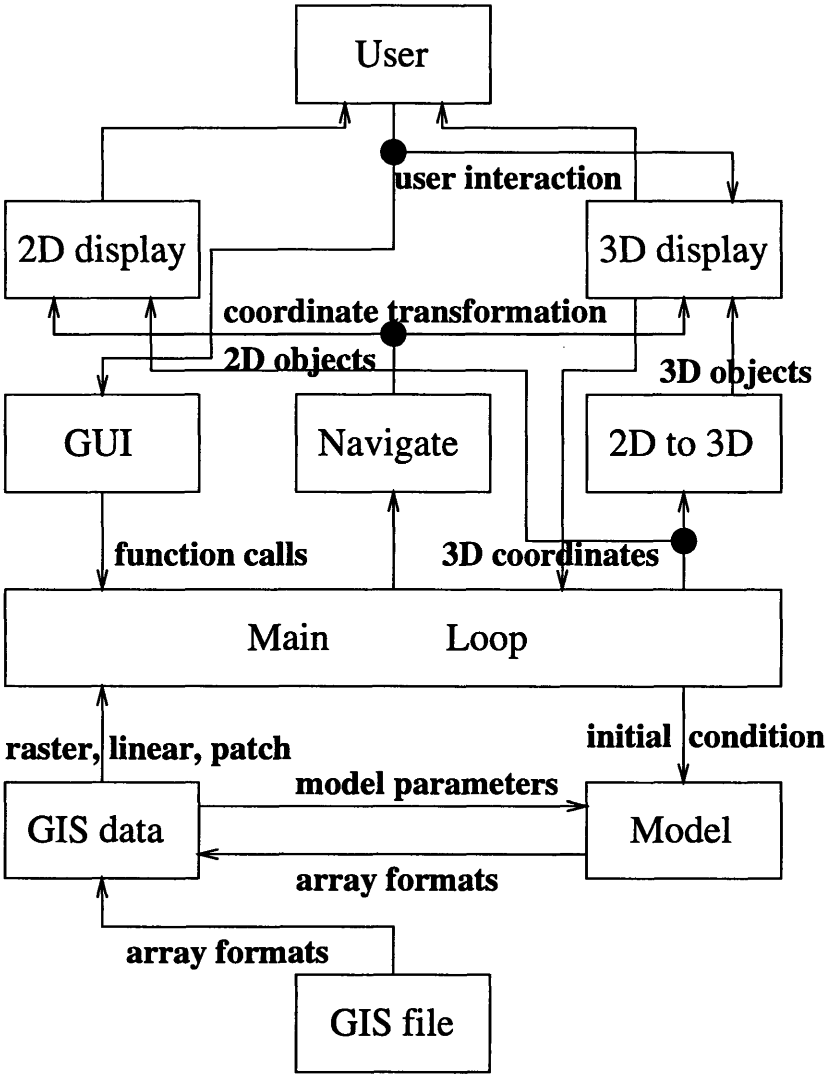
Almost all contemporary software products are required to modualize their subroutines. In this system, although all programs are written by one person, it is still necessary to modualize it for the convenience of realizing and debugging. The entire system contains 6 modules.

The 2D data module loads GIS data files into the system, updates changes of the data based on user's input, and converts the GIS data into several 2D data types: raster type, linear type, and patch type. To represent different GIS data in different types facilitates direct manipulations on these data and transformations of them into 3D objects that can be displayed in the 3D scene.

The 2D to 3D data module converts the 2D data into the 3D data. The 2D data is represented in one of the data types described above: raster, linear, and patch; the 3D data belongs to one of the file formats that can be directly loaded by using Performer's built-in file loading functions. Most 3D data are represented by a set of polygons, the coordinates of vertices, the vectors of the normal direction, the color and the texture of each polygon are indicated in the 3D data. For different type of data (raster, linear, patch), different algorithms are used to transform them into the same 3D data format.

The navigate module contains functions that allow the user to navigate through the 3D environment - fly, drive, automatic drive, and perspectives. Movements in 3D environment are represented by a series of matrix transformations that change the coordinates and viewing angles in the 3D scene. These transformations can be up-

Figure 3-1: GISVE system architecture



dated by polling the mouse movement in real time (fly, drive), or made by a designated path or pattern which is set by the user previously (automatic drive, perspectives).

The model module contains the numerical models described above: flood model, soil erosion generated pollution model, and fire diffusion model. The input parameters and initial conditions of these models come either from the user's direct input or available GIS data.

The GUI (Graphical User Interface) module creates a 2D display window and all function buttons on the window. It is created with the help by using function calls of Motif, an X Window toolkit for building the GUI. There are icons on the 2D window to emulate the buttons, when the user presses a button by clicking the mouse on it, the function attached to this button will be executed. The module also contains functions to generate 2D GIS data display.

The main loop module is in charge of controlling and coordinating all the modules to take care of the entire system. It includes two parts: initialization part and infinite loop part. The initialization part contains functions of doing all initialization processes - loading GIS data, creating 2D window and function buttons, building 3D scene graph, reseting model parameters, and initializing state variables of the system. The infinite loop part updates state variables in a short period according to the user's input, for example, change the 3D coordinates and viewing angles according to the movement of mouse, or change the 2D GIS data and 3D objects when user adds or deletes data. This loop will not end until the user chooses the "Exit" function.

## 3.2 GIS data module

The GIS data module is in charge of loading and updating the 2D GIS data in the system, and transforming these data into other data types that are more easily handled when the data is converted into 3D objects or when the data is used as parameters of model input. This module contains several functions:

1. *Load the GIS data files into the system.*

2. *Convert raster data type into linear data type.*

3. *Convert raster data type into patch data type.*

4. Update GIS data in the system.

### 3.2.1 GIS data files loading

To create a file adapter that is compatible with most GIS data file formats and can transform these file formats into different data types is both difficult and beyond the scope of this research. GIS file formats can be categorized in two types: vector and raster. Vector data stores more information than the raster data, for example, direction or local correlation; it can be quite useful in some applications, however, the representations are so diverse that it lacks a general representation for vector files. Raster data format, in spite of less information it contains, is more general in representing different data layers. In current stage GISVE system only accepts one type of GIS file format: ASCII format. All data is represented by a matrix, the column corresponds to longitute or vertical line, the row corresponds to latitude or horizontal line, each entry corresponds to the GIS data value at that position, it can be elevation in feet, vegetation type, population density, or other information at that position. The major advantage of using ASCII raster data format is the easiness to handle, the programmer need not to write a converter to transform between complicated GIS file formats. The other advantage is that many current existing GIS tools, such as ARC/INFO [ARC90], or MacGIS ([20]), can convert different GIS file formats, such as DXF, TIN, into the ASCII format. This makes using ASCII file loading more convenient since the user can transform vector files into ASCII files on these GIS tools, and then load the ASCII files into the GISVE system.

Several layers of GIS data files are loaded into the GISVE system. Some of them are displayed on 2D and 3D windows, some are only used as parameters of simulation models:

- *Elevation data* The value at each cell represents the average elevation in feet of that cell.
- *Road data* The value at each cell represents the road type at the cell, 0 is no road, 1 is the primary road, 2 is the secondary road.
- *Water body data* The value at each cell represents the water body type at that cell, 0 is dry land, 1 is pond, 2 is stream, 3 is wetland.
- *Tree cover data* The value at each cell represents the tree type at that cell, 0 is no tree, 1 is deciduous tree, 2 is evergreen tree, 3 is mixed forest.
- *Soil data* The value at each cell represents the soil type at that cell. However, it has very detailed classifications of soil types by US Soil Survey, different place has different soil types in terms of this taxonomy, thus, generally it is not a good way to represent different soil types by different numbers if the target place covers a large area. In this research, the place the GIS data depict is Petersham, Massachusetts; the soil type classifications are as follows: 1-shallow bedrock, 2-muck, 3-whiteman, 4-brookfield, 5-paxton loam, 6-paxton stony loam, 7-canton sandy loam, 8-canton fine sandy loam, 9-canton stony-fine sandy loam, 10-canton stony loam. (In the classifications by US Soil Survey the first words of many soil types represent the places they are found.)
- *Buildings data* The value at each cell represents the building type at that cell, 0 is without building, 1 is commercial and services, 2 is gravel pit, 3 is dump, 4 is cemetery, 5 is residential.

The scale of each cell and the number of cells in each row or column should be defined in the beginning. In this case, each cell is a 30 meters times 30 meters square, there are 125 cells in each row and column.

### 3.2.2 Linear data type

Although raster data is easy to manipulate, it is limited in use. One reason is that it represents the average value within the cell, if the sampling interval in space is large, the resolution will be very low and the data is incapable of showing finer structures. For example, if a river goes through a cell, although it only occupies a small part of the whole cell, the entire cell will be taken as a river cell and cannot be distinguished which part the river goes through. Another reason is that raster data cannot represent relations between different clusters, for example, flow direction, relative position, or a path sequence. To solve this problem, raster data is converted into two different data types: linear and patch data types. Linear data type is used for representing streams, roads, and other linear data. The basic idea is to represent the network of streams (roads) as a collection of branches. Each branch is a sequence of raster cells, each cell connects to its subsequent cell. A branch can be isolated or connected to other branches. Every branch is assigned an index number, if it connects to other branches, the index numbers of the branches it connects and at what positions it connects will be denoted. These branches can be either directed (like streams) or undirected (like roads); if they are directed, the cell at each branch sequence should start at the source of the branch, if they are undirected, it doesn't matter if the source or the sink cell begins. The raster data can be converted to the linear data of the same layer and vice versa: given a cell, the index number of the branch where the cell lies on, the sequence number this cell is situated at the branch, numbers of other branches which this branch connects, and the connection points are all known. Similarly, given a branch index number, coordinates of all cells in the sequence of that branch will also be known.

To do the conversion from raster data into linear data, a recursive tracing algorithm is used. Suppose cells are located within grids, and the cell which is directly at the top, bottom, right, or left of the other cell is taken as a neighbor of that cell (notice a cell on the top-right corner of the other cell is not a neighbor of that cell, so are the cells at other corners).

Figure 3-2: Neighboring relationship of linear data

| | | |
|---|---|---|
| | neighbor | |
| neighbor | target | neighbor |
| | neighbor | |

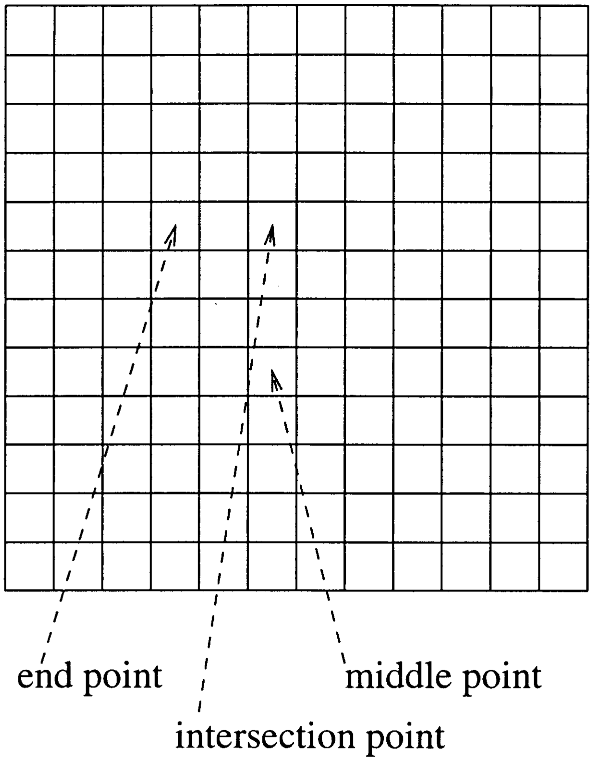
Two neighbor cells are connected if they have the same data values (for example, both are streams, the water body values are 2). Notice for a cell with linear data value, if it has one neighbor with the same value, it is the start or end cell of a branch, if it has two neighbors with linear data value, it is in the middle of a branch, if it has more than two neighbors with the same linear data value, it is the connection point to other branches.

Using this property, start from a cell with linear data value and has one neighbor with the same value, trace the cells along the branch to the connection point; then choose other branches at this connection point, and recursively trace all these branches of this tributary system; then start at the branch of another tributary system and trace it again; finally all branches of the linear data are traced and recorded. If the linear data is directed, data from other layers are used to help judging the direction, for example, the elevation data is used to determine the direction of stream flow.

### 3.2.3 Patch data type

Another important data type is patch data. It is used in the case that there exists an area with homogeneous property, so if we want to change the property (attribute)

Figure 3-3: Three different types of points of linear data



of that area, we only need to change the entire area rather than cell by cell. For example, when we want to cut a forest on the 2D map, we cut the trees of the patch. In GISVE system, patch data is used in representing tree cover and pond data.

The pond data is unordered and undirected, it contains an array of coordinates of its cells, an array of coordinates of its boundary cells, and index numbers of streams connected to (either flow in or flow out) the pond. The conversion is also bidirectional, pond data can be obtained from raster data and vice versa.

It uses trace algorithm to convert raster data into patch data, with some difference from the algorithm of linear data case. At first find out a boundary cell of a patch, then start looking for boundary cells of this patch by tracing along its boundary. The difference between this tracing algorithm and the algorithm used in linear data case is the definition of neighbors and the criteria for determining boundary cells. The neighbors in patch tracing include the corner cells. Thus the tracing criteria becomes: find the neighbor cells with the same patch data value and which have at least one neighbor cell with the non-patch data value; these cells are boundary cells.

Figure 3-4: Neighboring relationship of patch data

| | | |
|---|---|---|
| neighbor | neighbor | neighbor |
| neighbor | target | neighbor |
| neighbor | neighbor | neighbor |

For example, start from a boundary cell of a pond, choose one of its neighbor pond cell which has at least one neighbor land cell, then keep going until all boundary cells of that pond are exhausted. Then start at another pond boundary and do the same procedure, until all ponds are traced.

Then start finding the interior cells of each pond. The direct method is by checking all the cells to see if they are inside of a polygon formed by connecting pond boundaries. Here a computational geometry algorithm is used [MS91, pages 236–237] to determine if a cell is inside the polygon: choose a point which is obviously outside the polygon (has very large coordinate), connect the outside point with the point we want to check. Count the number of edges of the polygon this connected line intersects with, if it's an odd number, the point is inside the polygon, if it's an even number, the point is outside the polygon. By using this method all interior cells of a patch can be determined. Assign each pond a different index number and put cells of that pond under its index number.

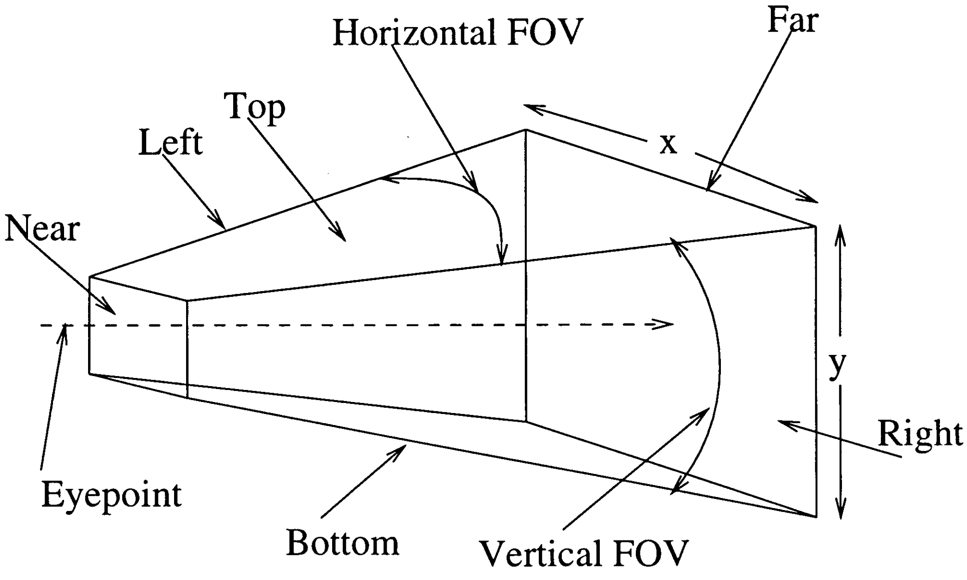
### 3.2.4 GIS data updating

Once GIS data files are loaded into the system, they are stored as 2D arrays, and converted into linear or patch data types or remain as raster type. These data in the system are dynamic. There are two ways to update the GIS data in the system:

1. *Data changed by the user* The user can choose add or delete functions on the 2D window to change the raster, linear, and patch data. The add/delete functions of raster data are straightforward: the user chooses cells on 2D map, and the system adds or deletes these cells. In add function of linear data, the user chooses a sequence of cells, the system automatically finds out all cells on the lines connecting two adjacent selected cells, add these cells to the raster data arrays and a new branch of linear data. In delete function of linear data, the system finds all existing branches connecting two selected cells, deletes all the branches in the middle, or part of the branches where the start point and end points are situated, from both linear data structure and raster data array. In add function of patch data, the system finds all interior cells of the polygon whose vertices are chosen by the user. Adds these cells into raster data array and converts them into a new patch data. In delete function of patch data, the system finds all existing patch data cells inside the polygon chosen by the user, deletes these cells from raster data array and converts them into a new patch data.

2. *Model simulation result* GIS data are both inputs and outputs of models: the change of GIS data modifies the input parameters of environmental models, hence the simulation result will change; the simulation result, however, influences the GIS data. So the overall process of modification-environmental impact forms a feedback loop. For example, the fire will erase houses and trees, or flood will increase water-covered area. In the three models chosen, the outputs are all of raster data type, they will directly update the raster data in the array.

Figure 3-5: Viewing frustum



After the 2D raster data is changed, it will be converted into a new 2D linear or patch data, which will transform into a new 3D object data. The updated data changes the display on 2D map, 3D scene, and environmental model inputs, so the data updating affects all operations of the system.

## 3.3 2D to 3D data modules

### 3.3.1 The overall structure of the scene graph

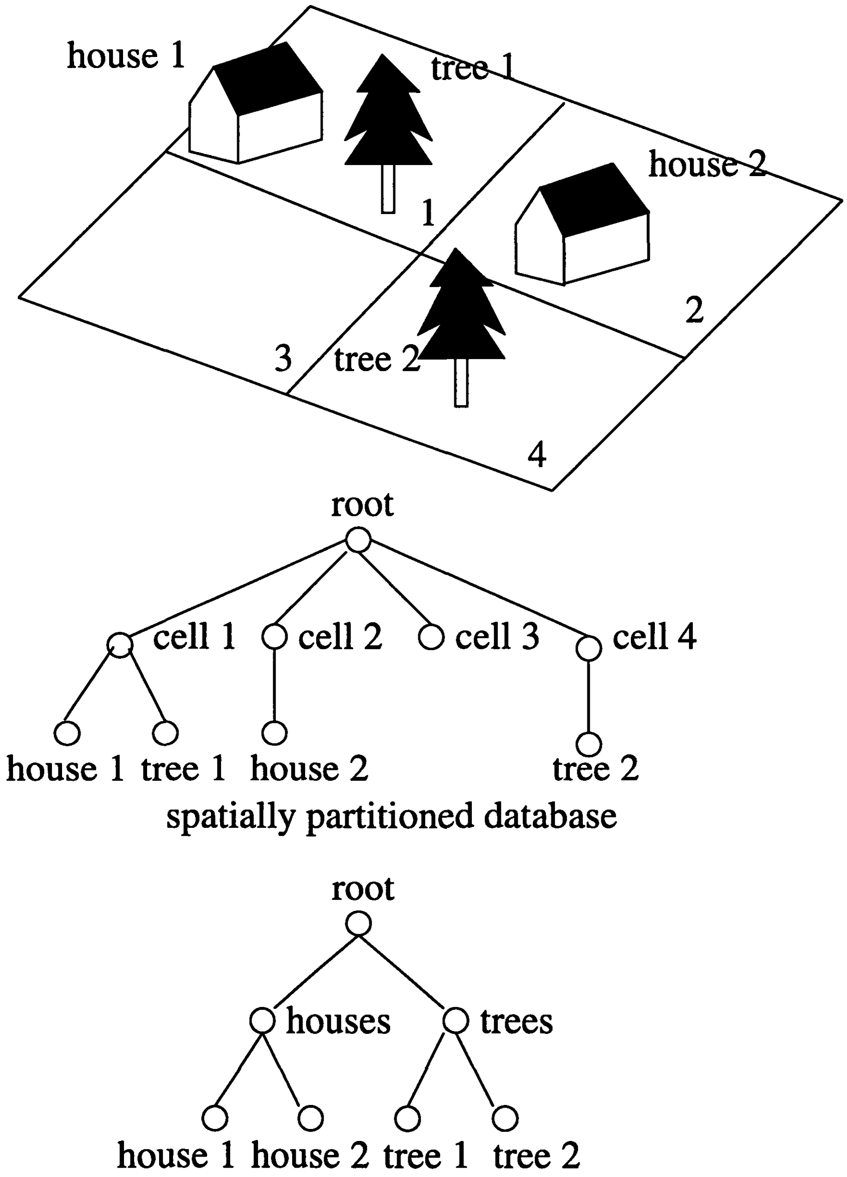
The GSIVE system used SGI Performer as its graphical toolkit [$F^+94$]. It provides function calls of drawing 3D objects, setting environmental and perspective parameters such as light sources and view volumes. It takes care of low level graphical processing.

A scene in Performer is represented as a hierarchical structure called graph. Each

graph contains several nodes, each node can have several child nodes. A node can represent a 3D object, many 3D objects clustering together, a light source, a coordinate system, or a level-of-detail (LOD) structure. Usually different objects in the scene are represented by different nodes in the graph, and their distances in the tree structure depends on their distances in the 3D space. For example, if a house and a tree are close to each other, they will have the same parent node, this parent node represents the group of all the objects of a small area where the house and the tree are situated.
However, considering the efficiency of loading GIS database, another juxtaposition of nodes is preferred. Different layers of GIS data are stored at different files, if we load each file into Performer one by one, then different layers are represented by different nodes; this is a shallow and wide structure, it only has two levels, one is the root node and the other has different GIS data layers, for example, one node represents all the houses in the scene.

The first choice of node allocation is consistent with object's spatial relationship, and it is more efficient in culling process (to determine which objects or which part of objects are within the viewing volume), but it's difficult to load data into these nodes. The second choice, on the contrary, is convenient in loading GIS database, but it needs more time in culling. Another important consideration in determining the structure of nodes is the efficiency of loading or removing one part of the entire scene: if the user only changes a part of the place, there is no need to load or remove entire data layer, it is preferred that as few nodes are influenced each time as possible.
The solution is the combination of these two structures. The scene graph is divided according to both the spatial relationship and the data types. The entire area is evenly divided into 25 blocks by grids, each block has has one fifth the length and width of the entire place. It takes 3 levels in the scene graph tree to build the GIS database in the 3D environment. The root (scene of the entire area) has five child nodes, each node represents a row in the area (5 blocks); each of these nodes also has five child nodes, these nodes represent individual blocks. Each block node contains

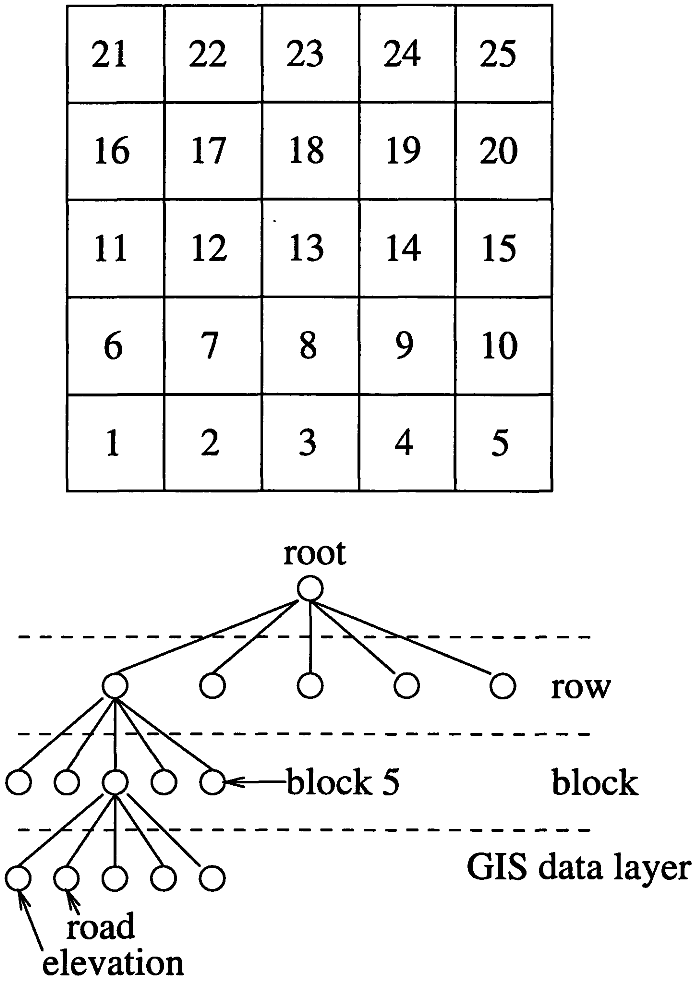
Figure 3-6: 3D database partition



spatially partitioned database

logically partitioned database

Figure 3-7: 3D database partition of GISVE



| | | | | |
|---|---|---|---|---|
| 21 | 22 | 23 | 24 | 25 |
| 16 | 17 | 18 | 19 | 20 |
| 11 | 12 | 13 | 14 | 15 |
| 6 | 7 | 8 | 9 | 10 |
| 1 | 2 | 3 | 4 | 5 |

a fixed number of child nodes, they correspond to different GIS layers, including elevation, streams, roads, ponds, houses, trees, flood, pollution, fire. When 2D data is modified, only the node representing a specific data layer (which is chosen by the user) and whose parent is the block node where the modified data is situated will be changed.

Functions converting 2D data to 3D objects are all block-based: they only do the conversions for the data within a specific block. This greatly reduces the processing time of 2D data functions and 2D to 3D functions. It also facilitates the culling process since spatial correlation is maintained. The only drawback occurs when loading the entire layer of GIS data. Compared to directly loading the entire layer into one node, the overhead of initialization time of loading all blocks is 25 times greater. However, compare the frequencies of use between the functions of modifying and loading data, the former is much higher than the later. For this reason, using blocks to load and change data is more preferable.

### 3.3.2 3D data format

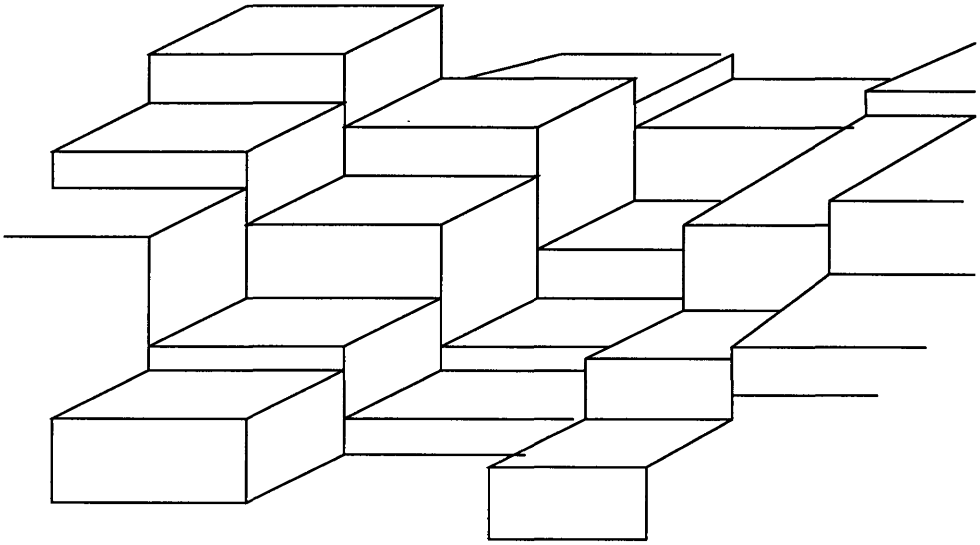
Performer accepts several 3D data formats, users can also write their own 3D database loaders for other file formats. Here the only available data is 2D GIS, all the 3D object files have to be created from these files. The SuperViewer (SV) format is chosen as the 3D data format of GISVE. This is one of the formats that the I3DM database modeling tool can read and write, and also supported by SGI Performer. Each file contains several models, each model consists of polygons. The polygon data includes the 3D coordinates of vertices, 3D normal directions of vertices, 2D texture coordinates, and the number of vertices. The file contains a header defining the color (RGB coordinates of ambient, diffuse, emissive, and specular colors, shineness) and texture (the texture file name).

The taxonomy of raster, linear, and patch data types is also used in classifying functions converting from 2D data into 3D data. The following paragraphs will discuss the implementation of these conversion functions based on different 2D data types.

### 3.3.3 Elevation to 3D function

Elevation data is different from other raster data because the terrain should be continuous in 3D graphics. We want to transform the elevation data file into polygons that represent the terrain. It is not easily done because the raster GIS contains no

Figure 3-8: Discontinuous representation of terrain



information of the finer structure within the cell, and using flat planes with certain heights for all cells to represent terrain will cause discontinuity at boundaries between cells.

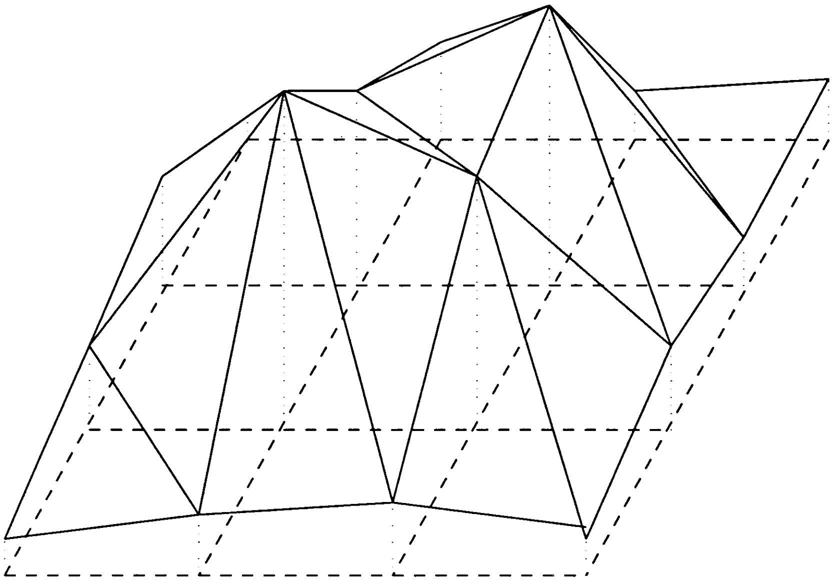
In order to avoid discontinuity, we use the TIN (triangular irregular network) structure in representing the terrain.

Each cell is composed of two triangles whose vertices are the vertices of the cell. The overall terrain is constituted with them; by drawing colors and textures on these triangles, the overall structure looks like a smooth surface.

### 3.3.4 Raster to 3D function

There are several raster GIS data layers: houses, trees, flood, pollution, and fire. The complexity of representing these data depends on the need of the application domain. For flood, fire, and pollution data, their 3D representations are simple. We attach

Figure 3-9: Triangular network representation of terrain

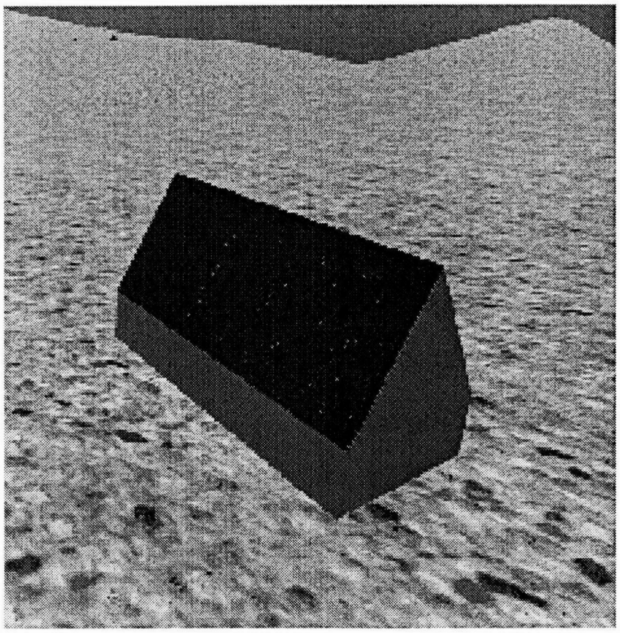


colors and textures on terrain surface to represent the area with specific values in these data layers, the meanings of the coded colors are explained in the legend; for example, a clump of red represents the area which is on fire. These layers, unlike houses or trees, are abstract representations of the attributes of a place, they lack concrete forms; for this reason, it makes sense to use symbols or colors to represent them.

For houses, the representation is a little more complicated. We use 6 polygons to represent roofs and walls. The roof is wedge shaped and colored red, the walls are gray without windows and doors. Each cell contains one house, the house sits at the center of the cell. Since the scale is 30 meters by 30meters per cell, it is obviously not true in the urban environment, but in a suburban area, this density is probably close to the real case.

The representation of houses is still too simple, it has no finer structures like door, window, chimney; and all the houses are of the same style and size. This may be true for a community designed by the same person or developed by the same plan. The

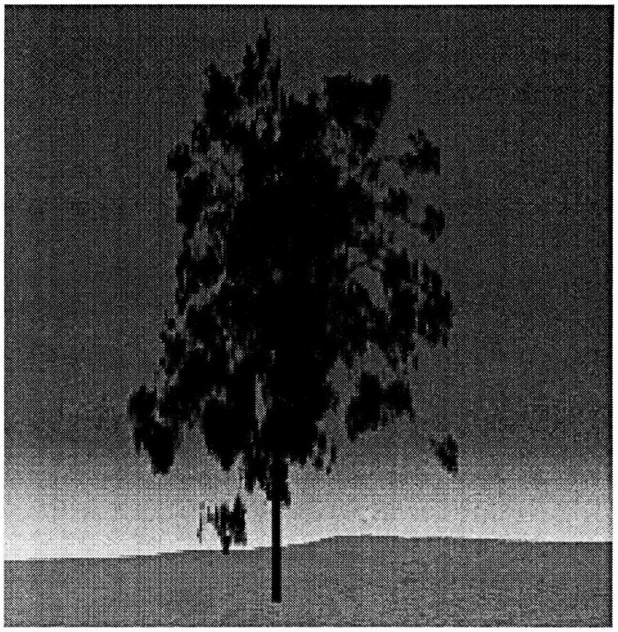
Figure 3-10: The house in the 3D environment



more important reason of allowing the homogeneity is that the main interest of this tool is for understanding the environmental impact of the area, rather than individual houses' architectural forms. Also, when more detailed structures are put in, the more polygons the scene has, the slower the response of the system to user's action will be. Thus the simple representations of houses are acceptable.

The problem of putting trees in the 3D scene is more complicated. Trees are one of the most often used materials in landscape design, they cannot be simplified by icons. We need a way to draw more realistic tree images in the 3D environment. It is very difficult to represent trees by polygons. There are algorithms generating trees using fractal geometry [Man83], the algorithms are easy to run in generating complex shapes like trees, but these shapes are to complex to put into the 3D scene, it has so many polygons that the rendering speed will be dragged to a sluggish pace. So using fractal geometry algorithms to generate tree polygons is not feasible.
The more feasible way is to use billboard nodes in the scene graph. A billboard is a type of graphical node that has several texture files attached at different transparent planes. Viewers see different facets of the same object when they view from different

Figure 3-11: The tree in the 3D environment



perspectives, thus generates cubic sense. Each tree has only 2 (or 4) quadrilaterals attached with different texture files which are the images taking from different viewing angles. This technique greatly reduces the number of polygons needed to achieve similar visual effect in plotting complicated objects.

However there are two problems in texture mapping. The first one is that the time of processing texture mapped polygons is about three folds of that of processing untexture mapped ones. This makes the response of updating the view after each movement very slow, even though the number of polygons are reduced. The second problem is that there are still too many polygons if we endow each tree two or polygons to depict the forest, the processing time will slow down drastically that the essential element of a virtual environment system - the real time response of the updating of visual display, will be ruined completely. So we still can not construct a forest from individual trees.

In order to overcome these difficulties, the level of detail method is used. A forest seen far away is only a patch of green, it's not necessary to draw individual trees for distant cells, rather, they can be replaced by colored polygons. Performer provides functions building level of detail nodes; each node contains several child nodes representing the same object viewing from different distances, when the distance between the viewer and object is within a certain range, the display automatically switches to the node corresponding to this range. In GISVE there are three levels of detail representing the trees: the first (the nearest) level is the node filled of individual trees, each tree is texture mapped; the second level is the node representing a volume attached with texture images of a forest at its boundary; the third (the most far) level is the node representing a patch of green.

The rendering speed increases greatly when the level of detail method is used, however, the problem is only halfly solved. Keeping all the polygon coordinates and color pixel values of the these three levels of all blocks on computer will fill up the memory and CPU time even though only one level is displayed at one time, the time response is still very poor. In order to further reduce polygons, we decide to put only one block of individual trees in the computer: create a node embedded with the dynamic coordinate system under the scene graph. Objects within this node can move independently, this node contains texture mapped polygons of twenty five trees within one block. When the viewers move and they are within the range of the forest, this node moves with them so they can see individual trees surrounding them; when the viewers are not in the tree cells or their altitudes are too high, this tree node becomes invisible. Since the only first level tree node which appears is this one, all other first level tree nodes of the twenty five blocks can be eliminated from the scene graph, thus, only two level of detail nodes need to be loaded for each block.

### 3.3.5 Linear to 3D function

The raster to 3D function directly converts 2D data into the Super Viewer file, then loads the file into the scene graph. If the 2D raster data is already converted into 2D linear data, the linear to 3D function is needed to convert the linear data into the SV file.

The linear data (streams, roads) is not cell based. If we use raster data type to represent roads or streams in the 3D scene, because the scale is enlarged, they will look like zig zag lines rather than smooth curves, which deviates from the real images of them. In order to draw the linear data in 3D scene, we have to create smooth strips. The linear data has more information than the raster data. For a branch of streams or roads, the linear data contains information about the number of vertices, 3D coordinates of these vertices, other branches it connects to, and the boolean value denoting if this branch is directed (stream) or not (road). To convert the linear data into 3D objects, the first step is to enlarge the scale to that of the 3D scene. Then spline this branch into a smooth curve. There are several spline algorithms which are commonly used in computer graphics. Cubic spline uses cubic polynomial curves to interpolate the sampled points that make the entire curve continuous and smooth (the first order derivatives are continuous at intersection points). Bézier curve combines several curves together by adopting the binomial distribution, it creates smooth curves around the "pivot points", which are sampled points [$F^+$90]. These algorithms don't quite fit the demand of the spline function here, because by using these two algorithms, the start and end points of splined curves of one branch don't overlap those of other branches even though these branches are connected; this results in converting two connected, piecewise lines into two separate and smooth curves; the topology of the linear is not conserved after the spline.

The more appropriate algorithm is the Hermite curve [$F^+$90, pages 482–488], it fetches every 4 vertices to create a series of cubic polynomial curves; the start and end points overlap with the original ones [$F^+$90]. (Appendix 1)
After the spline has been finished, the formula of the polynomial corresponding the smoothed curve segment of four consecutive points is known; sample ten points on each polynomial curve, this makes the number of new sampled points 10/4=2.5 times of original points.

The next step is to transform lines into strips, i.e. add width on the linear data. A point on the line is transformed into a pair of points, which are separated at a distance of the width of linear data (the width of streams or roads, which is less than

Figure 3-12: Mismatch of splined curves when end points do not intersect

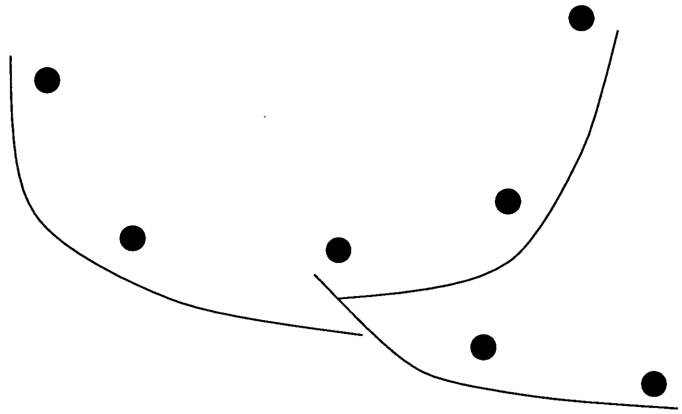


Figure 3-13: Linear data spline

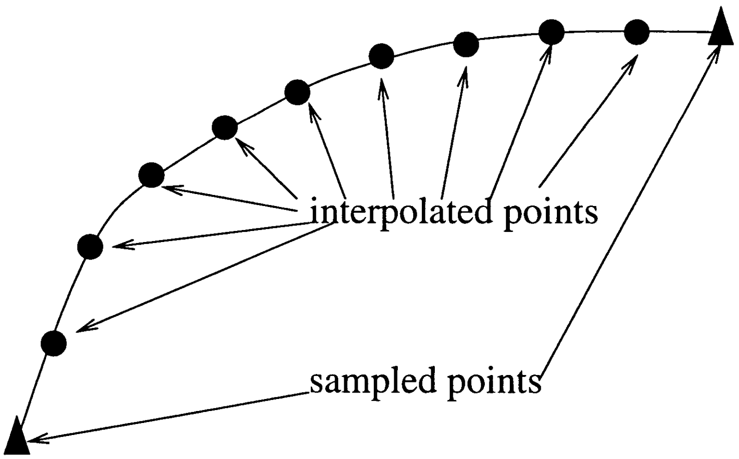
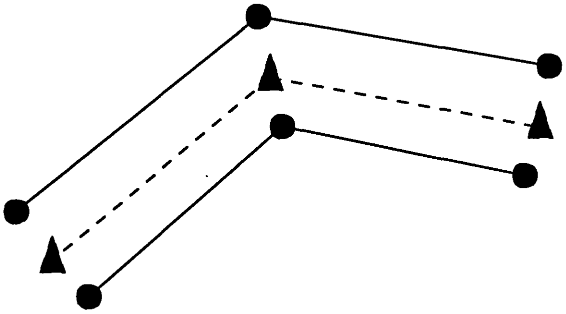


Figure 3-14: Strip on plane

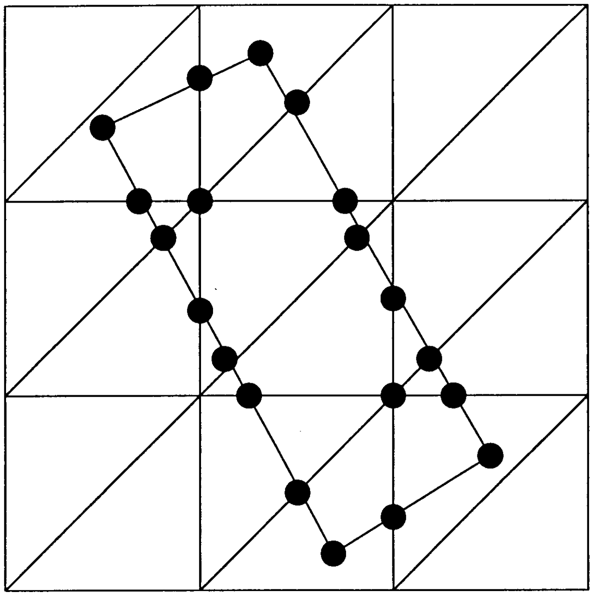


the cell width); the lines connecting this point pair are orthogonal to the original line or equally divides the angle between two adjacent lines of the splined linear data.

Then put this 2D strip on the 3D space to fit the terrain. Because we use TIN structure to represent the terrain, this process is equivalent to: to map an area on the $(x, y)$ plane to the triangular network surface. We have to find out all the triangles the strip contains and which parts of these triangles belong to the strip area. At first we want to find out all the vertices constituting the strip polygons on the terrain surface, they must be within one of the following types: the vertices of elevation triangles whose projections on the $(x, y)$ plane are within the strip area, the intersection points of strip edges to triangle edges, and the vertices of the strip. Then collect all the points on the same triangle together, reorder these points (counterclockwisely or clockwisely), then draw the new 3D polygons. The heights at these points can be calculated by knowing the plane equation of the triangle the points are situated.

The linear to 3D function needs many more calculations than the raster to 3D function, thus it takes longer to load or modify linear data. Notice the linear to 3D conversion is also block based, it only converts the branches that are within the blocks

Figure 3-15: The boundary points of a strip



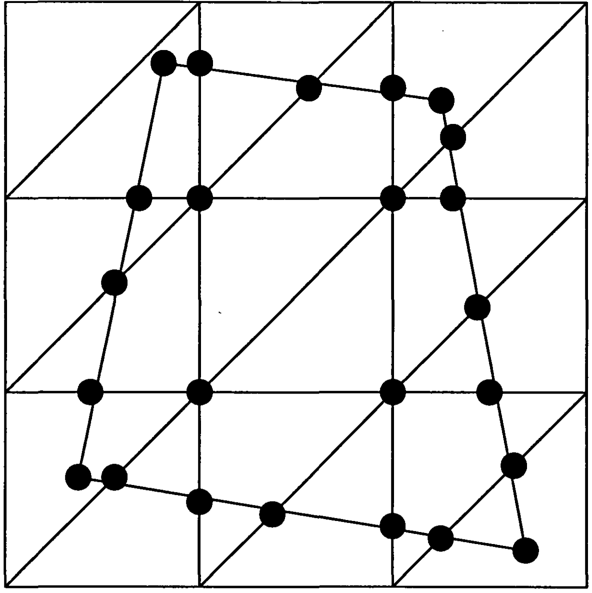
been chosen; this makes the response faster when modify linear data.

### 3.3.6 Patch to 3D function

The only patch data needed to convert into 3D object is pond. The pond is represented by texture mapped blue polygons with smooth edges. Forests are taken as raster data as described above.

The most important information of patch is its boundary, which is already found when converting the raster data into the patch data. Boundary cells and interior cells of a patch are treated separately. For an interior cell, since the entire cell is within the patch, it can be drawn in 3D window the same way as the raster cells. For a boundary cell, we want to know the part of this cell belonging to the pond. The algorithm of putting the boundary cells to fit the terrain surface is similar to that of putting the strips on the terrain. At first enlarge the scale; then find out the vertices of the polygons which are the intersections of the patch area and the elevation triangles where the boundary lines of the patch lay on. Reorder these vertices and construct these polygons.

Figure 3-16: The boundary and interior points of a patch



## 3.4 Navigate modules

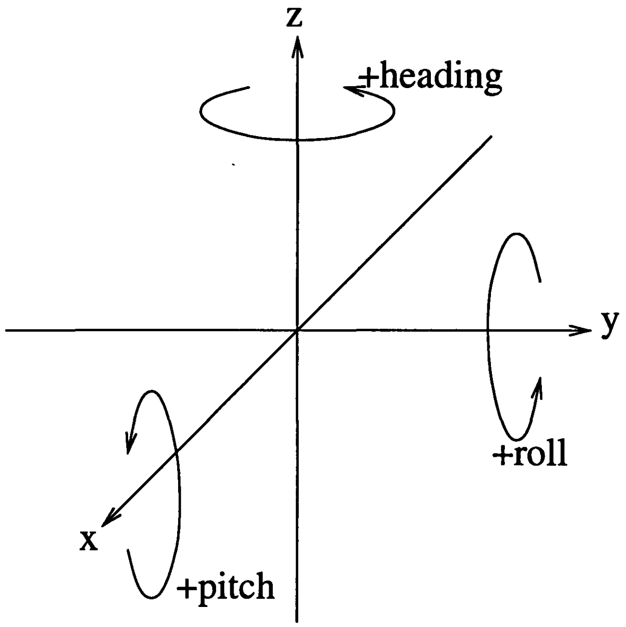
### 3.4.1 Coordinate transformations

Interactive navigation is one of the most important parts in virtual environment that endows the user the feeling of reality. In order to do that, the computer should fetch the current position and moving direction of the viewer via the user interface such as the mouse or the head tracker. Then the computer will use these values to manipulate the coordinate transformations; these transformations change the positions of objects in the user's point of view, so will give the user the feeling of movement in the 3D space.

In three dimensional space, the position of the viewer is represented by the $x, y, z$ components of the viewer's coordinate; the viewing angle of the viewer at this position is also represented by three numbers: heading, row, and pitch. Heading is the shift angle turning around the $z$ axis, row is the shift angle turning around the $y$ axis, and pitch is the shift angle turning around the $x$ axis.

When all objects of the environment are fixed and the viewer stares at an object at a specific position and angle, in the viewer- centered coordinate system, it is equivalent to shift the origin to the viewer's current position and rotate the three axes to

Figure 3-17: Viewing angles in 3D space



make the line of sight pointing to the object being stared. In Performer's setting, the line of sight is the positive direction of $y$ axis. After the position $(x, y, z)$ and viewing angles (heading, row, pitch) are determined, the specific view has been formed. By updating the viewing position and angle successively the sense of movement is generated.

We can use homogeneous coordinate to represent a point in 3D space. It is a four-component vector: [F+90, pages 213–226]

$$\begin{pmatrix} x & y & z & 1 \end{pmatrix}^T$$

The fourth component is usually set to be 1; all vectors with the same proportional number

$$\begin{pmatrix} Wx & Wy & Wz & W \end{pmatrix}^T$$

correspond to the same point in 3D space for all W.

Translations and rotations can be represented as transformation matrices multiplications to the coordinate vectors. For the translation from $(x, y, z)$ to $(x + d_x, y + d_y, z + d_z)$, the transformation matrix is

$$T = \begin{pmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.1}$$

For heading rotation, the transformation matrix is

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.2}$$

For pitch rotation, the transformation matrix is

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

For row rotation, the transformation matrix is

$$R_y(\alpha) = \begin{pmatrix} \cos\alpha & 0 & \sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.4}$$

The overall coordinate transformation is the product of these four transformations.

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3.5}$$

$$= T * R_z(\theta) * R_x(\phi) * R_y(\alpha) * \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3.6}$$

$$\tag{3.7}$$

So when the translation vector and three viewing angles are known, the overall coordinate transformation can be determined. However, these matrices are non-commutative, so the order of operations should be specified in the beginning.

If objects can move independently from the viewer, other coordinate systems should be introduced. Each moving object has a dynamic coordinate system (DCS), which transforms independently from the viewer's coordinate system. If objects are fixed, their coordinates will change only according to the viewer's transformation matrix, in this case they are under the same static coordinate system (SCS).

In order to update the viewing position and angle or the transformation matrix, the computer should know the viewer's moving direction and attitude through some user interfaces. This interface can be a head tracker mounted on a helmet which tracks the rotation of head, or a wired glove which tracks the position of hand in 3D space. As mentioned in section 2.3.5, it is not absolutely necessary for a system aiding landscape design to equip these multi-modal interfaces. The GISVE uses the mouse as the sensory input device, it maps the movement of the mouse into the movement in the 3D space.

In GISVE, the navigate functions are updated every loop. Each loop the system changes the view by doing coordinate transformations. These transformations are based on the position and moving direction of the mouse or the coordinates of a predetermined path or position. The frame rate is set at 30Hz, so the perspective updates every 33ms, which is fast enough to create the sense of continuous movement. However, when the system is heavily loaded the frame rate will decrease to an extent that discontinuity can be perceived.

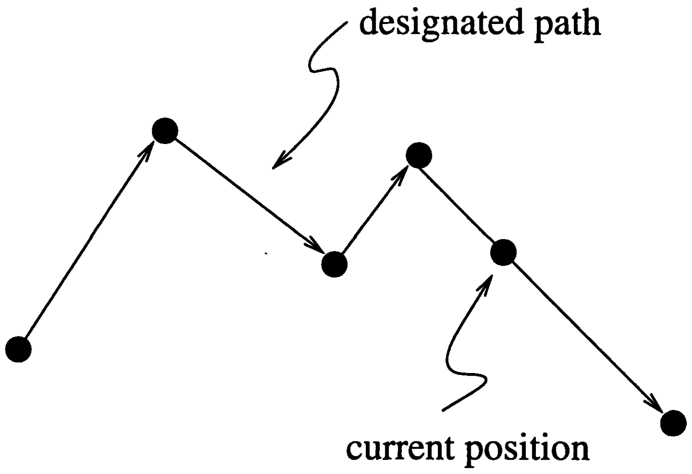
### 3.4.2 Fly function

The fly function allows the user to render in 3D space; it provides the largest degree of freedom among all navigate functions. It simulates the manipulation of the stick in the cockpit: move the mouse right turns the user right in the virtual environment, and move the mouse left turns the user left. The forward movement of the mouse corresponds to the pitch down movement, and the backward movement of the mouse corresponds to the pitch up. Acceleration or deceleration is done by pressing the left or right buttons. The fly function is built in the infinite main loop of GISVE. Every loop the system polls the status of mouse to get the transformation matrix of the mouse movement, then applies this matrix to update the view. The fly function also changes the position icon of the 2D window: it updates the coordinates on the 2D plane and moves the position icon to the cell that the viewer is situated in the 3D environment.

### 3.4.3 Drive function

The drive function is similar to the fly function except it only allows the movement along the terrain. The operation is also close to fly function, but the pitch up and pitch down will not change the altitude of the viewer. The $z$ coordinate of the viewing position is set to be the surface elevation plus a shift value, so by using the drive function changes of terrain are more easily detected. The position icon on the 2D window is also updated according to the movement in the 3D environment, as the fly

Figure 3-18: Automatic drive function



function does.

### 3.4.4 Automatic drive function

Both fly and drive functions are manipulated in the 3D environment. Although they provides the maximum degree of freedom, they are sometimes confusing for users who are not familiar with the system or the place being displayed. It needs navigate functions that allow the user to manipulate on the 2D map and to browse on the 3D environment. Automatic drive function satisfies such needs. By setting a path on the 2D map and applying the automatic drive function, the viewer moves automatically along the designated path. The path is selected by clicking a sequence of points on the 2D map; the moving direction is the vector connecting two adjacent points on the selected path, and the speed of moving is a built-in value, which can be set by the user.

### 3.4.5 Fix position function

Fix position is an easy but important navigate function. The viewer chooses a point on the 2D map and applies this function, then he/she will go to that position with a fixed viewing angle (the line of sight is toward the positive direction of $y$ axis of the original coordinate system). The position icon is fixed at that position. The user can choose the point by using mouse or keyboard input.

### 3.4.6 Perspectives function

Perspectives function offers the turning around perspectives. When applying this function, the user will stand at a fixed position and turn around, so he/she sees in the panorama view. It is required because sometimes demonstrating various views looking from one place are important in deciding good places to build houses or other constructions. This function is implemented by updating the heading (yaw) angle periodically while leaving other viewing angles and position coordinates invariant.

## 3.5 Model modules

### 3.5.1 Basic concepts

In addition to the navigation in 3D space, GISVE system provides functions which allow the user to see the temporal change of environment. Model modules are the core of the dynamic functions. By manipulating the environmental model functions, the environmental impact due to natural or anthropogenic factors is displayed on both 2D map and 3D environment.

There are several concerns listed as follows about the choice of environmental models implemented in the system.

- The environmental models should cover the important parts of landscape design. Here the "important" is from environmental concern. Most of them are related to water and soil: flood area, soil loss, ground water contamination, etc. Besides, fire is also an important concern in some cases.

- The input parameters of environmental models should be able to be directly obtained from or derived by the available GIS data. For the system we build so far, this implies that the input parameters should be within one or several of the following categories: elevation, water body distribution, road, forest distribution, soil type, and houses. Many environmental models need to input hundreds of parameters, these models, although more accurate and useful, are not appropriate to implement in GISVE because the input parameters are not able to be incorporated in the system.

- The model should be based on distributed database and generate distributed results rather than aggregate inputs and outputs. Since the aim of implementing models is to display its spatial consequences, the model which only shows average values over the entire area is less interesting. For example, a model calculating the hydrograph of an area does not need 2D or 3D map to demonstrate the result, while the model showing the water level of each tributary system has good reasons to display in GISVE. Since the GIS data used is cell-based, the model is also run cell by cell.

- The model needs not to be complicated, it is even less desirable when we have accurate but slow running performances. If the running time of simulation is more than ten minutes, the sense of interactivity of GISVE system will be lost. Some may argue that "simple" models will not generate significant results, and even worse, the inaccurate results may cause serious effect when making important decisions about environmental change. However, there are two reasons that support us to adopt simple environmental models comfortable. The first reason is that this system is designed for the preliminary stage of landscape design, in the other words, the function is to give the user a quick impression, rather than precise prediction, of the environmental impact. The second reason is that the GISVE system is still at developing stage, it demonstrates the potential of application, rather than affirms that it is mature for practical uses. In later software development stages more sophisticated models can be incorporated, but currently the most important issue is how the result of model simulation can be visualized.

- There are two modes of running simulations in terms of the time response. One mode is the fix duration simulation: it runs the simulation for a fixed time of duration, the duration is set by the user. For example, the user wants to know the flood water level after the first half hour of heavy rain, he/she sets the duration to be 30 minutes. After running the simulation for 30 minutes, the 2D and 3D windows shows the scene 30 minutes after the beginning, the dynamic process from the start to end is not shown. This mode is useful if the user only wants to know the result. Another mode, which is the dynamic simulation, updates the scene after a fixed period of time. This mode is useful when the user cares about the dynamic process of environment.

According to these concerns, three models are chosen to put in the GISVE system: flood model, soil erosion carried pollution model, and fire diffusion model.

### 3.5.2 Flood model

The flood model selected is CAS2D model. This model was developed by Julien, Saghafian, and Ogden [JSO95]. It is GIS based, raster distributed, and all parameters can be either obtained from GIS database available or set by the user (including surface, elevation, soil infiltration rate, rain amount and duration, and stream channel depth and width). It calculates the surface runoff volume of each cell at each step, then obtains the accumulated water level of each cell.

The flood model is based on continuity equations, infiltration equations, and momentum equations. The change of water level can be attributed to several terms: rainfall, soil infiltration, overland flow and stream flow. The land cells, stream cells and pond cells are treated separately.

The effective rainfall rate is the rainfall rate minus the infiltration rate.

$$i_e = i - f \tag{3.8}$$

The infiltration rate $f$ can be calculated from the Green and Ampt equation:

$$f = K(1 + \frac{H_f M_d}{F}) \tag{3.9}$$

$f$ = infiltration rate, $K$ = hydraulic conductivity, $H_f$ = capillary pressure head at the wetting front, $M_d$ = soil moisture deficit, $F$ is the total infiltration depth, which is the integral of f over time. These parameters are dependent upon the soil class. Table 3.1 is the measurement result of these parameters from Rawls, Brakensiek and Miller [R+83].

The soil moisture deficit $M_d$ is the effective porosity $\theta_e$ minus the initial soil moisture content $\theta_i$. In this paper, the initial soil moisture content is set to be 0 for all cells without losing generality.

Since we use finite difference method to solve the differential equations of this model, at each time step, the change of infiltration rate can be expressed as follows:

$$f^{t+\Delta t} = \frac{1}{4\Delta t}\{(2K\Delta t - 2F^t) + [(2K\Delta t - 2F^t)^2 + 8(KF^t + KH_f M_d)2\Delta t]^{1/2}\} \tag{3.10}$$

While $F^{t+\Delta t} = F^t + f^t \Delta t$

The continuity equations of land cells and stream cells have different forms. For land cells, the equation is:

$$\frac{\partial h}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} = i_e \tag{3.11}$$

$h$ is the surface flow depth, $q_x$ is the unit discharge of overland flow in $x$ direction with the unit of area/time, $q_y$ is the unit discharge in $y$ direction, $i_e$ is the effective

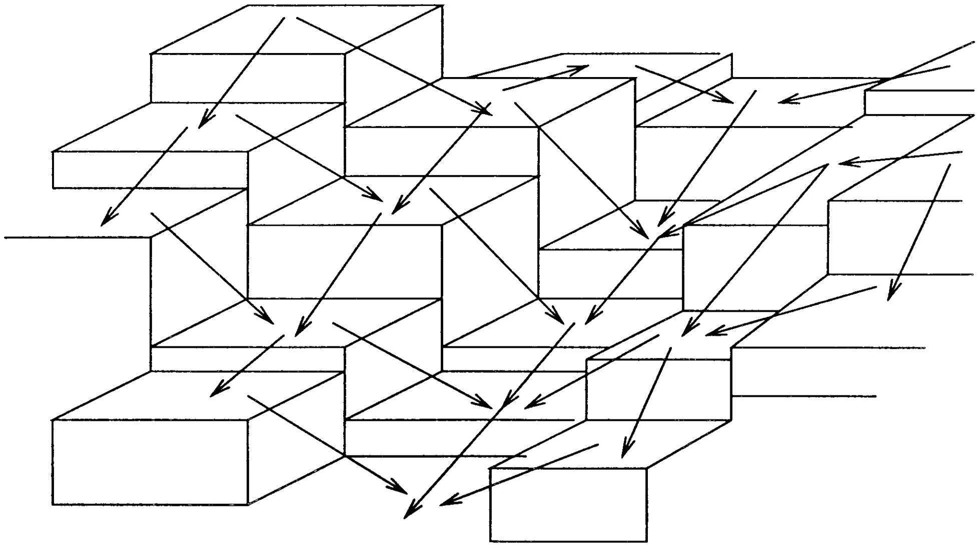
[1] Numbers within standard deviation

[2] Insufficient sample to determine parameters

Table 3.1: Green and Ampt parameters

| Soil texture class | horizon | Sample size | Total porosity, $\phi$ in $cm^3/cm^3$ | Effective porosity, $\theta_e$, in $cm^3/cm^3$ | Wetted front capillary pressure, $\psi_f$ in $cm^3$ | Hydraulic conductivity, K, in $cm^3/hr$ |
|---|---|---|---|---|---|---|
| Sand | | 762 | 0.437(0.374-0.500)[1] | 0.417(0.354-0.480) | 4.95(0.97-25.36) | 11.78 |
| | A | 370 | 0.452(0.396-0.508) | 0.431(0.375-0.487) | 5.34(1.24-23.06) | |
| | B | 185 | 0.440(0.385-0.495) | 0.421(0.365-0.477) | 6.38(1.31-31.06) | |
| | C | 127 | 0.424(0.385-0.463) | 0.408(0.365-0.451) | 2.07(0.32-13.26) | |
| Loamy sand | | 338 | 0.437(0.363-0.506) | 0.401(0.329-0.473) | 6.13(1.35-27.94) | 2.99 |
| | A | 110 | 0.457(0.385-0.529) | 0.424(0.347-0.501) | 6.01(1.58-22.87) | |
| | B | 49 | 0.447(0.379-0.515) | 0.412(0.334-0.490) | 4.21(1.03-17.24) | |
| | C | 36 | 0.424(0.372-0.476) | 0.385(0.323-0.447) | 5.16(0.76-34.85) | |
| Sandy loam | | 666 | 0.453(0.351-0.555) | 0.412(0.283-0.541) | 11.01(2.67-45.47) | 1.09 |
| | A | 119 | 0.505(0.399-0.611) | 0.469(0.330-0.608) | 15.24(5.56-41.76) | |
| | B | 219 | 0.466(0.352-0.580) | 0.428(0.271-0.585) | 8.89(2.02-39.06) | |
| | C | 66 | 0.418(0.352-0.484) | 0.389(0.310-0.468) | 6.79(1.16-36.95) | |
| Loam | | 383 | 0.463(0.375-0.551) | 0.434(0.334-0.534) | 8.89(1.33-59.38) | 0.34 |
| | A | 76 | 0.512(0.427-0.597) | 0.476(0.376-0.576) | 10.01(2.14-46.81) | |
| | B | 67 | 0.512(0.408-0.616) | 0.498(0.382-0.614) | 6.40(1.01-40.49) | |
| | C | 47 | 0.412(0.350-0.474) | 0.382(0.305-0.459) | 9.27(0.87-99.29) | |
| Silt loam | | 1206 | 0.501(0.420-0.582) | 0.486(0.394-0.578) | 16.68(2.92-95.39) | 0.65 |
| | A | 361 | 0.524(0.444-0.610) | 0.514(0.425-0.603) | 10.91(1.89-63.05) | |
| | B | 267 | 0.533(0.430-0.636) | 0.515(0.387-0.643) | 7.21(0.86-60.82) | |
| | C | 73 | 0.470(0.409-0.531) | 0.460(0.396-0.524) | 12.62(3.94-40.45) | |
| Sandy day loam | | 498 | 0.398(0.332-0.464) | 0.330(0.235-0.425) | 1.85(4.42-108) | 0.15 |
| | A | -[2] | - | - | - | |
| | B | 198 | 0.393(0.310-0.476) | 0.330(0.223-0.437) | 26.10(4.79-142.30) | |
| | C | 32 | 0.407(0.359-0.455) | 0.332(0.251-0.413) | 23.90(5.51-103.75) | |
| Clay loam | | 366 | 0.464(0.409-0.519) | 0.309(0.279-0.501) | 20.88(4.79-91.10) | 0.10 |
| | A | 28 | 0.497(0.434-0.560) | 0.430(0.328-0.532) | 27.00(6.13-118.9) | |
| | B | 99 | 0.451(0.401-0.501) | 0.397(0.228-0.530) | 18.52(4.36-78.73) | |
| | C | 55 | 0.452(0.412-0.492) | 0.400(0.320-0.480) | 15.21(3.79-61.01) | |
| Silty clay loam | | 689 | 0.471(0.418-0.524) | 0.432(0.347-0.517) | 27.30(5.67-131.50) | 0.10 |
| | A | 65 | 0.509(0.449-0.569) | 0.477(0.410-0.544) | 13.97(4.20-46.53) | |
| | B | 191 | 0.469(0.423-0.515) | 0.441(0.374-0.508) | 18.56(4.08-84.44) | |
| | C | 39 | 0.475(0.436-0.514) | 0.451(0.386-0.516) | 21.54(4.56-101.7) | |
| Sandy clay | | 45 | 0.430(0.370-0.490) | 0.321(0.207-0.435) | 23.90(4.08-140.2) | 0.06 |
| | A | - | - | - | - | |
| | B | 23 | 0.435(0.371-0.499) | 0.335(0.220-0.450) | 36.74(8.33-162.1) | |
| | C | - | - | - | - | |
| Silty clay | | 127 | 0.479(0.425-0.533) | 0.423(0.334-0.512) | 29.22(6.13-139.4) | 0.05 |
| | A | - | - | - | - | |
| | B | 38 | 0.476(0.445-0.507) | 0.424(0.345-0.503) | 30.66(7.15-131.5) | |
| | C | 21 | 0.464(0.430-0.498) | 0.416(0.346-0.486) | 45.65(18.27-114.1) | |
| Clay | | 291 | 0.475(0.421-0.523) | 0.385(0.269-0.501) | 31.63(6.39-156.50) | 0.03 |
| | A | - | - | - | - | |
| | B | 70 | 0.470(0.426-0.514) | 0.412(0.309-0.515) | 27.72(6.21-123.7) | |
| | C | 23 | 0.483(0.441-0.525) | 0.419(0.294-0.544) | 54.65(10.59-282.0) | |

Figure 3-19: Overland runoff routing scheme



rainfall. When this equation applies to the raster based GIS, the spatial differentiation corresponds to the difference between neighboring cells. Each cell has two neighbors in $x$ direction and two in $y$ direction (except boundary cells), so the differential equation above can be expressed as difference equation in terms of time step and cell difference:

$$h^{t+\Delta t}(j,k) = h^t(j,k) + i_e\Delta t - Q^t(j,k)\Delta t \qquad (3.12)$$

$$Q^t(j,k) = [\frac{q_x^t(j \to j+1) - q_x^t(j-1 \to j)}{W} + \frac{q_y^t(k \to k+1) - q_y^t(k-1 \to k)}{W}] \qquad (3.13)$$

$q_x^t(j \to j+1)$ is the flow in $x$ direction from the cell $(j,k)$ to $(j+1,k)$, similarly to $q_x^t(j-1 \to j)$, $q_y^t(k \to k+1)$, and $q_y^t(k-1 \to k)$; $h^{t+\Delta t}(j,k)$ is the water level of cell $(j,k)$ at time $t+\Delta t$; W is the size of a cell.

The overland flow is determined by diffusive wave momentum equation and Man-

Table 3.2: Manning roughness coefficient

| | |
|---|---|
| *Natural stream channel* | |
| Gravel bottom with cobles and boulders | 0.040-0.050 |
| Cobble bottom with boulders | 0.050-0.070 |
| Unlined open channel,soil bottom | 0.020-0.025 |
| Grassy swale, 12-inch grass | |
| *Overland flow surface* | |
| Compact clay | 0.030 |
| Dense sod or shrubs | 0.035-0.040 |

ning equation. The diffusive equation is :

$$S_{fx} = S_{ox} - \frac{\partial h}{\partial x} \tag{3.14}$$

$S_{fx}$ is the friction slope in the $x$ direction, $S_{ox}$ is the land surface slope in the x direction. Expressed in discrete time difference forms, they are

$$S^t_{fx}(j-1 \to j) = S_{ox}(j-1 \to j) - \frac{h^t(j,k) - h^t(j-1,k)}{W} \tag{3.15}$$

$$S_{ox}(j-1 \to j) = \frac{E(j-1,k) - E(j,k)}{W} \tag{3.16}$$

Where $E$ represents the ground surface elevation of each cell.

Manning equation formulates the relation between surface runoff, slope, and elevation:

$$q_x = \alpha_x h^\beta \tag{3.17}$$

In SI units,

$$\alpha_x = \frac{S^{1/2}_{fx}}{n}; \beta = \frac{5}{3} \tag{3.18}$$

$n$ is Manning roughness coefficient, which is highly dependent on land use: the more the impermeable surface, the less the roughness coefficient, so the more amount of surface runoff. Table 3.2 lists $n$ for several kinds of land uses [Mar91].

In discrete form, the equation becomes

$$q_x^t(j-1 \to j) = \frac{1}{n(j-1,k)}[h^t(j-1,k)]^{5/3}[S_{fx}^t(j-1 \to j)]^{1/2}\text{if } S_{fx}^t(j-1 \to j) \geq 0 \quad (3.18)$$

$$q_x^t(j-1 \to j) = \frac{-1}{n(j,k)}[h^t(j,k)]^{5/3}[-S_{fx}^t(j-1 \to j)]^{1/2}\text{if } S_{fx}^t(j-1 \to j) < 0 \quad (3.19)$$

It is similar to the flow of $y$ direction:

$$q_y^t(k-1 \to k) = \frac{1}{n(j,k-1)}[h^t(j,k-1)]^{5/3}[S_{fy}^t(k-1 \to k)]^{1/2}\text{if } S_{fy}^t(k-1 \to k) \geq 0 \quad (3.20)$$

$$q_y^t(k-1 \to k) = \frac{-1}{n(j,k)}[h^t(j,k)]^{5/3}[-S_{fy}^t(k-1 \to k)]^{1/2}\text{if } S_{fy}^t(k-1 \to k) < 0 \quad (3.21)$$

At each time step, calculate $q_x$ and $q_y$ from equations (3.18) to (3.21) and $f_t$ from equation (3.10), then substitute them into equation (3.12) to obtain $h(t+\Delta t)$.
For stream cells, the continuity equation becomes a little bit different:

$$\frac{\partial A_x}{\partial t} + \frac{\partial Q}{\partial x} = q_1 \quad (3.22)$$

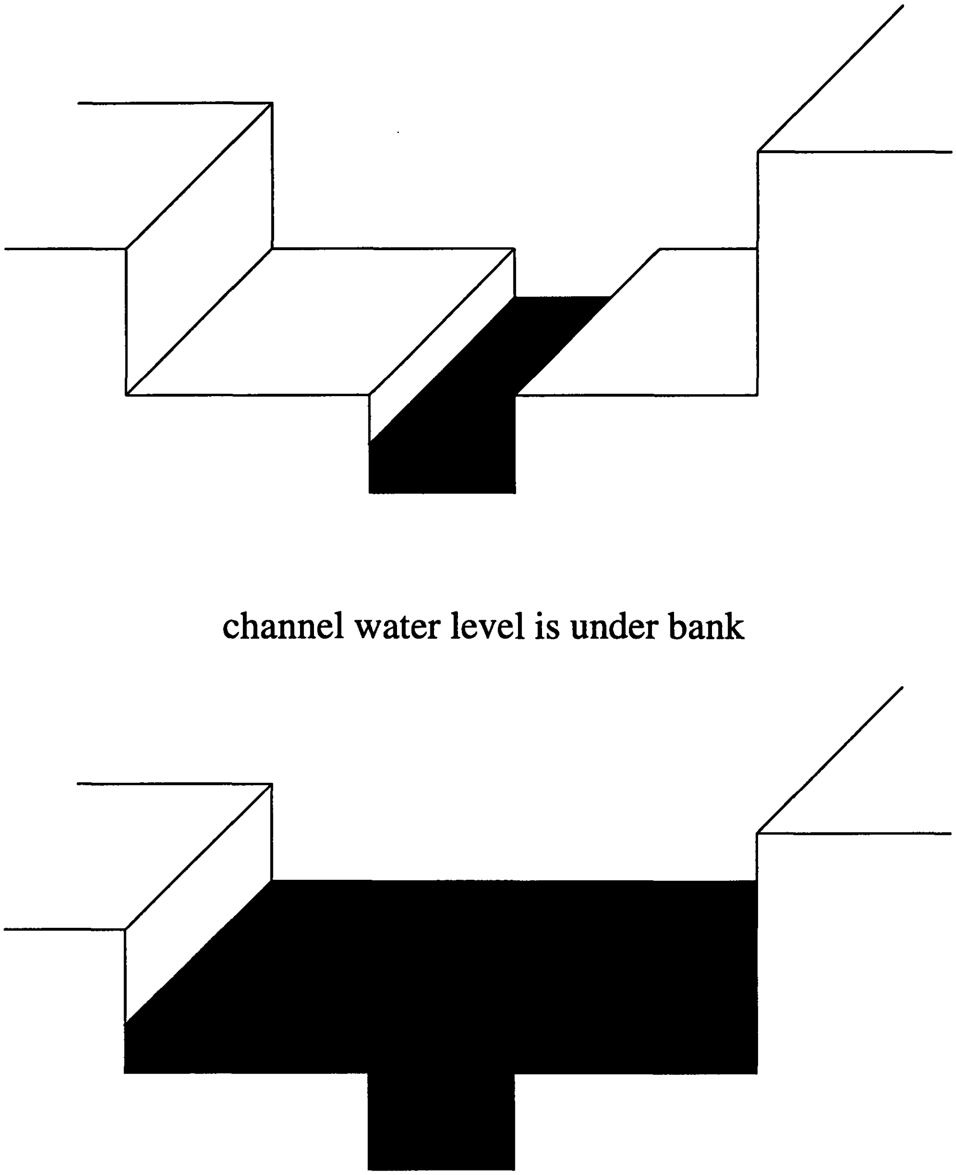
Where $A_x$ is channel flow cross section (the unit is area), $Q$ is total discharge in the channel, $q_1$ is lateral inflow rate per unit length. The difference between the continuity equation of channels and that of lands is that the stream flow is one dimensional rather than two. Here we assume the widths of streams are equal, so the cross section measures the water level in the channel. The discharge in the channel can also be expressed by Manning equation:

$$Q = \frac{1}{n} A_x R^{2/3} S_f^{1/2} \tag{3.23}$$

where $R$ is channel hydraulic radius, $S_f$ is the channel friction slope.
From previous discussion (section 3.2.2), it is known that the stream data will not cover the entire cell, so is it in 3D environment. Each stream cell actually contains stream part and land part. This makes us treat stream cells and land cells differently. The width of streams are set equal (1/10 of the cell width, 3 meters), the remaining part outside the stream within a stream cell is treated as land. The depth of stream bank is also set to be a constant (2 meters). When stream water in the channel is under the bank, the overland flow and the stream flow are treated separately, the lateral flux can only flow into the channels. When stream water is over bank, the stream water level less the bank depth equals to the land water level of the stream cell; here the stream cell is treated identically as the land cell.

Another concern about the stream flow is that we assume the stream water level is at equilibrium before the rainfall. In other words, different altitudes of adjacent cells are not considered to be factors of causing surface water runoff, only the additive flows due to the rainfall and the lateral flow into the channel is concerned.
For ponds, the water level inside a pond is identical. Additive rainfall and lateral inflow will increase the water level of all the cells within the pond; because these inflows will be distributed evenly over the entire patch, the rate of water level increase is relative low.
The advantage of using this model is that important parameters can be obtained from GIS database or directly set by the user. Slope is derived from elevation, infiltration rate can be determined by soil type, Manning roughness coefficient is determined by land use, rainfall intensity and duration is set by user. So the simulation result sufficiently utilizes the information provided by GIS without making too many assumptions (although there are still some, like constant bank width and depth).

Figure 3-20: Water body cell runoff routing scheme



channel water level is under bank

channel water level is over bank

### 3.5.3 Pollution model

The pollution model is based on AGNPS, a non-point source pollution model [YO+89], and soil erosion equations [NO94]. The universal soil loss equation (USLE) is used to estimate the soil erosion amount:

$$S_L = E_I KLSCP \tag{3.24}$$

$S_L$ is soil loss, the unit is tons/hectare, $E_I$ is energy factor, which is the sum of rainfall and surface runoff factors:

$$E_I = 0.5R_r + 7.5Qq^{1/3} \tag{3.25}$$

The rainfall contribution $R_r$ depends upon the rainfall profile (hyetograph); it is the sum of the products of logarithmic values of intensity and duration of each interval, multiply the maximum intensity of the storm.

$$R_r = \sum_i [(2.29 + 1.15 \ln X_i) D_i] I \tag{3.26}$$

$i$ = rainfall hyetograph time interval,
$D_i$ = rainfall during time interval (cm),
$I$ = maximum 30-min rainfall intensity of the storm (cm/hr),
$X_i$ = rainfall intensity (cm/hr).

Here the rainfall profile is set to be uniform over the entire duration of rainfall, so only one interval is calculated.
The contribution from surface runoff is determined by both runoff volume and rate. In equation (3.25), $Q$ is runoff volume (cm), $q$ is maximum runoff rate (cm/hr). Other terms in USLE are dependent on topography, soil type, and land use. $K$ is soil erodibility factor, which mainly depends upon soil type. (Table 3.3).

$LS$ is slope-length factor, which depends upon the topography:

Table 3.3: Soil erodibility factor K

| Technical class | $K$ for organic matter content(%) | | |
|---|---|---|---|
| | 0.5 | 2 | 4 |
| Sand | 0.05 | 0.03 | 0.02 |
| Fine sand | 0.16 | 0.14 | 0.10 |
| Very fine sand | 0.42 | 0.36 | 0.28 |
| Loamy sand | 0.12 | 0.10 | 0.16 |
| Loamy fine sand | 0.24 | 0.20 | 0.16 |
| Loamy very fine sand | 0.44 | 0.38 | 0.30 |
| Sandy loam | 0.27 | 0.24 | 0.19 |
| Fine sandy loam | 0.35 | 0.30 | 0.24 |
| Very fine sandy loam | 0.47 | 0.41 | 0.35 |
| Loam | 0.38 | 0.34 | 0.29 |
| Silt loam | 0.48 | 0.42 | 0.33 |
| Silt | 0.60 | 0.52 | 0.42 |
| Sandy clay loam | 0.27 | 0.25 | 0.21 |
| Clay loam | 0.28 | 0.25 | 0.21 |
| Silty clay loam | 0.37 | 0.32 | 0.26 |
| Sandy clay | 0.14 | 0.13 | 0.12 |
| Silty clay | 0.25 | 0.23 | 0.19 |
| Clay | | 0.13-0.2 | |

$$LS = L^{1/2}(0.0138 + 0.00974S + 0.00138S^2) \tag{3.27}$$

$L$ = length in meters from the point of origin of the overland flow to the point where the slope decreases to the extent that deposition begins or to the point at which runoff enters a defined channel. In this application it is set to be the cell length. $S$ = the average slope in percentage over the runoff length.

$C$ is vegetation factor, which depends upon the type of vegetation cover of land. (Table 3.4).

In this application, however, the type of vegetation is not indicated in GIS database, so it is divided by land use: cells with trees have lowest $C$, cells with houses have highest $C$.

$P$ is erosion control factor which also depends upon the land use. (Table 3.5, 3.6).

All these parameters are raster-based, they vary cell by cell. Energy factors are calculated from the rainfall amount and surface runoff flow of each cell, the runoff flow is obtained in the process of calculating the flood water level; other parameters are determined from GIS data and remain unchanged during the simulation.

We want to calculate the pollutant amount of each cell carried by soil loss from neighbor cells, in other words, the pollution caused by soil deposition to this cell. In order

Table 3.4: Cropping management factor C

| **Land cover or land use** | *C* |
|---|---|
| Continuous fallow tiled up and down slope | 1.0 |
| Shortly after seeding prior to harvesting | 0.3-0.8 |
| *For crops during main part of growing season* | |
| Corn | 0.1-0.3 |
| Wheat | 0.05-0.15 |
| Cotton | 0.4 |
| Soybean | 0.2-0.3 |
| Meadow | 0.01-0.02 |
| *For permanent pasture, idle land, unmanaged woodland* | |
| *Ground cover 85%-100%* | |
| As grass | 0.003 |
| As weeds | 0.01 |
| Ground cover 80% | |
| As grass | 0.01 |
| As weeds | 0.04 |
| Ground cover 60% | |
| As grass | 0.04 |
| As weeds | 0.09 |
| *For managed woodland* | |
| Tree canopy of 75%-100% | 0.001 |
| Tree canopy of 40%-75% | 0.002-0.004 |
| Tree canopy of 20%-40% | 0.003-0.01 |

Table 3.5: Values of P for agricultural lands

| | | **Strip cropping and terracing** | |
|---|---|---|---|
| **Slope(%)** | **Contouring** | **Alternate meadows** | **Closegrown** |
| 0-2.0 | 0.6 | 0.3 | 0.45 |
| 2.1-7.0 | 0.5 | 0.25 | 0.40 |
| 7.1-12.0 | 0.6 | 0.30 | 0.45 |
| 12.1-18.0 | 0.8 | 0.40 | 0.60 |
| 18.1-24.0 | 0.9 | 0.45 | 0.70 |
| > 24 | | -1.0- | |

Table 3.6: Values of P for construction sites

| **Erosion control practice** | *P* |
|---|---|
| ***Surface condition with no cover*** | |
| Compact,smooth,scraped with bulldozer or scraper up and down hill | 1.30 |
| Same as above,except raked with bulldozer and root raked up and down hill | 1.20 |
| Compact,smooth,scraped with bulldozer or scraper across the slope | 1.20 |
| Same as above,except raked with bulldozer and root raked across the slope | 0.90 |
| Loose as disked plow layer | 1.00 |
| Rough irregular surface,equipment tracks in all directions | 0.90 |
| Loose with rough surface > 0.3m depth | 0.80 |
| Loose with smooth surface < 0.3m depth | 0.90 |
| ***Structures*** | |
| ***Small sediment basins*** | |
| 0.09 ha basin/ha | 0.50 |
| 0.13 ha basin/ha | 0.30 |
| ***Downstream sediment basin*** | |
| With chemical flocculents | 0.10 |
| Without chemical flocculents | 0.20 |
| ***Erosion-control structures*** | |
| Normal rate usage | 0.50 |
| High rate usage | 0.40 |
| Strip building | 0.75 |

to do that, we have to determine the initial condition of pollutant distribution. Each cell has a certain amount of pollutant concentration; here we didn't denote specifically what kind of pollutant these cells possess, it can be nitrogen or phosphorous compound, bacteria, heavy metal and so on. We only assume it is generated from human activities, so the places where human activities are frequent such as house cells or road cells possess higher concentration of pollutant. The pollutant amount of soil deposition of a cell is the sum of the pollutant amount carried by soil erosion from all its neighbor cells. The pollution carried by soil erosion is the pollutant concentration times the soil loss:

$$P = SL * C \tag{3.28}$$

$P$ is pollutant content carried by soil erosion (g/cm cube), $SL$ is soil loss with the unit tons/ha = 0.001 $g/cm^3$, $C$ is the concentration of pollutant respect to soil mass, it can be g/g or ppm.

In GISVE the soil erosion model is linked with with the flood model. As mentioned, surface water runoff flow is calculated using momentum equation and Manning equation in the flood model. The flow is used by both the flood model and the soil pollution model: in the flood model, runoff flow is used for calculating the flood level, while in the soil pollution model, runoff flow is used for calculating the soil erosion. The flood level and the pollutant amount are both quantitative values, they are visualized by using different colors to represent different levels. In the flood model, since usually we only care about whether a specific area is flooded, we only use one level representation: if a cell's flood level exceeds a threshold value, it displays blue color on 2D and 3D windows, which means this area is flooded. In pollution model, however, we need to know the extent of pollution, the level of pollution is an important information; we use 4 different colors to represent different pollution levels: 10-50, 50-80, 80-110, and greater than 110, the unit is kilograms per cell.

### 3.5.4 Fire diffusion model

The fire diffusion model is based on cellular automata algorithm [TG94]. A cellular automation is a discrete dynamic system whose behavior is specified in terms of a local relation [TM88]. The algorithm is cell-based, the relation between the values of neighboring cells represents their interactions. Usually this relation is simple and local, which only adjacent cells interact with each other. When running the model, both spatial and temporal variations are displayed on the 2D map. The interesting part of cellular automata is that the simple and local relation between cells sometimes generates complex results which are able to describe global behaviors of a system.

The initial condition of fire is set by the user. The user chooses one or multiple fire spots on 2D map, fire will spread from these source spots. The probability of one cell catching on fire depends upon several factors: the number neighbor cells on fire, land use of that cell (forest, house, or crop land), the water body distribution (if the cell is stream cell or pond cell), and time of burning. The quantitative rules are as follows.

1. Count the number of neighbor cells on fire $n_1$, and the number of neighbor cells which are water bodies (streams, ponds) $n_2$.

2. Set the threshold value for the probability of catching on fire for each cell.
   If the cell is water body, the threshold value is set 0, in other words, fire will not spread in water bodies.
   If the cell has houses, the threshold value is $0.5n_1 - 0.1n_2$.
   If the cell has trees, the threshold value is $0.3n_1 - 0.15n_2$.
   Otherwise the threshold is $0.2n_1 - 0.15n_2$.

3. Randomly choose a number (between 0 and 1) for each cell, if this number is less than the threshold value, that cell will catch on fire at the next time step.

4. If the cell is on fire, trees and houses are burned down, this will removes the tree cells or house cells from the 2D and 3D windows.

5. For each cell on fire, count the time from the beginning of catching fire. If the time is over some duration, fire at that cell extinguishes.

## 3.6 Graphical user interface modules

### 3.6.1 Basic concepts

The Graphical User Interface (GUI) is a bridge between human and machine. It fetches input from the user, and displays the information which computer generates to the user in an understandable form . In GISVE, GUI becomes the only part of human-machine interface.

There are two windows which function as GUI: 2D window and 3D window. As discussed, they have different functions in manipulating the system. 2D window shows the map, provides the global view of the entire place, indicates current position and moving direction of the user, provides functions allowing the user to manipulate and change GIS data or set the initial conditions for model simulation. 3D window provides a graphical environment that allows the user to render through with a sense of reality, supplies different modes of navigation, provides different perspectives of the same object, and allows the user to see the scene at finer scales.

The most important input device is the mouse. The user clicks buttons on the 2D window to load or modify data layer, he/she also moves the mouse on the 3D window to navigate through this environment. The interaction between mouse movement and 3D coordinate transformation has been discussed in section 3.4. In the following section, the GUI on the 2D window is the main topic.

### 3.6.2 2D user interface

The 2D window of GISVE is written by using Motif, an X window toolkit which supplies function calls building GUI [JR93].

A widget is an abstract data structure used by Motif. It can be used to create various elements of the user interface, such as scroll bars, function buttons, labels, draw areas, pop-up windows, and so on. Like the encapsulation of the class in object oriented programming languages, a widget contains both data and function calls. When a specific event triggers this widget (for example, the mouse is clicked on the function button, or the size of the draw area is changed), specific function calls will be executed. This is how the functions GISVE offer are selected on the 2D window.

The outlook of 2D window is in figure 4-3. Menu bars are on the top row. It has six items. Load menu contains load/unload functions of various data layers, including elevation, roads, streams, ponds, houses, trees, flood, pollution, fire, and exit function. When click the button once the data is loaded into the 2D and 3D windows, click the same button again it is unloaded from both windows. Choose menu contains functions of choosing GIS data, including roads, streams, ponds, houses, trees, and fire; the user chooses the data he/she want to change, then applies the change function (add or delete) to this data layer. Different data types have different choosing methods: for the raster data (house, fire), the user clicks the mouse on the cell, the raster data on that cell is chosen; for the linear data, the user chooses the start and end points of the linear data he/she want to change (when the data contains several segments, the start and end points at each segment should be chosen); for the patch data, the user chooses the vertices of the polygon he/she want to change, then the cells inside the polygon are all selected. Navigate menu contains navigate functions which are discussed at section 3.4: fly, drive, automatic drive, fix position, and perspectives. Change menu contains two functions: add and delete. Add function adds the data chosen by user to the designated data layer, delete function deletes that data. The add and delete functions of raster data are straightforward: they add or remove individual cells chosen by the user from a specific data layer. The add and delete functions of patch data are manipulated on polygons: the user chooses vertices of a polygon (by using the choosing function), then a specific layer of data of all the cells within this polygon is added or removed. Both add and delete functions of linear data are manipulated on segments with a little difference between them. In the add function, the user chooses end and intermediate points of a path he/she wants to add in a specific data layer. All the cells on the selected piecewise-linear curve are added to this layer. In the delete function, the user chooses points at current existing linear structures such as streams or roads. Find out all the cells on the branches connecting consecutive points and remove these cells from the linear data. If there is no branch connecting two consecutive points, there is nothing to delete. Set menu contains set point function and set path function. Set point function is used for setting the coordinate of the point used by the fix position function, set path function is used for setting the coordinates of the path for autodrive function. The coordinates can be set by mouse or by keyboard. Model menu contains various model simulation functions discussed in section 3.5: flood model, fire model, and pollution model. Each model function contains the items of start, end, reset, and choosing initial condition.

Besides the menu bars, the 2D window includes a legend showing what the different colors represent, and a 2D map showing GIS data layers.

## 3.7 Application main loop

So far all modules performing necessary functions have been introduced, what we need now is a module that coordinates all of these modules to achieve the integral performance of the GISVE system. It is done by the application main loop module. The application main loop contains two parts: initialization submodule and infinite loop submodule. Initialization submodule, as its name suggests, initializes the system: loading GIS database, building 2D and 3D windows, initializing Performer and Motif libraries; these operations are executed only once when the system starts. Infinite loop submodule contains procedures which are executed at each time a new frame of 3D screen appears; it polls the status of both 2D and 3D windows and then decides which function is to be executed and the moving direction at 2D and 3D map for the next frame. The loop doesn't end until the "Exit" button on 2D window is clicked.

### 3.7.1 Initialization submodule

There are several works the initialization submodule should do. The first work is to load the GIS data files into the system. As discussed in section 3.2.1, the system accepts ASCII format and stores the data of each layer in a two-dimensional array. GIS data files are loaded into the system only once, the entities of the array stored in the system may change later (by the change either from user input or from the simulation results), and the array may be loaded or unloaded into the 2D and 3D windows. Each time when the 2D array is modified, it is converted into an appropriate data type (raster, linear, patch), no matter if this data is loaded on the 2D and 3D windows or not.

The second work is to do the data conversion from the 2D array into appropriate 2D and 3D data types. As discussed in section 3.2.1, the data structure of 3D scene is a tree, the entire area is divided into 25 blocks, each block is represented by a node, and it contains several child nodes including elevation, roads, and so forth. We also mentioned that each block is processed separately for the efficiency of culling. The initialization module has to convert the entire data array into the 2D data and 3D objects of all these 25 blocks. Later when some blocks are modified, we can change the data only within these blocks.

The third work is to build the 2D window, this is done by initializing the Motif functions. Each widget is built and the functions associated with widgets are specified, for example, the flood model module is linked with the flood model button at the GUI.

The forth work is to build and initialize the 3D window, this is done by calling various initialization functions of Performer. Several parameters, including frame rate, viewing frustum, level of detail distances, are set at this submodule.

### 3.7.2 Infinite loop submodule

After the initialization is done, the system enters the infinite loop until the "Exit" function is called. Each loop begins when a new frame comes at the 3D window, here the frame rate is set at 30 Hz, so the loop is updated every 33 ms. There are several works to be done during the interval between frames: the update of coordinate, the polling of GUI, and the refresh of model simulation.

The update of coordinate is dependent upon the current navigate mode. When the mode is fly or drive, the coordinate is updated by polling the moving direction of the mouse, new coordinate transformation matrix is calculated and the new position at 2D and 3D windows are displayed. When the mode is automatic drive or perspectives, the coordinate is updated by calculating the transformation matrix of an already existing path or moving direction. The fix-position mode has a fixed coordinate and viewing angle.

The polling of GUI can be fulfilled when specify links between widgets and function calls at initialization stage. At each inter-frame interval, if an event comes (a function button is clicked, the window is covered by other windows or recovered, and so on), the function call associated with the widget being triggered is executed.

When the model function is called in real time dynamic mode, the simulation is executed one time step at each inter-frame interval, rather than finishing it without interrupt, so at each loop the model function is refreshed once. It takes too long to change the scene modified by simulation each time step; the better way is to update the scene every fixed period of time. In this application the scene update period is chosen to be 100 inter-frame intervals, which is about 3.3 seconds.

# Chapter 4

# Verification, validation, and evaluation

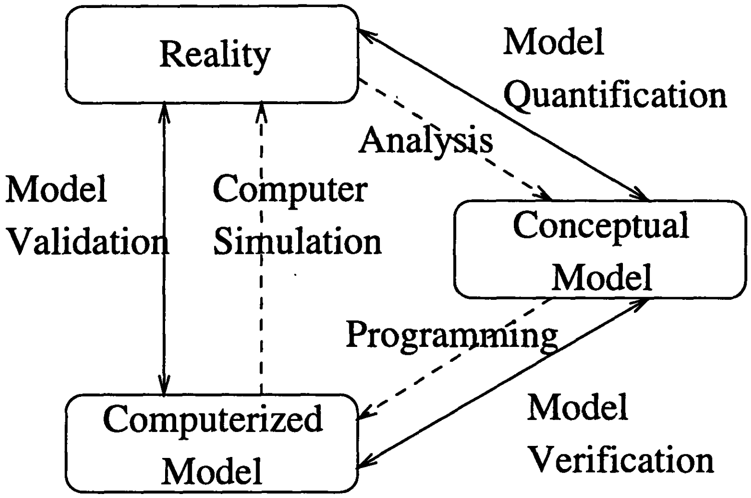
## 4.1 Basic concepts

For software development, half of the work is concentrated on generating ideas, goals, and specifications, then implement the software according to the specifications; the other half work, however, is concentrated on verifying, evaluating, and validating the product. So far the first half of work has been discussed. In this chapter the methodology and procedure of validation-modification are the main topic.

It is necessary to clarify the definitions about evaluation, verification, and validation. For a simulation software like GISVE, there are three models closely linked to achieve the overall functions [KA93, pages 1–5]. Reality is the real environment or system that the software is built to simulate. Conceptual model is the theory, concept, or mental image based on human's knowledge and cognition toward the reality. Computerized model is the content and effect computer generates from implementing the conceptual model; it contains both the simulation result by running calculations, and sensory I/O's representation of the virtual environment.

From this scheme, verification is defined as the substantiation that the computer program implementation of a conceptual model is correct and performs as intended. Validation, however, is the substantiation that a computer model, within its domain

Figure 4-1: Simulation model



of applicability, possesses a satisfactory range of accuracy consistent with the intended application of the model [KA93, pages 1–3]. The term evaluation used in this thesis is close to the concept of validation, both denoting processes to acquire the knowledge about whether the computer system achieves the functions in the domain of application. The difference is that here we use the word "evaluation" to denote the process that domain experts are invited to see the demonstration and use the system, then express their opinions about the performance of the system. And we use the word "validation" in a more general sense, including the explanation of the evaluation result and the justification of system specifications.

The processes of obtaining the knowledge about the system performance of GISVE contain the processes of verification, evaluation, modification, and validation. The first step is to generate the conceptual models and specifications based on needs of this application domain, this part is discussed in chapter 2. The second step is, after the system is implemented, to verify that the system satisfies the required specifications and features. The third step is to invite the domain experts to participate the evaluation process, from their comments during the demonstration session and the responses to the questionnaires filled out later, to obtain the knowledge that if the system qualifies as a useful tool in the domain of application. The fourth step is to modify and improve the system according to the previous evaluation result. Then repeat the evaluation and modification stages. In the software development of a commercial product, these two stages should be repeated many times until most of objectives are attained and drawbacks are improved. However, since GISVE is a prototype system, only two evaluation-modification stages are performed. The final step is to run a postulated landscape design project by using the GISVE system, this project, as we name it test project, is used to test and validate the performance of the system.

## 4.2 System verification

This section discusses how the prototype system conforms the specifications and features which are set in chapter 2. The limitations and the parts which are not achievable currently are also discussed in terms of computer resource consideration.

### 4.2.1 2D display and data manipulation

As expected, 2D display and data manipulation functions are achieved in GISVE system. They work as well as other 2D GIS tools, except the number of data layers is fixed and the choices of functions are limited.

**Data layers in the GISVE system**

There are ten layers of GIS data in the GISVE system: elevation, roads, streams, ponds, houses, trees, flood, pollution, fire, and soil type (not displayed). It can have more layers of data added in the system, and the idealist case is that the system can load arbitrary number of data layers. This is infeasible because the 3D representations depend on the specific properties of the 2D layers. We have to determine the relation between the 2D and 3D representations before the system is implemented, this restricts us not to be able to load arbitrary data layers before we are familiar with the properties of these data types. Considering the useful purposes to landscape design and complexities of putting more data into the system, it is found that these ten layers of data are essential for giving a rough assessment and quick impression of the environment of a place, especially for the preliminary stage of landscape design purposes.

### 2D display

The display is raster based. Since the 2D window size is not large and the cell number is high compared to its size (125 by 125 cells), the resolution of 2D window is high enough that discontinuity cannot be perceived. Coded colors are used to represent different data layers: roads are represented by gray, water bodies by blue, houses by fire brick red, etc. We try to match these colors with the colors traditionally used in representing these data layers. It is also necessary to make colors in 2D and 3D windows consistent to avoid confusion of the users. Elevation and soil type are not displayed on 2D window. The main reason is that these two types of data are either continuous (elevation) or have many classes (soil type), this increases the difficulty of representing different levels of the data by using different colors. For elevation, since the terrain is displayed on the 3D window, its 2D display can be eliminated; for soil type, we still lack an appropriate way that can both represent the variety of different classes and avoid confusion.

A major function of 2D map is for aiding the users to identify their position and direction to avoid disorientation in the 3D environment. This function is achieved by creating a position icon on the 2D map which indicates current position and viewing direction. When the user moves in 3D environment, the position icon moves corresponding to the same position and direction on the 2D map; it facilitates users to render through the environment.

### Data loading functions

The load/unload option of each data layer provides the users sufficient freedom in putting information they need into the environment or removing unnecessary information from it. There are two choices on the order of overlaying different layers on the same map. One choice is that all layers follow a fixed order, for example, road is always on top of water; another choice is that the lattest loaded layer sits on the top, like a stack. Both juxtapositions have pros and cons, but the former provides a fixed and more stable relationship between different layers. The GISVE system adopts the first one.

### Data modification functions

Functions allowing users to change the elements of the environment constitute one of the interactive features of the system. As mentioned, because of the difficulty for either users or the computer to manipulate 3D objects directly, all change functions should be done on the 2D window. However, to manipulate objects on 2D map is not straightforward. For different data types, it is necessary to have different types of manipulations. If the data is raster type, like houses, pollution sources or fire spots, it is appropriate to allow the user to add or delete data cell by cell. If the data is linear type, like streams or roads, a more appropriate way to let the user to add data is by choosing the end points of segments, then fill out all points along these segments automatically; and let the user to delete data by choosing two points, then delete all points along the line between these two points. If the data is patch type, like forests or ponds, it's more convenient to let the user to choose the vertices of a polygon, add or delete all the points inside the polygon. In GISVE, conforming to these three types, the change function is a two-step procedure. The first step is to choose the points and the data layer to be changed, the second step is to change the data by either adding or deleting the selected points. This corresponds to two function buttons on the 2D window. The user does the choose and change separately.

### 4.2.2 3D display

We have encountered more problems in developing the 3D display, and some of its performances still do not attain the level within satisfactory extent. Visual quality and interactivity are important concerns in discussing 3D display performances. In this section we discuss visual quality.
Visual quality is highly emphasized in building a CAD tool for landscape design, this point is phrased in previous sections discussing both specific goals and task analysis. The 2D window doesn't have many visual quality problems since the display is raster based and resolution is relatively high. In 3D environment, however, visual quality is an important issue. When transforming 2D raster data into 3D polygons, we have to notice:

1. How to preserve the correctness and smoothness of data when the scale and dimension change.

2. How to balance the tradeoff between visual quality and computer resources.

#### Correctness and smoothness preservation

Geographic data is obviously not scale invariant, the sizes of objects vary when looking at them with different scales. This arouses a problem when incorporating 2D and 3D windows representing the same objects with different scales. The resolution on the 2D window is the size of a cell, in this case, it is 30 meters by 30 meters. Within a cell the distribution of GIS objects are undetermined. The 3D window simulates a person's viewpoint when he/she flies or walks through the environment, so the smallest scale of objects on 3D window should be about the same size as a person. If we want to see finer structures within the cell, such as the bank widths of streams, the trunks and leaves of trees, or windows of houses, we cannot obtain the information about these structures from the 2D data because they are below the scale of finest 2D resolution. Hence there are many new variables added into the 3D environment which cannot be determined by GIS database.

The first problem is about the correctness of transformed data, for example, a cell marked "house" on 2D map probably mean there is only one house within the cell (although still not quite clear yet), but a cell marked "tree" cannot denote there is only one tree within the cell, it means "a bunch of trees", but how many? The accurate number of trees within a cell is not important in this application, so we can just put a random number of trees in each cell, but another parameter distribution within a cell is critical for all other data layers, that is elevation. The elevation data at each cell of 2D GIS map denotes the average values of elevation within this cell, however, these values are discrete. There are several ways to make them continuous, one way is to put cubes on each cell, the height of each cube is the corresponding elevation, but the overall topography looks like a staircase, which is very unrealistic, especially when the scale is large and the viewer is close to surface. Another way is to use triangular networks to represent the terrain. The elevation values represent the heights at the vertices of these triangles, rather than the average values within the cell. This gives us a more smooth and continuous configuration of terrain; when the texture is mapped on these polygons, it generates a much more realistic sense about the topography than the staircase model. However, the correctness problem is still unsolved. The point is no matter which method we use, the correctness below the range of the the finest resolution cannot be promised, we can only assure the data in 30m*30m scale is correct.

The second problem is about the smoothness of data. The cell size is 30m*30m, so the linear and patch data which look smooth on the 2D window will display zig-zag shapes at 3D window. The method to smooth curves is to spline the curve on 2D plane at first, then transform the splined the curve to fit the topography. Since all linear data are strips rather than lines, they all have width. So we have to transform the splined strip (rather than line) onto topographical surface. For patch data, the boundary lines are drawn on the topographical surface also, which divide elevation triangles where the edges of the patch are located into two parts: the part within the patch and the part outside the patch. The result shows the 3D display after the smoothing is much better than before the processing.

#### Tradeoff between visual quality and computer resource

The tradeoff between visual quality and computer resource, especially the run time, is very important in most virtual environment systems. The more polygons being put in one scene, the more picturesque and real quality it achieves, but it also consumes more memory and time. In GISVE the tree layer consumes the greatest number of polygons. It is mentioned that realistic tree profiles cannot be built simply from polygons, so various skills are developed or adopted: the first one is by using billboard nodes to reduce the number of polygons depicting individual trees, the second one is by using level of detail nodes to reduce unnecessary polygons appear in the viewing volume, the third one is by dividing the whole area into several sub-areas and put all elements within the same subarea together in scene graph tree to improve the culling efficiency, the fourth one is by creating a movable tree patch around the viewer to avoid texture mapping of all other cells. All these improvements generate a result which balances the visual quality and system response time. Although we can build more complicated models to make trees look more realistic or further simplify the model to enhance the response speed, the other end will become unsatisfiable.
The same thing happens to the choice between model complexity and response time. We decided to choose simplified environmental models because they reduced simulation time greatly. This system is not built to do complicated and complete simulations of environmental impact, hence we can reduce the model within a certain extent that generates interesting results and doesn't retard other system functions.

### 4.2.3 Navigate functions

Navigate functions of GISVE system have achieved expected functions that provide the user more freedom in navigating through the 3D environment.

### Fly function

Fly function provides the freedom in the 3D environment, but compared to other navigate functions, it is easier to confuse users. One problem is about the speed of response when manipulating the mouse. When the system is heavily loaded, there is a certain amount of time delay between moving the mouse and updating the 2D and 3D windows. The time response problem will be discussed later. The other problem is about the difficulty of manipulation. At short distances, it is not easy to move the viewer to the position and direction he/she wants to go, for example, to change viewing directions without changing current position, or to move along the road without adjusting direction and speed by the user. These drawbacks are compensated by providing other navigate functions such as automatic drive or turn around perspectives. In general, the fly function is used for providing a bird eye's view to the environment, this goal is successfully achieved in GISVE system.

### Drive function

Drive function has the same operation method as fly function does except it only allows the user to move along the terrain surface. The drive function implemented in GISVE system also satisfies its original specifications, thus verifies its functionality.

### Automatic drive function

This function provides the user more conveniency with less freedom in navigating through the environment. It satisfies specifications set for navigate functions by allowing the user to move through the environment. It works as a complement of fly and drive functions because the easier manipulations of this function reduce the confusion and disorientation when moving in the environment So far in GISVE system the combination of two types of navigate functions: the user has more freedom but less conveniency, and the user has more conveniency but less freedom, maintains user-machine interaction within an acceptable range.

### Fix point function

Although the fix point function is simple and the freedom it provides is low, it turns out to be one of the most useful navigate functions in the experience of using the GISVE. It is mainly used when the user "gets lost" in the 3D environment, he/she can choose a point on 2D map as a reference point and go back to that point. It also satisfies its specifications.

## 4.2.4 Model functions

Several concerns should be noticed in verifying model functions of GISVE. We have to know if the simulation result is correct, or at least makes sense according to our knowledge about the environmental impact it simulates. We have to know if this model is important to the application domain of GISVE system. We have to know if the two modes of simulation- fix time duration and dynamic simulation, have achieved their objectives. The models implemented in the system will be picked up to verify that they meet these three criteria.

### Flood runoff model

The main reason why we cannot verify if the simulation result of flood runoff model is correct or not is because there are still unknown parameters that we have to set arbitrarily, such as the widths and depths of river banks, and the rainfall profile (hyetograph) of the target place. However, the simulation result shows the consistency with the common knowledge about the flood profile after heavy rain: flood invades areas along streams or in valleys at first, then expands to other areas (figure

51). So the correctness of the flood model is within the acceptable range of the application domain.

The importance of predicting flood area in landscape design has been indicated from the domain experts during the interview session. This belongs to the range of the validation work which was done when setting the system specifications.

### Pollution model

The problem of verifying the correctness of flood model also occurs at pollution model. Furthermore, there are more unknown parameters which should be determined at the development stage in pollution model. Simulation results show that the most serious erosion-carried pollution occurs along the road and near the downtown area. This makes sense when looking at qualitative properties of erosion-carried pollution profile; but if the quantitative measurement is needed, the model used is not accurate enough to provide reliable information.

The importance of identifying areas where more erosion-carried pollutant is accumulated is unquestionable in landscape design process. Furthermore, landscape designers tend to take the overall hydrological cycle into consideration; this includes storm water runoff, ground water, precipitation, organic pollution such as septic or agricultural pollution, non-organic pollution such as industrial pollution. To build a model including all these factors is still a challenge to environmental scientists and is beyond the scope of this thesis research, currently there are only two models included. Which models should be added and how accurate these models should depend on the specific application of this tool.

### Fire model

The accurateness of the cellular automata model of fire diffusion depends on the function describing the local relation of neighbor cells. This function can be adjusted freely and seems difficult to verify which function causes results closer to real cases.

This model only provides a window to visualize the basic trend for the diffusion of fire. Similar to previous two models, accurate quantitative information is still not achievable yet.

The importance of fire factor in landscape design depends on the target place. In areas near grassland or forest which have arid seasons, fire event is a very important factor in landscape and urban design; while if the target place is in the humid temperate forest, such as the target area of the test project described later, fire is not a very big concern. However, the fire model is picked up because of its easiness to simulate and visualize, it is used to displaying the feasibility to put more sophisticated models in the system in predicting more complicated phenomena.

#### Fix duration simulation v.s. dynamic simulation

It is mentioned that two modes of model simulation are used: one mode is to set the simulation duration and run the model until the end of the duration, another mode is to run the model at one time step for each inter frame interval. These two modes are used for different purposes. In running the dynamic simulation mode, it seems to take too long to obtain results of interest; for example, when the user wants to see the dynamic change of flood area for the first half hour from the beginning of rain, he/she has to wait for 30 minutes. If he/she uses fix duration mode, the final result will display after several minutes of simulation, but all the intermediate processes will not appear. It needs another way that can both run the simulation faster and allow the user see the dynamic processes. Animation mode is invented to provide this feature, it will be discussed in modification section.

### 4.2.5 Time response

Time response is crucial in user-machine interactivity. If the delay from the user inputs a command (either by mouse or by keyboard) to the system responds to this command is long, then it takes more effort and time for the user to manipulate the system, i.e., reduce the performance of the system. There are several "critical paths" which take longer time to execute when GISVE is running:

- The procedures of transforming original GIS files into 2D and 3D linear data need more time compared to other 2D data manipulation and 2D to 3D data conversion functions. As discussed in section 3.3.5, the transformation from 2D linear data to 3D objects needs more calculations than raster to 3D or patch to 3D functions. It contains the processes of tracing all linear data branches cell by cell, enlarge the scale, spline these branches into more smooth curves, extend the lines into the strips, and fit those strips on the terrain polygons. These processes become the bottleneck of the system manipulation when we load or change the linear data. So far it still lacks a good way to enhance the speed of these processes; one improvement which has been done is to localize the linear data: when we change it, only the part within some blocks will be modified, other parts remain unchanged, this avoids doing the transformation processes for the linear data of the entire target place.

- The time response of manipulating fly or drive functions is relatively slow when all data layers are loaded into the environment. The crucial point is the number of polygons in the 3D scene, especially for the tree layer. Various skills in reducing the tree polygon numbers have been described in previous sections, these skills greatly reduces the time rendering 3D objects. Either by adding polygons or by using texture mapping to represent 3D objects increases the delay: time response and visual quality form the tradeoff. This cannot be easily solved unless the memory space or CPU speed of the platform expands.

- Model simulation also takes a certain amount of time. This delay is reasonable since most simulation models do not have real time response property. If the user is interested in looking at the dynamics the system evolves during the simulation, he/she can either run the dynamic simulation mode, which updates the simulation results on displaying windows every several seconds, or run the fix duration mode then apply the animate function to see the dynamic change of previous simulation.

## 4.3 Evaluation and modification

Evaluation processes are necessary in building any engineering products. During the evaluation processes not only the knowledge about the performances of the current system is obtained, but also potential applications in the future are indicated. There are two evaluation sessions given by the same domain expert, the processes, results, and modifications corresponding to the evaluation results will be discussed as follows. Shirley Kressel, a landscape architect, was invited to participate the two evaluation sessions. Each session lasts for one or two hours. It includes demonstration and filling out a prepared questionnaire. (The questionnaires and their responses of the two evaluations are attached in appendices B and C). The evaluation results and opinions come from both the dialogue during the session and the written comment on the questionnaire.

### 4.3.1 First evaluation

The first evaluation session was held on October 13th, 1995, at the VETT (Virtual Environment Technologies for Training) lab of RLE (Research Laboratory of Electronics), MIT. The first part was a brief introduction about the purpose for application, system features, system functions, and user interface. After the introduction we demonstrated the system operation to the evaluator. The 2D and 3D windows were displayed, most functions were manipulated: loading various GIS data layers into the 2D and 3D scenes; navigating through the 3D environment by flying, driving, automatic driving, fix point and perspectives functions; choosing and changing the raster, linear, and patch data, and simulating the flood model in fix duration mode. Considering the time and effort taking to train the evaluator to use this system, it was decided that the software developer manipulated the system during the session and the evaluator sit aside asking questions and giving comments. Comments from the evaluator were written down. After the demonstration session, a prepared questionnaire was handed out to the evaluator, which was returned to the software designer a week later.

According to the evaluator's oral and written opinions, we categorize two types of responses corresponding to different parts of the system: positive and negative. They are listed as follows.

- Positive evaluations

  1. The part that she gave the highest positive evaluation is the potential applications in site decision processes. She mentioned if the visual quality was improving and more environmental information was incorporated, then this system could be widely used in deciding which places were suitable for development. It is useful not only in landscape project, but also in urban or suburban planning or regional development. She mentioned that this tool could not only display various information to the designer in an appropriate way, but also could provide the public a visualization tool in forming common opinions and discussing issues closely related to themselves.

  2. Another important value of this system she mentioned was the ability to transform abstract or quantitative data types into concrete and understandable graphical forms. For example, the extent of pollution can be expressed by different colors rather than a table of numbers. This type of representation enhanced the public to access the environmental knowledge. Furthermore, if all the development proposals were required to run their plans on this system and displayed the environmental impacts on the screen that everyone could see, then this tool could play an important role in the official processes of reviewing a landscape, city or town plan.

3. In the questionnaire, she expressed high extent of satisfaction about the navigate functions GISVE provides. Similar to other experts being consulted during the task analysis, she thought the different perspectives the system provided were very helpful and necessary for a computer aided design tool. The interactivity of fly and drive functions in her evaluation also achieved satisfaction.
4. She thought it was essèntial and necessary to put environmental models into GISVE that made such system a tool not only for viewing visual impacts but also for environmental impacts.

- Negative evaluations

1. The visual quality is ranked below satisfaction, especially in the 3D window. She expressed the discontent about the dissimilarity between objects on computer screen and real scene. It includes: terrain didn't show the variation of ground cover; roads didn't show important features such as widths and curbs; streams didn't show widths and contours of valley; ponds didn't show the change of topography and vegetation; houses needed more details such as driveways and garages; trees needed to show individual ones. To sum up, she required improvements in visual quality by adding more details to 3D objects if this system would be used in practical projects.
2. Due to the unfamiliarity of the system, she didn't understand what the purposes of choose functions were for, so she didn't answer questions about choose functions. Although this is not a negative evaluation, it would be a concern in redesigning the user interface or giving a clearer description about the manipulation of the system.
3. She mentioned that the change functions only change the geometry of objects; in practical cases all relevant parameters should change corresponding to single modification. For example, to add one pond in the environment changes the topography, vegetation covers, and soil type of surroundings, these modifications cannot be displayed in GISVE yet.

4. Concerning environmental model simulation, she thought that was really important, but the models embedded in the GISVE system were not complete and accurate enough to provide the quantitative information of assessing the whole environmental impact to a place. There were problems with the flood model at the first evaluation session that displayed wrong results, so the evaluation of model modules couldn't be done.

To sum up, from the result of the first evaluation session, the system features and application potentials were confirmed by the evaluator, but the implementation of these features, such as visual quality, change functions, model functions, were not satisfactory yet; some erroneous parts need to be corrected.

### 4.3.2 Modifications

Several modifications of the system have been made before the second evaluation session. Most of these modifications are based on the result of the first evaluation.

- The errors of flood model are corrected so that it generates results which make sense. It is shown that during the heavy rain, the valley along the river is flooded at first, then expands to the flat area. Still use only one color to draw the flood cell, if the water level of this cell is beyond a threshold, then draw the blue color to this cell.

- The pollution model is linked with the flood model because the flow surface runoff in calculating soil erosion should be obtained from the flood model. The pollution was represented by multi-threshold color coding.

- We use the same colors in representing the same objects on the 2D and 3D windows. It is for avoiding the confusion when using the system.

- The scale of vertical elevation is enlarged five folds that the variation of topography is more easily identified.

- A 2D animation function is added in the system. After the user runs the fix duration simulation function, he/she can choose animation function to see the evolution of variables by running a sequence of frames on 2D window.

Two major improvements are still not done yet: 3D visual quality and environmental models. We understand that in order to improve these two factors, we need much more time and effort working on this project. Considering the time frame of the entire project and the resources of the computer, we decide to leave these works alone in the current stage. Since it is a prototype rather than the real product, we can add more sophisticated 3D graphics and environmental models in future development stages.

### 4.3.3 Second evaluation

The second evaluation was held on November 22th, 1995, at the same place. It was focused on the parts which were not covered or had low grades during the first session. A test project was run during this session to compare the environmental impacts generated by two development proposals. Another prepared questionnaire was issued after the session and was sent back to software developer several days later.
The concept and detail of test project will be discussed in the next section. Here we discuss the comments and responses from the evaluator.

- She has seen the improvement in visual quality of terrain. The main reason was the enlargement of vertical scale.
- After explanation the purposes and operations of set functions became clear.
- Model functions were still below satisfactory level. The models used were not accurate enough. The pollution model needed to show soil deposition as well as erosion.

- Her basic attitude towards this system was the same as that during the first evaluation session. More elements needed to put into the system when this system would be for practical uses.

## 4.4 Project testing

In order to validate a simulation system, besides the evaluation process, a testing process that invites the user to run the system to do a "virtual task" is also necessary at the last stage of software development. In the flight simulator this testing process is called pilot experiment. We borrow the concept of pilot experiment to GISVE in the final validation process. Here the testing process is called "project testing". A task to be executed is a landscape project. The conclusion or final result of this project will be drawn by using various functions GISVE provides. It shows that GISVE facilitates the working process of the test project and makes the user easier to fetch and manipulation information than conventional ways. This validates the use of the system.

### 4.4.1 Definition and goal of the test project

At first we have to choose the test project. Here are the features that we expect this project should have.

- This project contains the environmental impact assessment, which is the main goal GISVE system is designed to aid. It should look for the proposal that causes the least environmental impact. In other words, here the environmental issue is greater than economic, aesthetic, social, and other concerns.
- As mentioned in section 2.2, this system is used in preliminary stage of landscape design. More detailed considerations, such as the vegetation covers being used,

the sizes and styles of houses, or construction technologies, will not be considered yet.

- The information used in running this project is restricted to that provided by GISVE. The main (or the only) information available is the GIS data and environmental models. The more information we have, the more considerations the conclusion will be endowed with, so the less probable this conclusion will result in serious environmental impacts. However, we can only include limited amount of information in the system because of the restrictions of both data collection and system specifications; on the other hand, to work with finite amount of information is the case in any project, we cannot wait until getting enough information.

From these requirements, the test project is defined as follows.
The target area of this project is in Petersham, which is situated in middle Massachusetts. It is a medium sized New England town with suburban environment. Harvard forest sits on both sides, where many research projects about forestry, ecology and botany of Harvard University are going on. The vegetation cover is the mixed type which includes evergreen and deciduous trees; this type is usually located at the temperate area with medium or high amount of precipitation. The topography is relatively flat (below 1000 ft). Some small hills called drumlins, which are the deposition of soil during the glacial movement, scatter through the entire area. The target place is a part of Petersham, it is a 4.5 kilometers by 4.5 kilometers area, including downtown Petersham, several ponds and wetlands, and a part of Harvard forest.

Various layers of GIS data are available from US Geological Survey, including elevation, roads, water bodies, tree covers, soil types, and buildings. Figure 4 shows a comprehensive plot of GIS data displayed on the 2D window. (Figure 4)

Figure 4-2: Petersham, Massachusetts

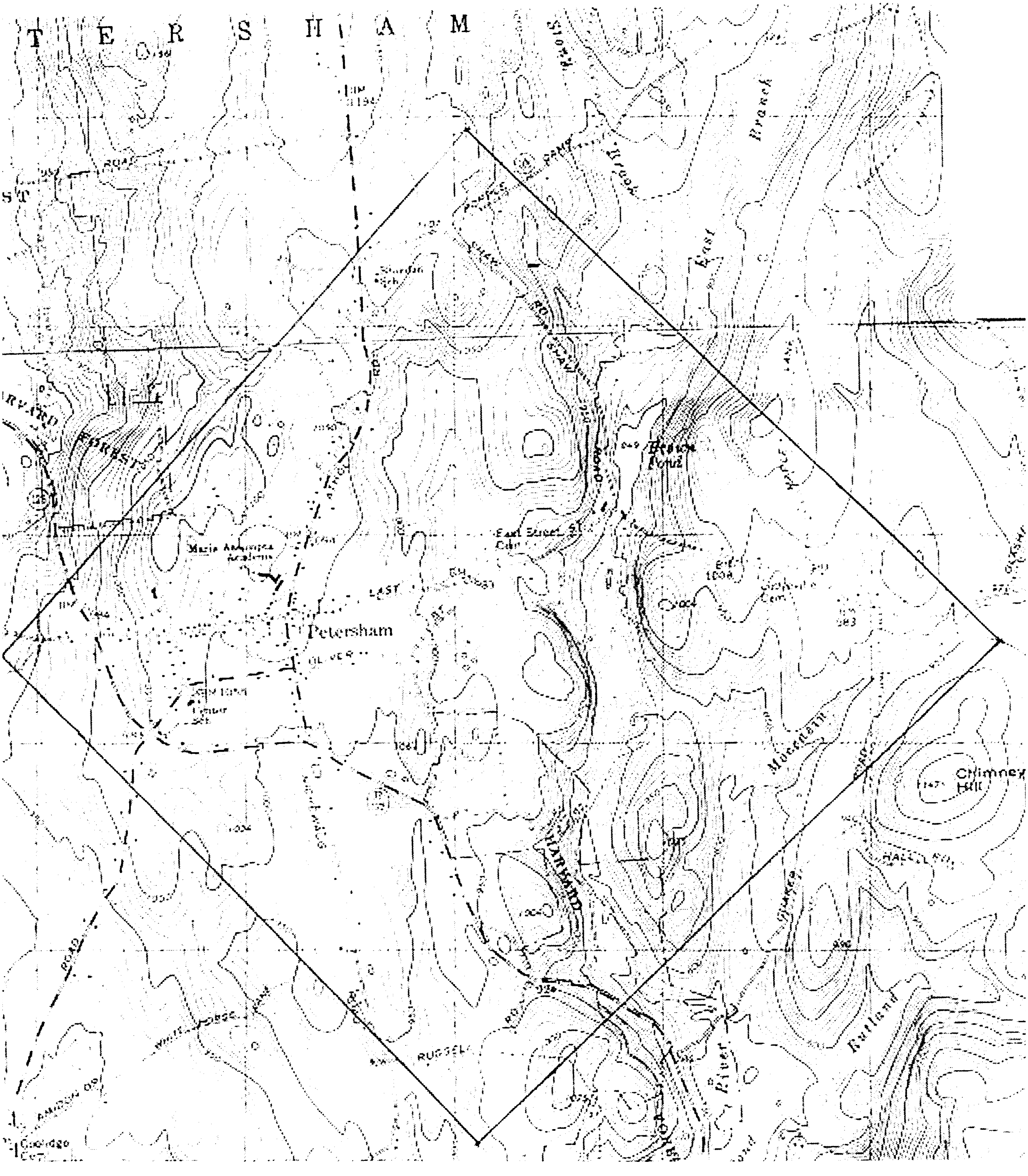
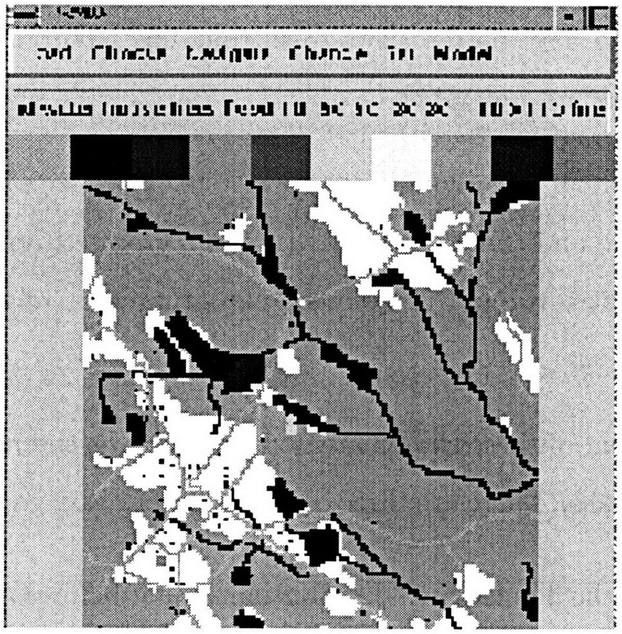


Figure 4-3: GISVE 2D window



Conceptually the project is simple: suppose a real estate developer wants to build one hundred new houses in the target area, where are the best places to build, and how to distribute these new houses in these potential areas? As mentioned before, environmental protection is the major concern, so the goal becomes: what kind of development can reduce the environmental impact to the minimum extent?

### 4.4.2 Criteria achieving the goal

In forming strategies of the developments in the beginning of design phase, it is necessary to give a comprehensive review of all factors at first. Although the main issue of this project is environmental protection, other important factors cannot be neglected. Several criteria are generated in terms of environmental, economical, comfortable, and safe concerns in choosing the appropriate sites for development.

- The sites should be away from the flood area, which means to avoid places on river valleys and wetlands.

- The sites should have convenient access to roads.

- The sites should have better views to surrounding environments, i.e., places at higher altitudes are preferable.

- The slope of these sites should not be too steep, which will increase both the soil erosion rate and the cost of construction. From criteria 3 and 4 it implies that better places to build new houses are situated at relatively high and flat sites.

- The development of these chosen areas should have least environmental impacts to the target place. There are three concerns in terms of environmental impacts:

    1. Compare the flooded area distribution to that before the new development, this development should cause the least increment to the flooded area.
    2. The development should cause the least increment in the area where soil erosion-generated pollution is beyond some threshold value.
    3. The development should have lowest probability of catching fire to the whole area after a spot of fire occurs.

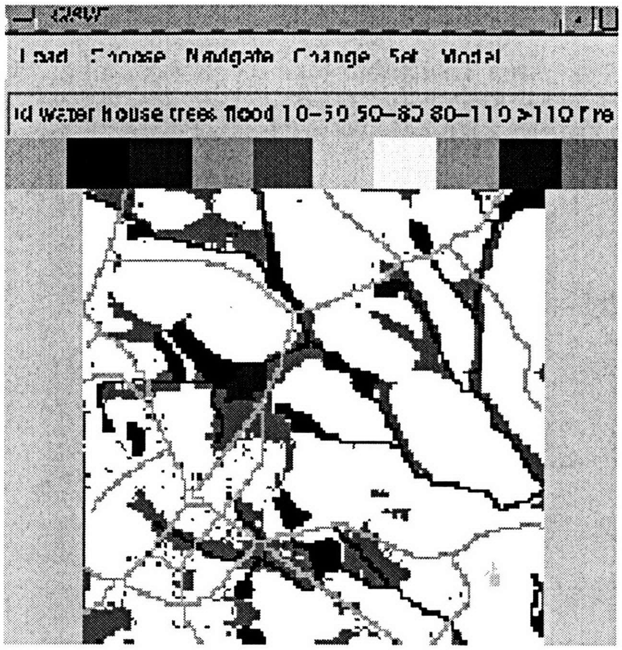
### 4.4.3 Processes of working on project using GISVE

Based on the objectives and criteria described above, we can use GISVE in deciding the development and distribution of houses in these sites.

#### Deciding appropriate sites

Load GIS layers of elevation and water bodies into the scene. Run the flood model, the initial conditions are set as: rainfall duration is 33 minutes, precipitation is 11 cm over the entire duration, and simulate it for 22 minutes. Look at the flood model simulation result on 2D and 3D maps; it is shown that flood is mainly concentrated along the river valleys.

Figure 4-4: Area 1 suitable for new development, 2D view


Load Choose Navigate Change Set Model
ld water house trees flood 10-50 50-80 80-110 >110 Fre

Add roads and houses layers into the scene. It is shown that existing houses are concentrated in the lower-left corner of the map, which is the location of downtown area. Find out locations on 2D map which are away from flooded area with easy access to roads. There are three locations which satisfy these conditions: one is at the lower-right corner of the map, one is at the upper-left part of the map, which is located between two tributary systems, one is at the upper-right part, which is a sub-center of the town with intermediate house density. These three areas are places suitable for building new houses.

Add tree cover layers into the scene. Avoid areas which a large patch of forest lies on. For example, the forest on the right part is Harvard forest, so it is not allowed for new development.

Navigate through these three areas on 3D window to look for appropriate places within the areas: places with higher elevation and flat slope are desirable. In area 1, such places are concentrated along the right side of the road which goes down to the bottom of the map. In area 2, such places are at the ridge between two tributary systems. In area 3, such places are concentrated in the sites at the lower left side.

Figure 4-5: Area 1 suitable for new development, 3D view

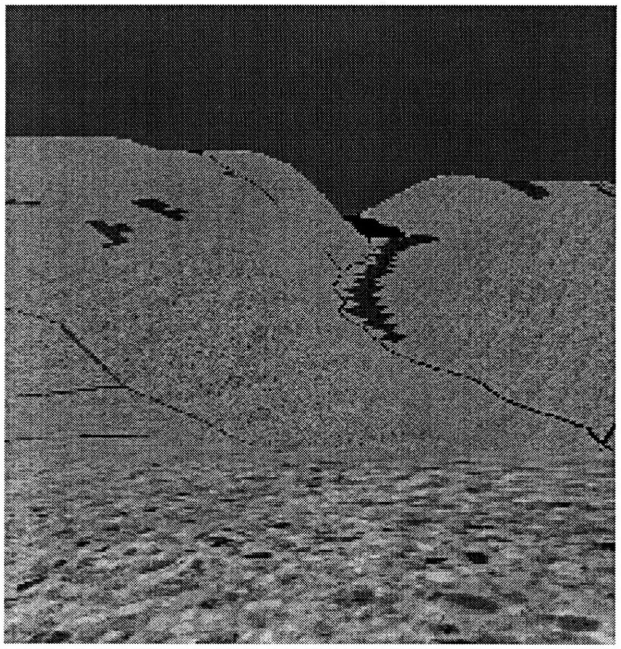


Figure 4-6: Area 2 suitable for new development, 2D view

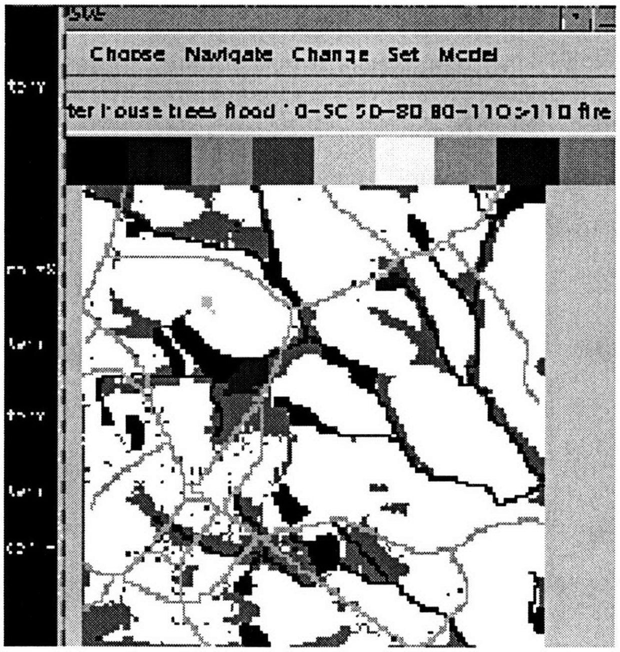

Choose Navigate Change Set Model
ter house trees Road '0-50 50-80 80-110 >110 fire

Figure 4-7: Area 2 suitable for new development, 3D view

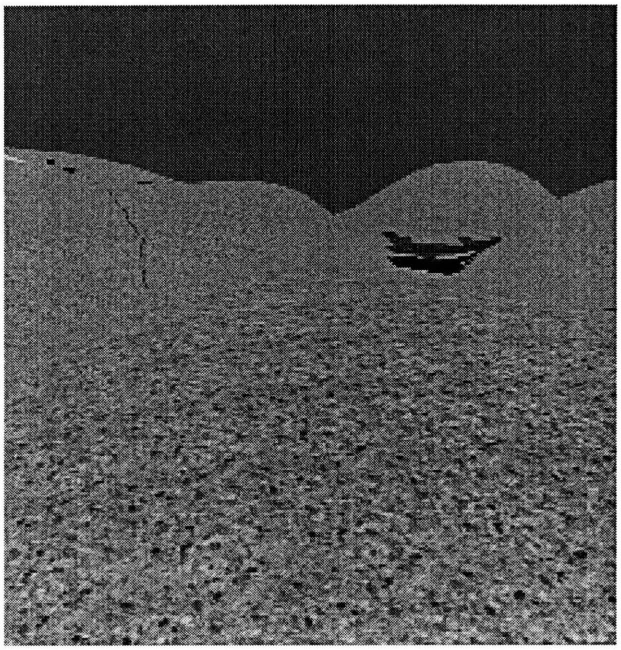


Figure 4-8: Area 3 suitable for new development, 2D view

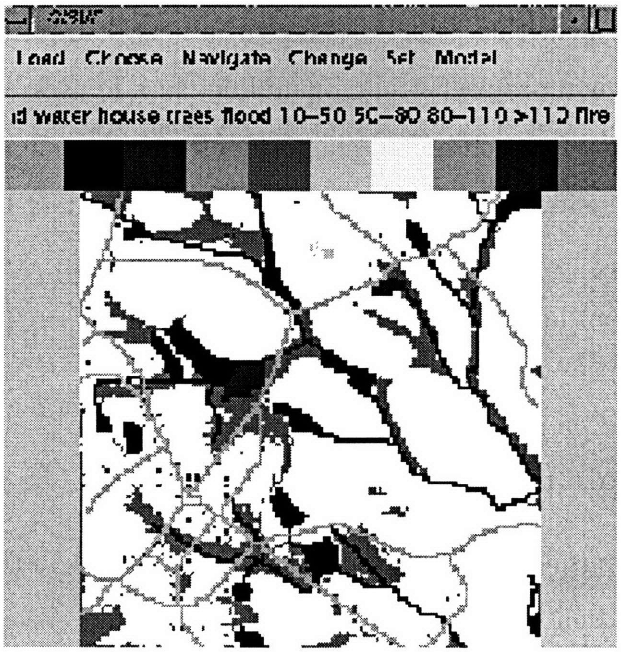
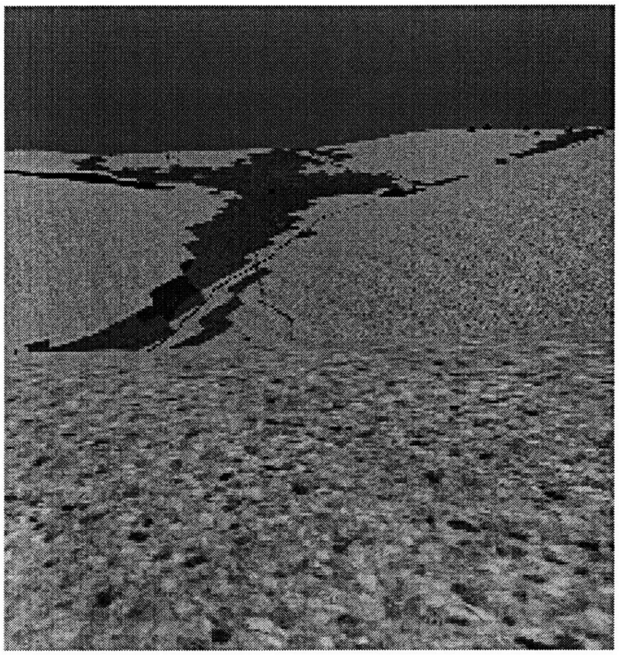

Load Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-9: Area 3 suitable for new development, 3D view



### Determine two strategies of development

Two strategies of development are generated in order to build 100 new houses within the target area. The first one is to distribute houses evenly into the three appropriate subareas. The second one is to concentrate these houses within one subarea. When implementing strategy 2, we decide to concentrate houses in area 3 because it is already partially developed, thus causes less environmental impact. Traditionally low density development is more preferable for middle class houses because it maintains more space and privacy for each house; cluster houses are beneficial in ecological concerns because they create less impermeable areas and cut less trees, thus result in less surface runoff and more open land. Here we want a more concrete measure and visual representation of how environmental impacts differ with these two proposals.

### Change and simulate the environment

Execute the first proposal. Use change-add functions to add thirty three new houses in each previously chosen sub-area and cut trees within the cells of new houses. Each house occupies one cell. Run the simulation of flood runoff and erosion-carried pollution models using the same initial conditions previously set (33 minutes in rain duration, 11 cm in precipitation, 22 minutes in simulation duration). Then run the fire diffusion model simulation by choosing one spot on fire initially.

Execute the second proposal and repeat the model simulations with identical initial conditions. Concentrate one hundred houses within area 3. Each cell being chosen contains 3 houses. Cut trees within the cell, and run the same simulations with the same initial conditions.

Also run the model simulations for the environment without changes. The results are used as the control to proposal 1 and proposal 2.

### Analyze and select proposals

The flood model simulation results are shown in figure 4-10 to figure 4-12. There are not great differences in terms of the flooded area for the three cases.

The erosion-generated pollution results are shown in figure 4-13 to figure 4-15. It is shown that the pollution range of each sub-area of new development in proposal 1 expands compared to the situation before the change. The pollution area in proposal 2 doesn't expand much since area 3 is already partially developed. The expansion of soil erosion is due to the increase of permeable area, depletion of trees, and increase of pollutant amount generated by houses.

The fire diffusion results are shown in figure 4-16 to figure 4-18. In terms of the damage costs of houses, proposal 1 is more preferable because the density of houses is low, so less houses are destroyed after a certain period of time. However, the overall area on fire doesn't change much for these two proposals, in other words, if taking the damage of forest into consideration, costs of both proposals are close.

To sum up, the area where flood or fire occurs doesn't change much as different development proposals are adopted. However, in terms of the consideration of soil erosion carried pollution, proposal 2 is better than proposal 1. So by the aid of GISVE, three elements of environmental impacts are compared, thus relevant information about these two proposals is acquired. Although this information is not complete, it is an important consideration in the final decision of development proposal.

Figure 4-10: 2D map of control flood result

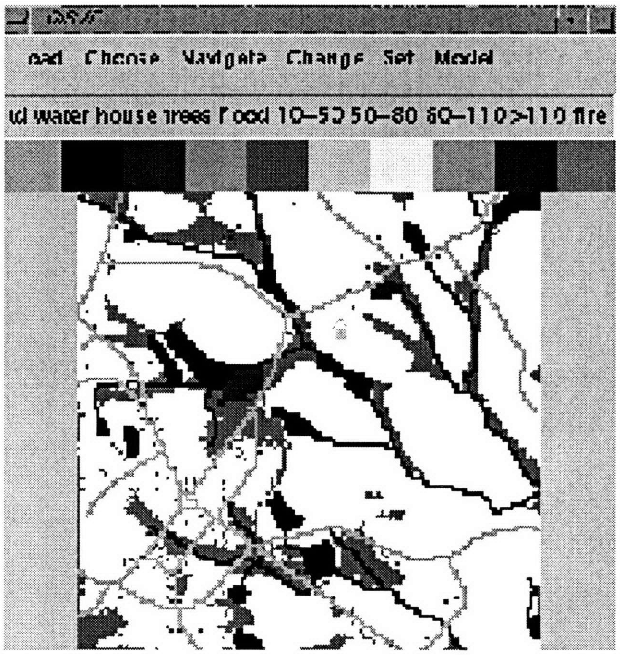

oad Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-11: 2D map of proposal 1 flood result

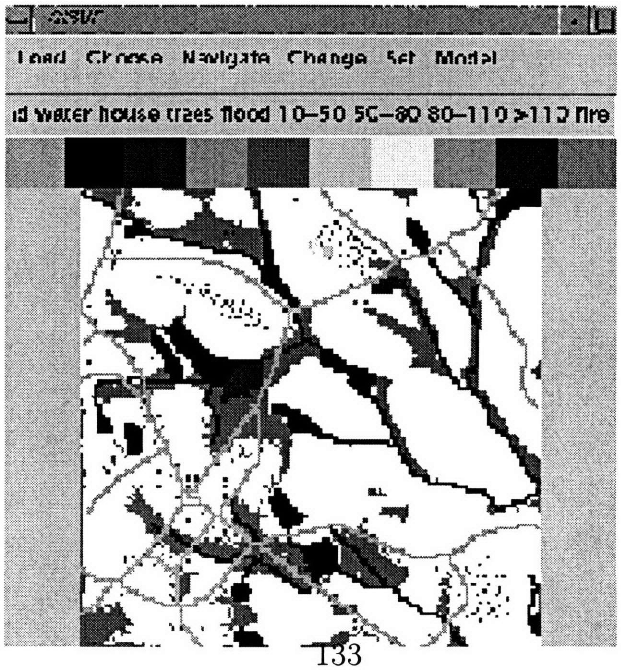

Load Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-12: 2D map of proposal 2 flood result

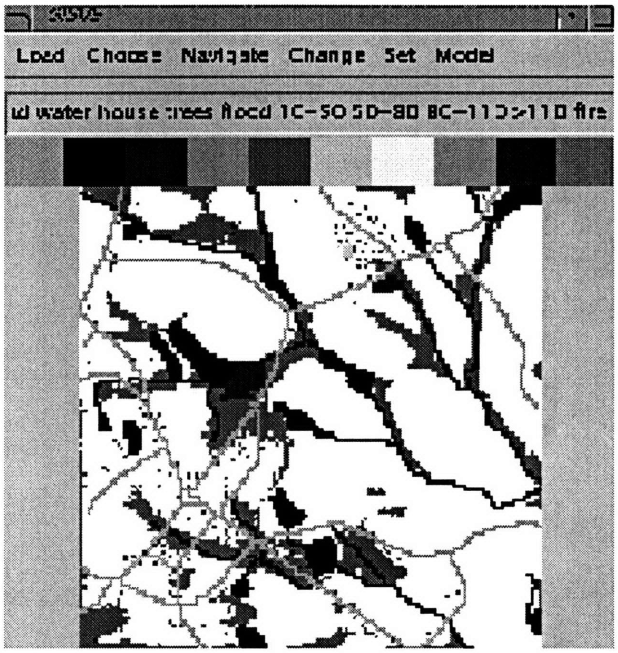

Load Choose Navigate Change Set Model
ld water house trees flood 10-50 50-80 80-110 >110 fire

Figure 4-13: 2D map of control erosion result

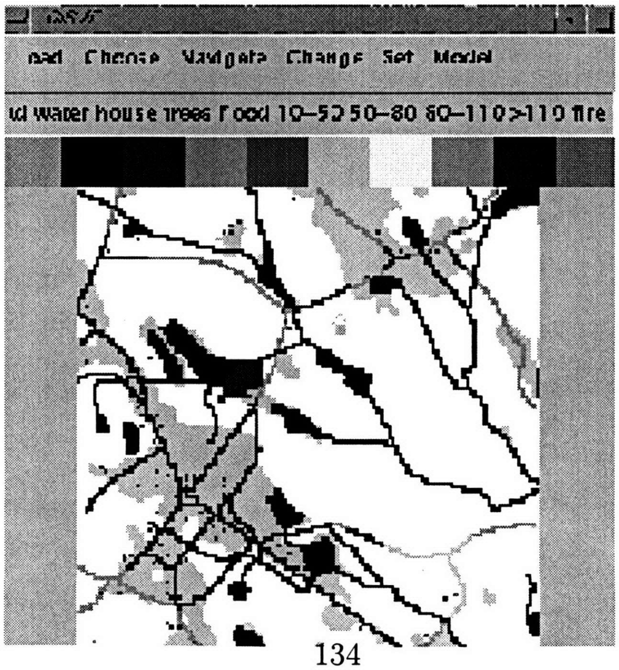

oad Choose Navigate Change Set Model
ld water house trees flood 10-50 50-80 80-110 >110 fire

Figure 4-14: 2D map of proposal 1 erosion result

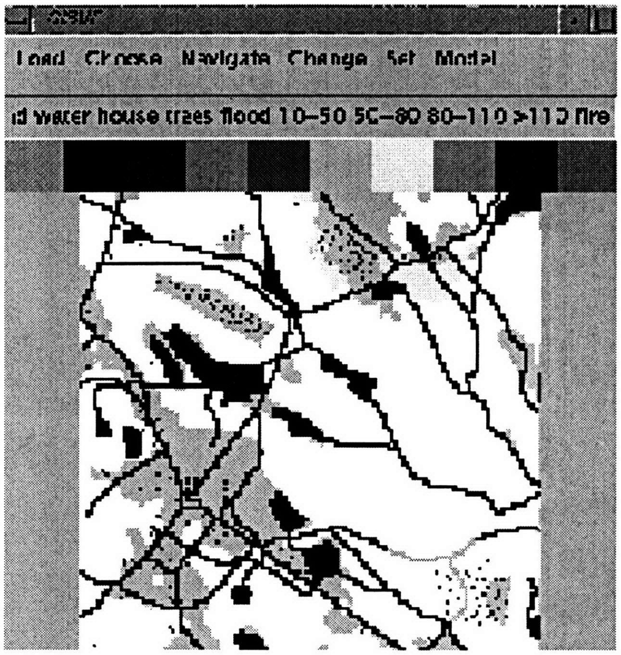

Load Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-15: 2D map of proposal 2 erosion result

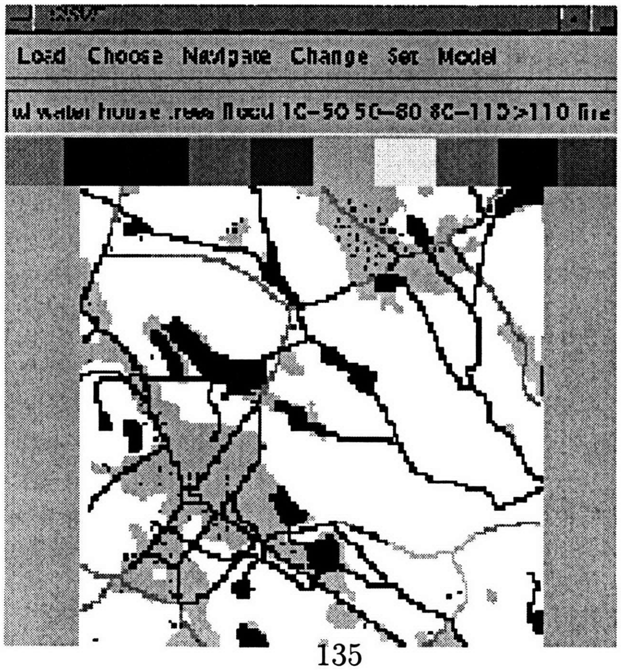

Load Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-16: 2D map of control fire result

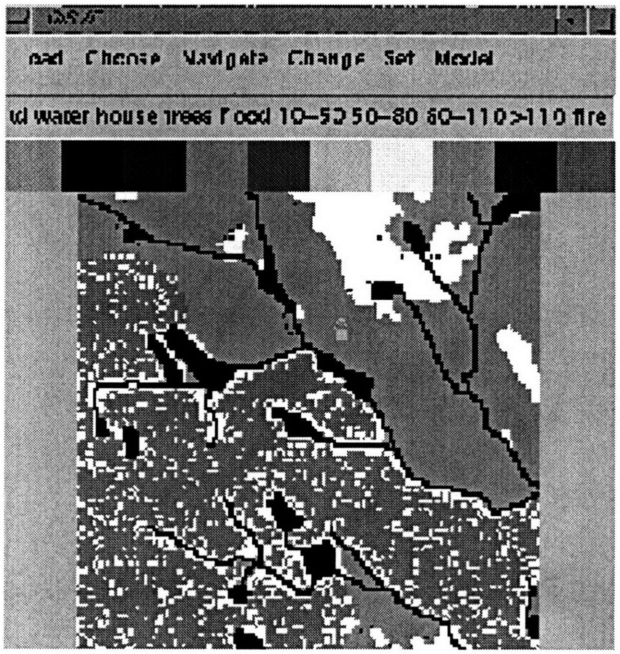

oad Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-17: 2D map of proposal 1 fire result

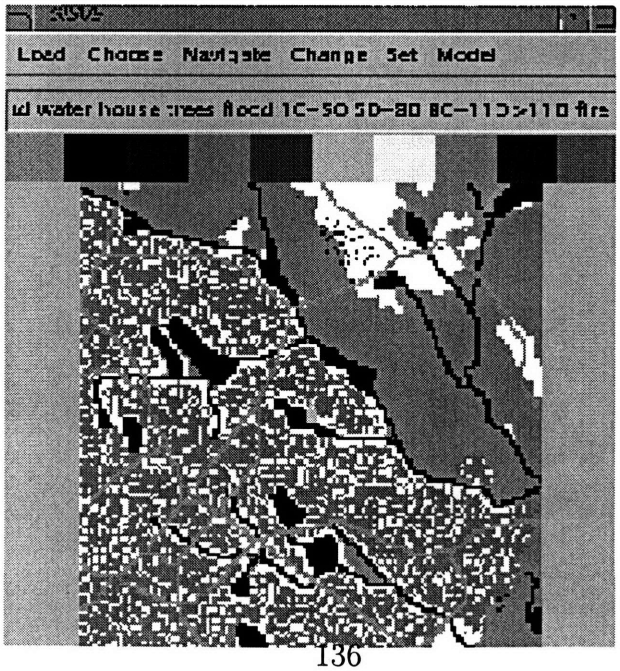
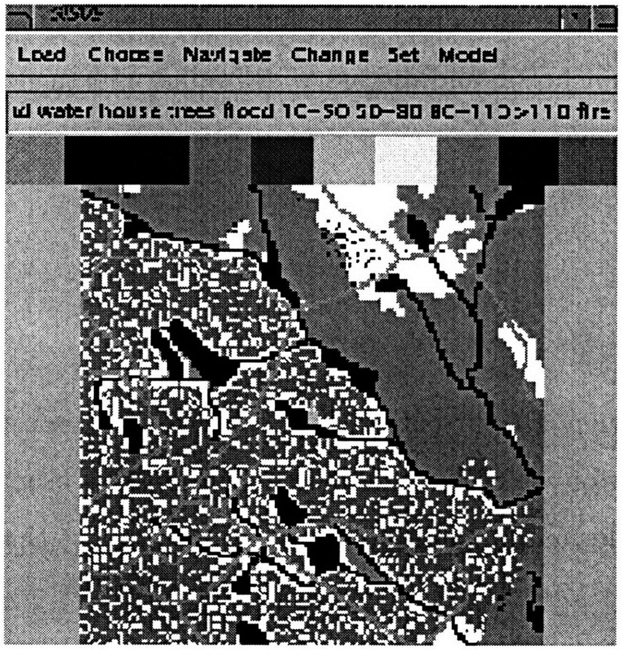

Load Choose Navigate Change Set Model
d water house trees flood 10–50 50–80 80–110 >110 fire

Figure 4-18: 2D map of proposal 2 fire result


Load Choose Navigate Change Set Model
ld water house trees flood 10-50 50-80 80-110 >110 fire

# Chapter 5

# Conclusion

So far all the development stages of GISVE have been discussed, including motivation, task analysis, specification determination, implementation, verification and validation. In this chapter, we discuss the future works about its improvement. Finally, we draw the conclusion and discuss what we have learned from the experience of implementing this system.

## 5.1 Future work

As mentioned in section 4.3, currently GISVE still can not be used in practical landscape projects. A lot of work has to be done to improve its user interface, capability of providing more information, and interactivity. We propose three directions for the future improvement.

### 5.1.1 Improvement of visual quality

Visual quality in 3D environment is the item with least satisfaction from the evaluation of domain experts. In order to be a tool for practical uses, visual quality should attain the level that the details and materials of objects can be displayed as well as their geometries.

There are several feasible ways to improve the visual quality. So far all 3D objects are generated from raster based GIS data. Because the resolution of grids is low, we cannot get the information about the shapes and materials of objects within a cell. If incorporate more information such as pictures (air photos or ground pictures) into the system, then more details are displayed. Another way is to get the 3D models from current existing databases and put these objects into GISVE's 3D environment. The most accessible 3D models are houses; we can obtain graphical models of houses with different sizes and architectural styles, then put them in the environment so that it looks more real because of the diversity of buildings. We can even build these models from the buildings of the target place; it depends upon that in the project if we need to take such effort to do this improvement. For structures like streams or forests, more parameters should be investigated to create variations in bank widths and depths, and vegetation species; more texture mapping will be used to reduce polygon number. There is a tradeoff between polygon number and rendering speed, so we have to decide which part is more important in different cases.

The effect of light and shade should be added into the environment if more accurate visual impact estimation is needed. This can be done by allocating the light sources and tracing the rays; although light effect is a complicated problem in computer graphics, a lot of research works have been done, so we can refer to these research works in improving our system.

Model simulation results can be represented in more sophisticated ways. Color coding is one way to represent different levels of the same variable. However, we can also use different icons to represent different variables, for example, use stop signs to represent the boundaries of prohibited area.

### 5.1.2 Enhancement of environmental models

As mentioned in section 4.3, environmental models embedded in the GISVE system are not accurate enough to demonstrate the quantitative results; and they only cover a small part of the environmental factors which should be considered in practical applications. There are several factors which are important to landscape design but are not plugged in the system yet:

### Complete the hydrological cycle

Aspects of hydrological cycles include surface runoff, infiltration, ground water flow, evaporation (caused by variation of temperature and humidity), transpiration (driven by plants), and precipitation. Theoretically all these parts are not separable. In GISVE, so far only surface runoff and infiltration factors are covered. To build a mathematical model simulating the entire hydrological process is still a great challenge for environmental scientists, but unless such complete model is implemented in the system, the user cannot obtain the overview of the hydrology of the target place. Relevant design works of sewers, septics, restoration wells, dams, and appropriate sites for building houses cannot be implemented if we lack the knowledge about this process.

### Wind effect

Wind turns out to be a very important factor in the micro-climate of both urban and rural areas. The model simulating the influence of topography and vegetation covers to wind should be included in the system. The strength and direction of wind can be represented by haptic devices or computer controlled fans to simulate the wind blow.

### Water quality

So far GISVE only contains a model simulating erosion carried non-point source pollution. In fact, people concern more about how pollution sources influence water quality of rivers, lakes, and ground water. In this sense, soil erosion is the first stage of the entire pollution process, subsequent stages include diffusion and transport in water, chemical and biological reactions (for example, eutrophication). So put models simulating all these stages is much more useful than only implementing erosion generated pollution.

**Succession of the ecosystem**

Ecosystem succession models are useful in preservation and in predicting trends of environmental change. They could be mathematical models simulating species interaction, or the record about the history of this ecosystem, especially vegetation cover. Put such mathematical models or past records in GISVE system can help users to view the change for the purpose of ecological preservation or historical restoration.

### 5.1.3 Creating an intelligent agent

Although we have seen how GISVE can help the landscape design process by displaying information in user-friendly forms and by running simulations to demonstrate the environmental impact, it still needs a lot of time for the user to manipulate this system. He/she has to determine what kind of GIS data and environmental models should be incorporated in the system, do the mechanical work of moving and clicking the mouse to communicate with the computer, and to decipher the simulation result on computer screen. For most users, this tool is not very friendly and powerful yet. A domain expert who was interviewed in task analysis has mentioned that he didn't want to use the mouse at all. He just wanted to tell the computer some rules to follow, then the computer should display results with the minimum cost in terms of these rules. For example, he wanted to find out a path with the least number of houses seen from this path. In other words, the ultimate requirement for a system aiding landscape design is that the computer can finish most time-consuming part of the job, then left final decision part to the user.

It turns out to build an intelligent agent with the following functions. The agent follows the commands from the user, these commands are the goals this project wants to achieve. For example, to minimize water pollution. The agent would have the ability to convert user's general, abstract goals into concrete steps to achieve these goals. This is very difficult because a simple word can imply a lot of work to do. Still use the example of water pollution minimization, pollution from all sources should be checked, models representing different stages the pollutant flows should be implemented. Each stage of checking the pollution sources and running the simulations should be defined by the agent. After determining these stages, the agent generates several alternatives and execute steps defined previously to these alternatives, it chooses an alternative (or some) which is closest to the general goal set by the user. To build such agent is much more difficult than the previous two improvements. This can be taken as the long term goal of a computer aided design tool. In short term, the intelligence of agent can be achieved by giving it more knowledge about environmental change and user's status, so it can display information and apply models without user's manipulation, thus simplify the manipulation of the system.

## 5.2 Summary

The progress of computer technology enhances the opportunities of cooperations among different disciplines, it also increases the communication between different groups of people. We pursue these directions to be the goals of GISVE. In order to do this, different information and knowledge are wrapped up within a system and displayed in visual forms. This system integrates GIS, 3D graphics, and environmental model simulation together. This is a innovative approach; although it is not yet mature enough for practical applications, and it still needs a lot of improvements, it indicates a new direction in expanding the application domain of VE technology and in linking with computers for environmental purposes.

# Appendix A

# Hermite spline curves

The spline problem is very important in computer graphics. Suppose there are a series of points in a plane or in higher dimensional spaces and we want to find out a curve connecting all these points. There are infinite number of such curves, for example, we can connect adjacent points directly to form a piecewise linear curve. However, if we add more constraints to the curve we want, for example, the $n$th derivative value at each point is continuous, then less curves satisfy these conditions. In computer graphics, when we draw the curve from these sampled points, we usually want the curve to look smooth, which means each segment is continuous to a certain order derivative.

A curve in high dimensional space can be represented by a one-variable parametric form. In 3D space, a curve can be represented as:

$$x = x(t), y = y(t), z = z(t) \tag{A.1}$$

Usually t is between 0 and 1; at start point t=0, at end point t=1. Usually $x(t)$, $y(t)$ and $z(t)$ are polynomial functions. The smoothness is determined by the order of derivatives which are continuous over the entire curve; the more smooth it is, the more conditions required to determine coefficients of polynomials.

The cubic polynomials that define a curve segment $Q(t) = \begin{pmatrix} x(t) & y(t) & z(t) \end{pmatrix}$ are of the form

$$x(t) = a_x t^3 + b_x t^2 + c_x t^1 + d_x \tag{A.2}$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t^1 + d_y \tag{A.3}$$

$$z(t) = a_z t^3 + b_z t^2 + c_z t^1 + d_z \tag{A.4}$$

It can be written as a matrix form:

$$T = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} \tag{A.5}$$

$$C = \begin{pmatrix} a_x & a_y & a_z \\ b_x & b_y & b_z \\ c_x & c_y & c_z \\ d_x & d_y & d_z \end{pmatrix} \tag{A.6}$$

$$Q(t) = T * C \tag{A.7}$$

When the curve segments are in the 3D space, there are 12 unknown coefficients to be determined for each segment; in other words, it needs 4 known vectors to determine these coefficients.

For Hermite curves, the 4 constraint vectors of each segments are two end points vectors $P_1$ and $P_4$ and two tangent vectors $R_1$ and $R_4$. The parametric form of a segment is the product of the polynomial vector, the coefficient matrix, and the matrix corresponding to constraints.

$$Q(t) = T * M_H * G_H \tag{A.8}$$

$M_H$ is called the Hermite basis matrix, which will be obtained later. $G_H$ is called

Hermite geometry vector, whose elements are constraint vectors.

$$G_H = \begin{pmatrix} P_{1_x} & P_{1_y} & P_{1_z} \\ P_{4_x} & P_{4_y} & P_{4_z} \\ R_{1_x} & R_{1_y} & R_{1_z} \\ R_{4_x} & R_{4_y} & R_{4_z} \end{pmatrix} \tag{A.9}$$

Take a look at the $x$ component, define $G_{H_x}$ as

$$G_{H_x} = \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix}_x \tag{A.10}$$

Then $x(t)$ can be expressed as

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x = T * M_H * G_{H_x} = \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} * M_H * G_{H_x} \tag{A.11}$$

The four constraints should be satisfied, so

$$x(0) = P_{1_x} = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix} * M_H * G_{H_x} \tag{A.12}$$

$$x(1) = P_{4_x} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} * M_H * G_{H_x} \tag{A.13}$$

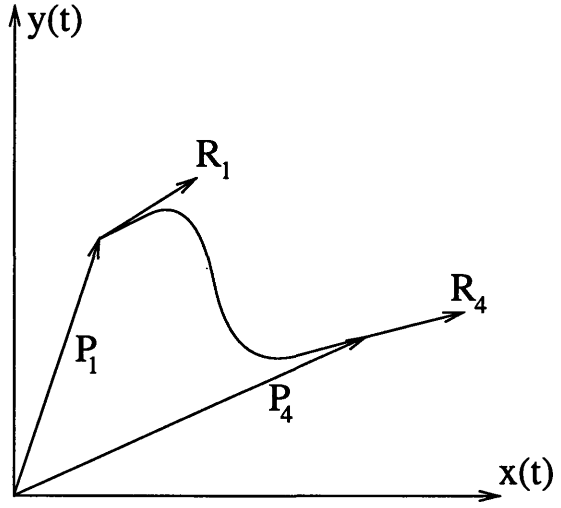
$$x'(0) = R_{1_x} = \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} * M_H * G_{H_x} \tag{A.14}$$

$$x'(1) = R_{4_x} = \begin{pmatrix} 3 & 2 & 1 & 0 \end{pmatrix} * M_H * G_{H_x} \tag{A.15}$$

So the four constraints can be written as a matrix form as

$$\begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix}_x = G_{H_x} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix} * M_H * G_{H_x} \tag{A.16}$$

Figure A-1: A Hermite curve connecting two points showing four constraint vectors



So

$$M_H = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{pmatrix}^{-1} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{A.17}$$

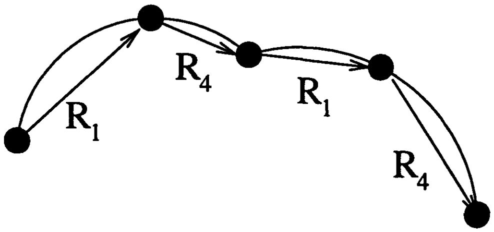
The same equation sustains for $y$ and $z$ components, so $M_H$ is the same for $y$ and $z$ components. Then curve segment can be written as:

$$\begin{aligned} Q(t) &= T * M_H * G_H \\ &= \begin{pmatrix} t^3 & t^2 & t & 1 \end{pmatrix} * \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} P_1 \\ P_4 \\ R_1 \\ R_4 \end{pmatrix} \\ &= (2t^3 - 3t^2 + 1)P_1 + (-2t^3 + 3t^2)P_4 + (t^3 - 2t^2 + t)R_1 + (t^3 - t^2)R_4 \end{aligned}$$

The Hermite curve segment between two end points is shown in Figure A-1.

It is seen that this curve passes the end points, which is different from other spline algorithms such as B-spline or Bézier curve that end points are not on the curve. So

Figure A-2: Hermite curves connecting multiple points



the Hermite curve is used in smoothing the linear data.

In order to spline all sampled points, group each three points into a segment, the constraint vectors of these three points are called $P_1$, $P_2$, $P_3$. By using these points, we can obtain the four constraints described above. The end point vectors are $P_1$ and $P_3$, the tangent vectors are $P_2 - P_1$ and $P_3 - P_2$, thus the parametric form can be written as:

$$Q(t) = (t^3 - t^2)P_1 + (-t^2 + 1)P_2 + (-t^3 + 2t^2)P_3 \tag{A.18}$$

To make sure end points are aligning on the curve, when there is one point remaining, connect this end points with the last curve created with a line. The splined curves of several sampling points are as follows. (Figure A-2)

# Appendix B

# Questionnaire and response of the first evaluation session

This is the questionnaire and evaluator's response to the first evaluation session.

## Questionnaire of GISVE

This questionnaire is for evaluating the GISVE system. In order to get a quantitative measure of system performance, use the following six-point scale:

5 - Extremely Satisfactory
4 - Moderately Satisfactory
3 - Between Satisfactory and Unsatisfactory
2 - Moderately Unsatisfactory
1 - Extremely Unsatisfactory
D - Don't Know

For answers less than "moderately satisfactory", please suggest how the features might be improved in the space of this question.
Some other questions need to be answered by short comments rather than numeral ranks.

## B.1 System features

### B.1.1 Two dimensional display

1. Compare to other GIS you ever used, how clear and understandable is the 2D display of this system ?

   *D.*

2. How suitable is the color code in representing different GIS data layers ?

   *OK, but need more ground cover categories.*

3. How does the position icon on 2D map work in helping you identifying the position ? Do you use this icon often ?

   *D.*

4. How does the elevation indicator work in helping you identifying the position ? Do you use the elevation indicator often ?

   *No - eye level shots most important; maybe more important to have elevation in urban or very hilly areas.*

### B.1.2 Three dimensional display

1. How is the visual quality of elevation (terrain) in the 3D window ?

   *Not accurate enough for site design - neither good for ground cover.*

2. How is the visual quality of roads in the 3D window ?

   *Does not show important features - width, curb, ...*

3. How is the visual quality of streams in the 3D window ?

   *Does not show widths, contours of valley, ...*

4. How is the visual quality of ponds in the 3D window ?

   *Need better terrain into adjacent to water bodies and vegetation.*

5. How is the visual quality of houses in the 3D window ?

   *For design, need more details, also, driveways, garages, ...*

6. How is the visual quality of trees in the 3D window ?

   *Saw only forest, which is OK, but need individual trees.*

7. How is the representation for flooded cells ?

   *Model seemed wrong when we did this.*

8. How is the representation for polluted cells ?

   *Did not see.*

9. How is the representation for fire cells ?

   *Did not see.*

10. How is the orientation sense in this visual environment, can you identify your current position and moving direction on the 3D window ?

    *Not well enough for design.*

## B.2 System functions

### B.2.1 Load/Unload functions

1. Please evaluate the necessity of load functions. If you feel some layers are not necessary, indicate which ones.

   *Don't know.*

2. How clear and understandable is the manipulation of load functions ?

   *Don't know.*

3. How is the time response of load functions ? If you find some data loading too slow to stop other functions of this system, please indicate which ones.

   *Don't know.*

### B.2.2 Choose functions

1. How clear and understandable is the manipulation of choose functions ?

   *Don't know.*

2. How is the time response of choose functions ? If you find some choose items so slow to stop other functions of this system, please indicate which ones.

   *Don't know.*

### B.2.3 Navigate functions

#### Fly function

1. Please evaluate the necessity of the fly function.

   *Would be useful if details were appropriate.*

2. How clear and understandable is the manipulation of the fly function ?

   *5.*

3. How is the time response of the fly function ?

   *5.*

#### Drive function

1. Please evaluate the necessity of the drive function.

   *Very important - eye level perspective is most like actual experience.*

2. How clear and understandable is the manipulation of the drive function ?

   *4.*

3. How is the time response of the drive function ?

   *5.*

4. How is the interactivity you feel in the environment by using the drive function ?

   *3.*

**Autodrive function**

1. Please evaluate the necessity of the automatic drive function.

   *Very useful.*

2. How clear and understandable is the manipulation of the automatic drive function ?

   *5.*

3. How is the time response of the automatic drive function ?

   *5.*

4. How is the interactivity you feel in the environment by using the automatic drive function ?

   *4.*

**Fix-position function**

1. Please evaluate the necessity of the fix-position function.

   *Necessary.*

2. How clear and understandable is the manipulation of the fix-position function ?

   *5.*

3. How is the time response of the fix-position function ?

   *5.*

### Perspective function

1. Please evaluate the necessity of the perspective function.

   *Necessary.*

2. How clear and understandable is the manipulation of the perspective function ?

   *5.*

3. How is the time response of the perspective function ?

   *5.*

## B.2.4 Change functions

### Add functions

1. Please evaluate the necessity of add functions.

   *Necessary, but must develop capabilities to show impacts, otherwise not really adding.*

2. How clear and understandable are the manipulations of add functions ?

   *Not clear.*

3. How is the time response of add functions ?

   *3.*

### Delete functions

1. Please evaluate the necessity of delete functions.

   *Necessary, but again, it is important that are impacts.*

2. How clear and understandable is the manipulation of add functions ?

   *Not clear.*

3. How is the time response of add function ?

   *3.*

### B.2.5 Set functions

#### Set point function

1. Please evaluate the necessity of the set point function.

   *Don't know - did we get to this ?*

2. How clear and understandable is the manipulation of the set point function ?

   *Don't know - did we get to this ?*

3. How is the time response of the set point function ?

   *Don't know - did we get to this ?*

#### Set path function

1. Please evaluate the necessity of the set path function.

   *Don't know - did we get to this ?*

2. How clear and understandable is the manipulation of the set path function ?

   *Don't know - did we get to this ?*

3. How is the time response of the set path function ?

   *Don't know - did we get to this ?*

### B.2.6 Model functions

#### Flood model function

1. Please evaluate the necessity of the flood model function.

   *Model seemed to be wrong for flood pattern.*

**Pollution model function**

1. Please evaluate the necessity of the pollution model function.

   *I didn't get to see this, but this is necessary to distinguish different kinds of "pollution" - soil washes off surface, toxic chemicals may travel underground, etc.*

**Fire model function**

1. Please evaluate the necessity of the fire model function.

   *Didn't see this.*

## B.3 Overall performance

### B.3.1 Usefulness of GISVE system

1. How useful do you think of this system in the landscape architecture design process ?

   *Has potential if can show interactive impacts of proposed design, this requires much more data and more complex modeling.*

2. For which kind of field or project in landscape architecture do you think will be the most beneficial from this system ?

   *If do this could be very useful to evaluate alternatives, for public meetings, development costs, environmental impact statements, etc.*

### B.3.2 Immersion of GISVE

1. How do you think the visual quality of the overall system ?

   *Not enough detail.*

2. How do you think of the interactivity of the overall system ? Do your actions in the environment be responded quickly ? If not, please indicate which actions

you need but have no response.

*Quickly enough - but not accurately !*

3. Consider other modal input/output devices (for example, speaker, head-mount display, speech recognizer), do you think it's necessary to put other devices on this system ? If so, what device is the most suitable ?

   *Not important at this point.*

### B.3.3 Spatial and temporal variations

1. How do you think this system achieves displaying spatial variations ?

   *1.*

2. How do you think this system achieves displaying temporal variations ?

   *1. Not real add/delete function unless show concomittant changes.*

# Appendix C

# Questionnaire and response of the second evaluation session

This is the questionnaire and evaluator's response to the second evaluation session.

## Questionnaire of GISVE

This questionnaire is for evaluating the GISVE system. In order to get a quantitative measure of system performance, use the following six-point scale:

5 - Extremely Satisfactory
4 - Moderately Satisfactory
3 - Between Satisfactory and Unsatisfactory
2 - Moderately Unsatisfactory
1 - Extremely Unsatisfactory
D - Don't Know

For Answers less than "moderately satisfactory", please suggest how the features might be improved in the space of this question.
Some other questions need to be answered by short comments rather than numeral ranks.

## C.1 3D display

1. How is the visual quality of the terrain in the 3D window ? Is there an improvement compared to the previous version you saw ?

   *Yes, is vertical scale exaggerated ?*

   *Does erosion model use steeper slopes or shallower slopes ?* [1]

2. How is the visual quality of water bodies (streams and ponds) in the 3D window ? Is there an improvement compared to the previous version you saw ?

   *Not improved water, but looks better in topography.*

## C.2 Change functions

1. How clear and understandable of add functions, after explanation by the system developer ?

   *4.*

2. How clear and understandable of delete functions, after explanation by the system developer ?

   *4.*

## C.3 Set functions

1. How clear and understandable of set point and path functions, after explanation by the system developer ?

[1] The vertical scale is enlarged, and soil erosion is based on enlarged slope values.

*4.*

# .4 Model functions

## 4.1 Flood model function

1. How clear and understandable of the flood model function, which includes the manipulation of setting initial conditions and the display of results in 2D and 3D windows ?

   *D.*

2. Compared to the previous version you saw, does the result of the flood model make more sense ? How accurate do you think, not quantitatively but qualitatively, does the result of model simulation show ?

   *D.*

3. How do you think about the time response of the flood model function ?

   *D.*

4. What do you think of using color coded patch to represent flooded area ? If it is not satisfactory, can you suggest some other representations of flood area ?

   *Flood area should be blue - it becomes water.*

## .4.2 Pollution model function

. How clear and understandable of the pollution model function, which includes the manipulation of setting initial conditions and the display of results in 2D and 3D windows ?

?. *Have to show deposition (siltation) as well as erosion areas - double damage sults from erosion.*

w accurate do you think, quantitatively rather than qualitatively, does the ılt of the pollution model function show ?

*2.*

3. How do you think about the time response of the pollution model functio

   *D.*

### C.4.3 Fire model function

1. How clear and understandable of the fire model function, which include manipulation of setting initial conditions and the display of results in 2D 3D windows ?

   *1.*

2. How accurate do you think, quantitatively rather than qualitatively, doe result of the fire model function show ?

   *1.*

3. How do you think about the time response of the fire function ?

   *D.*

4. What do you think of using color coded patch to represent fire area ? If not satisfactory, can you suggest some other representations of soil erosion ?

   *3.*

## C.5 Project testing

1. Do you think GISVE provides correct information in determining the building new houses ? If not, please indicate which part is wrong.

   *Must broaden scope of environmental impacts, both on site and region*

2. Do you think the environmental model simulation results of the two in the test project provide useful and correct information in eval environmental impacts caused by these two strategies ? If not, w

wrong or less useful ?

*Not yet - takes much more ecological info + performance standards.*

3. Do you think GISVE is a useful tool in aiding the site decision process of the test project ?

   *Not yet, but I think it could be with much more refined inputs + calculations.*

4. What will you suggest for GISVE system if it is going to be a software really used by landscape architects, rather than an experimental prototype ?

   *As we discussed - you must work with landscape architects + ecologists to load in more, much more refined logic tree + calculations + look at regional impacts - traffic, suburban development, air pollution, etc.*

Thank you very much for your cooperation.

# Bibliography

[ARC90] *Understanding GIS: the ARC/INFO method.* Environmental system research institute, Relands, California, 1990.

[Bed92] Philip B. Bedient. *Hydrology and floodplain analysis.* Addison-Wiley, 1992.

[BHT90] Michael Begon, John L. Harper, , and Colin R. Townsend. *Ecology - individuals, populations and communities.* Blackwell scientific publications, 1990.

[D+87] Colin G. Drury et al. Task analysis. In Gaviel Salvendy, editor, *Handbook of human factors*, pages 370–401, New York, 1987. Wiley.

[DM94] Nathaniel I. Durlach and Annes S. Mavor, editors. *Virtual reality, scientific and technologial challenges.* National Research Council, 1994.

[DZ94] Steven M. Drucker and David Zeltzer. Intelligent camera control in a virtual environment. In *Graphics interface '94*, pages 190–198, 1994.

[Erv92] Stephen M Ervin. Visualizing n-dimensional implications of two-dimensional design decisions. In *Proceedings visualization '92.* IEEE computer society press, 1992.

[F+90] James D. Foley et al. *Computer graphics - principles and practice.* Addison-Wesley publication company, 1990.

[F+94] Sharon Fischler et al. *IRIS Performer Programming guide.* Silicon Graphics, 1994.

[Goo93] Michael F. Goodchild. The state of gis for environmental problem solving. In Michael F. Goodchild, editor, *Environmental modeling with GIS*, pages 8–15. Oxford university press, 1993.

[HD$^+$93] Ian W. Hunter, Tilemachos D. Doukoglou, et al. A teleoperated microsurgical robot and associated virtual environment for eye surgery. *Presence*, 2(4):265–280, 1993.

[JR93] Eric F. Johnson and Kevin Reichard. *Power programming Motif.* Management information source, Inc., 1993.

[JSO95] Pierre Y. Julien, Bahram Saghafian, and Fred L. Ogden. Raster-based hydrologic modeling of spatially-varied surface runoff. *Water resource bulletin*, 31(3):523–536, June 1995.

[KA93] Peter L. Knepell and Deborah C. Arangno. *Simulation validation, a confidence assessment methodology.* IEEE computer society press, 1993.

[Lau86] Michael Laurie. *An introduction to landscape architecture.* Elsevier Science Publishing Co., Inc, 1986.

[Lot24] Alfred J. Lotka. *Elements of physical biology.* Williams and Wilkins Co. Inc., Baltimore, 1924.

[M$^+$90] McCullough et al. Iries in a landscape: an experiment in dynamic interaction and teaching design studio. In *The electronic design studio.* MIT press, 1990.

[Mai93] David R. Maidment. Gis and hydrologic modeling. In Michael F. Goodchild, editor, *Environmental modeling with GIS*, pages 8–15. Oxford university press, 1993.

[Man83] Benoit B. Mandelbrot. *The fractal geometry of nature.* W.H. Freeman, San Francisco, 1983.

[Mar91] William M. Marsh. *Landscape planning-environmental applications.* John Wiley and Sons, Inc., 1991.

[Mit95] William J. Mitchel. *City of bits.* MIT press, 1995.

[MS91] B.M.E. Moret and H.D. Shapiro. *Algorithms from P to NP, design and efficiency*, volume 1. The Benjamin/Cummings publishing company, Inc., 1991.

[NB93] Robert A. Nisbet and Daniel B. Botkin. Integrating a forest growth model with a geographic information system. In Michael F. Goodchild, editor, *Environmental modeling with GIS*, pages 8–15. Oxford university press, 1993.

[NO94] Vladmir Novotny and Harvey Olem. *Water quality, prevention, identification, and management of diffuse pollution.* Van Norstrand Reinhold, New York, 1994.

[Pio95] Nicholas J. Pioch. Officer of the deck:validation and verification of a virtual environment for training. Master's thesis, Massachusetts Institute of Technology, 1995.

[PT92] Ken Pimentel and Kevin Teixeira. *Virtual reality, through the new looking glass.* Intel/Windcrest/McGraw-Hill, INC., 1992.

[R$^+$83] Walter J. Rawls et al. Green-ampt infiltration parameters from soil data. *Journal of hydraulic engineering, ASCE*, 109(1):62–70, 1983.

[SO92] Gerhard N. Schmitt and Werner Oechslin. Computer aided architectural design futures. In *CAAD futures '91: computer aided architectural design futures: education, research, applications*, pages 9–16, Vieweg, Germany, 1992. Braunschweig/Wiesbaden.

[Ste93] Louis T. Steyaert. A perspective on the state of environmental simulation modeling. In Michael F. Goodchild, editor, *Environmental modeling with GIS*, pages 8–15. Oxford university press, 1993.

[TG94] David M. Theobald and Mark D. Gross. Eml: a modeling environment for exploring landscape dynamics. *Computer, environment, and urban systems*, 18(3):193–204, 1994.

[TM88] T. Toffoli and N. Margolus. *Cellular automata machine: a new environment for modeling.* MIT press, 1988.

[Tom90] C. Dana Tomlin. *Geographic Information Systems and Cartographic Modeling.* Prentice Hall, Inc., 1990.

[VK93] Stephen J. Ventura and Kyehyun Kim. Modeling urban nonpoint source pollution with a geographic information system. *Water resource bulletin*, 29(2):189–198, April 1993.

[YO+89] R.A. Young, C.A. Onstad, et al. Agnps: a nonpoint source pollution model for evaluating agricultural watersheds. *Journal of soil and water conservation*, pages 168–173, 1989.

[ZL93] David Zeltzer and Jeng-Feng Lee. Research on improved sonar displays: a human/machine processing system. Technical report, Sensory Communication Group, Research Laboratory of Electronics, Massachusetts Institute of Technology, August 1993.