COMPUTERIZED SIMULATIONS FOR INTRODUCTORY GEOGRAPHY INSTRUCTION : ORIENTEERING

by

Nirupama Thiruvengadam

Project submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

©Nirupama Thiruvengadam and VPI & SU 1993

APPROVED:

Dr. Clifford A. Shaffer, Chairman

Edward A. Fox

Dr. Lawrence W. Carstensen

Dr. Edward A. Fox

August, 1993
Blacksburg, Virginia

c.2

LD
5655
V851
1993
T547

COMPUTERIZED SIMULATIONS FOR INTRODUCTORY GEOGRAPHY INSTRUCTION : ORIENTEERING

by

Nirupama Thiruvengadam

Committee Chairman: Dr. Clifford A. Shaffer

Department of Computer Science

(ABSTRACT)

Computer-aided education (CAE) techniques offer good means of satisfying the need for laboratory instruction in geography. These techniques provide graphical manipulation of representations of geographical processes in a manner that stresses dynamic processes over static processes. This project focuses on the design and development of the Orienteering module. This is a surrogate travel program which shows the view one sees while moving through a mixed woods and fields area. This module uses Intel's Digital Video Interactive (DVI) Technology to enhance the orienteering experience by providing a simulation of forward motion and a simulation of panoramic, seamless view at every location of the terrain. The student uses a compass presented on the screen to navigate across the terrain. The Orienteering module is one of the six computerized laboratory modules of projet GeoSim, which is being developed to support introductory geography instruction.

ACKNOWLEDGEMENTS

I wish to extend my sincere thanks to Dr. Clifford A. Shaffer, my committee chairman and advisor whose guidance and suggestions were instrumental in the completion of this project. I would also like to thank Dr. Lawrence W. Carstensen and Dr. Edward A. Fox for serving on my committee and for their constant willingness to help.

I would also like to take this opportunity to thank Ghaleb Abdulla for his help in capturing the images and our discussions about the technicalities of the DVI system. Thanks are due to Mark Lattanzi for his assistance with Project GeoSim's general purpose database reader.

Thanks to my friends Sangita Betrabet and Praveen Kumar Paripati, for the countless invigorating discussions and their invaluable support.

I am grateful to my parents and sisters for the constant encouragement they have given me always. Special thanks to my friend, philosopher, and husband Srinivas who showed me that I could dare to dream and gave me strength to make my dreams come true.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

This paper describes the prototype design and implementation of the Orienteering module of Project GeoSim. This module is a surrogate travel simulation that teaches map orientation skills. It has been implemented using Intel's Digital Video Interactive (DVI) which provides the capability to simulate motion using still frame images, thereby making the simulation realistic.

1.1 Project GeoSim

The study of geography is an important component of a university education. Geography is unique among disciplines in its focus on the spatial aspects of physical processes and human actions. Geographic knowledge is essential for understanding local and global problems relating to environment, urban development, transportation, agriculture, trade, politics, and technology development, to name a few. Nevertheless, dozens of studies point out the incompetence of US students in the understanding, practice, and basic knowledge of geography [Cars93]. Recognition of the need for improved geography education is resurging throughout the United States, but is still far behind other sciences in both universities and secondary schools.

A multi-disciplinary group from the Departments of Computer Science and Geography at Virginia Tech has initiated Project GeoSim to create computer-aided education (CAE) software for the teaching of introductory geography [Shaf90]. GeoSim applies the immense capabilities of CAE, simulation, and Geographic Information Systems (GIS) to the teaching of geography, beginning at the first geography course. Thus, the most exciting aspects of geography are made available to the widest possible range of students in a manner that stresses dynamic processes over static processes. This project will provide an integrated series of self-contained computer modules that serve as lab exercises compatible with introductory geography courses.

1.2 Orienteering

Orienteering is one module of Project GeoSim. This module uses a large database of pictures to simulate the view one would see while moving through the countryside in a mixed woods and fields area. The instructor sets up the desired course using a map of the orienteering terrain by placing markers to define the course. The student begins to find her way around the course with the aid of a hand held map and information presented on the computer screen. At the end of the session, the time taken by the the student to cover the course is recorded and displayed.

Orientation skills are defined as the ability to determine one's position and the location of distant objects in relation to that position, and the ablility to use a compass and map to navigate through an unknown area. Orienteering is a competitive sport first developed in Sweden in 1918 [Smit82]. It challenges a person to find her way through unfamiliar terrain using maps and a compass. The basis of the competition is efficient route selection between markers usually hung from tree branches. The locations of the markers are shown on an orienteering map, and participants are to navigate to all the markers as quickly as possible using the features on the map as a guide [Kjel76].

A simulation of orienteering should prove useful both for education and for research into the techniques of pathfinding. For educational purposes, it is often difficult to set up and carry out an outdoor exercise for a lack of time, location, and transportation [Cars89]. This module illustrates to students interrelationships among spatially distributed features. Locating oneself on a map is a key process in effective map use, whether it is a topographic, thematic, or highway map. Students may experience the loss of orientation (being lost) in a safe manner that will encourage them to be more aware of their surroundings. The skills taught in this module are truly life skills that carry beyond the course into everyday interactions with the environment.

The simulation will also support other exercises related to position finding. In one scenario, the computer places the student in an unknown location (instructor selected from the 385 possible locations). Then with the aid of a map, compass, and the images presented, the student must locate her position as quickly as possible. Another scenario is that the computer places the student randomly within the study area. The student must locate her position and travel to a second position.

It is nearly impossible to produce conclusive results on pathfinding abilities because of the vast complexity present in the "real" environment and an inability to record the decision-making process. A simulation can aid by eliminating the need to travel outdoors to gain experience in traversing unknown terrain. In addition, the simulation can control the number and type of stimuli presented to subjects during a laboratory session [Cars89].

1.3 The Videodisc Prototype

The material and the figures in this section have been adapted from the paper "Videodiscs and Surrogate Travel / The Map-Environment Interface", by Lawrence W. Carstensen Jr. and Allan B. Cox, which appeared in *Cartographica*, Vol 26, Nos 3 & 4, 1989.

A surrogate travel program for orienteering has been developed, used, and evaluated in classes for more than three years at Virginia Tech [Cars89]. This employs a videodisc system for direct access of images on a frame-by-frame basis.

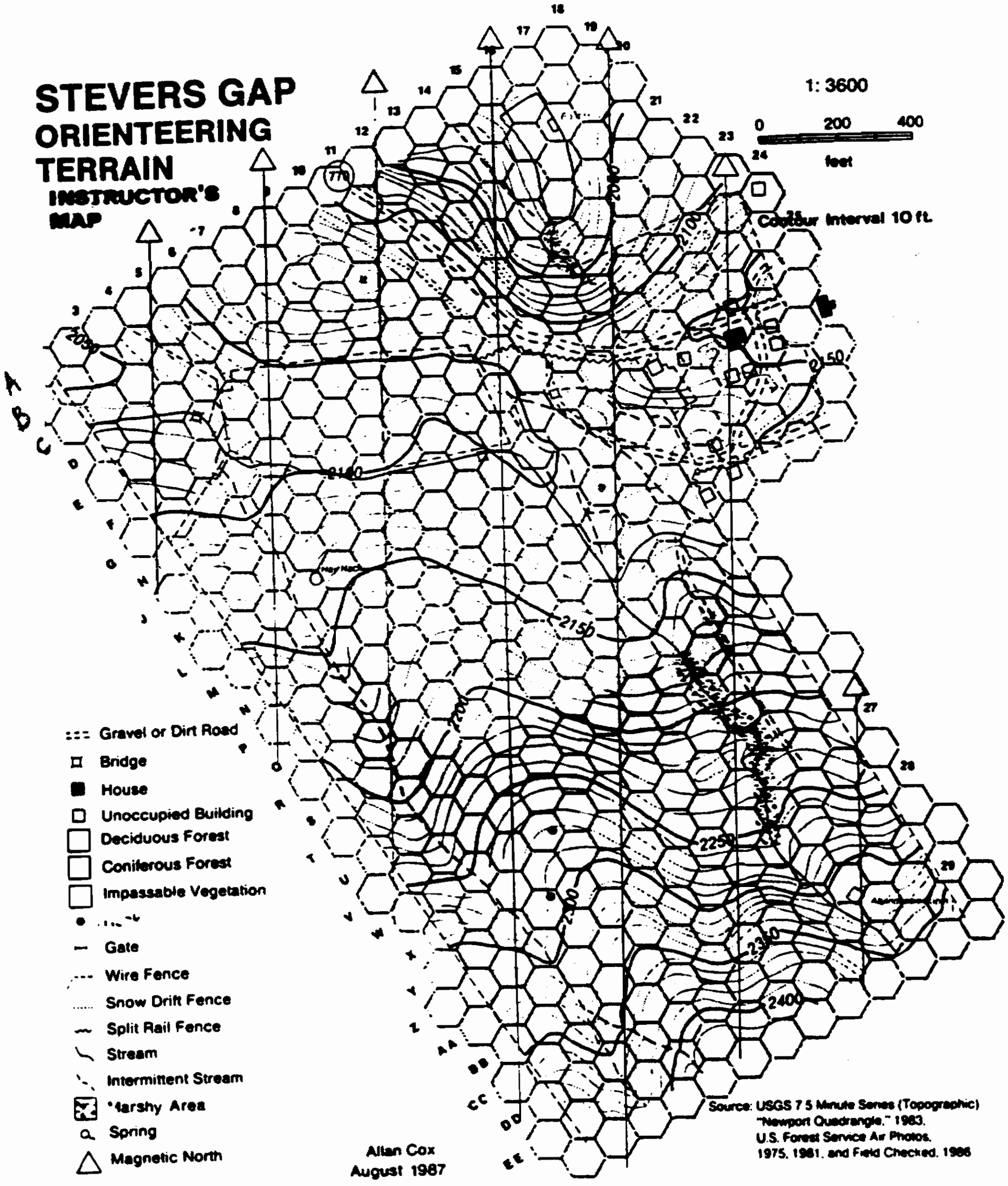
1.3.1 Spatial Partitioning

The study area, a mixed woods and fields terrain, was divided into a regular tessellation of cells in order to partition space into a finite number of origins and destinations to achieve a workable number of paths. Using a hexagonal tessellation provides six equidistant neighbors for each cell. It allows movement in north and south directions as well as at $60^o, 120^o, 240^o$, and 300^o. The hexagonal tessellation has the advantage of being less familiar to users. Movement based on a square grid would be apparent to the user. The hexagonal grid, because it does not form perpendicular rows and columns, is more difficult to decipher and use in movement. As such, a hexagonal grid should produce a more challenging and realistic experience for the user than a square grid. The distance between the grid centers, which is the distance that would be covered by each movement in the simulation, was selected to be 100 feet [Cars89]. Figure 1.1 shows the instructor's copy of the orienteering map with the hexagonal overlay.

Each still frame image on the videodisc has a number associated with it, called the frame number. This number can be used to directly access a still frame image from the videodisc. Each cell in the orienteering map has a name (such as N19) and also a number associated with it, called the cell number. The cell number for cell N19 is 230. The cell numbers are used internally by the videodisc prototype for storing data about the neighbors of each cell. Figure 1.2 shows the azimuths and frame numbers for views taken from cell N19. Solid lines represent valid movement directions and dotted lines represent invalid movement directions. Figure 1.3 shows map position N19 (whose cell number is 230), and all its neighbors. Cell 208 can be reached by moving forward from cell 230 at 0^o azimuth, cell 231 at 60^o azimuth, cell 244 at 120^o azimuth and so on.

1.3.2 Use of the System

A typical session of the simulation begins by having the instructor select the number and location of the markers to be found by the student during a session. Therefore, the difficulty of the course for any session is up to the instructor. All courses begin and end with cell L23 (see Figure 1.1). The instructor then transfers the selected marker locations to the student's copy of the orienteering map (which is similar to the instructor's copy, except that it does not contain the hexagonal tessellation).



Source: USGS 7 5 Minute Series (Topographic) "Newport Quadrangle," 1983. U.S. Forest Service Air Photos. 1975, 1981, and Field Checked, 1986

Figure 1.1: Instructor's copy of the Orienteering Map

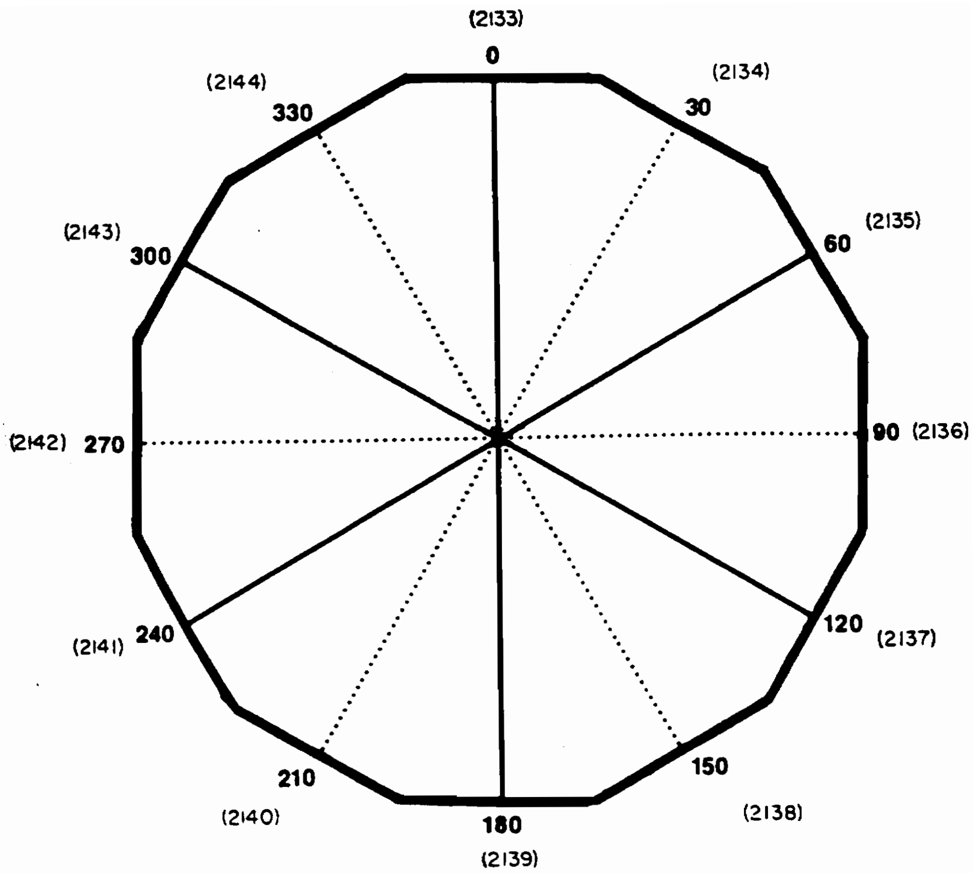


Figure 1.2: Frame numbers and azimuths for views taken from cell N19

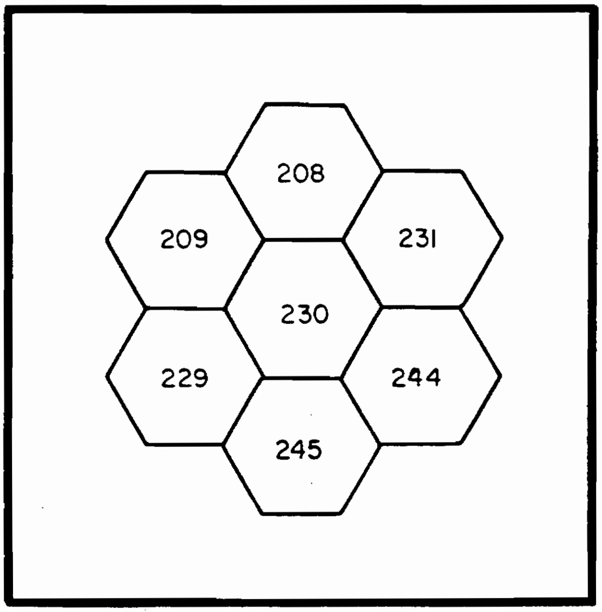


Figure 1.3: Neighbors of cell N19

The computer monitor shows the image corresponding to the starting location. It also contains a compass that shows the angle corresponding to the currently displayed image and indicates by arrow color whether or not that direction is valid for forward movement. Then, the computer is ready for input from the student. The following keys are used by the student to accomplish a simulation of rotation and forward movement : left arrow rotates the view 30° counter-clockwise, right arrow rotates the view 30° clockwise, and up arrow simulates forward movement of the student 100 feet in the direction which she is currently facing.

The student's goal is to place on the computer screen the image corresponding to the position and orientation from which the marker is accessible. This simulates the student moving to the marker in the terrain. By repeatedly doing this for each marker, the student traverses the entire course.

1.4 DVI Prototype Description

The Digital Video Interactive (DVI) implementation of the Orienteering module will use the same set of images as used in the Videodisc Prototype, but will enhance the orienteering experience through the flexible technology associated with CD-ROM and DVI. DVI technology has the capability to store the images not as low resolution television images, but at 512 × 480 resolution, presenting much sharper images and allowing smaller, yet important features such as fences to be clearly resolved. This capability to store better images will be made use of in the future. The prototype currently uses the still frame images from the videodisc. Also, DVI image manipulation functions will more realistically simulate motion between still frame images.

The primary objectives of this prototype are

1. to implement the basic Pan and Zoom image manipulation routines using DVI Technology and still frame images. These two routines will provide the student with a simulation of turning her head and looking around and a simulation of forward motion through the terrain, respectively. These routines can then be integrated with routines for the placement of markers by the instructor, manipulation of markers as the student navigates to a marker, and recording of time and path taken by the student.
2. to illustrate the feasibility of using the sophisticated DVI Technology to enhance the orienteering experience and to provide a closer-to-reality travel simulation program.

This prototype implementation was developed using the "C" programming language [Kern88] and ActionMedia 750 Software Library [Luth91]. The library provides Graphics, System, Real-Time Executive, and Standard functions necessary for the development of a multimedia application (see Section 1.5). These functions have been used in the implementation of the Pan and Zoom routines in this module.

1.5 DVI Technology

The material and the figure in this section have been adapted from "Digital Video Interactive" by Arch C. Luther, McGraw-Hill, 1991.

Digital Video Interactive (DVI) Technology makes possible the first systems which truly merge personal computers and television. Digital Video refers to a system in which all of the image information is in digital bits — a computer data format, which can be displayed or manipulated by a computer. Digital Video specifically excludes analog video, where images are represented by continuous-scale electrical signals.

DVI Technology actually consists of four unique elements

- A custom VLSI chip set
- A specification for a runtime software interface
- Some video/audio data file formats
- Compression and decompression algorithms

Using these four elements, DVI Technology can be implemented on any computer system that has sufficient power and storage capability. Hardware must be designed for that system to interface the DVI chips and to provide other features called for by the runtime software specification. Software must then be written in the native language of the host system to implement the DVI runtime software interface.

The DVI hardware consists of a single board called the ActionMedia 750 Delivery Board. The heart of DVI technology, the i750 chip set, is designed onto this board. The chip set consists of the pixel processor 82750PA and the display processor 82750DA. The board also includes 2 megabytes of VRAM for use by the i750 chip set, an interface for CD-ROM, and a complete audio processing system. The ActionMedia 750 Capture Board allows digitizing of video still images, or motion video, and capture of digital audio, either with video or separately. The Delivery Board Audio uses a Texas Instruments TMS320C10 digital signal processing chip for the audio coprocessor.

The DVI runtime software functions themselves are written in C for the most part, with assembly language used where necessary for better performance. An application for DVI technology is best written in the C language, because all the capabilities of the technology will then be easily accessible.

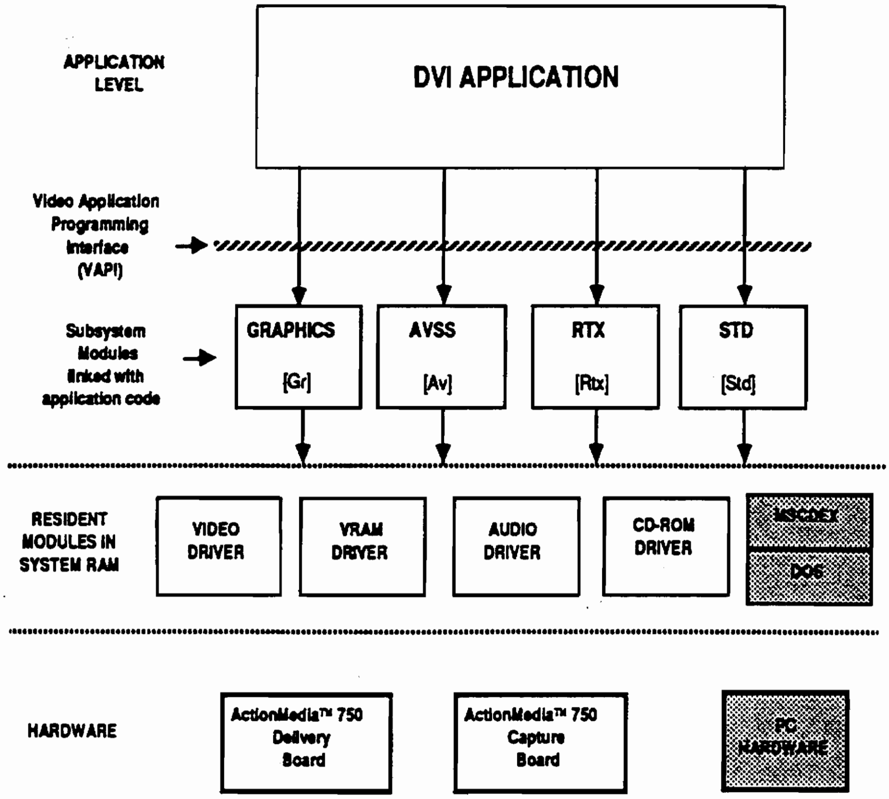

APPLICATION LEVEL
DVI APPLICATION
Video Application Programming Interface (VAPI)
Subsystem Modules linked with application code
GRAPHICS [Gr]
AVSS [Av]
RTX [Rtx]
STD [Std]
RESIDENT MODULES IN SYSTEM RAM
VIDEO DRIVER
VRAM DRIVER
AUDIO DRIVER
CD-ROM DRIVER
MSCDEX
DOS
HARDWARE
ActionMedia™ 750 Delivery Board
ActionMedia™ 750 Capture Board
PC HARDWARE

Figure 1.4: DVI Runtime Software Architecture

Figure 1.4 is a block diagram of the DVI runtime software showing the different modules and their relationship to the hardware and to one another. The shaded boxes are standard PC hardware and software. The clear boxes are parts of DVI hardware and software. The hardware is accessed directly by drivers which are installed at boot-up and remain resident thereafter. The four drivers which deal with the interface to VRAM, the i750 chips, audio, and the CD-ROM interface are not accessible to application programs except through the subsystem modules. The rest of the DVI runtime software is contained in the subsystem modules, which become part of an application program through linking. The driver modules create virtual devices for use by the higher-level software [Luth91].

The following are five classes of routines in the DVI Software Libraries. These can be used by the application to access the drivers :

1. AVSS Routines : The Audio/Video Support System (AVSS) consists of routines that control the playback of audio and video data files from RAM, hard disk, or CD-ROM. With AVSS, audio/video sequences can be played back at various rates, started and stopped, repositioned on the display screen, played at different volumes etc.

2. Graphics Routines : The Graphics routines provide the general purpose interface to the Intel 82750 PA coprocessor. The PA is used to provide

   - Traditional graphics functions such as drawing points, circles, and lines.
   - Special DVI Technology capabilities such as bitmap manipulation, image processing, video effects, and real-time decompression of compressed motion video.

   The Graphics routines also provide control of the 82750 DA, which can be programmed to display in many different resolutions and pixel formats.

3. RTX Routines : The Real-Time Executive (RTX) is a simple multi-tasking operating system that runs "on top of" MS-DOS. RTX is used to control the multi-tasking process of playing simultaneous video and audio, and can also be used for other application-specific purposes.

4. Standard Routines : The Standard routines are used to perform generally useful tasks, such as buffered file I/O.

5. System Routines : The System routines activate and deactivate the Software Libraries [Inte91].

The DVI Technology system is a low-cost product which is capable of providing sophisticated high-quality video compression/decompression. Reducing the amount of data needed to reproduce images saves storage space and increases access speed.

Chapter 2

PROTOTYPE IMPLEMENTATION

2.1 General Description

The Orienteering module is a surrogate travel program using a large database of pictures to simulate the view seen while moving through the countryside in a mixed wood and fields area. The images and a compass are displayed on the DVI screen. The student provides keyboard input to the system, primarily through three keys, viz. left arrow for a view 30^o to the left of the currently displayed view, right arrow for a view 30^o to the right of the currently displayed view, and up arrow for a view 100 feet forward in the same direction. The compass is updated with every change of view.

The desired height and location of the viewport are read from a configuration file. The width of the viewport is calculated using the desired height and the ratio of the height of an actual image to the width of an actual image. The cell number and the angle of view for the starting location of the simulation are also read from the configuration file (see Section 3.2.1). Image files in 16 bits-per-pixel compressed DVI format are decompressed, scaled according to the viewport size, and loaded as required during the simulation.

The data for cropping overlapping images to eliminate redundancy is read into memory and stored in a suitable data structure (see Section 3.2.2). Similarly, the data required for "zooming" forward to a new location is also stored in a suitable data structure (see Section 3.2.3). A log file *output.log* is initialized to record any errors encountered during execution of the simulation.

2.2 Bitmaps

The arrangement of adjacent pixels in a contiguous region of memory is called a *bitmap.* This prototype uses three such bitmaps for storage and manipulation of still images. The work buffer is a bitmap that can hold one cropped image and one uncropped image, placed adjacent to each other. The other two bitmaps are screen bitmaps, each of size 512 × 480, which will be displayed on the screen.

The term *view* is used to refer to the actual view that one would see in the orienteering terrain. The term *image* is used to refer to the still frame image of the view. An image may exist on the videodisc, on the hard disk, or in a VRAM bitmap.

The work buffer always holds two images. On its left side, it holds a cropped image of a view at a certain location at a certain angle. It always holds on its right side, an uncropped image of the view to the right of the image stored in the left side of the buffer (see Figure 2.1). For example, if the left side holds the image corresponding to cell number 62 at 90^o azimuth, then the right side contains the uncropped image corresponding to cell number 62 at 120^o azimuth. This holds true for all images for a certain location and also wraps around to provide a simulation of a panoramic view. So, if the left side holds an image corresponding to 330^o azimuth, the right side holds the image corresponding to 0^o azimuth.

2.3 Pan Function

Each cell of the hexagonal grid has twelve images associated with it. The database holds these images, which overlap to provide a panoramic and potentially seamless view from that cell. However, this overlap between adjacent images has to be cropped to provide a virtually seamless pan. The function described in this section crops the images as required and allows the user to pan across images.

The two screen bitmaps provide a *double buffered* approach to implement a seamless pan. The contents of only one screen bitmap will be displayed at one time. The other bitmap

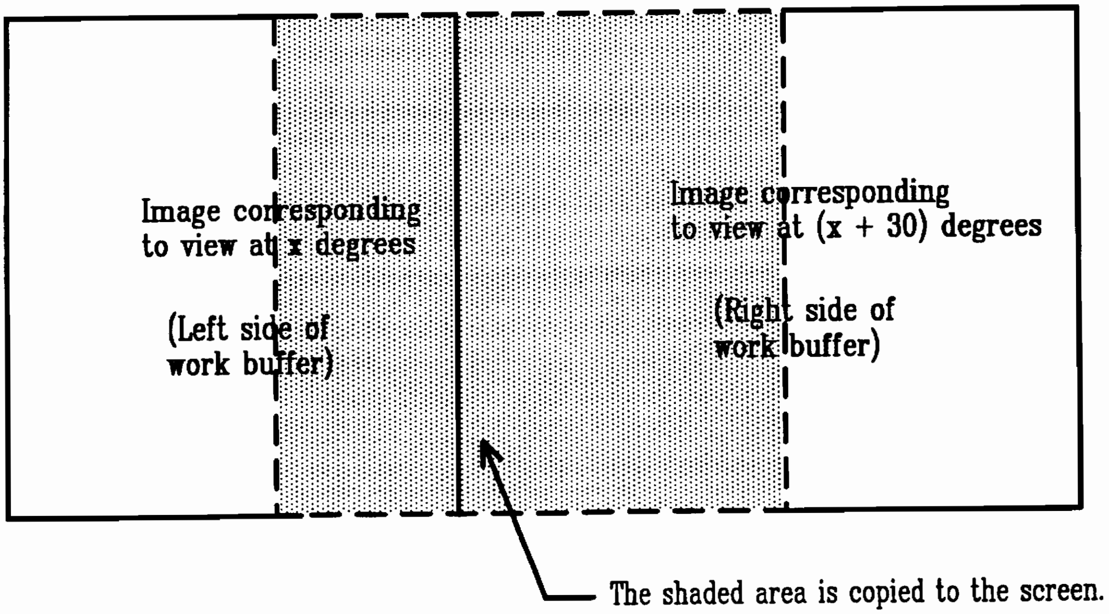


Figure 2.1: Layout of work buffer

is referred to as the off-screen buffer and is available for image loading or manipulation. The key concept of the pan function is to alternate between the two screen buffers and the loading of an image into the buffer that is not currently being displayed. A portion of the work buffer can be displayed on the DVI screen by copying it into the off-screen buffer and then switching the 82750 DA to the off-screen buffer, thus making it on-screen. Thus, the screen can be thought of as a glass window that slides on the work buffer. The screen will display the portion of the work buffer that is copied into it (see Figure 2.1).

The student can pan left by pressing the left arrow key or pan right by pressing the right arrow key. Pressing the spacebar while the pan is in progress (i.e., the old image is sliding out and the new image is coming in), will suspend the pan temporarily. This feature is very useful in a situation where the student wants to locate a landmark on the orienteering map when she sees it on the DVI screen. Thereafter, the pan can be resumed in the same direction or in the opposite direction by pressing the appropriate arrow key. The pan can also be resumed in the same direction by pressing the spacebar again.

Let us assume that at a given point, the left side of the work buffer contains the image corresponding to cell number 62, 90^o azimuth, and that the right side contains the image corresponding to cell number 62, 120^o azimuth. A pointer, denoted by offset, indicates the starting position in the work buffer from which a portion as wide as the viewport has been copied onto the DVI screen.

Given a user input of left arrow, one of the following three cases will apply :

- Case 1 : The image displayed on the screen is that of 120^o azimuth. In other words, offset now points to the place in the work buffer where the 90^o image ends and the 120^o image begins. Since the student wants the image to the left of the currently displayed image to be shown to her, the 90^o image has to slide onto the viewport from the left side, "pushing" the 120^o image all the way to the right until it is out of the viewport. To accomplish this, offset is decreased by a constant amount, chosen to be 20 pixels for a viewport size of 333 × 300. This translates to an "animation rate" of about 20 frames per second. Next, a portion of the work buffer as wide as the viewport is copied into the off-screen buffer. This buffer is then displayed, making this the on-screen buffer and the other off-screen. This step is repeated until offset becomes zero. At this time, the whole image corresponding to 90^o azimuth is displayed on the screen. The effect to the student is one of turning her head slightly to the left and seeing the view of 90^o azimuth (see Figure 2.2).

- Case 2 : The image displayed on the screen contains part of the 90^o image and part of the 120^o image. This resulted from the student pressing the spacebar key during a previous pan, to suspend the pan temporarily. This case is also handled like Case 1 above since this can be considered as an intermediate stage in the left pan on Case 1.

- Case 3 : The image displayed on the screen is that of 90^o azimuth. In other words, offset now points to the left end of the work buffer. Since the student wants the image to the left of the currently displayed image to be shown to her, the image corresponding to 60^o azimuth has to slide on to the viewport from the left side, "pushing" the 90^o

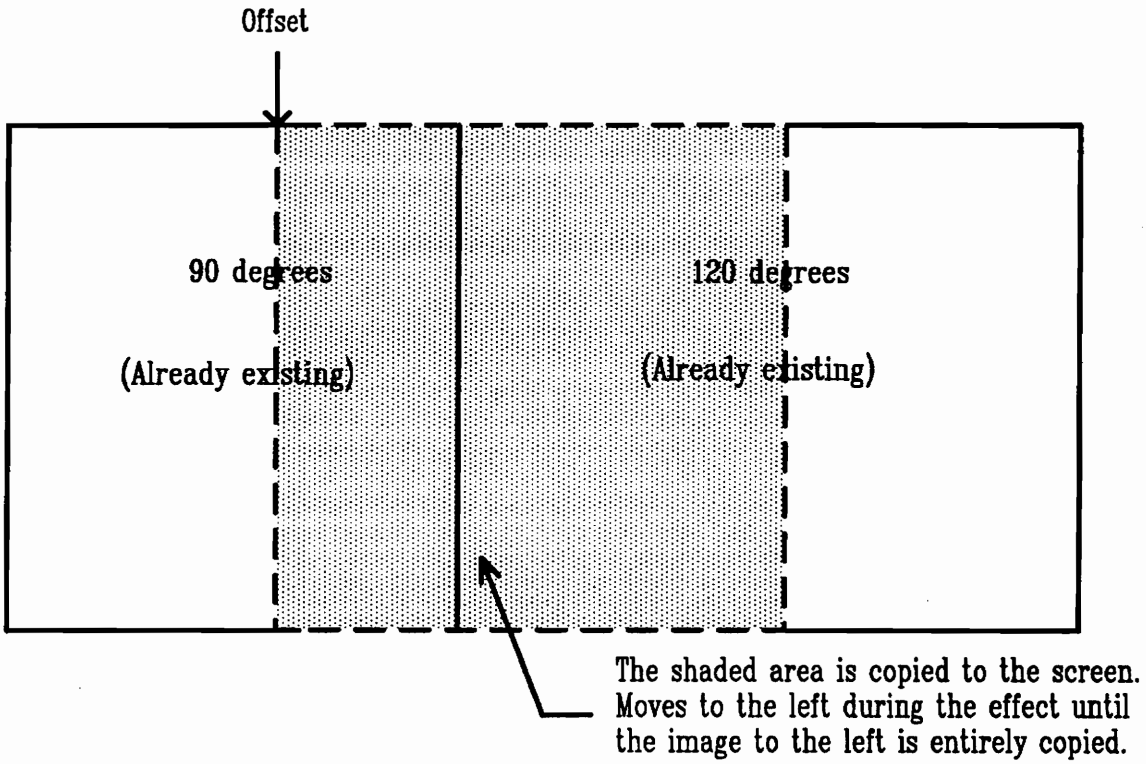
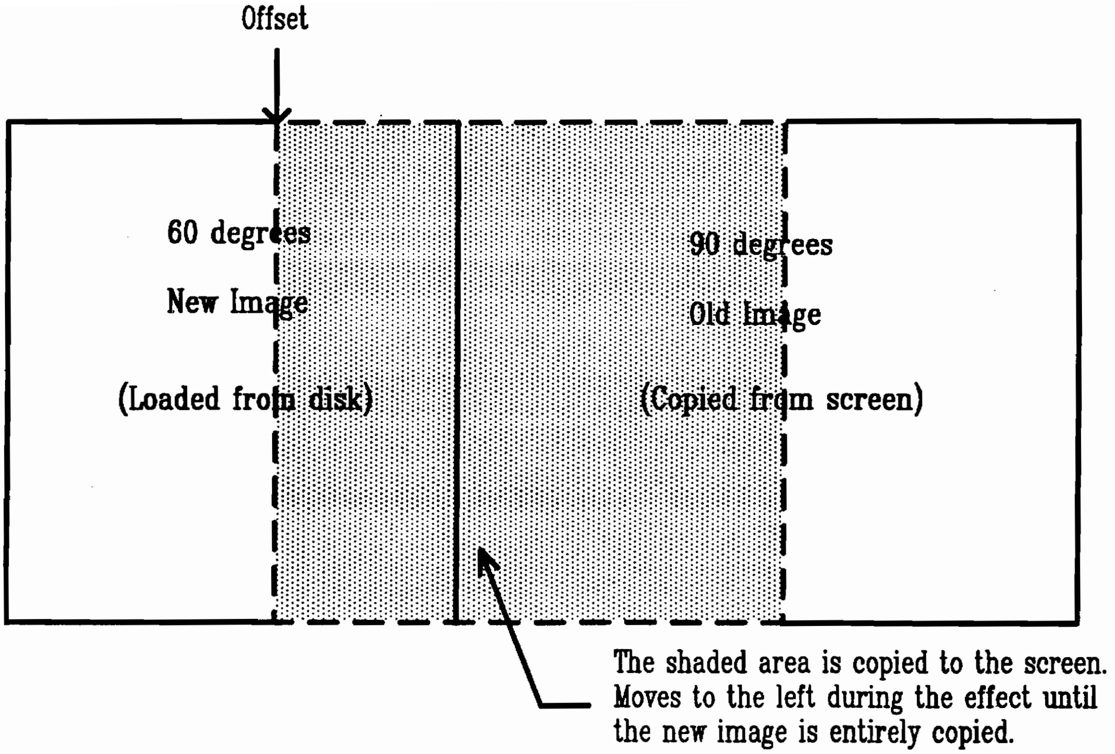

Offset
90 degrees
(Already existing)
120 degrees
(Already existing)
The shaded area is copied to the screen.
Moves to the left during the effect until
the image to the left is entirely copied.

Figure 2.2: Left Pan for Case 1 and Case 2



Offset
60 degrees
New Image
(Loaded from disk)
90 degrees
Old Image
(Copied from screen)
The shaded area is copied to the screen. Moves to the left during the effect until the new image is entirely copied.

Figure 2.3: Left Pan for Case 3

image to the right until it is out of the viewport. The steps below are followed:

1. The image on the screen (90^o) is copied onto the right side of the work buffer.
2. The new image file (60^o) is decompressed in an off-screen buffer, cropped, scaled to the size of the viewport, and loaded onto the left side of the work buffer.
3. Offset is now reset to point to the place where the 60^o image ends and the 90^o image begins.
4. Now, offset is decreased by a constant amount, chosen to be 20 for a viewport size of 333 × 300. Next, a portion of the work buffer as wide as the viewport is copied into the off-screen buffer. This buffer is then displayed, making this the on-screen buffer and the other off-screen. This step is repeated until offset becomes zero. At this time, the whole image corresponding to 60^o azimuth is displayed on the screen. The effect to the student is one of turning her head slightly to the left and seeing the view of 60^o azimuth (see Figure 2.3).

The above discussion describes the operations done during left pan, that is, turning in the counter-clockwise direction. Given a user input of right arrow, a right pan simulating turning in the clockwise direction, is initiated by the system. The implementation of the right pan is very similar to that of the left pan.

2.3.1 Updating the Compass

Since the image presented to the student keeps changing while left or right pan is in progress, the needle of the compass is also redrawn in red (to indicate that forward motion is not allowed) every time offset is updated and the image "shifts". Once the new image is entirely displayed on the screen, the angle of the new image is checked to see if forward motion is allowed from that direction. Forward motion is allowed for $0^o, 60^o, 120^o, 180^o, 240^o$, and 300^o. The needle of the compass turns green if and only if the new image corresponds to one these angles.

2.4 Zoom Function

If the student wishes to move forward in the current direction, the current image has to be manipulated in some way to simulate forward motion before the corresponding image in the cell ahead is displayed. The key concept employed in the zoom module is that objects appear bigger as one walks towards them and objects in the peripheral vision drop out of sight as one walks by them. The function described in this section simulates these effects.

2.4.1 Scenario

Let the image on the screen correspond to cell number 62 at 60^o azimuth. If the student presses the up arrow to move forward, the student would move into cell number 61 and would be looking at the image corresponding to 60^o azimuth. The student should then be able to move forward again, pan left, or pan right.

2.4.2 Zoom Effect

The effect of objects becoming bigger and peripheral objects dropping out of sight is achieved by scaling increasingly smaller portions of the current image to the size of the viewport, with the center of the zoom effect being selected so as to blend as well as possible with the corresponding image in the cell ahead. For example, let the width of the viewport be V_WIDTH, the height of the viewport be V_HEIGHT, and the zoom rectangle specifications read from the zoom database (see Chapter 3) be (x_1, y_1, x_2, y_2). The rectangle specifications are then scaled if necessary. Let the new specification be (x_{s1}, y_{s1}, x_{s2}, y_{s2}) Then, the number of pixels to discard at each zoom step in order to scale is computed using the number of scales, NUM_OF_SCALES. The higher the value of NUM_OF_SCALES, the longer it takes for the zoom operation to complete; the lower the value of NUM_OF_SCALES, the bigger seems each "jump" from the current image to the image in the cell ahead. The optimum value was found to be 3. Let *left-chop, right-chop, top-chop*, and *bottom-chop* denote the number of pixels to discard each time from the left, right, top, and bottom borders respectively. The zoom effect is achieved by executing the following loop :

```
for i = 1 to NUM_OF_SCALES
- take the smaller image from the work buffer
  obtained by chopping off
      (i*left-chop)   pixels from the left border
      (i*right-chop)  pixels from the right border
      (i*top-chop)    pixels from the top border
      (i*bottom-chop) pixels from the bottom border
- scale it to size V_WIDTH x V_HEIGHT
- display the scaled image on the screen
end for;
```

This gives the effect of walking forward. After this process is complete, the new image is displayed. The new image (cell number 61 at 60^o azimuth) is decompressed, scaled, and copied onto the left end of the work buffer. An *iris transition effect* is used to display the new image on top of the existing image. This transition opens up the new image from the center and is called an iris effect because it is similar to opening up a camera lens iris, although in this case the iris is rectangular [Luth91].

The use of the iris transition serves two purposes : the final scaled-up image seen before the new image is displayed and the new image itself are very similar but may not match perfectly. The discrepancies, however minor, may be easily noticeable since the final scaled-up image remains on the screen for about 1 second while the new image is cropped, decompressed, and loaded. The iris effect "softens" the jump to the new image. Secondly, it also gives a signal to the user that the zoom operation is complete.

After the iris transition is complete, the image corresponding to the view to the right of the new image (that is, cell number 61 at 90^o azimuth) is decompressed, scaled, and loaded into the right side of the work buffer. This image is loaded so that the system is ready for a subsequent pan or forward motion.

Chapter 3

THE DATABASE

The Orientation module requires data to crop the images before displaying them. The videodisc prototype (see Section 1.3) developed at Virginia Tech used images that had some overlap with images of adjacent views. In other words, if a house was seen on the extreme right of the image corresponding to cell number 230 at 90^o azimuth, then the same house was also seen on the extreme left of the image corresponding to 120^o azimuth. An estimated 5% overlap typically exists between adjacent images. This was useful to the students using the videodisc prototype in establishing 'context' when a new image is simply displayed on the screen, replacing the old image.

Since the DVI implementation of the Orienteering prototype takes advantage of the image manipulation routines offered by DVI Technology, a virtually seamless pan is not difficult to implement. However, the overlap between adjacent images has to be eliminated. So, each image has to be cropped by an amount that depends on the amount of overlap with its adjacent image. It was decided that each image would be cropped on the right side, so that if a cropped 90^o image is placed to the left of an uncropped 120^o image, the combined image would be seamless. Unfortunately, the crop amount is not constant for all images. So, it is necessary to collect crop data for every image available on the videodisc and use these data in cropping an image before loading it into the work buffer.

Zoom data for every cell is also required to render a realistic effect of forward motion. For a given cell, we store the set of cell numbers for its six neighbors and a rectangle specification for each neighboring cell, called the zoom rectangle. The zoom rectangle is totally contained within the current image. It represents that specific portion of the current image which is visually most similar to the image that will be displayed on the screen upon zooming. Since the slope and the features widely vary across images, this rectangle specification will not be constant for all images. For example, if the current image consists only of very distant objects, the rectangle will encompass almost all of the current image because there will be no perceptible difference by zooming forward. On the other hand, if the current image has a "close" object, the rectangle will be considerably smaller than the current image because the object may have to partially or completely drop out of sight during and after the zoom operation. So it is necessary to collect the rectangle specification data for every image at $0^o, 60^o, 120^o, 180^o, 240^o$, and 300^o azimuth (forward motion is allowed only in these directions).

The zoom effect is produced by successively scaling smaller concentric rectangles from the image onto the viewport until the zoom rectangle takes up the entire viewport. The new image is then displayed using the iris effect.

3.1 Image Capture

The Stevers Gap Orienteering Terrain Videodisc contains over 4600 still images. This section describes the process of capturing a still image from the videodisc into a file on the hard disk. This procedure was adopted to capture the still images for the Orienteering module.

3.1.1 Connections

The cable from the video output of the videodisc player is connected to a Composite Video Adapter (CVA). The CVA itself is connected to the red connector on the capture cable of the DVI system [Capt91].

3.1.2 Capture of Uncompressed Image

1. The videodisc is loaded in the player.

2. The DVI still image capture utility *VCapt* is executed.

3. The frame number of the required image is “search”ed for and displayed on the DVI screen.

4. The color, contrast, tint, and brightness of the image are adjusted by using the menu displayed on the system screen.

5. The image is then captured in 512 × 480, 16-bpp format. It is then saved to a file on the disk [Prod91]. The filename is of the form

   ```
   <cell number>_<angle of view>.<format>
   ```

   For example, 62_120.i16 is a valid name.

3.1.3 Image Compression

The DVI Image Conversion Utility *VImCvt* is executed in order to convert the uncompressed image obtained above (i16 format) to a compressed image (c16 format) [Prod91].

3.2 Input Files

In addition to the images, the Orienteering module prototype presently requires three input files. All three input files contain plain ASCII text and are read by using the general purpose database reader which is used by all modules of project GeoSim. This general reader scans the data file for keywords and extracts the data associated with these keywords for use in the module. The advantage of the general reader is that it allows tabulated, documented, human readable, general file format to be converted to machine readable format [Kris91].

3.2.1 Configuration File

The file *config.gcf* contains the *viewheight*, *viewxorig*, and *viewyorig* key words. The data available along with these key words control the size and placement of the viewport. The file also has the *startcellno* and *startangle* key words which determine the cell number and the initial angle to begin the session with. In addition, it also has the *vim* key word which has associated with it the full pathname of the directory where all the image files are located on the hard disk. The data in the configuration file can be modified to change the look of the simulation.

3.2.2 Crop Data File

The file *crop.gaf* contains one record for each cell. Crop data was determined for a small subset of the 385 cells in the orienteering terrain. Each record consists of the cell number, followed by the crop widths for each of the twelve views in the following order : $0^o, 30^o, 60^o, 90^o, 120^o, 150^o, 180^o, 210^o, 240^o, 270^o, 300^o$, and 330^o. Non-availability of crop data for a particular view is indicated by a NO_DATA flag.

The program *cropdata* is used to determine crop data for all twelve views associated with one particular cell. The interface of this program and the manner in which it displays the images is similar to the pan function described in Section 2.3. This facilitates the collection of crop data that provides close-to-realistic pan effect.

Initially, the input file *cropdata.inp* contains the cell number followed by twelve 493's, since the width of an uncropped image is 493 (excluding the grey border around the actual image). The first execution of the program shows the images with the overlap. By panning across all the images and visually examining the seam between each pair of adjacent images, the amount of overlap can be estimated. This overlap has to be deducted from the existing value for that particular view in the input file. The process is then repeated to successively refine the values until a virtually seamless pan for that cell is achieved. The crop values from the input file are then manually transfered to *crop.gaf*, the crop data file used by the Orienteering module.

3.2.3 Zoom Data File

The file *zoom.gaf* contains the zoom rectangles for the $0^o, 60^o, 120^o, 180^o, 240^o$, and 300^o azimuth images for a small subset of the 385 locations. In addition, it also contains the cell numbers that can be arrived at if the student opts to move forward in one of the six valid directions. Non-availability of rectangle specification data for a particular direction is indicated by a NO_DATA flag in the destination cell number.

The program *zoomdata* is used to determine, for each cell, the zoom rectangles for the six images from which forward motion is valid. This program is similar to the zoom function described in Section 2.4. The difference is that this program has a hot key which shows the entire zoom operation again. Therefore, during a single execution of the program, one can repeatedly view the zoom operation and easily get better estimates of the zoom rectangles.

The input file *zoomdata.inp* contains the cell number and the angle of view v for which zoom rectangle has to be determined. This is followed by zoom data for each of the six valid directions. This data consists of the cell number of the new cell zoomed into and the zoom rectangle specification of the image corresponding to angle v.

First, the DVI Still Image Viewing Utility *Vshow* is used to alternately view the current image and the new image to be zoomed into (in the direction of angle v). A rough estimate of the zoom rectangle is made by visual inspection. The coordinates of the top left and the bottom right corners are then manually entered into the file *zoomdata.inp.* This is done for each of the six valid directions.

Using this input file, the program *zoomdata* is executed to see if the zoom effect is realistic. The hot key described above is used in this process. Any refinement to the zoom rectangle is done by modifying *zoomdata.inp* and iteratively running *zoomdata* until satisfactory values are obtained.

The above process is done for each value of v which is a valid direction for forward motion. The zoom data from the input file are then manually transfered to *zoom.gaf*, the zoom data file used by the Orienteering module.

Chapter 4

CONCLUSIONS AND FUTURE WORK

The Orienteering module prototype provides the basic routines to provide a seamless panoramic view at any location, and to provide simulated motion in valid directions, using the image database from [Cars89]. The prototype also provides the student with a compass on the screen which helps her in finding her way through the simulated course.

The prototype uses the decompression routines available in DVI Technology to minimize storage space and to improve access speed. The size and location of the view window which shows the images on the DVI screen can be varied. This prototype employs the general reader, which is common to all modules of project GeoSim. The system logs all error messages into a log file that can be viewed after the session is completed.

This simulation has the advantage of being available in all weather conditions. Also, it eliminates physical differences among students so results are based strictly upon mental decision-making processes.

The complete process of cropping, decompressing, scaling, and loading an image takes about 1 second. The Pan and the Zoom operations take about 3 seconds each. The use of compressed images cut down response time by about 1 second, with no discernible change in the quality of the images.

Programs to collect crop data and zoom data were developed and used to determine data for a small subset of the cells in the terrain. The interface of these programs and the manner in which they display the images is very similar to the actual simulation. This facilitates the collection of data that renders close-to-realistic effects.

4.1 Future Developments

The Orienteering module is currently in the "proof of concept" phase. Possible enhancements to the prototype could include the display of an altimeter along with the compass, which would indicate the elevation of the current location. Marker routines for the placement and manipulation of markers and feedback routines which keep track of the path and time taken by the student could also be developed. The simulated drive to the orienteering site, as available in the videodisc prototype can also be incorporated in the session to provide some practice in position finding to the student.

The use of compressed images in the 9-bpp format as opposed to the current compressed 16-bpp format may result in less storage space and faster response. Image files are currently stored on the hard disk. Future versions could directly read images from a CD-ROM.

The new version of DVI Software Libraries for MS-Windows can be used to enhance the interface. Mouse input can also be supported, in addition to keyboard input. On-line help can be included.

To facilitate the collection of data for the Pan and Zoom modules, an image editor can be designed using the DVI Software Libraries. This would support mouse input and will use the x and y coordinates obtained from the click of the mouse to calculate the data associated with each image.

REFERENCES

[Capt91] *Intel ActionMedia II Capture Module Installation Guide*, Intel Corporation, Santa Clara CA, 1991.

[Cars89] L.W. Carstensen, and A.B.Cox, "Videodiscs and Surrogate Travel / The Map-Environment Interface", *Cartographica*, Vol. 26 Nos 3 & 4, 1989.

[Cars93] L.W. Carstensen, C.A. Shaffer, R.W. Morrill, and E.A. Fox, **GeoSim**: A GIS Based Simulation Laboratory for Introductory Geography, To appear in *Journal of Geography.*

[Inte91] *Intel DVI Technology ActionMedia 750 Software Library Overview*, Intel Corporation, Santa Clara CA, 1991.

[Kern88] B.W. Kernighan, and D.M. Ritchie, *The C Programming Language*, Prentice-Hall Inc., Englewood Cliffs NJ, 1988.

[Kjel76] B. Kjelstrom, *Be Expert With Map and Compass, The Complete Orienteering Handbook*, Charles Scribner's Sons, New York NY, 1976.

[Kris91] S. Kriss, "Computerized Simulations for Introductory Geography Instruction : International Population", Masters Project, Department of Computer Science, Virginia Tech, Blacksburg VA, 1991.

[Luth91] A.C. Luther, *Digital Video in the PC Environment*, McGraw-Hill, New York NY, 1991.

[Prod91] *Intel ActionMedia 750 Production Tool Reference*, Intel Corporation, Santa Clara CA, 1991.

[Shaf90] C.A. Shaffer, V. Miranda, J.C. Wang, S.K. Kriss, L.W. Carstensen, R.W. Morril, and E.A. Fox, "Computerized Simulations for Introductory Geography Instruction", *Proceedings of Applications of Geographic Information Systems, Simulation Models and Knowledge-based Systems for Landuse Management*, Blacksburg VA, November 1990, 479-480.

[Smit82] R. Smith, *The Penguin Book of Orienteering*, Penguin Books, New York NY, 1982.