Introduction

Overview

This introduction provides an overview of NCSA Height-Color Visualizer. The organization and use of this manual are described and notational conventions explained.

About NCSA Height-Color Visualizer

NCSA has developed Height-Color Visualizer as a non-interactive batch utility to enable users to easily create 3D animations from time-dependent data.

Height-Color Visualizer, a 3D visualization tool, allows users to view bivariate data in a simple manner. Input data consists of two 2D arrays each having a single variable at every node. These arrays can be image data, floating-point data, or any combination thereof. Using the Height-Color Visualizer, you manipulate the first of the two arrays to create the height geometry. The indices of this array form the X-Y coordinates, and the variable at each node forms the Z coordinate in a cartesian coordinate system. The second of the two arrays is assigned a color at every node through one of many available visualization methods. This resultant color field is draped over the height geometry. Both fields are assumed to have the same input dimensions establishing a unique one-to-one correlation of the respective data values in each of the two data sets. Thus, the analysis of the correlation of data becomes as easy as associating regions of varying height with coincident regions of color.

Special Features

The features of NCSA Height-Color Visualizer 1.1 allow you to:

- control orientation of geometry including independent XYZ translation, rotation, and scaling
- control light source parameters including arbitrary XYZ location of light source and RGB color of light source
- specify material properties including ambient, diffuse, and specular coefficients and specular high-light size which allows control over how the light source interacts with the viewed object
- pre-define convenient, alternative defaults
- choose from multiple illumination models

- choose color interpolation methods
- specify axes and axes labels
- choose wireframe, polygon, and relief representations of height-color visualization
- save hardware rendered images to disk for 3D rendered animation sequences (including full hidden surface removal)
- choose full support of HDF 8-bit and 24-bit raster image formats, HDF 2D scientific data format, and any combination thereof

System Requirements

To run Height-Color Visualizer, you will need a Silicon Graphics™ 4D series workstation with at least 24 bits of color, hardware Z-buffer, 8M RAM or more, and version 4D1-3.1 OS or newer. A very limited subset of the functionality of Height-Color Visualizer will run on systems with less than 24 bits of color, but this is not recommended.

Use of This Manual

Organization of This Manual

This manual is organized into three chapters. Each page in a chapter is given a unique number that consists of the chapter number, a period, and the number of that individual page (beginning with the first page of the chapter). For example, page 2.3 is the third page of the second chapter. Each chapter is divided into sections, and most sections are divided into subsections.

Manual Contents

This manual is organized into the following chapters:

Chapter 1, “Getting Started,” provides a tutorial and describes Height-Color Visualizer and its functionality. A detailed description of all command line parameters is included in this chapter.

Chapter 2, “Creating Animations,” offers suggestions and examples in setting up batch jobs to create 3D rendered animations from time-dependent and time-independent data.

Chapter 3, “Technical Notes,” provides users with an understanding of the geometric structures and techniques used to transform their input data into a 3D height-color visualization.

Form of Presentation

The material in this manual is presented in text, program lists, screen displays, command line parameters, and entry format notation.

Text

Portions of this manual refer to other sections or chapters of the manual where related topics are discussed. These cross references usually indicate the title of sections or chapters enclosed in quotation marks, such as, See Chapter 1, “Getting Started.”

Program Lists and Screen Displays

Program lists and screen displays in this manual are presented in `courier type`.

Command Line Parameters

Command line parameters requiring arguments are denoted as the parameter followed by the argument list enclosed in <> with a space separating each argument. For example, -r <rx ry rz> indicates that the -r parameter requires three arguments: rx,ry,rz.

Command line parameters which do not require any arguments are denoted as the parameter followed by its description enclosed in (). For example, -p (print current settings) indicates that the -p parameter does not require any arguments and is used as a flag to print the current settings.

Entry Format Notation

Throughout this manual, several explanations instruct you to make entries by typing on the keyboard. These entry instructions are printed in **`courier bold type`** and appear within a screen display, paragraph, or on a separate line.