IZIC: a Portable Language-Driven Tool for Mathematical Surfaces Visualization

Robert Fournier Norbert Kajler Bernard Mourrain
SEMIR SAFIR *

2004 routes des Lucioles
06565 Sophia-Antipolis Cedex, France

Abstract
This paper presents IZIC, a stand-alone high-quality 3D graphic tool driven by a command language. IZIC is an interactive version of ZICLIB, a 3D graphic library allowing efficient curve and surface manipulations using a virtual graphic device. Capabilities of ZICLIB include management of pseudo or true colors, illumination model, shading, transparency, etc. As an interactive tool, IZIC is run as a Unix server which can be driven from one or more Computer Algebra Systems, including Maple, Mathematica, and Ulysse, or through an integrated user interface such as CAS/PI. Connecting IZIC with a different system is a very simple task which can be achieved at run-time and require no compilation. Also important is the possibility to drive IZIC both through its freely-reconfigurable menus-buttons user interface, and through its command language, allowing for instance the animation of surfaces in a very flexible way.

Keywords: Computer Algebra, meshed surfaces, graphic tool, rendering, TCL.

1 Introduction

Most of the surface plotting tools used with Computer Algebra Systems are those provided with Axiom, Maple, Mathematica, etc. Being part of commercial products these plotters are neither available separately nor callable from other systems. Also, they do not include important features such as shadowing rendering, transparency, or surfaces composition. In fact, few attempts were made in the Computer Algebra community to develop powerful portable plotting tools. One of the most notable exception appears to be SIG [Wan90], a curve and surface plotting engine under X11 developped in 1990 by Wang at Kent State University. As a surface plotter, SIG is fairly competitive with respect to other plotting engines available at that time, allowing for instance hidden-line removal on surface mesh displays. As a portable tool, SIG include two parts: a system-independent visualization engine called Xgraph and a system-dependent set of

*SAFIR is a common project to INRIA, Université de Nice-Sophia Antipolis, and CNRS.
functions, called mgraph, which performs computations of data needed by Xgraph from a given Computer Algebra System. The communication between Xgraph and mgraph uses a file to transfer data and a minimal protocol to initiate displays. Clearly, Wang advocates in [Wan90] that more work is needed to improve communications between these two components.

On the other hand, many plotting engines were developed outside the Computer Algebra community. Most of these tools are either not powerful enough to allow quality display of mathematical surfaces, or necessitate the use of optimized hardware for 3-D manipulations. This includes respectively freely available tools like Gnuplot or commercial applications such as Explorer.

In this paper, we will present IZIC, a graphic server, that can be driven from outside and is based on the C-library ZICLIB. Commands scripts can be sent to this tool from one or more Computer Algebra Systems, or any other scientific tools, in order to display the geometrical objects and act on their representations. Conversely to SIG, IZIC can be dynamically connected to one or more Computer Algebra Systems and offer the power of a real command language to drive the surface display and/or customize its menu-button user interface.

The first part of this paper is related to the library and algorithms used in order to obtain nice and efficient rendering of surfaces. The second part deals with the incorporating of this code in an interpreter which can be driven from outside, including an easily configurable menu-button user interface. The third part presents the way geometrical objects are manipulated formally in a Computer Algebra System and how to link this one with our graphic server.

2 ZICLIB

ZICLIB [Fou92] is a software graphic library written in C by RObert Fournier at INRIA Sophia-Antipolis. By using only memory allocation, ZICLIB can run on any Unix system and requires no specific hardware. All existing applications based on ZICLIB, such as ZICVIS, ZVIS, and IZIC, also use standard Xlib drawing routines, a common simple language defining input data format, and various X11 toolkits, respectively: Athena, OSF/Motif, and TK.

The basic data structure of ZICLIB is a main memory workspace named zicimage, which is used as an abstract device to send graphic requests. This resource will be referenced as a Virtual Graphic Device in this paper.

Each Virtual Graphic Device has its own characteristics, including the size of a screen in pixels, its depth, and the availability of a software Z-buffer. Three predefined depths are available: 1 for monochrome, 8 for pseudo colors and 24 for true colors. A projection type - perspective or parallel - and its definition is also associated to a Virtual Graphic Device. Another characteristic is the definition of an illumination model [FVDFH90], including intensity of an ambient light and a light sources list. All the characteristics of a Virtual Graphic Device can be given at creation time or modified dynamically.

The three dimensional graphic elements which are the bricks of constructions, are curves and surfaces. Curves are defined by sequence of points, and surfaces are defined by structured or unstructured meshes. Graphic requests are available to draw these graphic elements "onto" a Virtual Graphic Device. Including text is also possible.