Reconstructing free form 3D objects from 2D images with PEX
by Xia You

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science
Montana State University
© Copyright by Xia You (1994)

Abstract:
This thesis develops a software which reconstructs free form 3D objects from 2D images based on Motif/Pex. It allows the users to edit, transform and render random shape 3D objects interactively through a friendly and consistent Motif style Graphic User Interface.

This software displays a series of 2D images and also provides several image processing methods which make it possible to enhance the quality of the images. 3D bodies are reconstructed with B-splines after the user defines a three-dimensional polygon structure data.

As the result, 3D human bodies were successfully reconstructed from medical MRI images. It is also applicable for other purposes 3D free form bodies rendering.

RECONSTRUCTING FREE FORM 3D OBJECTS

FROM 2D IMAGES WITH PEX

by

Xia You

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY
Bozeman, Montana

November 1994

ii

APPROVAL

of a thesis submitted by

Xia You

This thesis has been read by each member of the
thesis committee and has been found to be satisfactory
regarding content, English usage, format, citations,
bibliographic style, and consistency, and is ready for
submission to the College of Graduate Studies.

11/4/94	_J Denbigh Starkey_
Date	Chairperson, Graduate Committee

Approved for the Major Department

11/4/94	_J Denbigh Starkey_
Date	Head Major Department

Approved for the College of Graduate Studies

11/8/94	
Date	Graduate Dean

STATEMENT OF PERMISSION TO USE

_____

Signature

Nov 9, 1994

Date

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis develops a software which reconstructs free form 3D objects from 2D images based on Motif/Pex. It allows the users to edit, transform and render random shape 3D objects interactively through a friendly and consistent Motif style Graphic User Interface.

This software displays a series of 2D images and also provides several image processing methods which make it possible to enhance the quality of the images. 3D bodies are reconstructed with B-splines after the user defines a three-dimensional polygon structure data.

As the result, 3D human bodies were successfully reconstructed from medical MRI images. It is also applicable for other purposes 3D free form bodies rendering.

CHAPTER 1

INTRODUCTION

In recent years, with the big progress in computer hardware and software, especially the availability of low-cost visualization-compatible workstations, networks, and powerful graphics software, Computer visualization has become more and more popular. On the other hand, the rapid growth of large-scale computing in the basic sciences and the steady accumulation of high-bandwidth data sources (radio telescopes, medical scanners, etc.) has also increased the interest in scientific visualization as a computational technology. Scientific Visualization has become a tool for discovery and understanding, a tool for communication and teaching. Mathematicians want their equations to be visualized; engineers want their design to be visualized in order to make the designing cheaper and faster; In a word, visualization has become so popular that more and more scientists, engineers, physicians are involved in this new area.

There is a project called Boron Neutron Capture Therapy (BNCT), which is an international development aimed at curing certain cancers. Idaho National Engineering Laboratories (INEL), the University of Utah, and Montana State University have been responsible for the development of the Computer Software for this project. There is an existing software system called Bnct_edit. This software first shows a two dimensional medical image, then it can display a three dimensional representation of a human head, brain, and other objects in the head, to assist in the treatment and experimental planning of radiation therapies by creating accurate patient geometries.

This software was first developed for the Apollo environment using GPR at the University of Utah; then it was ported to X-Windows and significantly enhanced by INEL and Montana State University. In this thesis, I attempted to adapt the methods used in the BNCT project to the PEXlib/Motif environment. PEXlib is a programming library for 3D graphics. It's the lowest-level and most direct interface for drawing 3D pictures in the X Window System. Widely and freely available, efficient, and supported by many workstations and terminal manufacturers, it is a popular choice for the graphics

interface of 3D applications. PEX is the 3D extension to X. It adds over 200 X protocol requests for defining and displaying 3D pictures. PEX provides all the common features found in most modern 3D graphics systems, but provides them in a way that's seamlessly integrated with X. PEX is a high-level graphics library. It allows a programmer to describe a graphic image in terms of familiar objects and attributes, without having to deal with the details of producing that image in windows. All the details of producing the picture are handled by the PEX server. The user merely specifies the geometry, the location, and some appearance attributes for the objects. PEXlib and Motif have just worked together for a few months at the Department of Computer Science at Montana State University, Bozeman. The software is called "Show-mri". As the name indicates, it manipulates medical images, then reconstructs 3D bodies of patients. It is also applicable for rendering other 3D free form bodies without major changes. I will use medical images to describe how this program works through this thesis. After executing the program, it will show some medical images with a unique Z-value for each image. Users can then edit B-spline contours interactively by clicking the button of the mouse through a friendly motif-style

Graphics User Interface (GUI). Finally, 3D bodies will be reconstructed and displayed by interpolating those B-splines. Users can observe the 3D bodies in different orientations, and can change the color of each body.

The system was developed on a DEC alpha machine with the OSF/1 operating system. A PEX terminal, which has 4M bytes of code memory and 8M bytes of data memory, was used for the program development. Because of the high portability of the Unix program, this software can easily port to other PEX platforms such as HP and SUN workstations.

There are three major parts to this thesis: the first part is Motif-style Graphics User Interface, which will be presented in chapter 2; the second part is Graphics which includes B-spline drawing, picking, 3D rendering, etc, some of which were done by Pexlib, some by Xlib. In chapter 4, I will describe the details; the last part is image processing part including some image-processing utilities which are presented in chapter 4. This thesis concludes with chapter 5.

CHAPTER 2

THE GRAPHIC USER INTERFACE OF SHOW_MRI

In this chapter, I will show the Graphic User
Interface of this package. I will not only present the
appearance of the Graphic User Interface of this program,
but also will discuss some detailed information about how
it was implemented with Xlib/Motif.

"Show_mri" was designed through a consistent Motif-
style Graphic User Interface. Motif has become an
industry standard for displaying windows with text and
graphics. Now after years of proprietary window
interfaces, applications and their users can count on a
consistent interface across almost all producers and
models of computers. It is no longer necessary to produce
separate versions of an application to run on different
proprietary window systems, and it is no longer necessary
for users to learn a different interface to an
application simply because it's running on a different
machine.

The Main window of "show-mri" system, shown in
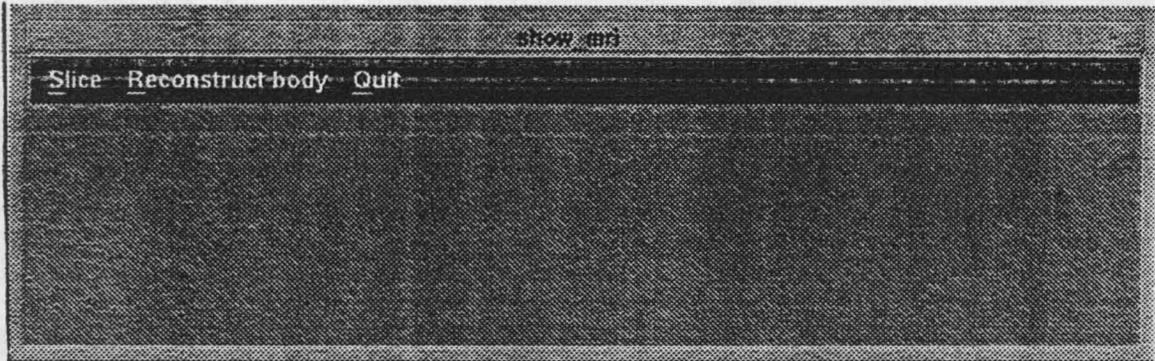**figure 1**, is the only window after the software is

executed.

Figure 1. Main window of "Show_mri"

At the top of the window, there is a Title Bar with a Title Area at the center and two push buttons for Minimize and Maximize at the right corner. Also, there is a window menu push button at the left corner; by pushing that button users can get a pull down menu for some window control commands. Below title bar there is a menu bar which is designed by the application programmer, which are the functions that I have implemented in this program. The next area is the drawing area, which was used to display the histogram in this program.

The menu bar is the most important part in this window, and provides the functions that are needed to use the software. In designing this menu bar, I used a general function to generate these menus with the following data structure to specify each menu item.

Table 1. Record for general menus (struct _menu_item)

Type	Name	Notes
*char	label;	the label for the item
*WidgetClass	class;	pushbutton, label, separator..
char	mnemo	mnemonic; NULL if none
*char	accelerator;	accelerator; NULL if none
*char	accel_text;	to be converted to compound string
*void	(callback) ();	routine to call; NULL if none
XtPointer	callback_data;	client_data for callback()
*_menu_item	subitems	pullright menu items, if not NULL

By using this general function, new functions can be added quite easily, which gives this program a good flexibility for expansion.

The first item is "Slice." Users can use both mouse or "alt-S" to activate this function, which is used to generate a window to display images for operations.

The second item is "Reconstruct body" activated by a clicking of a button of the mouse or "alt-R," which is used to reconstruct bodies after drawing contours on the image windows.

The last item is "Quit" for quitting this program, activated by mouse or "alt-Q."

There may be several windows on the screen at the same time. Users can use the mouse to move around and focus on the desired window. Most jobs to operate this software can be done by using the mouse except on a few

occasions when you will need to use key stroking.

Generate Slices of Image

When the "Slice" button has been pushed, there will be a pulldown menu. Currently there is just one item: "Make New slice" on the menu. After that, a window of the image will be displayed on the screen after you choose a Z-value and an image-file name.

Figure 2 show the dialog-popup window for users to enter a Z-value, which is the Z value of the slice image normally within the range 0.0 to 1.0.
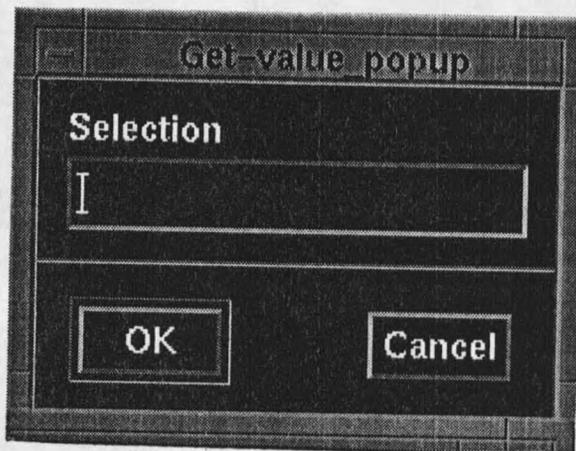

Figure 2. Dialog-popup Window for Entering the Z-value

Figure 3 is the dialog-popup window to choose the image file users want to display. Users can use this tool