Subject: Re: project_vol() in IDL
Posted by davidf on Wed, 25 Feb 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Grant W. Petty (gpetty@rain.atms.purdue.edu) writes:

> Today's my first day using IDL and I must say...

Sigh...

My next book is going to be entitled How To Do Really, Really Hard Things on Your First Day Using IDL. I'm going to use examples I have collected from this newsgroup. :-)

> However, I have already run into what seems to be an annoying
> limitation in project_vol() and am wondering whether I'm just
> overlooking something.

I don't as a rule try to give advice without also attempting to give some kind of an answer, but I've always believed rules were made to be broken.

Here is my advice. Sounds like you want to do some fairly complex 3D kinds of things. I would NOT be doing these in direct graphics, which is what Project_Vol and Voxel_Proj use. IDL in this incarnation is really a 2 and a half D application. As long as you are committed to learning something new, I would plunge in and try to learn how to do this using the new object graphics, which were designed *specifically* for this kind of 3D application.

This topic is NOT covered in my book because, frankly, I haven't been able to figure out exactly how it works from the documentation that is supplied with IDL. (I am currently scratching my head over Ken Bowman's polygon object problem. I can get *almost* there, but not quite, and I end up doing "experiential programming" in which I make random changes in my programs, hoping beyond hope that something will start to make sense. I hate it.)

What I *have* found is that the people at RSI are wonderfully helpful with giving me advice when I ask for it. And there are clearly people there who understand how this object graphics stuff works. I think if we all work together on this that sooner or later we will be able to get together some simple examples that demonstrate the principles. The fact that there are two object graphics questions here in two days certainly makes me think the time is right for *me*

to learn it better than I do now.

Just looking, for example, at the Volume Object I see that
it has four different "composite functions" for how to determine
the value of the projected pixel. One of these is an "alpha sum",
which sounds to me like what you want. There is also depth cueing
and various opacity and lighting options. I would pester the
folks at RSI until you learn how it works, and then publish a
good example here. :-)

The alternative is to look at the Project_Vol source code, which
is written in IDL and is available in the lib subdirectory, and
make the modifications to it yourself. This is probably not a
first-day-with-IDL kind of a job, but I didn't think the source
code looked impossibly hard, either.

> P.S. As long as I'm posting to this NG, I might as well mention one
> other suggested improvement to IDL that immediately comes to mind:
> command line history/editing/completion capabilities, a la the UNIX
> tcsh shell. It's a pain to have to retype an entire lengthy command
> from scratch when all I want to do is change one parameter!

Now here is a question I *DO* know how to answer. :-)

There should be no reason to re-type a lengthy command from
scratch. IDL has a command "history" buffer that is accessible
using the UP arrow key. Simply recalling the command and
editing it before hitting the Carriage Return will do the
job. The history buffer can be increased in size to more
than the default 20 commands if need be. (I was going to
give you a reference in my book, but I see that section
somehow got deleted in my overzealous editing. I'm going
to correct this in the next printing, even if it means
adding a couple of more pages.)

Sometimes the terminal window on UNIX machines has to be
configured properly to get command recall working. Let me
know if this is happening to you, and we can fix it.

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438