
Subject: Re: Object rendering with dynamic views
Posted by [davidf](#) on Tue, 23 Jan 2001 03:26:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ben Tupper (pemaquidriver@tidewater.net) writes:

- > I am wondering about the best way to manage objects graphics
- > when the display consists of changing and unchanging
- > models. I have done it three different ways in the past,
- > but I never considered the merits of each (until now.)

As anyone who has followed this newsgroup for any length of time knows, I've gained almost all my IDL expertise by doing things in the most godawful possible way first, and then later--MUCH later, usually--figuring out a better way to do it.

My only redeeming quality is that I tend to perceive failure as good information. :-)

(Well, except as it pertains to tennis, but that's another story.)

- > (1) The first way is that outlined in the online manual
- > (also David has a nice page about this.) This method
- > involves just one view. The view is first rendered with
- > the static portion of the graphics exposed and the dynamic
- > parts hidden using :
- > myDraw->DRAW, myView, /Create_Instance
- > Then the view is made transparent, the static portion is
- > hidden and the dynamic portion is exposed. After that the
- > view is drawn using:
- > myDraw->DRAW, myView, /Draw_Instance
- > Any subsequent changes in the dynamic atoms/models are
- > rendered using the DRAW_INSTANCE keyword.
- >
- > (2) The second way is to create two overlapping views, the
- > static underneath and the dynamic on top with the
- > TRANSPARENT keyword set. Then draw each view using the
- > CREATE_INSTANCE and DRAW_INSTANCE keywords as needed. This
- > is a method discussed (a long time ago) on the newsgroup,
- > but I can't find it documented anywhere.
- >
- > (3) Put all the atoms/models in one view and render the
- > whole thing as one. Grind-grind-grind. I use this method
- > when in a hurry to write code, but I really don't want to
- > look at it.

Alright, 99 percent of us do it the third way, too. But, really, who needs it?

In fact, I don't have enough expertise to judge between methods 1 and 2. I do it the first way. It works. End of story for me. :-)

I do know that in object graphics there are ALWAYS several ways to do something. Most of the time I understand one of the methods, and I just scratch my head at the other. Which method do you think I use? I've never **noticed** that one method is particularly better than another. Although occasionally one method will be significantly **faster** than another in real-world applications. I don't know if you would find such a difference in methods 1 and 2 above, but I doubt it.

I think, unfortunately, you just have to code it up and try it.

> Can someone explain the relative merits/pitfalls of each of
> the methods (in particular the first two?)

I'm sure **someone** can. I'm just not that person. :-)
I do think, however, that if your static scene is complicated to render, that creating an instance of it is almost guaranteed to be faster.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Object rendering with dynamic views
Posted by [Mark Hadfield](#) on Tue, 23 Jan 2001 03:53:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Ben Tupper" <pemaquidriver@tidewater.net> wrote in message
news:3A6CED48.98337741@tidewater.net...

> Hello,

>

> I am wondering about the best way to manage objects graphics

- > when the display consists of changing and unchanging
- > models. I have done it three different ways in the past,
- > but I never considered the merits of each (until now.)
- >
- > (1) The first way is that outlined in the online manual
- > (also David has a nice page about this.) This method
- > involves just one view. The view is first rendered with
- > the static portion of the graphics exposed and the dynamic
- > parts hidden using :
- > myDraw->DRAW, myView, /Create_Instance
- > Then the view is made transparent, the static portion is
- > hidden and the dynamic portion is exposed. After that the
- > view is drawn using:
- > myDraw->DRAW, myView, /Draw_Instance
- > Any subsequent changes in the dynamic atoms/models are
- > rendered using the DRAW_INSTANCE keyword.
- >
- > (2) The second way is to create two overlapping views, the
- > static underneath and the dynamic on top with the
- > TRANSPARENT keyword set. Then draw each view using the
- > CREATE_INSTANCE and DRAW_INSTANCE keywords as needed. This
- > is a method discussed (a long time ago) on the newsgroup,
- > but I can't find it documented anywhere.
- >
- > (3) Put all the atoms/models in one view and render the
- > whole thing as one. Grind-grind-grind. I use this method
- > when in a hurry to write code, but I really don't want to
- > look at it.
- >
- > Can someone explain the relative merits/pitfalls of each of
- > the methods (in particular the first two?)

No, but if you would like to compare them and report back to the group, that would be much appreciated :-)

I can only offer my \$0.02 worth:

When I last tried to compare these different methods I wasn't clever enough to think of method 1. (Obviously I should read the manual more carefully. Where is this described, anyway?) I tried method 2 because that seemed to be what was implied by the documentation for IDLgrWindow::Draw. Method 1 certainly seems to be more elegant than method 2, doesn't it?

But what I did find is that on my machine (a Wintel PC with ho-hum graphics hardware) method 2 was *much* slower than method 3. So I haven't tried "instancing" since. It has been suggested to me that the result might have been different had I been using the software renderer (RENDERER=1) instead of the hardware renderer.

I think this area deserves a careful examination. (Thanks for volunteering Ben.) I suspect you will find that performance is important, but that the ranking of the different methods will vary widely depending on OS, graphics hardware, IDL settings and day of the week.

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: Object rendering with dynamic views
Posted by [davidf](#) on Tue, 23 Jan 2001 04:20:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> I think this area deserves a careful examination. (Thanks for volunteering
> Ben.) I suspect you will find that performance is important, but that the
> ranking of the different methods will vary widely depending on OS, graphics
> hardware, IDL settings and day of the week.

I think you forgot the phase of the moon. :-)

Cheers,

David

P.S. By the way, I had an interesting e-mail conversation this week with someone producing object graphics pie charts, of all things. She was very carefully cleaning up memory, but when she ran her program in a loop it crashed consistently on the 4th or 10th or 16th or whatnot iteration. :-(

What in the world!?

Turns out it was the hardware renderer. This has gotten to be so prevalent that my FIRST suspect in ANY object graphics problem (especially those that would seem to be completely unrelated) is to check the graphics renderer. In my opinion you can only get predictable results using software rendering.

--

David Fanning, Ph.D.
Fanning Software Consulting

Subject: Re: Object rendering with dynamic views
Posted by [Mark Hadfield](#) on Tue, 23 Jan 2001 04:43:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.14d6b47246375088989d28@news.frii.com...
> P.S. By the way, I had an interesting e-mail conversation
> this week with someone producing object graphics pie charts,
> of all things. She was very carefully cleaning up memory,
> but when she ran her program in a loop it crashed consistently
> on the 4th or 10th or 16th or whatnot iteration. :-(
>
> What in the world!?
>
> Turns out it was the hardware renderer. This has gotten
> to be so prevalent that my FIRST suspect in ANY object
> graphics problem (especially those that would seem to
> be completely unrelated) is to check the graphics renderer.
> In my opinion you can only get predictable results using
> software rendering.

That's very interesting.

I should think the root cause of a flaky hardware renderer would be a flaky video driver. So once your correspondent might want to try tracking down & installing the latest version of the video driver, then trying the hardware renderer again.

I've never had any crashes related to the graphics renderer. I have found the hardware renderer on my machine somewhat faster (no more than 2x) than the software renderer. But I have (as I mentioned) pretty ho-hum graphics hardware. I was just about to check these things out on my new PC but I corrupted the hard disk! (I am too ashamed to tell the group how I did it.)

Doesn't IDL provide endless opportunities to tinker!

Mark Hadfield
m.hadfield@niwa.cri.nz <http://katipo.niwa.cri.nz/~hadfield/>
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand

Subject: Re: Object rendering with dynamic views
Posted by [davidf](#) on Tue, 23 Jan 2001 05:45:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Mark Hadfield (m.hadfield@niwa.cri.nz) writes:

> I should think the root cause of a flaky hardware renderer would be a flaky
> video driver. So once your correspondent might want to try tracking down &
> installing the latest version of the video driver, then trying the hardware
> renderer again.

Not so likely, in this case, as we both experienced the same problem with different graphics cards (I think). In any case, my graphics drivers are completely up to date. Another way of saying this is that they work, and I sure as hell am not going to install another driver on a machine that has not crashed in a week. Even I'm not that masochistic. :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Object rendering with dynamic views
Posted by [btt](#) on Tue, 23 Jan 2001 13:45:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
```

Thanks Mark and David,

```
<br>&nbsp;
```

```
<br>&nbsp;
```

```
<p>Mark Hadfield wrote:
```

```
<blockquote TYPE=CITE>No, but if you would like to compare them and report
back to the group, that
```

```
<br>would be much appreciated :-)
```

```
<p>I can only offer my $0.02 worth:
```

```
<p>I think this area deserves a careful examination. (Thanks for volunteering
<br>Ben.) I suspect you will find that performance is important, but that
the
```


ranking of the different methods will vary widely depending on OS,
graphics

hardware, IDL settings and day of the week.</blockquote>
I don't know the exchange rate but you \$0.02 is always worth the money. Volunteer? I should have kept my trap shut.
<p>My urge is to abandon the object graphics in this case and use direct
graphics. I have bumped into this question because I am converting
a direct graphics routine that I wrote a long time ago (it is very fast)
to object graphics. Now it is mostly complete, but the interactive
business is getting to be a bit of an event-processing-rendering bottle
neck.

<blockquote TYPE=CITE>
<pre>My only redeeming quality is that I tend to perceive
failure as good information. :-)</pre>
</blockquote>

<p>
David, Have you ever read the wonderful Emily Dickinson poem about
the sweetness of success to those who rarely experience it? If not,
I'll dig it out and send it to you.

<p>Thanks again,
<p>Ben
<p>--

Ben Tupper

Bigelow Laboratory for Ocean Sciences

180 McKown Point Rd.

W. Boothbay Harbor, ME 04575

btupper@bigelow.org

 </html>

Subject: Re: Object rendering with dynamic views
Posted by [Pavel A. Romashkin](#) on Tue, 23 Jan 2001 17:28:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

I am afraid my input is not worth even \$0.02, but after I made a fairly
good (to me anyway) OG plotting tool (Display), I ran into a strong
limitation of OG - lack of useful Contour routine. Again, all I do is 2D
plotting (for now), and OG might seem as an overkill, but it was very
easy to program using wide assortment of methods available already.
Now, the speed of rendering for me only kicks in after having > 100k
objects to render, and I am using method 3, because I have no static
elements at all.

The second holdback I ran into was lack of simple way to use OG visuals
in any sort of quality output. For some reason, the same plot in DG PS
output is fully vector and can be tweaked in Adobe or Corel, while OG
output in any form is made of a bunch of rectangular scalable blocks at

best. I could be doing something wrong, but I tried all possible settings and can't get OG output to be editable. And, no offense meant, IDL's visuals (at least those I have the patience to achieve) are not exactly publication ready.

For these reasons, I am moving backwards to DG, but of course it will be objects anyway. I'll have to write a few methods for ROI and all that, but I really want to have a nice Contour and Map capabilities combined with line plots. For as long as I have no true 3D needs, that should do and be fast to render.

Cheers,
Pavel

P.S. The reason I brought this up is that rendering speed of OGis not the only thing that comes to my mind. Output (whichm after all, is what visualization is all about) limitations became important once I had full-blown onscreen visualization going.

Subject: Re: Object rendering with dynamic views

Posted by [btt](#) on Tue, 23 Jan 2001 18:36:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Pavel,

I agree that this might be best handled with direct graphics. The trouble I have bumped into is that when I move the cursor over a data point within the scatter plot, something else will occur in a secondary (and separate) separate draw widget that already contains object graphics. I can't be sure that the display system will always be at least 24 bit (mixing object graphic widgets and direct graphic widgets on an 8bit monitor in the same session produces a mess.) Geez, if I could be sure that the routine would never be used on an 8 bit display, then I would definitely switch to a mix of graphics types (each in its own draw widget.)

Although I'd like to take Mark's challenge to test out the two OOG systems, I think that I must retreat a bit and modify my secondary widget display: it will be able to do either DG or OOG depending upon the situation. Then I will use direct graphics for the primary display/interaction. It's the path of least resistance for now... and the deadline is, well, let us see what time it is...

Thanks,

Ben

"Pavel A. Romashkin" wrote:

> I am afraid my input is not worth even \$0.02, but after I made a fairly
> good (to me anyway) OG plotting tool (Display), I ran into a strong
> limitation of OG - lack of useful Contour routine. Again, all I do is 2D
> plotting (for now), and OG might seem as an overkill, but it was very
> easy to program using wide assortment of methods available already.
> Now, the speed of rendering for me only kicks in after having > 100k
> objects to render, and I am using method 3, because I have no static
> elements at all.
> The second holdback I ran into was lack of simple way to use OG visuals
> in any sort of quality output. For some reason, the same plot in DG PS
> output is fully vector and can be tweaked in Adobe or Corel, while OG
> output in any form is made of a bunch of rectangular scalable blocks at
> best. I could be doing something wrong, but I tried all possible
> settings and can't get OG output to be editable. And, no offense meant,
> IDL's visuals (at least those I have the patience to achieve) are not
> exactly publication ready.
> For these reasons, I am moving backwards to DG, but of course it will be
> objects anyway. I'll have to write a few methods for ROI and all that,
> but I really want to have a nice Contour and Map capabilities combined
> with line plots. For as long as I have no true 3D needs, that should do
> and be fast to render.
>
> Cheers,
> Pavel
>
> P.S. The reason I brought this up is that rendering speed of OG is not
> the only thing that comes to my mind. Output (which after all, is what
> visualization is all about) limitations became important once I had
> full-blown onscreen visualization going.

--

Ben Tupper
Bigelow Laboratory for Ocean Sciences
180 McKown Point Rd.
W. Boothbay Harbor, ME 04575
btupper@bigelow.org
