Subject: IDL graphics w/ no display
Posted by KM on Fri, 22 Oct 2004 22:21:07 GMT
View Forum Message <> Reply to Message

Hi Again,

I used IDL daily from 1998 through 2002, but haven't since then, and forgot a bit. I just started in on a big project using IDL, and am looking for some suggestions/advice.

We are using IDL as a backend to produce map graphics. It is a cross-platform program (Win/Mac) and the client wants it looking mac-ish on the Mac side, hence IDL is the backend, not the frontend. In fact, the deployment environment might not even have X11 installed! The front-end is also cross-platform, so on the Win side I also won't access the display with IDL, only with the frontend, just because this approach uses the same code for both Win and Mac.

So I am producing images in IDL in Z buffer, writing PNGs to disk, and then telling the frontend (via a socket) that the image is ready. It reads it in and display it in a mac/win-like UI.

But there are some limitations to this model.

1) Z buffer is limited to 256 colors. This is not a major limitation, but it does exist

2) Z buffer text looks terrible.

3) front-end doesn't have native/fast EPS render support

I cannot use the "blow up * 4" trick for Z buffer text [http://dfanning.com/graphics_tips/zfonts.html] because it is too slow. I am using a very fast machine, but doing all the map stuff x4 introduces a 1 second lag.

I think I cannot use object graphics because I am working with map projections, map_continents, etc. and these don't work with object graphics, right?

Maybe I could do the map/grid/continents at regular scale in Z, read them out, blow up z*4, print the title, colorbar labels, etc., read them out, rebin to regular size, and then "print" the labels in the image with a where statement. This might save a bit of time but seems like quite a hack. Maybe thats what I have to do with no access to the display. :/

I don't think callable IDL would help. That could replace the socket

and file->disk part, but it wouldn't help with the display. Plus, it sounds complex.

Can anyone offer any advice/suggestions?

Thanks,

-k.

Subject: Re: IDL graphics w/ no display
Posted by KM on Wed, 27 Oct 2004 18:32:13 GMT
View Forum Message <> Reply to Message

On Wed, 27 Oct 2004, Ben Tupper wrote:

> Ken Mankoff wrote:
>
> (1) About OG without a display device.
>
> I think you canuse the IDLgrBuffer as the destination drawing device. The
> following works without X11 (MacOSX).

Yeah that code works. But the only reason I am considering OG is because of anti-aliasing. But it turns out only IDLgrText supports anti-aliasing! I had hoped all lines would be aliased, not just the text...

> (2) I think that the iMap might be disguising how easy it can be
> to transform map polygons into the OG realm. I recall a thread on
> this newsgroup where somebody tessalated the Polygons for Alaska
> and then threw them into an object graphics destination. (I think
> it was Karl - but I haven't found that thread.) With the advent
> of the MAP_PROJ_***** routines, it sure seems like RSI has freed
> mapping from the DG world. So, somewhere in iMap, the CIA map
> data (or the Shape file data) that comes with IDL is unpacked and
> then formed into OG polygons. That said, I have no experience
> with OG mapping although I started down that path once in an idle
> moment.

It might be this:
http://tinyurl.com/445dg

My new question is this: If I am producing static images written to disk, and making heavy use of the ma_set, map_grid, etc code, and _not_ doing interactive stuff, does it make sense to use OG or DG?

The only advantage to OG is pretty text, not even pretty lines.

-k.

Subject: Re: IDL graphics w/ no display
Posted by JD Smith on Wed, 27 Oct 2004 18:48:28 GMT
View Forum Message <> Reply to Message

On Wed, 27 Oct 2004 14:32:13 -0400, Ken Mankoff wrote:

>
> On Wed, 27 Oct 2004, Ben Tupper wrote:
>
>> Ken Mankoff wrote:
>>
>> (1) About OG without a display device.
>>
>> I think you canuse the IDLgrBuffer as the destination drawing device. The
>> following works without X11 (MacOSX).
>
> Yeah that code works. But the only reason I am considering OG is
> because of anti-aliasing. But it turns out only IDLgrText supports
> anti-aliasing! I had hoped all lines would be aliased, not just the
> text...
>
>> (2) I think that the iMap might be disguising how easy it can be
>> to transform map polygons into the OG realm. I recall a thread on
>> this newsgroup where somebody tessalated the Polygons for Alaska
>> and then threw them into an object graphics destination. (I think
>> it was Karl - but I haven't found that thread.) With the advent
>> of the MAP_PROJ_***** routines, it sure seems like RSI has freed
>> mapping from the DG world. So, somewhere in iMap, the CIA map
>> data (or the Shape file data) that comes with IDL is unpacked and
>> then formed into OG polygons. That said, I have no experience
>> with OG mapping although I started down that path once in an idle
>> moment.
>
> It might be this:
> http://tinyurl.com/445dg
>
> My new question is this: If I am producing static images written to
> disk, and making heavy use of the ma_set, map_grid, etc code, and
> _not_ doing interactive stuff, does it make sense to use OG or DG?
>
> The only advantage to OG is pretty text, not even pretty lines.

Personally I would target the postscript device in direct graphics,
and then use ImageMagick's "convert" to convert to PNG, ala:

convert -antialias -density 150x150 map.eps map.png

Maybe more overhead than you want to assume.

JD

Subject: Re: IDL graphics w/ no display
Posted by David Fanning on Wed, 27 Oct 2004 18:50:46 GMT
View Forum Message <> Reply to Message

Ken Mankoff writes:

> The only advantage to OG is pretty text, not even pretty lines.

And RSI seems commited to exploiting this single advantage. :-(

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http:/www.dfanning.com/
Phone: 970-221-0438, IDL Book Orders: 1-888-461-0155

Subject: Re: IDL graphics w/ no display
Posted by KM on Wed, 27 Oct 2004 20:46:57 GMT
View Forum Message <> Reply to Message

On Wed, 27 Oct 2004, JD Smith wrote:
>
> Personally I would target the postscript device in direct
> graphics, and then use ImageMagick's "convert" to convert to PNG,
> ala:
>
> convert -antialias -density 150x150 map.eps map.png
>
> Maybe more overhead than you want to assume.

Its a little bit slow, but maybe worth it for the quality.

This app is going out to the public, Win and Mac worlds, with
unknown installation environments. So everything I use gets bundled

in my installer, and must be cross platform.

ImageMagick probably has the right license, but I tihnk its a bit
big/complex to roll into my app. Do you know of a convert-like
utility that is cross platform and smaller?

-k.

Subject: Re: IDL graphics w/ no display
Posted by Liam Gumley on Mon, 08 Nov 2004 16:28:12 GMT
View Forum Message <> Reply to Message

Hey Ken,

I don't know if this came up in the thread, but Xvfb might solve your
problem on that Mac side. It's a virtual frame buffer for X, and it
allows you to run X11 graphics programs when a display device is not
available. I've had very good luck with it. RSI has a tech tip with more
information at

http://www.rsinc.com/services/techtip.asp?ttid=2382

You might have to hunt around a bit to find a binary. It may not help on
the Windows side, however.

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

KM wrote:
> Hi Again,
>
> I used IDL daily from 1998 through 2002, but haven't since then, and
> forgot a bit. I just started in on a big project using IDL, and am
> looking for some suggestions/advice.
>
> We are using IDL as a backend to produce map graphics. It is a
> cross-platform program (Win/Mac) and the client wants it looking mac-ish
> on the Mac side, hence IDL is the backend, not the frontend. In fact,
> the deployment environment might not even have X11 installed! The
> front-end is also cross-platform, so on the Win side I also won't access
> the display with IDL, only with the frontend, just because this approach
> uses the same code for both Win and Mac.
>
> So I am producing images in IDL in Z buffer, writing PNGs to disk, and
> then telling the frontend (via a socket) that the image is ready. It

> reads it in and display it in a mac/win-like UI.
>
> But there are some limitations to this model.
>
> 1) Z buffer is limited to 256 colors. This is not a major limitation,
> but it does exist
>
> 2) Z buffer text looks terrible.
>
> 3) front-end doesn't have native/fast EPS render support
>
> I cannot use the "blow up * 4" trick for Z buffer text
> [http://dfanning.com/graphics_tips/zfonts.html] because it is too slow.
> I am using a very fast machine, but doing all the map stuff x4
> introduces a 1 second lag.
>
> I think I cannot use object graphics because I am working with map
> projections, map_continents, etc. and these don't work with object
> graphics, right?
>
> Maybe I could do the map/grid/continents at regular scale in Z, read
> them out, blow up z*4, print the title, colorbar labels, etc., read them
> out, rebin to regular size, and then "print" the labels in the image
> with a where statement. This might save a bit of time but seems like
> quite a hack. Maybe thats what I have to do with no access to the
> display. :/
>
> I don't think callable IDL would help. That could replace the socket and
> file->disk part, but it wouldn't help with the display. Plus, it sounds
> complex.
>
> Can anyone offer any advice/suggestions?
>
> Thanks,
>
> -k.
