
Subject: how to keep an image object in the window?

Posted by [yy](#) on Sun, 24 Sep 2006 17:59:58 GMT

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

Hi there,

I have a basic question about object. In my simulation, I want to display some 3D data, read the image displayed in the window and save the 2D image as one movie frame. When I display the image by using object graphics, I found that it will be erased by other windows which happened to be on top of it. In that situation, what I read in will be the content in the top window not the 3D data I displayed. This is really bad for me b/c limited by the storage space, it takes more than a day to run the whole program and get all the data. If I have to keep the displaying widow always on top, I can't do any other things while the program is running and I have to even turn off the screen saver. I don't know if there is some way to solve this problem. I found that even the examples given in IDL help has this problem. Could anybody help me about this? Thanks a lot!

Jingyi

Subject: Re: how to keep an image object in the window?

Posted by [yy](#) on Tue, 26 Sep 2006 17:32:55 GMT

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

But I did find that if I read the image from the window and display it using the direct graphics, the image quality becomes really bad. I don't know if it's because I didn't take the snapshot properly.

When do you usually use hardware rendering?

Jingyi

Rick Towler wrote:

> As you know, Michael's suggestion is the way to go.
>
> The only thing to consider is that when drawing to an IDLgrBuffer IDL
> uses the software renderer. In your case this doesn't matter since
> you're rendering IDLgrVolume objects which force IDL to use the software
> renderer anyways. But sometimes you want/need to use the hardware
> renderer. In these cases you have to render to IDLgrWindow and the only
> solution to your type if problem is to just start recording your
> animation and take the rest of the day off :) er, I mean go to the
> library and catch up on the literature...
>
> -Rick
>

>
> Michael Galloy wrote:
>> yy wrote:
>>> Hi there,
>>> I have a basic question about object. In my simulation, I want to
>>> display some 3D data, read the image displayed in the window and save
>>> the 2D image as one movie frame. When I display the image by using
>>> object graphics, I found that it will be erased by other windows which
>>> happened to be on top of it. In that situation, what I read in will be
>>> the content in the top window not the 3D data I displayed. This is
>>> really bad for me b/c limited by the storage space, it takes more than
>>> a day to run the whole program and get all the data. If I have to keep
>>> the displaying widow always on top, I can't do any other things while
>>> the program is running and I have to even turn off the screen saver. I
>>> don't know if there is some way to solve this problem. I found that
>>> even the examples given in IDL help has this problem. Could anybody
>>> help me about this? Thanks a lot!
>>>
>>> Jingyi
>>>
>>
>> I have had weird issues with things like that too. I can't remember
>> exactly what was going on, but since you just want to get frames for
>> your movies and you're not looking at the frames anyway, try rendering
>> them to an IDLgrBuffer instead of an IDLgrWindow.
>>
>> Mike
>> --
>> www.michaelgalloy.com

Subject: Re: how to keep an image object in the window?
Posted by [Karl Schultz](#) on Tue, 26 Sep 2006 17:42:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

"In your case this doesn't matter since
you're rendering IDLgrVolume objects which force IDL to use the software
renderer anyways."

IDLgrVolume uses raycasting to generate an intermediate
and private IDLgrImage and then renders the IDLgrImage to the device,
using either hardware or software rendering, depending on how the device
was set up.

Raycasting is very slow, and is a software rendering process, but the
result may actually be displayed with hardware rendering. So, putting an
IDLgrVolume into a scene doesn't cause an automatic switch to the software
renderer.

Karl

On Mon, 25 Sep 2006 16:54:48 -0700, Rick Towler wrote:

```
> As you know, Michael's suggestion is the way to go.
>
> The only thing to consider is that when drawing to an IDLgrBuffer IDL
> uses the software renderer. In your case this doesn't matter since
> you're rendering IDLgrVolume objects which force IDL to use the software
> renderer anyways. But sometimes you want/need to use the hardware
> renderer. In these cases you have to render to IDLgrWindow and the only
> solution to your type of problem is to just start recording your
> animation and take the rest of the day off er, I mean go to the
> library and catch up on the literature...
>
> -Rick
>
>
> Michael Galloy wrote:
>> yy wrote:
>>> Hi there,
>>> I have a basic question about object. In my simulation, I want to
>>> display some 3D data, read the image displayed in the window and save
>>> the 2D image as one movie frame. When I display the image by using
>>> object graphics, I found that it will be erased by other windows which
>>> happened to be on top of it. In that situation, what I read in will be
>>> the content in the top window not the 3D data I displayed. This is
>>> really bad for me b/c limited by the storage space, it takes more than
>>> a day to run the whole program and get all the data. If I have to keep
>>> the displaying widow always on top, I can't do any other things while
>>> the program is running and I have to even turn off the screen saver. I
>>> don't know if there is some way to solve this problem. I found that
>>> even the examples given in IDL help has this problem. Could anybody
>>> help me about this? Thanks a lot!
>>>
>>> Jingyi
>>>
>>
>> I have had weird issues with things like that too. I can't remember
>> exactly what was going on, but since you just want to get frames for
>> your movies and you're not looking at the frames anyway, try rendering
>> them to an IDLgrBuffer instead of an IDLgrWindow.
>>
>> Mike
>> --
>> www.michaelgalloy.com
```

Subject: Re: how to keep an image object in the window?
Posted by [Rick Towler](#) on Tue, 26 Sep 2006 19:08:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Karl Schultz wrote:

> "In your case this doesn't matter since
> you're rendering IDLgrVolume objects which force IDL to use the software
> renderer anyways."
>
> IDLgrVolume uses raycasting to generate an intermediate
> and private IDLgrImage and then renders the IDLgrImage to the device,
> using either hardware or software rendering, depending on how the device
> was set up.
>
> Raycasting is very slow, and is a software rendering process, but the
> result may actually be displayed with hardware rendering. So, putting an
> IDLgrVolume into a scene doesn't cause an automatic switch to the software
> renderer.
>

Thanks for clarifying. Makes sense too. After I posted that I thought
that I was probably assuming too much.

<sigh>

Do you texture a "billboard" with the output of the raycaster?

-Rick

> Karl
>
> On Mon, 25 Sep 2006 16:54:48 -0700, Rick Towler wrote:
>
>> As you know, Michael's suggestion is the way to go.
>>
>> The only thing to consider is that when drawing to an IDLgrBuffer IDL
>> uses the software renderer. In your case this doesn't matter since
>> you're rendering IDLgrVolume objects which force IDL to use the software
>> renderer anyways. But sometimes you want/need to use the hardware
>> renderer. In these cases you have to render to IDLgrWindow and the only
>> solution to your type if problem is to just start recording your
>> animation and take the rest of the day off er, I mean go to the
>> library and catch up on the literature...
>>
>> -Rick
>>
>>
>> Michael Galloy wrote:

>>> yy wrote:
>>>> Hi there,
>>>> I have a basic question about object. In my simulation, I want to
>>>> display some 3D data, read the image displayed in the window and save
>>>> the 2D image as one movie frame. When I display the image by using
>>>> object graphics, I found that it will be erased by other windows which
>>>> happened to be on top of it. In that situation, what I read in will be
>>>> the content in the top window not the 3D data I displayed. This is
>>>> really bad for me b/c limited by the storage space, it takes more than
>>>> a day to run the whole program and get all the data. If I have to keep
>>>> the displaying widow always on top, I can't do any other things while
>>>> the program is running and I have to even turn off the screen saver. I
>>>> don't know if there is some way to solve this problem. I found that
>>>> even the examples given in IDL help has this problem. Could anybody
>>>> help me about this? Thanks a lot!
>>>>
>>>> Jingyi
>>>>
>>> I have had weird issues with things like that too. I can't remember
>>> exactly what was going on, but since you just want to get frames for
>>> your movies and you're not looking at the frames anyway, try rendering
>>> them to an IDLgrBuffer instead of an IDLgrWindow.
>>>
>>> Mike
>>> --
>>> www.michaelgalloy.com

Subject: Re: how to keep an image object in the window?
Posted by [Karl Schultz](#) on Tue, 26 Sep 2006 23:24:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 26 Sep 2006 12:08:49 -0700, Rick Towler wrote:

> Karl Schultz wrote:
>> "In your case this doesn't matter since
>> you're rendering IDLgrVolume objects which force IDL to use the software
>> renderer anyways."
>>
>> IDLgrVolume uses raycasting to generate an intermediate
>> and private IDLgrImage and then renders the IDLgrImage to the device,
>> using either hardware or software rendering, depending on how the device
>> was set up.
>>
>> Raycasting is very slow, and is a software rendering process, but the
>> result may actually be displayed with hardware rendering. So, putting an
>> IDLgrVolume into a scene doesn't cause an automatic switch to the software
>> renderer.

>>
>
> Thanks for clarifying. Makes sense too. After I posted that I thought
> that I was probably assuming too much.
>
> <sigh>
>
> Do you texture a "billboard" with the output of the raycaster?
>
> -Rick

The raycaster outputs an RGBA image, where the A channel has opacity information. It also outputs a "depth array". The RGBA image is drawn like any other image via `IDLgrImage`, which currently uses a texture-mapped polygon. The depth information is also written to the depth buffer so that the volume "depth" is communicated to the device in what would otherwise be a simple 2D image rendering step. This allows you to, for example, create an isosurface of the volume and draw it in the same view as the volume itself. The volume rendering, depending on opacity, is then able to obscure the isosurface where appropriate. `iVolume` is a good place to play with this, since `iVolume` has an operation that will generate the isosurface and then display it in the same view.

Karl
