
Subject: Re: 3d device coordinates from a 3D polyline....

Posted by [Karl\[1\]](#) on Thu, 04 Oct 2012 20:16:34 GMT

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

On Thursday, October 4, 2012 10:23:58 AM UTC-6, (unknown) wrote:

> On Wednesday, October 3, 2012 2:29:02 PM UTC-6, Karl wrote:

>

>> On Tuesday, October 2, 2012 3:32:02 PM UTC-6, Mike Galloy wrote:

>

>>

>

>>> On 10/2/12 3:02 PM, Karl wrote:

>

>>

>

>>>

>

>>

>

>>>> It should also be possible to write a general-purpose function that

>

>>

>

>>>

>

>>

>

>>>> takes a "leaf" graphics object and walks up the scene graph,

>

>>

>

>>>

>

>>

>

>>>> computing the single 4x4 combined matrix and returns it. You would

>

>>

>

>>>

>

>>

>

>>>> then use that single matrix to transform your points.

>

>>

>

>>>

>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> In a way, you are duplicating the entire transform that IDL applies
>
>>
>
>>>
>
>>
>
>>>> to the points via the underlying graphics system (OpenGL). I don't
>
>>
>
>>>
>
>>
>
>>>> remember if there is a way to get this transform directly from IDL -
>
>>
>
>>>
>
>>
>
>>>> don't think so. And someone out there may have already written an
>
>>
>
>>>
>
>>
>
>>>> IDL function to do this. But, I don't know of any.
>
>>
>
>>>

>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>>> Isn't this the ::getCTM() method or am I misunderstanding the situation?
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Mike
>
>>
>
>>>
>
>>
>
>>> --
>
>>
>
>>>
>
>>
>
>>> Michael Galloy
>
>>
>
>>>

```
>
>>
>
>>> www.michaelgalloy.com
>
>>
>
>>>
>
>>
>
>>> Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
>
>>
>
>>>
>
>>
>
>>> Research Mathematician
>
>>
>
>>>
>
>>
>
>>> Tech-X Corporation
>
>>
>
>>
>
>>
>
>>
>
>> yep, that's it.
>
>
>
> Actually, this IS a 3D perspective view I'm working with. I have a polyline in a 3D perspective
scene. One end of the polyline is at the center of my 3D coordinate system (ie, [0,0,0]) and the
other end is at (say) [+10,0,0]. As I rotate it around with a trackball the 2D projection in the window
can assume any
>
> 'size' (from a single pixel dot to a line of length 10) and any orientation (0 to 360 if you like).
>
>
>
```

> I'm amazed there isn't an inbuilt function to tell me what these 2D window coordinates are - but there you go, nothing like spending a couple of weeks fiddling with IDL - it's fun right ?
>
>
>
> I'll take a look a Michael's object graphics chapter. For me that is the ideal sample chapter....
>
>
>
> Cheers
>
>
>
> George.

It shouldn't take a couple of weeks. As Michael pointed out, there is the GetCTM method that will give you the Current Transform Matrix for the given object. You would multiply your desired 3D points by this matrix to (hopefully) get the window coordinates. It should probably work for both ortho and perspective projections.

I just forgot about the GetCTM method. That long description that I gave is probably pretty close to what GetCTM does under the covers.
