Subject: 3d device coordinates from a 3D polyline....
Posted by George.millward on Fri, 28 Sep 2012 21:30:56 GMT
View Forum Message <> Reply to Message

Hi there,

This must be straightforward but I'm lost in the help system:

I have am idlgrpolyline which I can rotate in a 3D view (with the trackball).
I want to know the 2D coordinates of this line in the device (ie,the 2D the projection in the window). Can't figure it out.

Any ideas ?

Cheers

George.

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by George.millward on Tue, 02 Oct 2012 16:46:16 GMT
View Forum Message <> Reply to Message

On Tuesday, October 2, 2012 3:22:12 AM UTC-6, alx wrote:
> Le lundi 1 octobre 2012 19:07:22 UTC+2, (inconnu) a écrit :
>
>> On Monday, October 1, 2012 6:32:00 AM UTC-6, David Fanning wrote:
>
>>
>
>>>> I have am idlgrpolyline which I can rotate in a 3D view (with the trackball).
>
>>
>
>>>
>
>>
>
>>>> I want to know the 2D coordinates of this line in the device (ie,the 2D the projection in the window). Can't figure it out.
>
>>
>
>>>
>
>>
>
>>>

>
>>
>
>>>
>
>>
>
>>> I'm no expert in this area, but I think the 3D to 2D
>
>>
>
>>>
>
>>
>
>>> conversions of the transformation matrix (which you
>
>>
>
>>>
>
>>
>
>>> can recover from the trackball) are well known. You
>
>>
>
>>>
>
>>
>
>>> can read the answer at the bottom of this article,
>
>>
>
>>>
>
>>
>
>>> for example:
>
>>
>
>>>
>
>>
>
>>>

>
>>
>
>>>
>
>>
>
>>>   http://math.stackexchange.com/questions/336/why-are-3d-
>
>>
>
>>>
>
>>
>
>>> transformation-matrices-4x4-instead-of-3x3
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Cheers,
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> David

>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> --
>
>>
>
>>>
>
>>
>
>>> David Fanning, Ph.D.
>
>>
>
>>>
>
>>
>
>>> Fanning Software Consulting, Inc.
>
>>
>
>>>
>
>>
>
>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

>
>>
>
>>>
>
>>
>
>>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
>>
>
>>
>
>>
>
>>
>
>>
>
>> Hmm,
>
>>
>
>>
>
>>
>
>> So there is nothing in the object graphics system like the 'CONVERT_COORD' routine ?
>
>>
>
>>
>
>>
>
>> George.
>
>
>
> In Object Graphics and New Graphics you can use "[XYZ]COORD_CONV" and
>
> "ConvertCoord" methods, respectively. The last one being quite similar to the "Convert_Coord" function in Direct Graphics.
>
> Alain.

Alain,

Thanks for your help.......I'm not using new graphics - this is all object graphics. I've been trying to understand how [XYZ]COORD_CONV in object graphics relates to all of this - but it's somewhat confusing.
I understand how XCOORD_CONV and YCOORD_CONV are used to map a 2D line to 2D normalized space - but I'm wanting the same for 3D...

Cheers

George.

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by Karl[1] on Tue, 02 Oct 2012 21:02:26 GMT
View Forum Message <> Reply to Message

On Tuesday, October 2, 2012 10:46:16 AM UTC-6, (unknown) wrote:
> On Tuesday, October 2, 2012 3:22:12 AM UTC-6, alx wrote:
>
>> Le lundi 1 octobre 2012 19:07:22 UTC+2, (inconnu) a écrit :
>
>>
>
>>> On Monday, October 1, 2012 6:32:00 AM UTC-6, David Fanning wrote:
>
>>
>
>>>
>
>>
>
>>>> > I have am idlgrpolyline which I can rotate in a 3D view (with the trackball).
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> > I want to know the 2D coordinates of this line in the device (ie,the 2D the projection in the window). Can't figure it out.

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> I'm no expert in this area, but I think the 3D to 2D
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> conversions of the transformation matrix (which you

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> can recover from the trackball) are well known. You
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> can read the answer at the bottom of this article,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> for example:

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> http://math.stackexchange.com/questions/336/why-are-3d-
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> transformation-matrices-4x4-instead-of-3x3

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Cheers,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>

```
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> David
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
```

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> --
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> David Fanning, Ph.D.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Fanning Software Consulting, Inc.

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>

>
>>
>
>>>
>
>>
>
>>> Hmm,
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> So there is nothing in the object graphics system like the 'CONVERT_COORD' routine ?
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> George.
>
>>
>
>>
>
>>
>
>> In Object Graphics and New Graphics you can use "[XYZ]COORD_CONV" and

>
>>
>
>> "ConvertCoord" methods, respectively. The last one being quite similar to the "Convert_Coord" function in Direct Graphics.
>
>>
>
>> Alain.
>
>
>
> Alain,
>
>
>
> Thanks for your help.......I'm not using new graphics - this is all object graphics. I've been trying to understand how [XYZ]COORD_CONV in object graphics relates to all of this - but it's somewhat confusing.
>
> I understand how XCOORD_CONV and YCOORD_CONV are used to map a 2D line to 2D normalized space - but I'm wanting the same for 3D...
>
>
>
> Cheers
>
>
>
> George.

[XYZ]COORD_CONV is a PROPERTY, not a method, for many IDLGr* Object Graphics classes. It is used to apply a transform to the raw vertex data stored in the object instance as the first part of the overall object-to-window transform that IDL performs when drawing the scene.

This is a simple scale and translate transform, so the values for all three ([XYZ]COORD_CONV) can be put into a 4x4 matrix and used to multiply all your 3D points stored in the object.

You'll also have to multiply the points by each 4x4 transform matrix stored in the TRANSFORM property in each IDLgrModel in your scene graph, working your way up to the view.

Then you need to apply a view transform using some of the properties (like VIEWPLANE_RECT) from the IDLgrView. Finally apply a view-to-window transform to get your 2D device coordinates.

Figuring out the last two are perhaps the trickiest, especially if the view projection is perspective. But it is possible.

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.

Karl

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by Michael Galloy on Tue, 02 Oct 2012 21:32:01 GMT
View Forum Message <> Reply to Message

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

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by Michael Galloy on Wed, 03 Oct 2012 14:49:05 GMT
View Forum Message <> Reply to Message

On 10/2/12 10:46 AM, George.millward@yahoo.com wrote:
> Thanks for your help.......I'm not using new graphics - this is all
> object graphics. I've been trying to understand how [XYZ]COORD_CONV
> in object graphics relates to all of this - but it's somewhat
> confusing. I understand how XCOORD_CONV and YCOORD_CONV are used to

> map a 2D line to 2D normalized space - but I'm wanting the same for
> 3D...

There are several examples of using [XYZ]COORD_CONV in the object
graphics chapter of my book, which also happens to be the sample chapter
that is freely available on the books website:

http://modernidl.idldev.com

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL: A Guide to IDL Programming (http://modernidl.idldev.com)
Research Mathematician
Tech-X Corporation

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by Karl[1] on Wed, 03 Oct 2012 20:29:01 GMT
View Forum Message <> Reply to Message

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.

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by George.millward on Thu, 04 Oct 2012 16:23:57 GMT
View Forum Message <> Reply to Message

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.

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.

Subject: Re: 3d device coordinates from a 3D polyline....
Posted by George.millward on Thu, 04 Oct 2012 20:21:24 GMT
View Forum Message <> Reply to Message

On Friday, September 28, 2012 3:30:56 PM UTC-6, (unknown) wrote:
> Hi there,
>
>
>
> This must be straightforward but I'm lost in the help system:
>
>
>
> I have am idlgrpolyline which I can rotate in a 3D view (with the trackball).
>
> I want to know the 2D coordinates of this line in the device (ie,the 2D the projection in the window). Can't figure it out.
>
>
>
> Any ideas ?
>
>
>
> Cheers
>
>
>
> George.

Karl,

Thanks for the suggestions. I'll give getCTM() a go and report back what I come up with. Yes, if getCTM is doing the job it should be 2 minutes , not 2 weeks.

Cheers for now,

George.
