Subject: Re: range ordering of triangular facets
Posted by nasalmon on Tue, 15 Jun 2004 21:31:20 GMT
View Forum Message <> Reply to Message

nasalmon@onetel.net.uk (Neil) wrote in message
news:<74039481.0405261005.7c6ebb50@posting.google.com>...
> "Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message
news:<10b9as5hf48h41a@corp.supernews.com>...
>> "Neil" <nasalmon@onetel.net.uk> wrote in message
>> news:74039481.0405251329.69e88002@posting.google.com...
>>> Does anyone have any idea how i can obtain a routine that will range
>>> order a series of triangular facets, such that each triangle can be
>>> rendered to appear on top of predecessing ones, without triangles
>>> appearing out of order? The assumption of course is that no triangles
>>> cross, but they can of course share common pairs of vertices?
>>>
>>> Unfortunatley, simple range ordering the facets according to the mean
>>> of the distance to the viewer doesnt always work, particularly for
>>> long triangles. This is annoying when rendering, as some triangles
>>> that should be on top lie underneath, leading to an imperfect image.
>>> This must be the bread and butter of some mathematicians, but it's
>>> certainly got me fooled.
>>
>> An analytical solution is the Binary Space Partition tree. There is a
>> sample implementation in Graphics Gems.
>>
>> If all you want to do is render these triangles, then Object Graphics or the
>> Direct Graphics 'Z' device will render them correctly using a depth buffer.
>>
>> Karl
>
>
> Karl,
>
> For reasons, i have decided to opt for Direct Graphics, and i need a
> routine that takes the "connectivity" and uses the geometry
> information from the "vertices" to give a range ordering index to the
> triangles, so when i render, no triangles are obscuring things they
> shouldnt be. This would seem such a general problem, there must be a
> simple IDL or C routine that i could slot into my programme.
>
> many thanks
> Neil

Karl / anyone

just checking this depth facet ordering stuff from Graphics Gems, i

downloaded the BSP code which is in C. Could anyone recommend a good text book for C and how can i get the C code to run in IDL?

Going back further in time to the mid 70's, before BSP, depth order was done using a method originally due to Newell and Sancha, where distances and cross overs of polygons where examined, followed by a sorting process. I have tried to develop such a code from Steve Harrington's book, Edition 1 from 1983, but cant get the code to work to my satisfaction. Is anyone familiar with the Newell / Sancha method and could say anything about the relative merits of this process versus that of BSP? For example how about the speed of these codes? Is anyone familar with Harrington's solution, or are there any other codings up of this method for access?

many thanks,
Neil
