Subject: Re: Realistic Illumination, IDL & OpenGL
Posted by Rick Towler on Sat, 31 Jan 2004 00:36:41 GMT
View Forum Message <> Reply to Message

"Neil" wrote in message...

> For Object Graphics to emulate reality, illumination mustbe
> accurately represented. Take for example an outdoor scene, where sky
> background emission above and up-welling emission below bathes an
> object in different intensities from all directions. Can this be
> accurately described in IDL or OpenGL?

It depends what you mean by accurately described. Lighting in openGL is more of an art than a science. Simply throwing in a few lights rarely leads to an acceptable rendition of a scene. Proper texturing and camera placement are key. If there is motion in the scene (either camera or actor) causing the lighting to change you can generate procedural textures. Take a look at a few ~2-3 year old 3d games to get an idea of what you can do with good texturing.

If you need to go further (or do it faster) you'll need to move beyond IDL.

> In IDL Object Graphics there is the possibility to bring in the
> "light" by way of obj_new "IDLgrLight". However, this can only be:
> type 0) Ambient; 1) point source; 2) collimated beam; 3) spot light
> (apertured). The number of lights is limited to 8. Is this
> illumination limited by IDL or the OpenGL on which IDL operates?

All openGL implementations support at least 8 lights, some support more. I don't know if there is a hard limit in IDL but my guess is that this limit is imposed so IDL will behave the same way regardless of the underlying gl implementation.

8 may seem like a small number but I think you'll find that it doesn't take a lot of lights to render an acceptable scene as long as you are creative with placement and textures.

> Gereral illumination in an outdoor environment has something of an
> angular distribution, for an overcast day at least. This might be
> represented in an alt-azimuth coordinate system; large amount of
> emission at zenith, with some kind of distribution coming down to the
> horizon, with some up-welling illumination from the ground. This could
> be either simulated by a very large number of point sources or ambient
> illumination with some angular distribution on it. Are there plans to
> introduce this kind of "light" into IDL in future versions?

If I understand you, this can be done in IDL right now. You just have to be creative. :)

Or you might want to consider pov-ray.

-Rick

Subject: Re: Realistic Illumination, IDL & OpenGL
Posted by nasalmon on Sat, 31 Jan 2004 08:40:35 GMT
View Forum Message <> Reply to Message

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:<bvetav$jfc$1@nntp6.u.washington.edu>...
> "Neil" wrote in message...
>
>
>> For Object Graphics to emulate reality, illumination mustbe
>> accurately represented. Take for example an outdoor scene, where sky
>> background emission above and up-welling emission below bathes an
>> object in different intensities from all directions. Can this be
>> accurately described in IDL or OpenGL?
>
> It depends what you mean by accurately described. Lighting in openGL is
> more of an art than a science. Simply throwing in a few lights rarely leads
> to an acceptable rendition of a scene. Proper texturing and camera
> placement are key. If there is motion in the scene (either camera or actor)
> causing the lighting to change you can generate procedural textures. Take a
> look at a few ~2-3 year old 3d games to get an idea of what you can do with
> good texturing.
The difficulty i have is that i am trying to use Object Graphics as a
science to speed my scene simulation, which i have already coded up in
Direct Graphics with good scientific precision. Somehow i need to be
quantitatively creative in Object Graphics to create the same result
as in Direct Graphics. Strategically for the further development of
Object Graphics, if note is taken of the science of reflections /
illumination a more accurate description of rendering will result.
However, i can seen this is not a priority of the gaming industry, but
IDL with is scientific background would be interested in enabling a
more scientifically precise Object Graphics, as opposted to art based
Object Graphics for computer gaming.
>
> If you need to go further (or do it faster) you'll need to move beyond IDL.
I may want to go further and faster, if so what would you recommend?

I suppose what i really want to do is to go faster with scientifically
precise graphics.
>
>> In IDL Object Graphics there is the possibility to bring in the

>> "light" by way of obj_new "IDLgrLight". However, this can only be:
>> type 0) Ambient; 1) point source; 2) collimated beam; 3) spot light
>> (apertured). The number of lights is limited to 8. Is this
>> illumination limited by IDL or the OpenGL on which IDL operates?
>
> All openGL implementations support at least 8 lights, some support more.
> I don't know if there is a hard limit in IDL but my guess is that this limit
> is imposed so IDL will behave the same way regardless of the underlying gl
> implementation.
>
> 8 may seem like a small number but I think you'll find that it doesn't take
> a lot of lights to render an acceptable scene as long as you are creative
> with placement and textures.
>
>> Gereral illumination in an outdoor environment has something of an
>> angular distribution, for an overcast day at least. This might be
>> represented in an alt-azimuth coordinate system; large amount of
>> emission at zenith, with some kind of distribution coming down to the
>> horizon, with some up-welling illumination from the ground. This could
>> be either simulated by a very large number of point sources or ambient
>> illumination with some angular distribution on it. Are there plans to
>> introduce this kind of "light" into IDL in future versions?
>
> If I understand you, this can be done in IDL right now. You just have to be
> creative. :)
>
> Or you might want to consider pov-ray.
Pov-ray? What on earth is that?
>
> -Rick

many thanks for your help
Neil

Subject: Re: Realistic Illumination, IDL & OpenGL
Posted by Karl Schultz on Mon, 02 Feb 2004 18:57:23 GMT

View Forum Message <> Reply to Message

"Neil" <nasalmon@onetel.net.uk> wrote in message
news:74039481.0401310040.101be823@posting.google.com...
> "Rick Towler" <rtowler@u.washington.edu> wrote in message
news:<bvetav$jfc$1@nntp6.u.washington.edu>...
>> "Neil" wrote in message...
>>
>>
>>> For Object Graphics to emulate reality, illumination mustbe
>>> accurately represented. Take for example an outdoor scene, where sky

>>> background emission above and up-welling emission below bathes an
>>> object in different intensities from all directions. Can this be
>>> accurately described in IDL or OpenGL?
>>
>> It depends what you mean by accurately described. Lighting in openGL is
>> more of an art than a science. Simply throwing in a few lights rarely
leads
>> to an acceptable rendition of a scene. Proper texturing and camera
>> placement are key. If there is motion in the scene (either camera or
actor)
>> causing the lighting to change you can generate procedural textures.
Take a
>> look at a few ~2-3 year old 3d games to get an idea of what you can do
with
>> good texturing.
> The difficulty i have is that i am trying to use Object Graphics as a
> science to speed my scene simulation, which i have already coded up in
> Direct Graphics with good scientific precision. Somehow i need to be
> quantitatively creative in Object Graphics to create the same result
> as in Direct Graphics. Strategically for the further development of
> Object Graphics, if note is taken of the science of reflections /
> illumination a more accurate description of rendering will result.

This would call for a global illumination model.

> However, i can seen this is not a priority of the gaming industry, but
> IDL with is scientific background would be interested in enabling a
> more scientifically precise Object Graphics, as opposted to art based
> Object Graphics for computer gaming.

This is a good point. However, there is a large demand for (object)
graphics that are precise in ways other than illumination and are
interactive as well, and that's where Object Graphics is today. Rendering
with more complex illumination models isn't as interactive, at least within
reason.

>> If you need to go further (or do it faster) you'll need to move beyond
IDL.
> I may want to go further and faster, if so what would you recommend?
>
> I suppose what i really want to do is to go faster with scientifically
> precise graphics.
>>
>>> In IDL Object Graphics there is the possibility to bring in the
>>> "light" by way of obj_new "IDLgrLight". However, this can only be:
>>> type 0) Ambient; 1) point source; 2) collimated beam; 3) spot light
>>> (apertured). The number of lights is limited to 8. Is this
>>> illumination limited by IDL or the OpenGL on which IDL operates?

>>
>> All openGL implementations support at least 8 lights, some support more.
>> I don't know if there is a hard limit in IDL but my guess is that this
limit
>> is imposed so IDL will behave the same way regardless of the underlying
gl
>> implementation.
>>
>> 8 may seem like a small number but I think you'll find that it doesn't
take
>> a lot of lights to render an acceptable scene as long as you are
creative
>> with placement and textures.
>>
>>> Gereral illumination in an outdoor environment has something of an
>>> angular distribution, for an overcast day at least. This might be
>>> represented in an alt-azimuth coordinate system; large amount of
>>> emission at zenith, with some kind of distribution coming down to the
>>> horizon, with some up-welling illumination from the ground. This could
>>> be either simulated by a very large number of point sources or ambient
>>> illumination with some angular distribution on it. Are there plans to
>>> introduce this kind of "light" into IDL in future versions?
>>
>> If I understand you, this can be done in IDL right now. You just have
to be
>> creative. :)

And I *don't* completely understand the requirement, but the mention of
"very large number of point sources" and "ambient illumination with some
angular distribution" makes me doubt that IDL/OpenGL can do it.

>>
>> Or you might want to consider pov-ray.
> Pov-ray? What on earth is that?

Persistance Of Vision RAYtracer. This is one of the most popular ray
tracers around. As far as I know, it is still free and runs on various
platforms. Because of its age, er, maturity, and the number of people
working on it, it is quite a full-featured program. Just Google for it.

It may not be too hard to write an IDL program that emits Pov-ray scene
descriptions that contain the global illumination information you need and
then let Pov-ray render it for you with all the illumination effects you
need. It won't be interactive, as the raytracing may take some time on the
order of minutes. A really cool IDL app would let you see the scene with
the OpenGL lighting model as an approximation while being interactive enough
to adjust things. When you are happy with it, push the button to generate
the Pov-ray file and spawn the tracer.

A radiosity-based renderer might be worth googling for too.

Karl

Subject: Re: Realistic Illumination, IDL & OpenGL
Posted by Rick Towler on Mon, 02 Feb 2004 20:01:02 GMT
View Forum Message <> Reply to Message

"Neil" wrote...
> "Rick Towler" wrote...
>> "Neil" wrote in message...
>>
>>
>>> For Object Graphics to emulate reality, illumination mustbe
>>> accurately represented. Can this be
>>> accurately described in IDL or OpenGL?
>>
>> It depends what you mean by accurately described. Lighting in openGL is
>> more of an art than a science.

> The difficulty i have is that i am trying to use Object Graphics as a
> science to speed my scene simulation, which i have already coded up in
> Direct Graphics with good scientific precision. Somehow i need to be
> quantitatively creative in Object Graphics to create the same result
> as in Direct Graphics. Strategically for the further development of
> Object Graphics, if note is taken of the science of reflections /
> illumination a more accurate description of rendering will result.
> However, i can seen this is not a priority of the gaming industry, but
> IDL with is scientific background would be interested in enabling a
> more scientifically precise Object Graphics, as opposted to art based
> Object Graphics for computer gaming.

The extent of the openGL lighting model that is currently exposed to
IDL programmers is very small so you are limited by IDL. Even if IDL exposed
the entirity of openGL's lighting model, openGL just isn't designd to do
this. OpenGL is designed for interactive applications and as a result a lot
of compromises have been made along the way. Lighting is one. Like I said,
there are many ways to fake it, but you're not interested in that :)

>> If you need to go further (or do it faster) you'll need to move beyond
IDL.
> I may want to go further and faster, if so what would you recommend?
>
> I suppose what i really want to do is to go faster with scientifically
> precise graphics.

Without details it is difficult to be certain but I think it would be safe
to say that you should focus on optimizing the core components of your
current application. Be it simply optmizing your current IDL code or
writing core components in C/C++. You could also just buy a faster computer
:)

>> Or you might want to consider pov-ray.
> Pov-ray? What on earth is that?

The persistence of vision ray tracer. Probably not a tool to do
scientifically precise graphics but the "lighting model" is far more
realistic. www.povray.org

-Rick