Subject: Re: [update]: artifacts with volume rendering
Posted by David Fanning on Wed, 26 Feb 2003 14:23:08 GMT
View Forum Message <> Reply to Message

Sebastian (s@visita2.die.upm.es) writes:

> I wrote a litte program that shows the artefacts. I creates a volume, cuts
> of a cube, and renders a short sequence. The artefacts are clearly
> recognizable on the cutting surfaces.

When I removed the INTERPOLATE=1 keyword, the artifacts
appear to disappear:

```
vol->SETPROPERTY, data0=volData,ZBUFFER=1,ZERO_OPACITY_SKIP=1, $
  OPACITY_TABLE0=(INDGEN(256) / 1);;;;;;;;;;INTERPOLATE=1
```

This was the first thing I thought of when I saw the
output, because those look like interpolation artifacts
(rounding errors, etc.) to me.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: [update]: artifacts with volume rendering
Posted by s[1] on Wed, 26 Feb 2003 15:15:45 GMT
View Forum Message <> Reply to Message

Hi David,

when you disable INTERPOLATE, there are less artefacts, but they still are
there.
Anyway, I must use INTERPOLATE for the real prog because otherwise the
image quality is much too bad.

Furthermore, using a bounding box, the artefacts do not appear (or at
least are much weaker).
To verify this, include

```
vol->getproperty, BOUNDS=bounds
```

```
  bounds[0] = bounds[3] / 3
  vol->setproperty, BOUNDS=bounds
```

somewhere in the volume creation code.

Best regards,

Sebastian

On Wed, 26 Feb 2003, David Fanning wrote:

> Sebastian (s@visita2.die.upm.es) writes:
>
>> I wrote a litte program that shows the artefacts. I creates a volume, cuts
>> of a cube, and renders a short sequence. The artefacts are clearly
>> recognizable on the cutting surfaces.
>
> When I removed the INTERPOLATE=1 keyword, the artifacts
> appear to disappear:
>
> vol->SETPROPERTY, data0=volData,ZBUFFER=1,ZERO_OPACITY_SKIP=1, $
> OPACITY_TABLE0=(INDGEN(256) / 1);;;;;;;;;;INTERPOLATE=1
>
> This was the first thing I thought of when I saw the
> output, because those look like interpolation artifacts
> (rounding errors, etc.) to me.

Subject: Re: [update]: artifacts with volume rendering
Posted by Karl Schultz on Wed, 26 Feb 2003 15:23:43 GMT
View Forum Message <> Reply to Message

"Sebastian" <s@visita2.die.upm.es> wrote in message
news:Pine.LNX.4.44.0302261420320.2560-100000@visita2.die.upm .es...
> I wrote a litte program that shows the artefacts. I creates a volume, cuts
> of a cube, and renders a short sequence. The artefacts are clearly
> recognizable on the cutting surfaces.

<snip program>

I think that this is simply caused by sampling error. You are doing trilinear interpolation, which means that the "empty" (zero) voxels adjacent to your cut surfaces are going to contribute to the computation of the "pixel" in the final image. The amount of the contribution of the "zero" voxel depends on how the ray passes through the voxels along the cutting edge. And that's why the "aliased" frequency you are seeing in the bands depends on the rotation angle.

I'm not a wizard in this area, but I know that sampling at twice the frequency can remove these aliasing problems. I changed your window size to 100x100 and made the volume 200x200x200 and that made the artifacts disappear or at least become less noticable.

I also think that this exposes a minor flaw with your workaround for lack of bounded cutting planes. If IDLgrVolume had bounded cutting planes, then the interpolator wouldn't use the voxels on the other side of the bounded cutting plane in the interpolation calculation and then you'd probably be OK. As it stands, the interpolator doesn't know that you don't want the zero voxels to contribute to the image at all, and so includes them in the interpolation.

Here's another idea, which I'm sorry that I didn't suggest sooner.

Use two 3D datasets in IDLgrVolume.

The first dataset is your real volume data. The second one is a mask volume. The voxels are 255 where you want voxels in the first dataset to be displayed and 0 where you don't. You would set the VOLUME_SELECT property to 1. This may not help the aliasing problem, but it might be a better way to cut cubes out of your volume - you wouldn't have to damage your original data.

Karl

Subject: Re: [update]: artifacts with volume rendering
Posted by s[1] on Wed, 26 Feb 2003 17:20:31 GMT
View Forum Message <> Reply to Message

On Wed, 26 Feb 2003, Karl Schultz wrote:
> ...
> I'm not a wizard in this area, but I know that sampling at twice the
> frequency can remove these aliasing problems. I changed your window size to
> 100x100 and made the volume 200x200x200 and that made the artifacts
> disappear or at least become less noticable.
>
Yes, by making the window small compared to the volume size the artefacts disappear - but what you really want is to have a large window and asmall volume or a zoom into a volume, so that's no solution.

> ...
> OK. As it stands, the interpolator doesn't know that you don't want the
> zero voxels to contribute to the image at all, and so includes them in the
> interpolation.
>

I somehow was hoping the ZERO_OPACITY_SKIP keyword would handle this.
>
> Here's another idea, which I'm sorry that I didn't suggest sooner.
> Use two 3D datasets in IDLgrVolume.
> The first dataset is your real volume data. The second one is a mask
> volume. The voxels are 255 where you want voxels in the first dataset to be
> displayed and 0 where you don't. You would set the VOLUME_SELECT property
> to 1. This may not help the aliasing problem, but it might be a better way
> to cut cubes out of your volume - you wouldn't have to damage your original
> data.
>
Thanks for the tip, but wouldn't this be a performance penalty? My technique only needs to multiple the dataset with the mask once, which is really fast, this technique needs that multiplication for every rendering.

Best regards,

Sebastian