Subject: IDL w/ 12-bit grayscale?
Posted by mchinand on Tue, 28 Mar 2006 16:45:12 GMT
View Forum Message <> Reply to Message

Does anyone have experience using IDL to display 12-bit grayscale images on hardware that supports it while maintaining full bit-depth of the images? I'm using IDL on Linux and one of the available X11 visuals is GrayScale with depth 12. I used .Xdefaults settings to force IDL to use this visual. TVSCL works but using the /words keyword with TVSCL results in the error that the hardware doesn't support 16-bits per pixel and would sometimes crash the X server. It seems the hardware must actually take 16-bits per pixel even though the depth resolution is only 12-bits.

Thanks,

--Mike

--
Michael Chinander, PhD
m-chinander@uchicago.edu
Department of Radiology
University of Chicago

Subject: Re: IDL w/ 12-bit grayscale?
Posted by David Fanning on Mon, 03 Apr 2006 14:22:37 GMT
View Forum Message <> Reply to Message

Mike Chinander writes:

> Does anyone have experience using IDL to display 12-bit grayscale images on hardware that
> supports it while maintaining full bit-depth of the images? I'm using IDL on Linux and one
> of the available X11 visuals is GrayScale with depth 12. I used .Xdefaults settings to
> force IDL to use this visual. TVSCL works but using the /words keyword with TVSCL results
> in the error that the hardware doesn't support 16-bits per pixel and would sometimes crash
> the X server.

TVSCL!? Doesn't that sort of defeat the whole
purpose of having 12-bit values? I would have
thought a TV of integer data, scaled into the
range of 0 to 2^12 (4096) would be something your
12-bit hardware would like.

> It seems the hardware must actually take 16-bits per pixel even though the
> depth resolution is only 12-bits.

I would hope so, or you are going to have to write your
own LIXUX kernel, too, probably. :-)

Cheers,

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

Subject: Re: IDL w/ 12-bit grayscale?
Posted by mchinand on Mon, 03 Apr 2006 17:00:56 GMT
View Forum Message <> Reply to Message

In article <MPG.1e9add86555d222d989bef@news.frii.com>,
David Fanning <davidf@dfanning.com> wrote:
>
> TVSCL!? Doesn't that sort of defeat the whole
> purpose of having 12-bit values? I would have
> thought a TV of integer data, scaled into the
> range of 0 to 2^12 (4096) would be something your
> 12-bit hardware would like.

I did use TV as well, from the discription of the /WORDS keyword, I was under the impression that the converion to byte was not done for TVSCL when /WORDS is set. You're right, TV is more appropriate, I first used TVSCL because I used it in a program I have that opens up a window that is the size of the image.

>
>> It seems the hardware must actually take 16-bits per pixel even though the
>> depth resolution is only 12-bits.
>
> I would hope so, or you are going to have to write your
> own LIXUX kernel, too, probably. :-)
>

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago

Subject: Re: IDL w/ 12-bit grayscale?
Posted by Karl Schultz on Mon, 03 Apr 2006 19:10:49 GMT
View Forum Message <> Reply to Message

On Mon, 03 Apr 2006 17:00:56 +0000, Mike Chinander wrote:

> In article <MPG.1e9add86555d222d989bef@news.frii.com>,
> David Fanning <davidf@dfanning.com> wrote:
>>
>> TVSCL!? Doesn't that sort of defeat the whole
>> purpose of having 12-bit values? I would have
>> thought a TV of integer data, scaled into the
>> range of 0 to 2^12 (4096) would be something your
>> 12-bit hardware would like.
>
> I did use TV as well, from the discription of the /WORDS keyword, I was under the impression
> that the converion to byte was not done for TVSCL when /WORDS is set. You're right, TV is
> more appropriate, I first used TVSCL because I used it in a program I have that opens up a
> window that is the size of the image.

/WORDS only makes sense on IDL Direct Graphics devices that support it.
The only device that does support it is the Z-Buffer device. That device
has an 8-bit color channel and a 16-bit depth channel.

From the docs:

To read the depth values in the Z-buffer, use the command:
a = TVRD(CHANNEL=1, /WORDS)
To write the depth values, use the command:
TV, a, /WORDS, CHANNEL=1
The TV, TVSCL, and TVRD routines write or read pixels directly to a
rectangular area of the designated buffer without affecting the other
buffer.

Yes, the docs imply that /WORDS causes a 16 bits per pixel transfer. And
this would be true on devices that support it and when transferring a
channel that is 16-bits wide.

current device is 'Z':

```
IDL> help, tvrd()
<Expression>    BYTE      = Array[640, 480]
IDL> help, tvrd(channel=1)
TVRD: Z depth buffer contains words.
Execution halted at: $MAIN$
IDL> help, tvrd(channel=1, /words)
<Expression>    INT       = Array[640, 480]
```

>
>
>>> It seems the hardware must actually take 16-bits per pixel even though
>>> the depth resolution is only 12-bits.
>>
>> I would hope so, or you are going to have to write your own LIXUX
>> kernel, too, probably. :-)
>>
>>
> Maybe Fï¿½ldy can work on this when he finishes FL.
>
>
> --Mike

Mike, what graphics card are you using. A DOME card?

The IDL Direct Graphics 'X' driver will probably require some work to
support 12-bit channels. The driver does support the GrayScale Visual
type, but probably initializes the first 256 entries of the Colormap to a
ramp from back to white, and does not touch the other Colormap entries.
Then, for images, it only writes the values [0-255] into the frame buffer.
It works, but you're not using all the bits.

Karl

Subject: Re: IDL w/ 12-bit grayscale?
Posted by mchinand on Wed, 05 Apr 2006 16:12:02 GMT
View Forum Message <> Reply to Message

In article <pan.2006.04.03.19.10.43.328000@rsinc.com>,
Karl Schultz <k____schultz@rsinc.com> wrote:
>
> Mike, what graphics card are you using. A DOME card?
>
> The IDL Direct Graphics 'X' driver will probably require some work to
> support 12-bit channels. The driver does support the GrayScale Visual
> type, but probably initializes the first 256 entries of the Colormap to a
> ramp from back to white, and does not touch the other Colormap entries.
> Then, for images, it only writes the values [0-255] into the frame buffer.
> It works, but you're not using all the bits.
>
> Karl

Yes, I think it was a DOME card. I tried it on two different systems, not sure if the other one was DOME as well. It would be great if the driver could use the full bit-depth of the hardware.

--Mike

--
Michael Chinander
m-chinander@uchicago.edu
Department of Radiology
University of Chicago