Subject: JPEG2000 compression ratios?
Posted by Matt Feinstein on Wed, 29 Sep 2004 12:14:19 GMT
View Forum Message <> Reply to Message

There's probably an 'obvious' answer to this, but I'm in need of a
hint...

I'd like to figure out the possible image compression ratios that
result from using different quality layers in JPEG2000-- but I don't
see any obvious way of getting that out of the IDLffJPEG2000 object.

Matt Feinstein

--
There is no virtue in believing something that can be proved to be true.

Subject: Re: JPEG2000 compression
Posted by Rick Towler on Wed, 13 Apr 2005 16:24:10 GMT
View Forum Message <> Reply to Message

Roberto Monaco wrote:
> I have a question regarding JPEG2000:
>
> How do you control compression ratio when you save an image as JPEG2000
> using the IDL object?

Looks like the N_LEVELS property.

> The following is curious: within a Photoshop plug-in I use to save JPEG2000
> images, "they" define a compression factor. If I save the same image with
> different compressions ratios (e.g. 1:1, 1:10, 1:40) I obviously get big
> differences in the resulting file sizes. But if you read these files into
> IDL objects, surprisingly all properties are exactly the same!! Same
> dimensions for the data array, same bit_depth, same number of layers, etc.
> ?? This does not make any sense to me ... I must be overlooking at something
> stupid?

Well, I know nothing about JPEG2000 but I would hope that all of the
properties that you have specified *would* be identical. The only
property that should change would be the compression factor. Don't know
where that would be though. Maybe the N_LEVELS property again? Don't
know if that would be set after reading a file or if it only applies to
writing.

-Rick

Subject: Re: JPEG2000 compression
Posted by Roberto Monaco on Wed, 13 Apr 2005 19:55:20 GMT
View Forum Message <> Reply to Message

>> I have a question regarding JPEG2000:
>>
>> How do you control compression ratio when you save an image as JPEG2000
>> using the IDL object?
>
> Looks like the N_LEVELS property.

Yes, from the documentation you would expect N_LEVELS to be responsible for compression.

This is the code I am using:

```
jp2 = obj_new('IDLffJPEG2000', jp2_file, /WRITE)
jp2->SetProperty, N_LEVELS=15
jp2->SetProperty, TILE_DIMENSIONS=[2048.,2048.]
jp2->SetData, image, /ORDER
obj_destroy, jp2
```

If I set N_LEVELS to 5 in the second line (the default) and run it again for the same image (same input file) I get both times jp2 files of the same dimension (31 MB) ?!?! I am obviously missing something here.

Perhaps one could define N levels and somehow store only some of these levels, discarding for example the 2 higher resolution levels? But I don't see a way of doing something like this, in SetData or elsewhere.

>> The following is curious: within a Photoshop plug-in I use to save JPEG2000
>> images, "they" define a compression factor. If I save the same image with
>> different compressions ratios (e.g. 1:1, 1:10, 1:40) I obviously get big
>> differences in the resulting file sizes. But if you read these files into
>> IDL objects, surprisingly all properties are exactly the same!! Same
>> dimensions for the data array, same bit_depth, same number of layers, etc.
>> ?? This does not make any sense to me ... I must be overlooking at something
>> stupid?
>
> Well, I know nothing about JPEG2000 but I would hope that all of the
> properties that you have specified *would* be identical. The only
> property that should change would be the compression factor. Don't know
> where that would be though. Maybe the N_LEVELS property again? Don't
> know if that would be set after reading a file or if it only applies to
> writing.

These images were created ouside IDL, in Photoshop saving the same TIFF image as JP2 using different compression ratios each time. Afterwards when I read them within IDL I would expect to see differences in their properties, since one corresponds to 1:1 (~30 MB of data) and another to 1:40 (~3 MB of data). But N_LEVELS, as ALL the other properties, are exactly the same for all cases...

By the way there is no compression property in 'IDLffJPEG2000' object structure.

Roberto

Subject: Re: JPEG2000 compression
Posted by Rick Towler on Wed, 13 Apr 2005 23:27:06 GMT
View Forum Message <> Reply to Message

Roberto Monaco wrote:
>>> I have a question regarding JPEG2000:
>>>
>>> How do you control compression ratio when you save an image as JPEG2000
>>> using the IDL object?
>>
>> Looks like the N_LEVELS property.
>
>
> Yes, from the documentation you would expect N_LEVELS to be responsible for
> compression.
>
> This is the code I am using:
> jp2 = obj_new('IDLffJPEG2000', jp2_file, /WRITE)
> jp2->SetProperty, N_LEVELS=15
> jp2->SetProperty, TILE_DIMENSIONS=[2048.,2048.]
> jp2->SetData, image, /ORDER
> obj_destroy, jp2
>
> If I set N_LEVELS to 5 in the second line (the default) and run it again for
> the same image (same input file) I get both times jp2 files of the same
> dimension (31 MB) ?!?! I am obviously missing something here.

Well, it works for me:

```
jp2 = obj_new('IDLffJPEG2000', 'test1.jp2', /WRITE)
jp2->SetProperty, N_LEVELS=0
jp2->SetData, BYTSCL(DIST(1600,1200))
obj_destroy, jp2
```

```
jp2 = obj_new('IDLffJPEG2000', 'test2.jp2', /WRITE)
jp2->SetProperty, N_LEVELS=15
jp2->SetData, BYTSCL(DIST(1600,1200))
obj_destroy, jp2
```

test1.jp2 is 614KB
test2.jp2 is 13KB

Further, querying the image works as advertised too:

```
jp2 = obj_new('IDLffJPEG2000', 'test1.jp2', /READ)
jp2->GetProperty, N_LEVELS=nl
PRINT, nl
obj_destroy, jp2

% Compiled module: $MAIN$.
IDL> .GO
           0

jp2 = obj_new('IDLffJPEG2000', 'test2.jp2', /READ)
jp2->GetProperty, N_LEVELS=nl
PRINT, nl
obj_destroy, jp2

% Compiled module: $MAIN$.
IDL> .GO
           15
```

>> Well, I know nothing about JPEG2000 but I would hope that all of the
>> properties that you have specified *would* be identical. The only
>> property that should change would be the compression factor. Don't know
>> where that would be though. Maybe the N_LEVELS property again? Don't
>> know if that would be set after reading a file or if it only applies to
>> writing.
>
>
> These images were created ouside IDL, in Photoshop saving the same TIFF
> image as JP2 using different compression ratios each time. Afterwards when I
> read them within IDL I would expect to see differences in their properties,
> since one corresponds to 1:1 (~30 MB of data) and another to 1:40 (~3 MB of
> data). But N_LEVELS, as ALL the other properties, are exactly the same for
> all cases...

Why do you think it is a problem that "ALL the other properties" are
exactly the same if you have taken the same image and saved it multiple

times, only changing the "compression factor"? What other properties should change?

My point was that I would assume that the image dimensions, color space, n_layers, offset and the other properties *wouldn't* change if you only changed the "compression factor" in photoshop.

> By the way there is no compression property in 'IDLffJPEG2000' object
> structure.

I know. I was relating N_LEVELS to what you called "compression factor" in photoshop.

-Rick

Subject: Re: JPEG2000 compression
Posted by Emmanuel Christophe on Thu, 14 Apr 2005 14:41:14 GMT
View Forum Message <> Reply to Message

Roberto Monaco a ï¿½crit :
> I have a question regarding JPEG2000:
>
> How do you control compression ratio when you save an image as JPEG2000
> using the IDL object?
>
> The following is curious: within a Photoshop plug-in I use to save JPEG2000
> images, "they" define a compression factor. If I save the same image with
> different compressions ratios (e.g. 1:1, 1:10, 1:40) I obviously get big
> differences in the resulting file sizes. But if you read these files into
> IDL objects, surprisingly all properties are exactly the same!! Same
> dimensions for the data array, same bit_depth, same number of layers, etc.
> ?? This does not make any sense to me ... I must be overlooking at something
> stupid?
>
> Thanks for any tips.
> Roberto
>
>

I got this problem before, I only manage to get lossless jpeg2000 compression directly with IDL. I went around this problem by calling one free implementation of jpeg2000 (using spawn): jasper

;creating source j2k file from idl data (image)

```
  im_jpeg=obj_new('IDLffJPEG2000',filenamesrc,/write)
  im_jpeg->SetProperty, bit_depth=16, signed=1, reversible=1
  im_jpeg->SetData, image
  OBJ_DESTROY, im_jpeg

;compressing file
  spawn,'jasper -f '+filenamesrc+'-F '+filenamenew+' -T jpc -O rate=0.2'
;reopening in IDL
  im_jpeg2=obj_new('IDLffJPEG2000',filenamenew,/read)
  imnew=intarr(ns,nl)
  imnew=im_jpeg2->GetData()
```

Hope this help,
Emmanuel
