Subject: color_quan - how for exactly 256 colors?
Posted by justspam03 on Thu, 16 Oct 2003 15:39:57 GMT
View Forum Message <> Reply to Message

Hi,

is there a way to do color quantization of a
true color image which contains exactly 256 colors?

color_quan seems no viable choice: the statistical
method does not produce exact results and the cube=6
option only works (does it? didn't test it) for up
to 216 colors.

Cheers
  Oliver

Subject: Re: color_quan - how for exactly 256 colors?
Posted by JD Smith on Fri, 17 Oct 2003 18:57:31 GMT
View Forum Message <> Reply to Message

On Fri, 17 Oct 2003 01:01:33 -0700, Oliver Thilmann wrote:

>> David Fanning <david@dfanning.com> wrote in message
>> news:<MPG.19f86c414eaf8be5989719@news.frii.com>...
>
>> What does this mean!? By definition, there will be no "exact" results
>> when you sample 16.7 million colors down to 256. It just, uh...,
>> mathematically can't be done. :-)
>
> What I mean is: I know that my image contains not more than 256
> different RGB colors (out of 16.7 million) - I created the RGB image
> from an indexed image and now I want to transform it back. This can be
> done exactly and I wondered whether IDL provides a method to get that
> done. Cheers,

Yes, with HISTOGRAM:

```
rgb_image=r+256L*(g+256L*b)
h=histogram(rgb_image,OMIN=om)
wh=where(h gt 0,cnt) # Should be fewer than 256
h[wh]=bindgen(cnt)
index_image=h[rgb_image]
colors=om+wh ; these are your <=256 colors
r_vec=colors AND 255L
g_vec=ishft(colors,-8) AND 255L
b_vec=ishft(colors,-16) AND 255L
```

tvlct,r_vec,g_vec,b_vec
tv,index_image

Probably not the most efficient method in the universe, given the
sparseness of the histogram, but it gets the job done.

JD

Subject: Re: color_quan - how for exactly 256 colors?
Posted by David Fanning on Fri, 17 Oct 2003 19:57:39 GMT
View Forum Message <> Reply to Message

JD Smith writes:

> Yes, with HISTOGRAM:

Ha! Maybe I'm beginning to understand this. My first thought
this morning, was "histogram". But, of course, I left it to
JD to work out the details. :-)

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: color_quan - how for exactly 256 colors?
Posted by JD Smith on Fri, 17 Oct 2003 21:35:56 GMT
View Forum Message <> Reply to Message

On Fri, 17 Oct 2003 11:57:31 -0700, JD Smith wrote:

> On Fri, 17 Oct 2003 01:01:33 -0700, Oliver Thilmann wrote:
>
>>> David Fanning <david@dfanning.com> wrote in message
>>> news:<MPG.19f86c414eaf8be5989719@news.frii.com>...
>>
>>> What does this mean!? By definition, there will be no "exact" results
>>> when you sample 16.7 million colors down to 256. It just, uh...,
>>> mathematically can't be done. :-)

>>
>> What I mean is: I know that my image contains not more than 256
>> different RGB colors (out of 16.7 million) - I created the RGB image
>> from an indexed image and now I want to transform it back. This can be
>> done exactly and I wondered whether IDL provides a method to get that
>> done. Cheers,
>
> Yes, with HISTOGRAM:
>
> rgb_image=r+256L*(g+256L*b)
> h=histogram(rgb_image,OMIN=om)
> wh=where(h gt 0,cnt) # Should be fewer than 256 h[wh]=bindgen(cnt)
> index_image=h[rgb_image]

I think I meant:
index_image=h[rgb_image-om]

Of course, I still haven't tested it...

JD

Subject: Re: color_quan - how for exactly 256 colors?
Posted by justspam03 on Sun, 19 Oct 2003 13:29:12 GMT
View Forum Message <> Reply to Message

>
> Of course, I still haven't tested it...
>
> JD

> index_image = h[rgb_image]
Impressive :)
Works like charm. Thank you!

Cheers
Oliver

Subject: Re: color_quan - how for exactly 256 colors?
Posted by David Fanning on Fri, 24 Oct 2003 15:34:06 GMT
View Forum Message <> Reply to Message

Folks,

Ok, I'm confused.

JD Smith wrote the other day in response to Mr. Thilmann:

>> What I mean is: I know that my image contains not more than 256
>> different RGB colors (out of 16.7 million) - I created the RGB image
>> from an indexed image and now I want to transform it back. This can be
>> done exactly and I wondered whether IDL provides a method to get that
>> done. Cheers,
>
> Yes, with HISTOGRAM:
>
>   rgb_image=r+256L*(g+256L*b)
>   h=histogram(rgb_image,OMIN=om)
>   wh=where(h gt 0,cnt) # Should be fewer than 256
>   h[wh]=bindgen(cnt)
>   index_image=h[rgb_image-om]
>   colors=om+wh ; these are your <=256 colors
>   r_vec=colors AND 255L
>   g_vec=ishft(colors,-8) AND 255L
>   b_vec=ishft(colors,-16) AND 255L
>   tvlct,r_vec,g_vec,b_vec
>   tv,index_image
>
> Probably not the most efficient method in the universe, given the
> sparseness of the histogram, but it gets the job done.

To which Oliver responded with this:

> Impressive :)
> Works like a charm. Thank you!

But,... it's not working like a charm for me. :-(

In fact, when I run this code, I find that index_image
is a LONG *vector*, not the 2D image I was expecting.
What am I missing here?

In line three:

```
  wh=where(h gt 0,cnt) # Should be fewer than 256
```

I used:

```
  wh=where(h gt 0,cnt) # 255
```

How does this "work like a charm"?

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: color_quan - how for exactly 256 colors?
Posted by JD Smith on Fri, 24 Oct 2003 15:55:43 GMT
View Forum Message <> Reply to Message

On Fri, 24 Oct 2003 08:34:06 -0700, David Fanning wrote:

> Folks,
>
> Ok, I'm confused.
>
> JD Smith wrote the other day in response to Mr. Thilmann:
>
>>> What I mean is: I know that my image contains not more than 256
>>> different RGB colors (out of 16.7 million) - I created the RGB image
>>> from an indexed image and now I want to transform it back. This can
>>> be done exactly and I wondered whether IDL provides a method to get
>>> that done. Cheers,
>>
>> Yes, with HISTOGRAM:
>>
>> rgb_image=r+256L*(g+256L*b)
>> h=histogram(rgb_image,OMIN=om)
>> wh=where(h gt 0,cnt) # Should be fewer than 256 h[wh]=bindgen(cnt)
>> index_image=h[rgb_image-om]
>> colors=om+wh ; these are your <=256 colors r_vec=colors AND 255L
>> g_vec=ishft(colors,-8) AND 255L
>> b_vec=ishft(colors,-16) AND 255L
>> tvlct,r_vec,g_vec,b_vec
>> tv,index_image
>>
>> Probably not the most efficient method in the universe, given the
>> sparseness of the histogram, but it gets the job done.
>
> To which Oliver responded with this:
>
>> Impressive :)
>> Works like a charm. Thank you!
>

> But,... it's not working like a charm for me. :-(
>
> In fact, when I run this code, I find that index_image is a LONG
> *vector*, not the 2D image I was expecting. What am I missing here?
>
> In line three:
>
> wh=where(h gt 0,cnt) # Should be fewer than 256
>
> I used:
>
> wh=where(h gt 0,cnt) # 255

Sorry, this is a Perl comment character slipping in... too much Perl'ing
for IDLWAVE lately (hidden IDLWAVE rumor of the week: an IDL6 version with
full doc support should be out early next week). It should read:

wh=where(h gt 0,cnt) ; Should be fewer than 256

That is, you'd better have fewer than 256 colors in your rgb image if
you'd like to create an exact indexed image from it. Also note that the
original r,g, & b were intended to be *images*, one for each color plane
of your 24bit image, i.e. I should have written:

rgb_image=r_image+256L*(g_image+256L*b_image)

Got Charm?

JD

Subject: Re: color_quan - how for exactly 256 colors?
Posted by David Fanning on Fri, 24 Oct 2003 16:17:19 GMT
View Forum Message <> Reply to Message

JD Smith writes:

> Sorry, this is a Perl comment character slipping in... too much Perl'ing
> for IDLWAVE lately (hidden IDLWAVE rumor of the week: an IDL6 version with
> full doc support should be out early next week). It should read:
>
> wh=where(h gt 0,cnt) ; Should be fewer than 256
>
> That is, you'd better have fewer than 256 colors in your rgb image if
> you'd like to create an exact indexed image from it. Also note that the
> original r,g, & b were intended to be *images*, one for each color plane
> of your 24bit image, i.e. I should have written:
>

> rgb_image=r_image+256L*(g_image+256L*b_image)
>
> Got Charm?

Oh, right. Well, yes, that does work nicely. :-)

But, of course, we still have the Color_Quan-like
color table and image. How about extra credit for getting us
back to the original color table vectors and 2D image?

What I wanted to use this for was getting a 2D image
from a pixmap. This would be *so* much more convenient
than having to draw the darn picture in the Z buffer. :-(

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
