Subject: Re: Stretching an image
Posted by David Fanning on Thu, 09 Dec 2004 19:01:49 GMT
View Forum Message <> Reply to Message

Julio writes:

> I'd like to know how to apply a stretch in an image using 2% of
> saturation. That's what ENVI does when I open any image.
>
> How can I do that using an IDL code?

I have no idea how ENVI does this, or even what
"2% of saturation" means, but I'm feeling reckless.
Here is how I would do it:

```
minImage = Min(image, Max=maxImage)
range = Float(maxImage) - Min(image)
onePercent = range * 0.01
low = minImage + onePercent
high = maxImage - onePercent
stretchedImage = BytScl(image, Min=low, Max=high)
TV, stretchedImage
```

Cheers,

David

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

Subject: Re: Stretching an image
Posted by jnettle1 on Thu, 09 Dec 2004 19:51:56 GMT
View Forum Message <> Reply to Message

Julio wrote:
> Hello there!
>
> I'd like to know how to apply a stretch in an image using 2% of
> saturation. That's what ENVI does when I open any image.
>
> How can I do that using an IDL code?
>
> Any comments wellcome,
>
> Thanks,

> Julio

Since you're running ENVI you can also used ENVI's routines, namely the
STRETCH_DOIT routine. The help file for this procedure tells you
exactly how to do what you want.

Jeff

Subject: Re: Stretching an image
Posted by KM on Thu, 09 Dec 2004 21:33:53 GMT
View Forum Message <> Reply to Message

On Thu, 9 Dec 2004, Julio wrote:
> I'd like to know how to apply a stretch in an image using 2% of
> saturation. That's what ENVI does when I open any image.

hist_equal() ?

-k.

Subject: Re: Stretching an image
Posted by David Fanning on Thu, 09 Dec 2004 21:36:48 GMT
View Forum Message <> Reply to Message

Ken Mankoff writes:

> On Thu, 9 Dec 2004, Julio wrote:
>> I'd like to know how to apply a stretch in an image using 2% of
>> saturation. That's what ENVI does when I open any image.
>
> hist_equal() ?

I don't think of that so much as a stretch as I do a squash. :-)

Cheers,

David

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

Subject: Re: Stretching an image
Posted by Liam Gumley on Fri, 10 Dec 2004 15:14:10 GMT
View Forum Message <> Reply to Message

Julio wrote:
> Hello there!
>
> I'd like to know how to apply a stretch in an image using 2% of
> saturation. That's what ENVI does when I open any image.
>
> How can I do that using an IDL code?
>
> Any comments wellcome,
>
> Thanks,
> Julio

I believe ENVI uses what is known as histogram clipping, as a way to remove outlier values from the stretch range. I describe an implementation of histogram clipping in Chapter 7 of my book. Here is the code from the book (imclip.pro):

```
;---start of imclip.pro---
FUNCTION IMCLIP, IMAGE, PERCENT=PERCENT

;- Check arguments
if (n_params() ne 1) then $
  message, 'Usage: RESULT = IMCLIP(IMAGE)'
if (n_elements(image) eq 0) then $
  message, 'Argument IMAGE is undefined'

;- Check keywords
if (n_elements(percent) eq 0) then percent = 2.0

;- Get image minimum and maximum
min_value = min(image, max=max_value)

;- Compute histogram
nbins = 100
binsize = float(max_value - min_value) / float(nbins)
hist = histogram(float(image), binsize=binsize)
bins = lindgen(nbins + 1) * binsize + min_value

;- Compute normalized cumulative sum
sum = fltarr(n_elements(hist))
sum[0] = hist[0]
for i = 1L, n_elements(hist) - 1L do $
  sum[i] = sum[i - 1] + hist[i]
sum = 100.0 * (sum / float(n_elements(image)))
```

```
;- Find and return the range
range = [min_value, max_value]
index = where((sum ge percent) and $
  (sum le (100.0 - percent)), count)
if (count ge 2) then $
  range = [bins[index[0]], bins[index[count - 1]]]
return, range

END
;---end of imclip.pro---
```

NOTE: The TOTAL function with the /CUMULATIVE keyword may be used in IDL 5.3 and higher to compute the cumulative sum.

To use this function, assuming you have an image array named IMAGE:

```
IDL> range = imclip(image)
IDL> tv, bytscl(image, min=range[0], max=range[1])
```

If you use IMCLIP in conjunction with my IMDISP program (see my website), you can just do this:

```
IDL> imdisp, image, range=imclip(image)
```

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Stretching an image
Posted by David Fanning on Fri, 10 Dec 2004 16:33:08 GMT
View Forum Message <> Reply to Message

Liam Gumley writes:

```
>  I believe ENVI uses what is known as histogram clipping, as a way to
>  remove outlier values from the stretch range. I describe an
>  implementation of histogram clipping in Chapter 7 of my book. Here is
>  the code from the book (imclip.pro):
>
>  ;---start of imclip.pro---
>  FUNCTION IMCLIP, IMAGE, PERCENT=PERCENT
>
>  ;- Check arguments
>  if (n_params() ne 1) then $
>     message, 'Usage: RESULT = IMCLIP(IMAGE)'
```

> if (n_elements(image) eq 0) then $
> message, 'Argument IMAGE is undefined'
>
> ;- Check keywords
> if (n_elements(percent) eq 0) then percent = 2.0
>
> ;- Get image minimum and maximum
> min_value = min(image, max=max_value)
>
> ;- Compute histogram
> nbins = 100
> binsize = float(max_value - min_value) / float(nbins)
> hist = histogram(float(image), binsize=binsize)
> bins = lindgen(nbins + 1) * binsize + min_value

Liam, why 100 bins? I would have thought the minimum number would be 256, to correspond with a possible byte image. Or is 100 just a nice round number that gives reasonable results?

Cheers,

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

Subject: Re: Stretching an image
Posted by Liam Gumley on Fri, 10 Dec 2004 18:41:09 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Liam Gumley writes:
>
>
>> I believe ENVI uses what is known as histogram clipping, as a way to
>> remove outlier values from the stretch range. I describe an
>> implementation of histogram clipping in Chapter 7 of my book. Here is
>> the code from the book (imclip.pro):
>>
>> ;---start of imclip.pro---
>> FUNCTION IMCLIP, IMAGE, PERCENT=PERCENT
>>
>> ;- Check arguments
>> if (n_params() ne 1) then $
>> message, 'Usage: RESULT = IMCLIP(IMAGE)'

>> if (n_elements(image) eq 0) then $
>>   message, 'Argument IMAGE is undefined'
>>
>> ;- Check keywords
>> if (n_elements(percent) eq 0) then percent = 2.0
>>
>> ;- Get image minimum and maximum
>> min_value = min(image, max=max_value)
>>
>> ;- Compute histogram
>> nbins = 100
>> binsize = float(max_value - min_value) / float(nbins)
>> hist = histogram(float(image), binsize=binsize)
>> bins = lindgen(nbins + 1) * binsize + min_value
>
>
> Liam, why 100 bins? I would have thought the minimum
> number would be 256, to correspond with a possible byte
> image. Or is 100 just a nice round number that gives
> reasonable results?
>
> Cheers,
>
> David

A nice round number that gives reasonable results.

Subject: Re: Stretching an image
Posted by KM on Fri, 10 Dec 2004 21:15:16 GMT
View Forum Message <> Reply to Message

On Thu, 9 Dec 2004, David Fanning wrote:
> Ken Mankoff writes:
>> On Thu, 9 Dec 2004, Julio wrote:
>>> I'd like to know how to apply a stretch in an image using 2% of
>>> saturation. That's what ENVI does when I open any image.
>>
>> hist_equal() ?
>
> I don't think of that so much as a stretch as I do a squash. :-)

OK, then how about 1/hist_equal()? :)
