
Subject: Re: Does IDL has histogram matching function?

Posted by [tianyf_cn](#) on Sat, 23 Nov 2002 08:34:05 GMT

[View Forum Message](#) <> [Reply to Message](#)

Maybe I want to process float type images. Or the output data values are in a narrow range. Does anyone have some ideas?

Thanks.

Yours,
Tian.

David Fanning <david@dfanning.com> wrote in message
news:<MPG.1835a3e2693e7288989a0b@news.frii.com>...

> David Fanning (david@dfanning.com) writes:

>

>> I expect it might take a day or so to write the code.

>> Do you have any money? :-)

>

> Ah, forget the money. This turned out to be too easy. :-)

>

> Here is a routine, named HISTOMATCH, that takes an image

> and a histogram that you would like to perform histogram

> matching to.

>

> *****

> FUNCTION HistoMatch, image, histogram_to_match

>

> ; Perform histogram matching according to the method of

> ; Gonzales and Woods in Digital Image Processing, pp 94-102

>

> ; image - The input image.

> ; histogram_to_match - The histogram used for histogram matching.

>

> ; Calculate the histogram of the input image.

>

> h = Histogram(Byte(image), Binsize=1, Min=0, Max=255)

> totalPixels = Float(N_Elements(image))

>

> ; Find a mapping from the input pixels to s.

>

> s = FltArr(256)

> FOR k=0,255 DO BEGIN

> s[k] = Total(h(0:k) / totalPixels)

> ENDFOR

>

> ; Find a mapping from input histogram to v.

>

```

> v = FltArr(256)
> FOR q=0,255 DO BEGIN
>   v[q] = Total(histogram_to_match(0:q) / totalPixels)
> ENDFOR
>
>   ; Find z from v and s.
>
> z = BytArr(256)
> FOR j=0,255 DO BEGIN
>   l = Where(v LT s[j], count)
>   IF count GT 0 THEN z[j] = (Reverse(l))[0] ELSE z[j]=0
> ENDFOR
>
>   ; Create the matched image.
>
> matchedImage = z[Byte(image)]
> RETURN, matchedImage
> END
> ,*****
> ,
>
> I'm certain JD or someone will point out to me how to
> use another Histogram to eliminate the Where function,
> but, hey, this is for free. I'm trying to make a living
> here. :-(
>
> Does it work!? I think so. I'm not sure.
>
> Try this. Let's see if we can match an image to the
> histogram formed by calculating the histogram of
> the histogram equalized image. (The result should
> be the same as the histogram equalized image, more
> or less.)
>
> ,*****
> ,
> PRO TestIt
> filename = Filepath('ctscan.dat', Subdir=['examples', 'data'])
> OpenR, lun, filename, /Get_Lun
> image = BytArr(256, 256)
> ReadU, lun, image
> Free_Lun, lun
>
> Window, XSize=3*256, YSize=256
> TV, image, 0
> TV, Hist_Equal(image), 1
> TV, HistoMatch(image, Histogram(Hist_Equal(image), Min=0, Max=255)), 2
> END
> ,*****
> ,
>

```

> IDL> TestIt
>
> Wow! And this was on the *first* try. *That* doesn't happen too
> often. :-)
>
> Try this:
>
> a = LonGen(255)
> b = a#b
> b = BytScl(b)
> Window, 1
> Plot, Histogram(b, Min=0, Max=255)
> Window, 2, XSize=256, YSize=256)
> TV, HistoMatch(image, Histogram(b, Min=0, Max=255))
>
> Still looks good, I think.
>
> OK, I'm waiting for feedback. :-)
>
> Cheers,
>
> David
