Subject: Re: Does IDL has histogram matching function?
Posted by aardvark62 on Fri, 06 Dec 2002 21:37:29 GMT
View Forum Message <> Reply to Message

I think I have found a more accurate algorithm for the FCN keyword.
The algorithm in hist_equal.pro IDL 5.5 and 5.6 is:

```
    if keyword_set(fcn_in) then begin
       y2 = bytscl(total(histogram(bytscl(fcn_in)), /CUM), TOP=top)
       p = y2[bytscl(p)]
    endif else begin
       p = BYTSCL(TEMPORARY(p), TOP = top)
    endelse
```

I changed that to:

```
    p = BYTSCL(TEMPORARY(p), TOP = top)

    if keyword_set(fcn_in) then begin
       y2 = bytarr(top + 1)
       f = bytscl(fcn_in, top=top)
       f = congrid(f, top+1)
       for i=0,top do begin ;invert curve f.
          y2[i] = (where(f ge i))[0]
       end

       p = y2[p]
    endif
```

With that change the resulting output fits the requested FCN more
closely. I used procedure HISTOGRAM_TEST (below) to assess output.
This version of HISTOGRAM_TEST has some fixes from the previous one
that I posted in this thread (I think).

The algorithm that I am proposing yeilds images with grayscales that
don't always reach TOP. This is dictated by the shape of the FCN
curve, and by the fact that caluculations are BYTSCLed. If you really
want to reach TOP, you would have to slightly distort the curve (or
operate at a higher resolution than bytes). I have not tested this,
but one way to reach TOP might be to change...

```
    f = bytscl(fcn_in, top=top)
```

...to...

```
    f = bytscl(fcn_in, top=top-1)
    f[n_elements(f)-1] = top
```

...which distorts the curve a little.

-Paul Sorenson

```
pro histogram_test, top=top
;
;Purpose: visually examine the output of HIST_EQUAL by plotting
;cumulative distribution functions.
;
filename = filepath('ctscan.dat', subdir=['examples', 'data'])
image = read_binary(filename, data_dims=[256, 256])

x = findgen(256)/255.      ;Ramp from 0 to 1.
y = exp(-((x-.5)/.2)^2)    ;Gaussian curve

fcn = total(y, /cumulative)
gauss_image = hist_equal(image, fcn=fcn, top=top) ;Request custom
dist.

d = total(histogram(gauss_image), /cum) ;Resulting distribution.

device, decomp=0
window, /free
loadct, 39
;
;Comepare the ideal requested distribution curve (in red) to the
actual
;resulting distribution curve (in white).
;
!p.multi = [0, 2, 1]
xrange = [0, top+10] ; Somewhat arbitrary.
;xrange = [0, 80] ; An interesting alternative.
plot, bytscl(d), xrange=xrange
oplot, bytscl(congrid(fcn, top+1)), color=254 ;red

print, 'max(gauss_image) = ', max(gauss_image)

uniform_image = hist_equal(image, top=top) ;Request uniform
distribution.

d = total(histogram(uniform_image), /cum) ;Resulting distribution.
;
;Comepare the ideal requested distribution curve (in red) to the
actual
;resulting distribution curve (in white).
;
plot, bytscl(d), xrange=xrange
oplot, bytscl(bindgen(top + 1)), color=254 ;red
```

end

David Fanning <david@dfanning.com> wrote in message
news:<MPG.184720e7d16d5cd5989a41@news.frii.com>...
> Paul Sorenson (aardvark62@msn.com) writes:
>
>> I'm not sure if the implementation of FCN is correct. I'm still thinking
>> about it.
>
> I don't know. According to what I read about this,
> the method is -- at best -- an approximation
> with digital images. It seems to be doing approximately
> what I expect it to. Close enough for government work,
> anyway.
>
> Cheers,
>
> David
