Subject: Re: Histogram Hot-shots Required
Posted by eddie haskell on Fri, 16 Jul 1999 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> What I can't figure out tonight is how to find out
> what bin that pixel is in, given that I know the pixel
> value.

David,

I believe reverse_indices are the way to go. First the short answer:

given:
A - the array containing the data (your image)
V - the data (pixel) value at the chosen location
R - the reverse_indices returned from the histogram function
    (and I don't care what you chose as a bin size, that is incorporated
     into R)

The number of the bin in which the data value resides can be calculated
as:

bin=(where(R ge ((where(((where(A eq V))[0]) eq
R[R[0]:*]))[0]+R[0])))[0]

and the number of elements in that bin is then:

num=R[bin+1]-R[bin]

Now the long answer:

reverse_indices gives you the number of elements in each bin and tells
you in which bin each element of your original array falls.

For example:
A=[1,2,3,4,3,2,2]
h=histogram(A,bin=2,reverse=R) ;note: binsize does not matter to this
 function, I chose 2 to make R smaller.
  now R = [3, 7, 10, 0, 1, 5, 6, 2, 3, 4]

What R is telling you is that the subscripts of the items in A in the
first bin are the ones found in R[3:7] or A[0,1,5,6] which are the
positions of the 1's and 2's in A. The second bin is R[7:10] or
A[2,3,4], the 3's and 4.

Choose a value that does exist in the array A: e.g., V = 3
First determine a location in A where V exists, for the sake of ease we

will choose the first occurance of V

```
w1=(where(A eq V))[0]   ;here w1 = 2
```

Since we know V has to exist in A we don't have to check for where returning -1

Then we find the location of that position in the vector R

```
w2=(where(w1 eq R[R[0]:*]))[0]+R[0]    ;here w2 = 7
```

R[0] happens to be the index of the first location of the portion of R where the separate array subscripts are stored. Adding R[0] to the where() returns the actual value. Again, we know the value exists so no need to check for a -1

The first part of R lists the indices in R of where the subscripts of A are found in each individual bin. Just look for the first value that is greater than or equal to w2.

```
bin=(where(r ge w2))[0]   ;here bin = 1 or the value of 3 is found in
the
 2nd bin.
```

The long line at the top condenses these steps into one confusing line. As before, finding the number of elements in that bin is straightforward.

```
num=R[bin+1]-R[bin]
```

I hope this helps you. As I noted before, odds are either this solution will not work for you or a better solution will soon present itself.

Cheers,
eddie

----- ---- --- --- ---- --- -- --- --- -- -- - - - -
|\   A G Edward Haskell
|\   Center for Coastal Physical Oceanography
|\ Old Dominion University, Norfolk VA 23529
|\   Voice 757.683.4816   Fax 757.683.5550
|\   e-mail haskell*ccpo.odu.edu
----- ---- --- ---- --- --- --- --- -- -- -- - - -   -