Subject: WHERE Function
Posted by bjp8350 on Wed, 10 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

I am using the WHERE function to return the pixel locations of a single object
in a binary image. The WHERE function returns a LONGWORD VECTOR that can be
used to subscript the image array. I need to calculate the distance between
each combination of border pixels to find the object's major axis.

How can I seperate the LONGWORD VECTOR into its X and Y components? so I can
use the distance formula. Or is there an easier way to find the distance
between each pixel?

Thanks in advance,
Ben
--
****
Ben Pryhoda
bjp8350@rit.edu
Rochester Institute of Technology

Subject: Re: WHERE Function
Posted by chris on Thu, 11 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

PRYHODA (bjp8350@osfmail.isc.rit.edu) wrote:
: I am using the WHERE function to return the pixel locations of a single object
: in a binary image. The WHERE function returns a LONGWORD VECTOR that can be
: used to subscript the image array. I need to calculate the distance between
: each combination of border pixels to find the object's major axis.

: How can I seperate the LONGWORD VECTOR into its X and Y components? so I can
: use the distance formula. Or is there an easier way to find the distance
: between each pixel?

There is probably an easier way, but here's how you can find x and y.

```
Nx = 4 & Ny = 10  ; Dimensions of the Array
im = findgen(Nx) # findgen(Ny)  ; Make an arbitrary array.
w = where( im eq 27)  ; w = longword type
x = w mod Nx
y = w / Nx
print, im(x,y)   ; check: should = 27.
```

You might consider playing with the dist(_) function, or with
shift.

-chris

Subject: Re: WHERE Function
Posted by chase on Thu, 11 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

>>>> > "Pryhoda" == PRYHODA <bjp8350@osfmail.isc.rit.edu> writes:
In article <1995May10.175428.15046@ultb.isc.rit.edu> bjp8350@osfmail.isc.rit.edu (PRYHODA) writes:

Pryhoda> How can I seperate the LONGWORD VECTOR into its X and Y
Pryhoda> components? so I can use the distance formula. Or is there an
Pryhoda> easier way to find the distance between each pixel?

I have a function the changes a one-dimensional index for an array
into its multi-dimensional version. It more generally performs
decoding for arbitrary dimensions/radix/bases.

It is not necessarily the best solution for your particular problem,
but I find it useful in a large variety of instances.

I have included the function below. See the EXAMPLE section in the
header.

I hope this is useful. If you have improvements, please let me know.

Thanks,
Chris

--- Begin included file ------
```
FUNCTION Decode, scl, dim, help=help, quiet=quiet
;+
; $Id: decode.pro,v 1.4 1995/05/10 16:20:27 chase Exp $
;
; NAME:
;
;   DECODE
;
; PURPOSE:
;
;   Decode a vector of scalars into the dimensions d1,...,dn in order
;   of increasing significance.  Useful for conversions between
;   different base/radix, time conversions, etc.  See EXAMPLE below.
;
; CATEGORY:
;
;   Mathematical Functions
```

```
;
; CALLING SEQUENCE:
;
;   Result = DECODE(Scl, Dim)
;
; INPUTS:
;
;   Scl - Vector of scalars to be decoded.
;
;   Dim - If a scalar, then it is used as a repeated base for the
;         decoding, i.e., D1,...,DN=Dim.
;         If a 1 dimensional vector, then it is taken as the
;         dimensions D1,...,DN.
;         If > 1 dimensional array, then the dimensions of the array
;         are used for D1,...,DN.
;         The dimensions increase in significance, i.e., the first
;         dimension is the least significant and the last dimension is
;         the most significant.
;
;         Dim need not be integral.
;
;         D1*D2*...*DN must be representable to the precision of a double
;         for the results to be accurate.
;
; KEYWORD PARAMETERS:
;
;   HELP - Provide help (this information). No other action is performed.
;
;   QUIET - Do not print warning when Scl is outside total dimensions.
;
; OUTPUTS:
;
;   Result - Array of size NxM where M is dimension of Scl.  Array has
;            the same type as Scl.  Result(*,i) is the decoding of the
;            scalar Scl(i). If Scl(i) is larger then the dimensioned
;            size, i.e. D1*D2*...*DN, then the modulus, Scl(i) mod
;            D1*D2*...*DN, is decoded.  Result(j-1,i) corresponds to
;            dimension Dj with Result(N-1,i) the most significant
;            digit of the decoding.
;
; PROCEDURE:
;
;   Let b0,...,bN be the decoding. Then Scl can be represented as:
;
;   Scl = D1*D2*...*D[N-1]*bN + ... + D1*D2*b3 + D1*b2 + b1
;
;       = D1(D2(...(D[N-1]*bN + b[N-1])...+ b2 ) + b2) + b1
;
```

```
;
;   with 0 <= bi < Di, and b2,...,bN integral.
;   If Scl is floating point then b1 may not be integral.
;
;   The representation is unique for Scl, i.e., there are not two
;   distinct decodings resulting in the same Scl.
;
; EXAMPLE:
;
;   scl = [20,63]
;   ; Conversion to base 16
;   print,decode(scl,16)
;
;   ; Convert times in seconds to second, minute, hour, day
;   t = [1.,60.,3600.d0,24.*3600d0] # [[32,55,10,2],[59,0,23,364]]
;   print,t
;   print,decode(t,[60,60,24,365])
;
;   ; Conversion to binary (base 2)
;   print,decode(scl,2)
;   ; Invert the decoding
;   print,2^indgen(5)#decode(scl,2)
;
;   ; Arbitrary decoding.  Generates a warning for decoding 63 in
;   ; which case (63 mod 3*4*5) = 3 is decoded.
;   print,decode(scl,[3,4,5])
;   print,[1,3,3*4]#decode(scl,[3,4,5])
;   print,[1,3,3*4,3*4*5]#decode(scl,[3,4,5,6])
;
;   ; Convert 1D index into a multi-dimensional index
;   w=dist(20,20)
;   a=max(w,i)
;   ; Get 2D index for max
;   print,decode(i,w)
;
; MODIFICATION HISTORY:
;
;       Mon Feb 27 16:13:20 1995, Chris Chase S1A
;       <chase@retro.jhuapl.edu>
;
;               Handles non-integral dimensions and inputs.
;
;       Mon Jul 18 15:58:18 1994, Chris Chase S1A <chase@jackson>
;               Fixed/cleaned up.
;
;       Mon Jul 26 12:17:56 1993, Chris Chase <chase@aphill>
;               Created.  Named for similar APL function.
;
```

```
;-
if keyword_set(help) then begin
    doc_library, 'decode'
    return, ''
endif
s = size(dim)
if (s(0) eq 0) then begin
    d = replicate(dim, ceil(alog(max(scl))/alog(dim)))
endif else begin
    if (s(0) gt 1) then d = s(1:s(0)) $
    else d = dim
endelse

nd = n_elements(d)

;; Use double to make sure it is big enough
dd = double(d)
for i=1, nd-1 do dd(i) = dd(i)*dd(i-1)
v = scl
if max(v/dd(nd-1)) gt 1 then begin
    if not keyword_set(quiet) then begin
        print, "Warning - function DECODE: scalar outside dimension " + $
          "bounds, decode of modulus returned."
    endif
    v(*) = v(*) mod dd(nd-1)
endif

index = replicate(v(0), nd, n_elements(v))
for i = nd-1, 1, -1 do begin
    f = long(v/dd(i-1))
    index(i, *) = f
    v = v-f*dd(i-1)
endfor
index(0, *) = v
return, index
end
```

--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu

Subject: Re: WHERE Function
Posted by rivers on Fri, 12 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <1995May10.175428.15046@ultb.isc.rit.edu>, bjp8350@osfmail.isc.rit.edu (PRYHODA) writes:
> I am using the WHERE function to return the pixel locations of a single object
> in a binary image. The WHERE function returns a LONGWORD VECTOR that can be
> used to subscript the image array. I need to calculate the distance between
> each combination of border pixels to find the object's major axis.
>
> How can I seperate the LONGWORD VECTOR into its X and Y components? so I can
> use the distance formula. Or is there an easier way to find the distance
> between each pixel?
>

Here is a function I wrote to do just that.

Mark Rivers (312) 702-2279 (office)
CARS (312) 702-9951 (secretary)
Univ. of Chicago (312) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars3.uchicago.edu (Internet)


```
function convert_index, index, array

;+
; NAME:
;       CONVERT_INDEX
; PURPOSE:
;       Converts a one dimensional array index into a vector of indices, whose
;       length is equal to the number of dimensions of the array. This is
;       useful when wanting to know, for instance, what row and column element
;       10034 corresponds to in a 200x150 2-D array. The routine is general and
;       can handle arrays with any number of array dimensions, up to the IDL
;       maximum of 7.
; CALLING SEQUENCE:
;       new_index = CONVERT_INDEX(index, array)
; INPUTS:
;   INDEX
;       A 1 dimensional array index to be converted. IDL can reference
;       multidimensional arrays using a simple 1 dimensional index.
;       Such an index is obtained, for instance from functions such as
;       MAX, MIN and WHERE.
;   ARRAY
;       The array to which this index applies. This routine only uses this
```

```
;       parameter to determine the array dimensions, it does not actually use
;       the data stored in the array.
; OUTPUTS:
;   NEW_INDEX
;       The function returns an array of indices, in increasing array index
;       order. NEW_INDEX has a maximum lenght of 7, since IDL arrays are limited
;       to 7 dimensions.
; EXAMPLE:
;       If ARRAY is a 4x3 array and INDEX=7 then this function will return
;       [3,1], since array element 7 (when ARRAY is viewed as a
;       one-dimensional array) is actually column 3, row 1 when ARRAY is viewed
;       as a 2-dimensional array.
; MODIFICATION HISTORY:
;       Created October 1990 by Mark Rivers
;-


nd = size(array)
ndims = nd(0)

denom = 1
for i=1, ndims do denom = denom * nd(i)
result = lonarr(7)

for i=ndims, 0, -1 do begin
  result(i) = index / denom
  index = index MOD denom
  denom = denom / nd(i)
endfor

return, result(0:ndims-1)
end
```

Subject: Re: where function
Posted by jeanh on Sun, 23 Jan 2011 21:12:50 GMT

View Forum Message <> Reply to Message

On 23/01/2011 3:26 PM, smuzz wrote:
> Hi --
>
> I am a beginner programmer trying to write a code involving the where
> function. My dataset includes call detections made by whales from 1
> month. I am trying to search within this dataset for a particular call
> type, let's call it a moan. Then I want to search 1 hour before or
> after a detected moan and tally up all the other whale calls I hear --
> this is the part I am stuck on. How do you use the where function to
> search 1 hour before or after? I am only familiar with using gt and

> lt.
>
> This is what I have so far...
>
> n=n_elements(auto) ; auto is my dataset with all whale calls
>
> for i=0, n-1 do begin
> k = where(auto(*).manual_species eq 0 and auto(*).manual_call_type eq
> 1 and auto(*).time gt stime and auto(*).time lt etime,kcount) ; this
> is looking for all moans within a specified start and end time of the
> 1 month study
>
> if(kcount gt 0) then begin
>
> j=where(auto(*).mdist lt 3.0 and auto(*).avg_amplitude ge 12.0 and
> auto(*).time gt stime and auto(*).time lt etime and
> auto(*).manual_species eq 7 and auto(*).manual_call_type eq
> 1,jcount) ; this is looking for all other whale calls within the
> dataset
>
> ....right now j is tallying up all the calls in the dataset, but now I
> want to add in a part asking for only calls that occur 1 hour before
> or after a moan detection?
>
> Any ideas?
>
> Thanks, smuzz

Hi Smuzz,

glad to see I am not the only one working this Sunday :-)

First, sorry, but the 1st loop is useless (for i=0, n-1 do begin)
... indeed, you are never using i!

Here is what you want to do (it could probably be optimized though)

1) Select the entries from the dataset that have the proper time (month)
and call type. Count the number of results

2) Loop through these results, and make your 2nd selection by
adding/removing time

3) process

---
to use your code, it would look like this

```
k = where(auto(*).manual_species eq 0 and auto(*).manual_call_type eq
1 and auto(*).time gt stime and auto(*).time lt etime,kcount) ; this
is looking for all moans within a specified start and end time of the
1 month study

for entry = 0, kcount -1 do begin
  linkedWhalesID = where(auto(*).mdist lt 3.0 and auto(*).avg_amplitude
ge 12.0 and
auto(*).time gt auto[k[entry]].time - 60 and auto(*).time lt
auto[k[entry]].time + 60 and
auto(*).manual_species eq 7 and auto(*).manual_call_type eq
1,jcount)

  WhalesToAnalyze = auto[linkedWhalesID]
endfor
```

Jean
