Subject: Re: Repeats and Triangulation
Posted by Craig Markwardt on Mon, 26 Feb 2001 22:31:05 GMT
View Forum Message <> Reply to Message

Ben Tupper <pemaquidriver@tidewater.net> writes:
> The REPEATS keyword to TRIANGULATION returns a 2,n element
> array of pairs indices of repeated values. If no values are
> repeated then REPEATS = [-1,-1]. My question is 'How do I
> pull out the repeated values efficiently?'
>

...

```
>   ;make the polygon descriptor (see IDLgrPolyLine)
>  List = Conn[Conn[0]:Conn[1]-1L]
>  Ptr = Ptr_NEW([N_elements(List),List])
>  For  i = 1, n_elements(X) -1 DO Begin
>   List = Conn[conn[i]:Conn[i+1]-1]
>   *Ptr = [*Ptr, n_elements(List), List]
>  EndFor
```

Umm, while the procedure does not document how it treats repeated
points -- which might be considered a documentation bug -- the format
of the adjacency list is very similar to the output from the
REVERSE_INDICES keyword of HISTOGRAM. That is, there are bins that
have no entries, and there is a way to ignore them. How about
inserting an IF clause which tests for this?

```
List = Conn[Conn[0]:Conn[1]-1L]
Ptr = Ptr_NEW([N_elements(List),List])
For  i = 1, n_elements(X) -1 DO IF CONN[i+1] GT CONN[i] then Begin
 List = Conn[conn[i]:Conn[i+1]-1]
 *Ptr = [*Ptr, n_elements(List), List]
EndIf
```

The above solution works for me.

Your other options are to:
 * pre-check your point list to remove repeats. This might be done
   similar to UNIQ.
 * use the REPEATS array to remove duplicates.

```
   DELMASK = lonarr(n_elements(x))
   DELMASK (REP(1,*)) = 1
```

   and then check DELMASK(i) EQ 1 before adding the point to your list.

Good luck,
Craig

--

```
 ------------------------------------------------------------ --------------
Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 ------------------------------------------------------------ --------------
```

Subject: Re: Repeats and Triangulation
Posted by Ben Tupper on Mon, 26 Feb 2001 23:48:02 GMT

View Forum Message <> Reply to Message

Thanks Craig,

I think I'll pursue you final suggestion. The tricky part there is that it is possible to have indices repeated not only within column 1 but also apprearing in column 2.

Ben

Craig Markwardt wrote:

```
>  Ben Tupper <pemaquidriver@tidewater.net> writes:
>>  The REPEATS keyword to TRIANGULATION returns a 2,n element
>>  array of pairs indices of repeated values.  If no values are
>>  repeated then REPEATS = [-1,-1].  My question is 'How do I
>>  pull out the repeated values efficiently?'
>>
>  ...
>>   ;make the polygon descriptor (see IDLgrPolyLine)
>>  List = Conn[Conn[0]:Conn[1]-1L]
>>  Ptr = Ptr_NEW([N_elements(List),List])
>>  For  i = 1, n_elements(X) -1 DO Begin
>>   List = Conn[conn[i]:Conn[i+1]-1]
>>   *Ptr = [*Ptr, n_elements(List), List]
>>  EndFor
>
>  Umm, while the procedure does not document how it treats repeated
>  points -- which might be considered a documentation bug -- the format
>  of the adjacency list is very similar to the output from the
>  REVERSE_INDICES keyword of HISTOGRAM.  That is, there are bins that
>  have no entries, and there is a way to ignore them.  How about
>  inserting an IF clause which tests for this?
>
>  List = Conn[Conn[0]:Conn[1]-1L]
>  Ptr = Ptr_NEW([N_elements(List),List])
>  For  i = 1, n_elements(X) -1 DO IF CONN[i+1] GT CONN[i] then Begin
>   List = Conn[conn[i]:Conn[i+1]-1]
>   *Ptr = [*Ptr, n_elements(List), List]
>  EndIf
```

>
> The above solution works for me.
>
> Your other options are to:
> * pre-check your point list to remove repeats. This might be done
>   similar to UNIQ.
> * use the REPEATS array to remove duplicates.
>   DELMASK = lonarr(n_elements(x))
>   DELMASK (REP(1,*)) = 1
>   and then check DELMASK(i) EQ 1 before adding the point to your list.
>
> Good luck,
> Craig
>
> --
> -- --------------
> Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> -- --------------

--
Ben Tupper
248 Lower Round Pond Road
POB 106
Bristol, ME 04539

Tel: (207) 563-1048
Email: PemaquidRiver@tidewater.net