
Subject: Re: chunk indexing like
Posted by [Wout De Nolf](#) on Fri, 08 Jan 2010 14:12:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Fri, 08 Jan 2010 14:08:45 +0100, Wox <spam@nomail.com> wrote:

> Hi IDLers,
>
> I know how to get
> from [2,3,1,0,5] to [0,0,1,1,1,2,4,4,4,4,4] (i.e. chunk indexing)
> but how do I get
> from [2,3,1,0,5] to [0,1,0,1,2,0,0,1,2,3,4]
> without loops that is.
>
> I have been juggling with histogram and total(.../cumulative) but I
> can figure this one out.
>
> Any ideas?

Ok, I got something working:

```
n=[2,3,1,0,5]
h=histogram(total(n>0,/CUMULATIVE,/int)-1,/BINSIZE,MIN=0,REVERSE_INDICES=ri)
nh=n_elements(h)
chinkind=ri[0:nh-1]-ri[0]
ind2=where([1,chinkind[1:]*]-chinkind[0:nh-2]) ne 0)
print,lindgen(nh)-ind2[chinkind]
```

```
IDL> 0 1 0 1 2 0 0 1 2 3 4
```

Subject: Re: chunk indexing like
Posted by [pgrigis](#) on Fri, 08 Jan 2010 16:16:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 8, 9:12 am, Wox <s...@nomail.com> wrote:

> On Fri, 08 Jan 2010 14:08:45 +0100, Wox <s...@nomail.com> wrote:
>> Hi IDLers,
>
>> I know how to get
>> from [2,3,1,0,5] to [0,0,1,1,1,2,4,4,4,4,4] (i.e. chunk indexing)
>> but how do I get
>> from [2,3,1,0,5] to [0,1,0,1,2,0,0,1,2,3,4]
>> without loops that is.
>
>> I have been juggling with histogram and total(.../cumulative) but I
>> can figure this one out.
>

```

>> Any ideas?
>
> Ok, I got something working:
>
> n=[2,3,1,0,5]
> h=histogram(total(n>0,/CUMULATIVE,/int)-1,/BINSIZE,MIN=0,REVERSE_INDICES=ri)
> nh=n_elements(h)
> chinkind=ri[0:nh-1]-ri[0]
> ind2=where([1,chinkind[1:]*]-chinkind[0:nh-2]) ne 0)
> print,lindgen(nh)-ind2[chinkind]
>
> IDL> 0 1 0 1 2 0 0 1 2 3 4

```

I wonder how does the IDL-magic version above compares with the more boring for-loop version below in execution speed?

I.e. is it worth to go the histogram-way, rewarding but fraught with danger? ;)

Ciao,
Paolo

```
n=[2,3,1,0,5]
```

```
res=intarr(total(n))
indexarr=findgen(max(n))
i2=0
```

```
.run
FOR i=0,n_elements(n)-1 DO BEGIN
  IF n[i] GE 1 THEN BEGIN
    res[i2:i2+n[i]-1]=indexarr[0:n[i]-1]
    i2=i2+n[i]
  ENDIF
ENDFOR
end
```

Subject: Re: chunk indexing like
 Posted by [rogass](#) on Sat, 09 Jan 2010 09:37:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi, Paolo indirectly shows that loops are sometimes not evil ;) I tried this one:

```
pro cr_test_chunk_indexing,n=n,verbose=verbose
n= keyword_set(n)? n : [2,3,1,0,5]
```

```

if keyword_set(verbose) then print,n

t0=systime(1)
l1=((l=lindgen(((nm=n_elements(n)>(max(n))))),nm) mod nm))[where((l lt
transpose(rebin(n,nm,nm,/sample)))]
print,'CR-Rebin/mod-approach: ',systime(1)-t0,' seconds
for ',n_elements(n), ' elements'
if keyword_set(verbose) then print,l1

t0=systime(1)
h=histogram(total(n>0,/CUMULATIVE,/int)-1,/
BINSIZE,MIN=0,REVERSE_INDICES=ri )
nh=n_elements(h)
chinkind=ri[0:nh-1]-ri[0]
ind2=where([1,chinkind[1:*]-chinkind[0:nh-2]] ne 0)
l1=lindgen(nh)-ind2[chinkind]
print,'Histogram-approach: ',systime(1)-t0,' seconds for ',n_elements
(n), ' elements'
if keyword_set(verbose) then print,l1

t0=systime(1)
res=intarr(total(n))
indexarr=findgen(max(n))
i2=0
FOR i=0,n_elements(n)-1 DO BEGIN
  IF n[i] GE 1 THEN BEGIN
    res[i2:i2+n[i]-1]=indexarr[0:n[i]-1]
    i2=i2+n[i]
  ENDIF
ENDFOR
print,'Loop-approach: ',systime(1)-t0,' seconds for ',n_elements(n),
' elements'
if keyword_set(verbose) then print,res

end

```

...and got this one:

```

IDL> cr_test_chunk_indexing, n=round(((i=10))*randomu(seed,i)),/
verbose
      9      6      3      0      6
1      2      8      7      5
Rebin/mod-approach: 0.00000000 seconds for , 10
elements
      0      1      2      3      4
5      6      7      8      0
1      2      3      4      5
      0      1      2      0      1

```

```

2    3    4    5    0
0    1    0    1    2
      3    4    5    6    7
0    1    2    3    4
5    6    0    1    2
      3    4

```

Histogram-approach: 0.00000000 seconds for , 10 elements

```

      0    1    2    3    4
5    6    7    8    0
1    2    3    4    5
      0    1    2    -6    -5
-4    -3    -2    -1    -1
-2    -1    -8    -7    -6
      -5    -4    -3    -2    -1
-7    -6    -5    -4    -3
-2    -1    0    1    2
      3    4

```

Loop-approach: 0.00000000 seconds for , 10 elements

```

      0    1    2    3    4    5    6    7
8    0    1    2    3    4    5    0
1    2    0    1    2    3    4
      5    0    0    1    0    1    2    3
4    5    6    7    0    1    2    3
4    5    6    0    1    2    3
      4

```

IDL> cr_test_chunk_indexing, n=round(((i=100))*randomu(seed,i))

Rebin/mod-approach: 0.0010001659 seconds for , 100 elements

Histogram-approach: 0.00000000 seconds for , 100 elements

Loop-approach: 0.00000000 seconds for , 100 elements

IDL> cr_test_chunk_indexing, n=round(((i=1000))*randomu(seed,i))

Rebin/mod-approach: 0.050000191 seconds for , 1000 elements

Histogram-approach: 0.017000198 seconds for , 1000 elements

Loop-approach: 0.0049998760 seconds for , 1000 elements

IDL> cr_test_chunk_indexing, n=round(((i=10000))*randomu(seed,i))

Rebin/mod-approach: 3.9480000 seconds for , 10000 elements

Histogram-approach: 2.0690000 seconds for , 10000 elements

Loop-approach: 0.46099997 seconds for , 10000 elements

So, sometimes histogram method fails. Loop is here the fastest one and my one is the slowest but right one without loops.

JD? Any improvements? :)

Regards

CR

Subject: Re: chunk indexing like
Posted by [rogass](#) on Sat, 09 Jan 2010 10:15:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ok just a little improvement:

change the rebin approach to this:

```
I1=((I=lindgen(((nm=n_elements(n)>(max(n))))),nm) mod nm))[where((I lt  
(rebin(transpose(n),nm,nm,/sample)))]
```

Now its faster than the histogram-approach for large indices...

Regards

CR

Subject: Re: chunk indexing like
Posted by [David Fanning](#) on Sat, 09 Jan 2010 15:10:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wox writes:

- > I know how to get
- > from [2,3,1,0,5] to [0,0,1,1,1,2,4,4,4,4,4] (i.e. chunk indexing)
- > but how do I get
- > from [2,3,1,0,5] to [0,1,0,1,2,0,0,1,2,3,4]
- > without loops that is.
- >
- > I have been juggling with histogram and total(.../cumulative) but I
- > can figure this one out.

Here is my problem with this, and with Chris's answers:
is this an intellectual exercise, or is there a reason
for wanting to do something like this? :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: chunk indexing like
Posted by [rogass](#) on Sat, 09 Jan 2010 16:50:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 9 Jan., 16:10, David Fanning <n...@dfanning.com> wrote:
> Wox writes:
>> I know how to get
>> from [2,3,1,0,5] to [0,0,1,1,1,2,4,4,4,4,4] (i.e. chunk indexing)
>> but how do I get
>> from [2,3,1,0,5] to [0,1,0,1,2,0,0,1,2,3,4]
>> without loops that is.
>
>> I have been juggling with histogram and total(.../cumulative) but I
>> can figure this one out.
>
> Here is my problem with this, and with Chris's answers:
> is this an intellectual exercise, or is there a reason
> for wanting to do something like this? :-(
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

David,
I will take this approach to address the different features within my
varying spectral feature database. So, it helps me too. Nevertheless,
the problem was interesting for me, because it also touches somehow
the missing possibility of IDL to do (arr
[indexarray1:indexarray1+x,indexarray3:indexarray3+y])[*]

However, I assumed that WOX wanted help or searched for a different

approach due to his/her problem. Some people here - especially you and your website - helped me and so I try to help others too if I can. It would be wonderful, if ITTVIS would reflect or solve the most common problems posted to this newsgroup in their upcoming IDL releases.

Regards

CR

Subject: Re: chunk indexing like
Posted by [David Fanning](#) on Sat, 09 Jan 2010 17:53:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

chris writes:

> It would be wonderful, if ITTVIS would reflect or solve
> the most common problems posted to this newsgroup in
> their upcoming IDL releases.

Yeah, and if everyone dropped a dime into the kitty whenever they used a tip from my web page, I wouldn't have to work for a living. ;-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: chunk indexing like
Posted by [rogass](#) on Sat, 09 Jan 2010 19:07:17 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 9 Jan., 18:53, David Fanning <n...@dfanning.com> wrote:

> chris writes:
>> It would be wonderful, if ITTVIS would reflect or solve
>> the most common problems posted to this newsgroup in
>> their upcoming IDL releases.
>

> Yeah, and if everyone dropped a dime into the kitty whenever
> they used a tip from my web page, I wouldn't have to work
> for a living. ;-)
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

LOL, maybe you should sell some kind of an 'IDL TIP Flatrate' to your then password protected website or maybe you should sell Coyote-cookies with 'IDL cooking tips' inside :) Nevertheless, I bought your recommendable book and your Snake-algorithm which helped me a lot to understand how object- and widget programming really works, I hope ;) So, maybe some of us understand the IDL way, but maybe you live the IDL way: David 'idl_assistant.exe' Fanning >:-)

Cheers

CR

Subject: Re: chunk indexing like
Posted by [Wout De Nolf](#) on Sun, 10 Jan 2010 12:22:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Sat, 9 Jan 2010 08:10:45 -0700, David Fanning <news@dfanning.com> wrote:

> Here is my problem with this, and with Chris's answers:
> is this an intellectual exercise, or is there a reason
> for wanting to do something like this? :-(

Why the :-(? Do I offend you by asking a question? Unexpectedly I did find an answer myself 1 hour later. So sorry for wasting your time.

Btw, I would never spam newsgroups with intellectual exercises. This particular problem popped up when implementing an ART tomographic reconstruction.

Subject: Re: chunk indexing like

Posted by [Wout De Nolf](#) on Sun, 10 Jan 2010 13:09:44 GMT

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

On Sat, 9 Jan 2010 01:37:18 -0800 (PST), chris <rogass@googlemail.com> wrote:

> So, sometimes histogram method fails.

You're right. Thanks for checking. It happens because of zero's in n.

The thing is, I need chunk indexing anyway so it would be nice to combine the two. In other situations, loops would be fine as you and Paolo showed.

Subject: Re: chunk indexing like

Posted by [Wout De Nolf](#) on Sun, 10 Jan 2010 13:52:14 GMT

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

On Sat, 9 Jan 2010 01:37:18 -0800 (PST), chris <rogass@googlemail.com> wrote:

```
> h=histogram(total(n>0,/CUMULATIVE,/int)-1,/
> BINSIZE,MIN=0,REVERSE_INDICES=ri )
> nh=n_elements(h)
> chinkind=ri[0:nh-1]-ri[0]
> ind2=where([1,chinkind[1:]*]-chinkind[0:nh-2]) ne 0)
> l1=lindgen(nh)-ind2[chinkind]
```

The code below solves the histogram approach error but even when excluding the histogram from timing (I need chunk indexing anyway), the loop approach is the fastest.

```
h=histogram(total(n>0,/CUMULATIVE,/int)-1,/BINSIZE,MIN=0,REVERSE_INDICES=ri)
nh=n_elements(h)
chunkind=ri[0:nh-1]-ri[0]
```

```
t0=systeme(1)
tmp=[1,chunkind[1:]*]-chunkind[0:nh-2]
ind2=where(tmp ne 0)
tmp=total((tmp-1)>0,/cum,/int)
l2=lindgen(nh)-ind2[chunkind-tmp]
print,'Histogram-approach: ',systeme(1)-t0,' seconds for
',n_elements(n), ' elements'
```

Timing:

CR-Rebin/mod-approach: 1.4690001 seconds for , 10000 elements
Histogram-approach: 2.3290000 seconds for , 10000 elements
Loop-approach: 0.82800007 seconds for , 10000 elements

Subject: Re: chunk indexing like
Posted by [David Fanning](#) on Sun, 10 Jan 2010 15:07:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Wox writes:

> Why the :-(? Do I offend you by asking a question? Unexpectedly I did
> find an answer myself 1 hour later. So sorry for wasting your time.

No, the frowny face is because I don't have enough imagination, apparently, to come up with an example of where this kind of thing is useful. I assume it must be, because it comes up over and over. But the question, as far as I know, has never been presented with its context. I'm one of those people who have a difficult time learning things out of context. So, sorry for wasting YOUR time! :-)

Cheers,

David

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: chunk indexing like
Posted by [Wout De Nolf](#) on Sun, 10 Jan 2010 16:05:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Sun, 10 Jan 2010 08:07:00 -0700, David Fanning <news@dfanning.com> wrote:

> But the
> question, as far as I know, has never been presented
> with its context.

Ok, you asked for it :-).

Suppose you have two lines crossing a NxN grid.

$t = \text{lindgen}(N+1) - 0.5$; the grid

$y_1 = m_1 * t + b_1$; line 1

$y_2 = m_2 * t + b_2$; line 2

Now you want to calculate the area of each pixel of the NxN grid that is covered by the area between the two lines. This is illustrated in (lines don't have to be parallel) <http://xrdua.ua.ac.be/public/tmp.jpg>

To do this I first take all intersections between the two lines and the hor. and vert. gridlines:

```
x=[t,(t-b1)/m1,t,(t-b2)/m2]
```

```
y=[y1,t,y2,t]
```

Then I add grid nodes that are in between the two lines

```
y1=round(y1)
```

```
y2=round(y2)
```

```
nadd=y2-y1
```

```
if total(nadd) ne 0 then begin
```

```
  xadd=chunkindex(nadd)
```

```
  yadd=chunklindgen(nadd) ; THIS WAS MY QUESTION
```

```
  x=[x,xadd-0.5]
```

```
  y=[y,y1[xadd]+yadd+0.5]
```

```
endif
```

Then I use TRIANGULATE to make triangles. Each triangle lies within 1 pixel so calculating the area and the center of the triangles will solve my original problem (an additional drizzling is used because 1 pixel will be the sum of several triangles). Some triangles have to be rejected too.

So the "chunked lindgen" (my question) is used to calculate y coordinates of grid nodes between the two lines. For each x there are zero or more grid nodes, hence the need for some "chunked lindgen".

Subject: Re: chunk indexing like

Posted by [David Fanning](#) on Sun, 10 Jan 2010 16:09:59 GMT

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

Wox writes:

> Ok, you asked for it :-).

>

- > Suppose you have two lines crossing a NxN grid.
- > `t=lindgen(N+1)-0.5` ; the grid
- > `y1=m1*t+b1` ; line 1
- > `y2=m2*t+b2` ; line 2

Ah, ha! I am working on a similar problem with map grids. Maybe this sheds some light on that...

Thanks! :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")
