Subject: Adding elements in an Array
Posted by bhattacharjee12 on Sun, 22 Sep 2013 18:31:49 GMT
View Forum Message <> Reply to Message

Hi,
I am new to IDl. I was hoping if you guys can help with my problem.
I will explain my problem with an example.
Lets say I have an array
x=[1,2,3,4,5,6,1,2,3]
& I want to average lets say every 3 elements and put the results in a new array say y
Where y would look like
y=[2,5,2]
Is there a nifty IDL function/easy way which would do that?
Thanks for the help in advance.
Ani

Subject: Re: Adding elements in an Array
Posted by David Fanning on Sun, 22 Sep 2013 18:46:21 GMT
View Forum Message <> Reply to Message

bhattacharjee12@gmail.com writes:

>
> Hi,
> I am new to IDl. I was hoping if you guys can help with my problem.
> I will explain my problem with an example.
> Lets say I have an array
> x=[1,2,3,4,5,6,1,2,3]
> & I want to average lets say every 3 elements and put the results in a new array say y
> Where y would look like
> y=[2,5,2]
> Is there a nifty IDL function/easy way which would do that?

IDL> x=[1,2,3,4,5,6,1,2,3]
IDL> x = reform(x, 3, 3)
IDL> b = Mean(x, dimension=1)
IDL> print, b
      2.00000   5.00000   2.00000

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by bhattacharjee12 on Sun, 22 Sep 2013 19:13:36 GMT
View Forum Message <> Reply to Message

Thanks a lot, David. Reform will be a great help.

Subject: Re: Adding elements in an Array
Posted by natha on Mon, 23 Sep 2013 17:46:09 GMT
View Forum Message <> Reply to Message

y = x[1:*:3]

cheers,

nata

Subject: Re: Adding elements in an Array
Posted by wlandsman on Mon, 23 Sep 2013 18:30:07 GMT
View Forum Message <> Reply to Message

On Monday, September 23, 2013 1:46:09 PM UTC-4, nata wrote:
> y = x[1:*:3]

That takes every 3rd element starting with element 1.   It gives the right answer in this case but just by coincidence, e.g. element 1 just happens to be the mean of the first 3 values, and element 4 happens to be the average of the second 3 values.

My solution would be

IDL> y = rebin(x,3)

--Wayne

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 19:37:57 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> That takes every 3rd element starting with element 1.   It gives the right answer in this case but just by coincidence, e.g. element 1 just happens to be the mean of the first 3 values, and element 4 happens to be the average of the second 3 values.
>
> My solution would be

>
> IDL> y = rebin(x,3)

It's not clear to me that nearest neighbor sampling is the same as the
"mean" value in general, although it works in this specific case.

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by wlandsman on Mon, 23 Sep 2013 20:14:06 GMT

On Monday, September 23, 2013 3:37:57 PM UTC-4, David Fanning wrote:

>
> It's not clear to me that nearest neighbor sampling is the same as the
>
> "mean" value in general, although it works in this specific case.
>

I agree.   That is why one shouldn't use the /SAMPLE keyword to REBIN() ;-)   But as the documentation for REBIN says

"Normally, REBIN uses bilinear interpolation when magnifying and neighborhood averaging when minifying."   REBIN only uses nearest neighbor sampling if the /SAMPLE keyword is set.

--Wayne

>
> Cheers,
>
>
>
> David
>
>

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

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 20:31:00 GMT

wlandsman writes:

> "Normally, REBIN uses bilinear interpolation when magnifying and neighborhood averaging when minifying."

I suppose it would take me a couple of days scratching my head to figure
this out!

Do you know this to be true, or are you just taking their word for it?
I'm not sure "neighborhood averaging" means what you appear to think it
means. If you are right, it certainly doesn't mean what I think it
means! And, in any case, I wouldn't bet the satellite on it until I ran
a couple of tests. :-)

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 20:36:34 GMT

David Fanning writes:

>
> wlandsman writes:
>
>> "Normally, REBIN uses bilinear interpolation when magnifying and neighborhood averaging
when minifying."
>
> I suppose it would take me a couple of days scratching my head to figure
> this out!
>
> Do you know this to be true, or are you just taking their word for it?
> I'm not sure "neighborhood averaging" means what you appear to think it
> means. If you are right, it certainly doesn't mean what I think it
> means! And, in any case, I wouldn't bet the satellite on it until I ran
> a couple of tests. :-)

I guess you are right. Wow! I learn something every day about IDL!

```
IDL> a = [10, 20, 5, 6, 18, 17, 1, 30, 2]
IDL> print, rebin(a, 3)
     11     13     11
IDL> print, rebin(a, 3, /sample)
     10      6      1
```

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by wlandsman on Mon, 23 Sep 2013 20:40:07 GMT

On Monday, September 23, 2013 4:31:00 PM UTC-4, David Fanning wrote:

>

> Do you know this to be true, or are you just taking their word for it?

I use this property of REBIN() all the time because astronomers are often interested in preserving the surface flux when resizing images.   And one can do a quick test by resizing to one element.

IDL> print,rebin([2.5,3.2,3.6,9.99],1)
      4.82250
IDL> print,mean([2.5,3.2,3.6,9.99])
      4.82250

Perhaps you are thinking of CONGRID() which does default to nearest neighbor sampling?
--Wayne

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 20:44:07 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> I use this property of REBIN() all the time because astronomers are often interested in
preserving the surface flux when resizing images.   And one can do a quick test by resizing to
one element.
>
> IDL> print,rebin([2.5,3.2,3.6,9.99],1)
>   4.82250
> IDL> print,mean([2.5,3.2,3.6,9.99])
>   4.82250

I hope you document this code heavily. I don't think a casual reader
will catch this "feature" without some help. I was thinking of Congrid,
because I use that more frequently than Rebin, but "averaging when
minifying" as the default behavior! That's scary to me. :-)

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by   on Mon, 23 Sep 2013 20:48:52 GMT

On 2013-09-23 22:36, David Fanning wrote:
> David Fanning writes:
>
>>
>> wlandsman writes:
>>
>>> "Normally, REBIN uses bilinear interpolation when magnifying and neighborhood averaging
when minifying."
>>
>> I suppose it would take me a couple of days scratching my head to figure
>> this out!
>>
>> Do you know this to be true, or are you just taking their word for it?
>> I'm not sure "neighborhood averaging" means what you appear to think it
>> means. If you are right, it certainly doesn't mean what I think it
>> means! And, in any case, I wouldn't bet the satellite on it until I ran
>> a couple of tests. :-)
>
> I guess you are right. Wow! I learn something every day about IDL!
>
> IDL> a = [10, 20, 5, 6, 18, 17, 1, 30, 2]
> IDL> print, rebin(a, 3)
>   11   13   11
> IDL> print, rebin(a, 3, /sample)
>   10   6   1

11 and 13 are not the mean values of [10, 20, 5] and [6, 18, 17],
respectively. 11 happens to be the mean of [1, 30, 2] but in general you
need floating point calculations:

IDL> print,rebin(float(a),3)
      11.6667   13.6667   11.0000

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 20:51:09 GMT

Mats Löfdahl writes:

> 11 and 13 are not the mean values of [10, 20, 5] and [6, 18, 17],
> respectively. 11 happens to be the mean of [1, 30, 2] but in general you
> need floating point calculations:
>
> IDL> print,rebin(float(a),3)
>   11.6667   13.6667   11.0000

Oh, I know that, but I was talking about something else entirely and
just using integer arrays to illustrate the point. :-)

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by   on Mon, 23 Sep 2013 20:54:35 GMT

On 2013-09-23 22:51, David Fanning wrote:
> Mats Löfdahl writes:
>
>> 11 and 13 are not the mean values of [10, 20, 5] and [6, 18, 17],
>> respectively. 11 happens to be the mean of [1, 30, 2] but in general you
>> need floating point calculations:
>>
>> IDL> print,rebin(float(a),3)
>>   11.6667   13.6667   11.0000
>
> Oh, I know that, but I was talking about something else entirely and
> just using integer arrays to illustrate the point. :-)

I'm sure you do. And did. But since the OP's example was with an integer
array I thought it might be pointed out. :o)

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 21:01:19 GMT

Mats Löfdahl writes:

> I'm sure you do. And did. But since the OP's example was with an integer
> array I thought it might be pointed out. :o)

Well, that's a good point. :-)

Although the answer would have been converted to a float for him if he
did it my way, rather than Wayne's. Let's hope he didn't understand
Wayne's method any better than I did. ;-)

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by   on Mon, 23 Sep 2013 21:10:45 GMT
View Forum Message <> Reply to Message

On 2013-09-23 23:01, David Fanning wrote:
> Mats Löfdahl writes:
>
>> I'm sure you do. And did. But since the OP's example was with an integer
>> array I thought it might be pointed out. :o)
>
> Well, that's a good point. :-)
>
> Although the answer would have been converted to a float for him if he
> did it my way, rather than Wayne's. Let's hope he didn't understand
> Wayne's method any better than I did. ;-)

I still don't understand it.

When you evaluate on a grid point, linear interpolation should give the
same result as nearest neighbor. So if the evaluation point is not the
center cell of the three here, then what is it?

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 21:15:43 GMT
View Forum Message <> Reply to Message

Mats Löfdahl writes:

> I still don't understand it.
>
> When you evaluate on a grid point, linear interpolation should give the
> same result as nearest neighbor. So if the evaluation point is not the
> center cell of the three here, then what is it?

Oh, man, you see where this is going, don't you? We are about to open an
ENTIRE can of worms over the very simplest question. OK, I turn you back
over to Wayne, who understands flux preservation and the workings of the
Congrid command a whole HELL of a lot better than I do. ;-)

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by   on Mon, 23 Sep 2013 21:16:32 GMT
View Forum Message <> Reply to Message

On 2013-09-23 23:15, David Fanning wrote:
> Mats Löfdahl writes:
>
>> I still don't understand it.
>>
>> When you evaluate on a grid point, linear interpolation should give the
>> same result as nearest neighbor. So if the evaluation point is not the
>> center cell of the three here, then what is it?
>
> Oh, man, you see where this is going, don't you? We are about to open an
> ENTIRE can of worms over the very simplest question. OK, I turn you back
> over to Wayne, who understands flux preservation and the workings of the
> Congrid command a whole HELL of a lot better than I do. ;-)

Except that it's not congrid... :o)

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 21:20:07 GMT
View Forum Message <> Reply to Message

Mats Löfdahl writes:

> Except that it's not congrid... :o)

I know, but I've been down this road before. It's going to come up.
Especially, the CENTER and MINUS_ONE keywords. My head is already
aching. :-(

Cheers,

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

Subject: Re: Adding elements in an Array
Posted by David Fanning on Mon, 23 Sep 2013 21:25:48 GMT
View Forum Message <> Reply to Message

David Fanning writes:

> I know, but I've been down this road before. It's going to come up.
> Especially, the CENTER and MINUS_ONE keywords. My head is already
> aching. :-(

Just as a preemptive strike, and so I can sit most of the following
discussion out, I want you to know I *am* doing the right thing in
cgResizeImage (which I use in place of Rebin and Congrid) when I am
resizing images. I wrote this code the last time I was dragged kicking
and screaming though the discussion which is sure to follow. :-)

Cheers,

David

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

Subject: Re: Adding elements in an Array
Posted by wlandsman on Mon, 23 Sep 2013 21:35:20 GMT

OK, here is why I think the behavior of REBIN is intuitive.   Like all good IDL explanations it (vaguely) involves HISTOGRAM.

ReBin means to change the binning size.   And, as with HISTOGRAM, when you change the binning size, the total sum over all bins must remain the same.   When you REBIN a 512 x 512 array to 256 x 256 array, the total amount of "stuff" in the pixels is conserved; it is just distributed over bigger pixels. (To answer Mat's question -- REBIN does *not* do linear interpolation when minifying, it does averaging.)

This gives me a chance to plug one of my favorite programs -- FREBIN
http://idlastro.gsfc.nasa.gov/ftp/pro/image/frebin.pro
This program also conserves the "stuff" in the array, but allows new bin sizes that are not necessarily an integral fraction of the old. --Wayne

Subject: Re: Adding elements in an Array
Posted by wlandsman on Tue, 24 Sep 2013 00:56:27 GMT

The priority for display programs when resizing images is not to conserve flux but to avoid spurious shifts and distortions.   A feature in the center of the image should remain in the center of the image after resizing, and yes cgResizeImage() does this correctly.

 And apologies that this thread has gotten quite far from the original question.   --Wayne

On Monday, September 23, 2013 5:25:48 PM UTC-4, David Fanning wrote:
> David Fanning writes:

> Just as a preemptive strike, and so I can sit most of the following
>
> discussion out, I want you to know I *am* doing the right thing in
>
> cgResizeImage (which I use in place of Rebin and Congrid) when I am
>
> resizing images. I wrote this code the last time I was dragged kicking
>
> and screaming though the discussion which is sure to follow. :-)
>
>
>
> Cheers,
>
>
>
> David
>

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

Subject: Re: Adding elements in an Array

Posted by   on Tue, 24 Sep 2013 01:50:26 GMT

On 2013-09-23 23:35, wlandsman wrote:
> OK, here is why I think the behavior of REBIN is intuitive.   Like all good IDL explanations it (vaguely) involves HISTOGRAM.
>
> ReBin means to change the binning size.   And, as with HISTOGRAM, when you change the binning size, the total sum over all bins must remain the same.   When you REBIN a 512 x 512 array to 256 x 256 array, the total amount of "stuff" in the pixels is conserved; it is just distributed over bigger pixels. (To answer Mat's question -- REBIN does *not* do linear interpolation when minifying, it does averaging.)

Right. I agree that the behavior is intuitive. It was just the
suggestion that it involves interpolation that I couldn't make sense of.

> This gives me a chance to plug one of my favorite programs -- FREBIN
> http://idlastro.gsfc.nasa.gov/ftp/pro/image/frebin.pro
> This program also conserves the "stuff" in the array, but allows new bin sizes that are not necessarily an integral fraction of the old. --Wayne

That's useful. Thanks!