
Subject: Re: sorting of a multi-dimensional array in only one "direction"/dimension
Posted by [Timm Weitkamp](#) on Thu, 02 Dec 2004 15:46:41 GMT

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

Today at 15:19 +0100, Benjamin Luethi wrote:

> I have (let's say) 5 images in an array `imgs=uintarr(1024,1024,5)`.
> I'd like to average pixel values over the 5 images, but only use the
> central three values of each position while discarding the minimum and
> maximum value.
>
> So far, I'm looping through all image positions (1024x1024!) [...]

There may be better ways, but this here does the job without a loop
(assuming you have IDL 5.6 or higher):

```
tot = TOTAL(imgs, 3)
min = MIN(imgs, DIMENSION=3)
max = MAX(imgs, DIMENSION=3)
avg = (tot - min - max) / 3.0
```

Cheers,
Timm

--
Timm Weitkamp <<http://people.web.psi.ch/weitkamp>>

Subject: Re: sorting of a multi-dimensional array in only one "direction"/dimension
Posted by [JD Smith](#) on Thu, 02 Dec 2004 16:10:14 GMT

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

On Thu, 02 Dec 2004 15:19:54 +0100, Benjamin Luethi wrote:

> Hi,
>
> I'm looking for an efficient way of sorting a multi-dimensional array in
> only one "direction"/dimension?
>
> I have (let's say) 5 images in an array `imgs=uintarr(1024,1024,5)`.
> I'd like to average pixel values over the 5 images, but only use the
> central three values of each position while discarding the minimum and
> maximum value.
>
> So far, I'm looping through all image positions (1024x1024!) and sorting
> the five elements in each position. (something like
> `imgs[x,y,*]=sort(imgs[x,y,*])`)
>

> Quote from JD Smith: "A typical rule of thumb: if you're looping over each
> data element individually, there's (probably) a faster way to do it."
> So, in this case that would be...? (something with histograms maybe?)
>
> thanks for your thoughts on this,

In your particular case, this is easy enough, since you're just
excluding one minimum and one maximum value:

```
m=max(imgs,DIMENSION=3,max_list,MIN=m,SUBSCRIPT_MIN=min_list )  
imgs[max_list]=0.0 & imgs[min_list]=0.0  
return,total(imgs,3)/3
```

You might also need to test for/exclude NaNs, but the basic idea is to
set the min and max to zero before totaling along the third dimension.
If you need to exclude NaNs, it's just:

```
m=max(imgs,DIMENSION=3,max_list,MIN=m,SUBSCRIPT_MIN=min_list ,/NAN)  
imgs[max_list]=0.0 & imgs[min_list]=0.0  
return,total(imgs,3,/NAN)/((total(finite(imgs),3)-2)>1)
```

What if you wanted to exclude more than one value? These types of
operations are actually not as easy as they should be in IDL. There
is a wonderful function inside of IRAF called IMCOMBINE (and you won't
find me singing IRAF's praises much), that is a very flexible combiner
of image stacks. Things IMCOMBINE can do:

- Combine with median or mean, or weighted mean.
- Use a reject mask to reject certain pixels
- MINMAX (nlow, nhigh) pixel rejection (e.g., this case is nlow=1,
nhigh=1)
- SIGCLIP reject via a sigma clipping algorithm, with adjustable low
side and high side sigma thresholds.
- AVSIGCLIP which takes advantage of the Poisson noise properties of
photosensors to compute sigmas directly from the median/mean.
- PCLIP or percentile clipping based on the histogram distribution at
each pixel.
- Many other thresholding rejections.

You can read about IMCOMBINE here:

<http://iraf.noao.edu/scripts/irafhelp?imcombine>

I have mentioned to RSI that they would do well to implement something
in IDL with similar functionality. Actually, standard array
inflation/comparison and WHERE can get you much of the way there. A
cheap addition which would go a bit further towards a full IMCOMBINE
would be to allow MIN and MAX to select the top N_max and/or bottom

N_min values along arbitrary dimensions. If you would find these types of image operations useful, I urge you to contact RSI and let them know that an IDL IMCOMBINE, or at the least a more flexible MIN/MAX, would interest you.

JD

Subject: Re: sorting of a multi-dimensional array in only one "direction"/dimension
Posted by [David Fanning](#) on Thu, 02 Dec 2004 16:50:27 GMT

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

Timm Weitkamp writes:

```
> There may be better ways, but this here does the job without a loop
> (assuming you have IDL 5.6 or higher):
>
> tot = TOTAL(imgs, 3)
> min = MIN(imgs, DIMENSION=3)
> max = MAX(imgs, DIMENSION=3)
> avg = (tot - min - max) / 3.0
```

This reminds me...

I'm making notes for a 3rd Edition of my book.
(Don't hold your breath, the outline is grandiose,
so who knows if it will really happen. IDL has gotten
so BIG!) One of my proposed chapters is "The IDL Way:
Writing Code to Confuse Your Boss and Amaze Your Friends".
This is the kind of material I'd like to have in there,
not so much to be instructive, but to give people an
idea of how these kinds of problems are solved.

I've saved these kinds of nuggets over the years,
but I can use more. Does anyone have a good story
to tell of how code that took days to run and pages
to write ran in seconds with five lines of IDL code?
(After consulting the IDL newsgroup, of course.)

I'd offer a prize, but I'm not drinking coffee or
beer anymore after returning from Germany. I am
completely disillusioned with what passes for both
in America. :-(

Cheers,

David

P.S. Any you can throw in bread, too. :-)

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

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