Subject: Fast local contrast calculations?
Posted by kagoldberg on Wed, 23 Oct 2013 18:28:34 GMT
View Forum Message <> Reply to Message

I'm looking for a quick way to calculate local contrast across a (2048,2048) image.
The end result could be a 64x64 element array, for example, and that would be fine.

Speed-wise, calculating local averages with rebin() are lightning fast. Even median(image,N) is pretty fast. But is there a similar high-speed way to get minimum and maximum values within each 'tile' without having to write a loop? It's as though I need a rebin_min() and rebin_max() where the min and max values are preserved within the sampled output array.

I'd love to hear any suggestions. Thanks, Ken

Subject: Re: Fast local contrast calculations?
Posted by Moritz Fischer on Thu, 24 Oct 2013 05:33:49 GMT
View Forum Message <> Reply to Message

Hi Ken,
my suggestion is to 'align' the tile elements with reform:

```
IDL> t = randomn(1, 6,6)
IDL> print, t
IDL> print, max(max( reform( t, 2, 6/2, 2, 6/2 ), D=1),D=2)
```

You could even remove the second max, by using transpose and another reform, but I think transpose will cost you...

Let me know what you end up with!

mo

Am 23.10.2013 20:28, schrieb kagoldberg@lbl.gov:
> I'm looking for a quick way to calculate local contrast across a
> (2048,2048) image. The end result could be a 64x64 element array, for
> example, and that would be fine.
>
> Speed-wise, calculating local averages with rebin() are lightning
> fast. Even median(image,N) is pretty fast. But is there a similar
> high-speed way to get minimum and maximum values within each 'tile'
> without having to write a loop? It's as though I need a rebin_min()
> and rebin_max() where the min and max values are preserved within the
> sampled output array.
>
> I'd love to hear any suggestions. Thanks, Ken
>

Subject: Re: Fast local contrast calculations?
Posted by kagoldberg on Fri, 25 Oct 2013 03:35:02 GMT
View Forum Message <> Reply to Message

Such genius, I don't even know where to begin, except: Thanks!