Subject: Re: "Max()" filter?
Posted by Dick Jackson on Thu, 07 Mar 2002 08:45:15 GMT
View Forum Message <> Reply to Message

"Dick Jackson" <dick@d-jackson.com> wrote in message
news:Neth8.6463$Qq1.33877@shaw-ty2...

> I'd like a filter that uses the Max() function, so that this array:
>
> 1 4 2 4 3
> 3 5 2 1 4
> 5 7 4 3 1
> 3 5 1 1 2
> 3 1 2 3 1
>
> filtered with width 3 gives:
>
> 5 5 5 4 4
> 7 7 7 4 4
> 7 7 7 4 4
> 7 7 7 4 3
> 5 5 5 3 3
>
> (I think! :-)

Sorry for answering my own posting, but something at this late hour reminded me that this sounds like a Dilate kind of thing. (maybe my pupils are dilating! :-) Sure enough, it's just the Grayscale Dilate operation. Have to be careful around the edges, padding with a ring of zeros, removing it afterward.

If you're curious, it's all described in online help for Dilate, but here's how it works (sorry for the terse code):

```
IDL> z0=byte(10*randomu(seed,5,5))
IDL> print,z0
   0   3   6   7   3
   4   1   0   4   5
   3   3   4   5   0
   6   3   7   5   2
   5   1   8   0   9
IDL> z00=bytarr(7,7) ; make array padded with ring of zeros
IDL> z00[1,1]=z0
IDL> ; now dilate with 3x3 kernel and strip off ring
IDL> print,(dilate(z00,Replicate(1B,3,3),1,1,/gray))[1:5,1:5]
   4   6   7   7   7
   4   6   7   7   7
```

```
  6  7  7  7  5
  6  8  8  9  9
  6  8  8  9  9
```

I imagine that using Erode will do the converse, what I'd call "Min()"
filtering.

Thanks for reading!

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: "Max()" filter?
Posted by the_cacc on Thu, 07 Mar 2002 11:23:24 GMT

View Forum Message <> Reply to Message

You could code this up quicker than it took to write the message...

Subject: Re: "Max()" filter?
Posted by Dick Jackson on Fri, 08 Mar 2002 17:19:36 GMT

View Forum Message <> Reply to Message

"trouble" <the_cacc@hotmail.com> wrote in message
news:5f9f0a23.0203070323.4c32551b@posting.google.com...

> You could code this up quicker than it took to write the message...

trouble,

Certainly a brute-force method with loops that go over every array element and do a Max operator over the appropriate nxn (or less) kernel is easy, but I think that it is somewhat inefficient for the 1024x1024 datasets I have in mind where the kernel might be around 30x30. In trying it, it runs in about 14 seconds.

Perhaps I should have added the word "efficient" somewhere, where I would like performance on the same order as the Smooth function, which takes 0.151 seconds. I believe Smooth saves huge amounts of time by overlapping calculations. I don't see how an efficient local max filter like this would be trivial to write in IDL, that's why I was asking.

The builtin grayscale dilate I mentioned before works conveniently, but I discover now that it's not really that fast, over 10 seconds in my little test.

However, I'd really like to have it work (efficiently) with floating-point values. Trying that now, I get results in over 18 seconds. Not bad, but not great, so if you have something up your sleeve, please post away.

Regards,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392