Subject: Re: Median filter the hard way
Posted by JD Smith on Tue, 21 Oct 2003 23:29:54 GMT
View Forum Message <> Reply to Message

On Tue, 21 Oct 2003 15:52:28 -0700, Dick Jackson wrote:

> "JD Smith" <> wrote in message
> news:pan.2003.10.17.22.22.14.73337.24537@as.arizona.edu...
>> On Fri, 17 Oct 2003 14:34:30 -0700, Dick Jackson wrote:
>>
>>> Here's an array, a:
>>>
>>> IDL> a=Float(Byte(RandomU(seed,7,7)*10)) IDL>
> a[2:4,2:4]=!values.F_nan
>>
>> [...]
>>
>>> The Convol function can be used to count up neighborhoods. If you
> need
>>> better counting around the edge, you could pad the array before
> calling
>>> Convol.
>>>
>>> IDL> print,Convol(Finite(a),Replicate(1B,3,3))
>>
>> Looks good, Dick. CONVOL's a bit heavy-handed for just counting: I'd
> use
>> smooth instead:
>>
>> IDL> print,smooth(finite(a)*9,3,/EDGE_TRUNCATE)
>>
>> [...]
>>
>> Notice it treats edges better (as far as this problem is concerned),
>
> Granted, if edge pixels are not going to be dropped anyway for having
> too few 'good' neighbors...
>
>> and should definitely be faster.
>
> A good guess, but in this case, I guess Convol can keep everything as
> Byte type and it is indeed faster:

This is the key (byte type). Change it to smooth(finite(a)*9b... and you should see similar performance. Obviously, in this case, we're limited by something other than the details of the addition.

JD