
Subject: Faster image median filtering

Posted by [Deckard++](#); on Fri, 24 Jun 2011 13:48:01 GMT

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

Hi,

I am confronted with a performance issue related to median filtering of images using the median() function of IDL. I have rather extended sets (several thousands) of 1500x1500 images, and I need to median filter each of them. So far I have been using this to filter them

```
img_filtered = median(img, boxsize)
```

The problem is that it take ~5 seconds per image for boxsize=15, which is ok for a few images but really time consuming when considering thousands of images. I have a few questions:

- 1- is there a way to speed-up median filtering that I am not aware of?
- 2- do you know if IDL implementation of 2D median() is the most efficient?
- 3- if not, are you aware of a DLM that would implement a faster median filtering of images?

Thank a lot in advance.

Best regards,

-- Arthur;

Subject: Re: Faster image median filtering

Posted by [wlandsman](#) on Fri, 24 Jun 2011 22:33:43 GMT

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

I think suspect that the IDL implementation of MEDIAN is reasonably optimal (e.g. using a rolling window). In the ancient days of IDL, MEDIAN() only worked on byte data, and I used to use a CALL_EXTERNAL to a C program for float data. But when support for floating data in MEDIAN was added, it was definitely faster than my CALL_EXTERNAL setup.

Having said that you might look at scaling your data to byte. The MEDIAN() is the midpoint of ordered data, and the order should be maintained after byte scaling (except that resolution is lost, so some nonequal values become equal). And MEDIAN is *much* faster on byte data.

--Wayne

Subject: Re: Faster image median filtering

Posted by [Juggernaut](#) on Tue, 28 Jun 2011 17:33:25 GMT

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

On Jun 24, 9:48 am, "Deckard++;" <arthur.vi...@gmail.com> wrote:

> Hi,
>
> I am confronted with a performance issue related to median filtering
> of images using the median() function of IDL. I have rather extended
> sets (several thousands) of 1500x1500 images, and I need to median
> filter each of them. So far I have been using this to filter them
>
> img_filtered = median(img, boxsize)
>
> The problem is that it take ~5 seconds per image for boxsize=15, which
> is ok for a few images but really time consuming when considering
> thousands of images. I have a few questions:
>
> 1- is there a way to speed-up median filtering that I am not aware of?
> 2- do you know if IDL implementation of 2D median() is the most
> efficient?
> 3- if not, are you aware of a DLM that would implement a faster median
> filtering of images?
>
> Thank a lot in advance.
>
> Best regards,
>
> -- Arthur;

If you have multiple cores on your computer try out the `idl_idlbridge` object in IDL and just run multiple instances of your median processing at once.

Subject: Re: Faster image median filtering

Posted by [Deckard++;](#) on Sun, 03 Jul 2011 13:53:55 GMT

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

On Jun 28, 7:33 pm, Bennett <juggernaut...@gmail.com> wrote:

> On Jun 24, 9:48 am, "Deckard++;" <arthur.vi...@gmail.com> wrote:
>
>
>
>
>
>> Hi,
>
>> I am confronted with a performance issue related to median filtering

>> of images using the median() function of IDL. I have rather extended
>> sets (several thousands) of 1500x1500 images, and I need to median
>> filter each of them. So far I have been using this to filter them
>
>> img_filtered = median(img, boxsize)
>
>> The problem is that it take ~5 seconds per image for boxsize=15, which
>> is ok for a few images but really time consuming when considering
>> thousands of images. I have a few questions:
>
>> 1- is there a way to speed-up median filtering that I am not aware of?
>> 2- do you know if IDL implementation of 2D median() is the most
>> efficient?
>> 3- if not, are you aware of a DLM that would implement a faster median
>> filtering of images?
>
>> Thank a lot in advance.
>
>> Best regards,
>
>> -- Arthur;
>
> If you have multiple cores on your computer try out the idl_idlbridge
> object in IDL and just run multiple instances of your median
> processing at once.

This looks really interesting! I had never heard of idl_idlbridge before. I will look into it because I am running my code on a 16-core computer. Actually I am quite surprised that the MEDIAN function is not optimised for multi-core.

Thanks,

-- Arthur;

Subject: Re: Faster image median filtering
Posted by [Deckard++](#); on Sun, 03 Jul 2011 13:54:55 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jun 25, 12:33 am, wlandsman <wlands...@gmail.com> wrote:

> I think suspect that the IDL implementation of MEDIAN is reasonably optimal (e.g. using a rolling window). In the ancient days of IDL, MEDIAN() only worked on byte data, and I used to use a CALL_EXTERNAL to a C program for float data. But when support for floating data in MEDIAN was added, it was definitely faster than my CALL_EXTERNAL setup.
>
> Having said that you might look at scaling your data to byte. The MEDIAN() is the midpoint of ordered data, and the order should be maintained after byte scaling (except that resolution is

lost, so some nonequal values become equal). And MEDIAN is *much* faster on byte data.

>

> --Wayne

Hi Wayne,

scaling my data to byte is not possible I think, but thanks for the tip.

-- Arthur;
