

Re: Rookie having problems with some filter code. Any help?

Re: Rookie having problems with some filter code. Any help?

Source: <http://sci.tech-archive.net/Archive/sci.image.processing/2005-12/msg00068.html>

- *From:* "GrahamH" <pknvdw@xxxxxxxxxxxxxx>
 - *Date:* Wed, 14 Dec 2005 09:46:25 GMT
-

"Don Bruder" <dakidd@xxxxxxxxxx> wrote in message
[news:439f9bd5\\$0\\$95973\\$742ec2ed@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx](mailto:news:439f9bd5$0$95973$742ec2ed@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)
> In article <dno3mv\$bae\$2@xxxxxxxxxxxxxxxxxx>,
> davem@xxxxxxxxxx (Dave Martindale) wrote:
>
>> Don Bruder <dakidd@xxxxxxxxxx> writes:
>>
>> >However, you bring up a valid point – one which I've been struggling
>> >with – How to cope with (A) color values that go negative while being
>> >"cooked", and (B) values that overflow the available 8 bits while being
>> >cooked. For situation B, a simple "value % 256" leaps to mind as the
>> >proper treatment. But coping with values that land in "negative-ville"
>> >as a result of being cooked, whether they are or are not possible to fit
>> >into 8 bits, is something that has me completely stumped.
>>
>> If you're just going to write out the image again, negative values
>> should be clamped to zero, and values greater than 255 should be clamped
>> to 255 – not wrapped around to zero.
>
> Hmm... So simple clamping is, despite the counterintuitiveness
> involved, the way to go? Without chasing every possibility, it seems
> likely to me that would end up with a whole slew of pixels whose
> color-component values are pegged at either 0 or 255, with comparatively
> little "middle ground".
>

No, this is exactly your problem. The negative coefficients in your kernel can result in negative output pixel values but these are represented as ~255. Rather than a small negative value, implying darker than black, you get a very bright value.

You apply clipping just after normalization, i.e. to the result for the whole kernel. It is important to hold your intermediate results sumr etc in signed int, as you do. Clip the int value to a valid BYTE (unsigned char) range then build your result color.

The kernel you have here is an edge detection filter. Areas of uniform color will become black and strong edges will be bright. This doesn't seem to

Re: Rookie having problems with some filter code. Any help?

Re: Rookie having problems with some filter code. Any help?

match your output image so perhaps you are adding the filter result to the input pixel value. This gives a result that is the original image with the edges enhanced and this is perhaps your intention. In this case you must add the signed int filter result to the source pixel and then apply clipping.

When calculating with BYTE values make sure intermediate results are cast to int.

You could use an averaging kernel – all coefficients = 1 as a test. This cannot produce negative values and should give you a blurred version of your input.

Graham

• *Follow-Ups:*

- ◆ [Re: Rookie having problems with some filter code. Any help?](#)
◇ From: Juho Pitkänen

• *References:*

- ◆ [Rookie having problems with some filter code. Any help?](#)
◇ From: Don Bruder
- ◆ [Re: Rookie having problems with some filter code. Any help?](#)
◇ From: jg . campbell . ng
- ◆ [Re: Rookie having problems with some filter code. Any help?](#)
◇ From: Don Bruder
- ◆ [Re: Rookie having problems with some filter code. Any help?](#)
◇ From: Dave Martindale
- ◆ [Re: Rookie having problems with some filter code. Any help?](#)
◇ From: Don Bruder

- Prev by Date: [MATLAB Video Acquisition](#)
- Next by Date: [PCI / Image Selection](#)
- Previous by thread: [Re: Rookie having problems with some filter code. Any help?](#)
- Next by thread: [Re: Rookie having problems with some filter code. Any help?](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)