Re: interpolation for a color image?

Source: http://sci.tech−archive.net/Archive/sci.image.processing/2008−05/msg00129.html

- *From*: aruzinsky <aruzinsky@xxxxxxxxxxxxxxxxxxxx>
- *Date*: Sun, 25 May 2008 15:48:20 −0700 (PDT)

Correction:

AFter giving it more thought, I expect it is faster but not much.

The problem is with the result. Consider an edge with red hue on one side and cyan hue on the other, both of equal Y. Your method will produce a result that is the same as box interpolation of RGB.

When people look at linear enlargements, they associate the blur with that of an unfocused eye lens because that is their life experience with blur. A lens blurs different colors independently and that is why linear enlargement should also do it independently. In the case of a red−cyan edge, the eye expects the blurred edge to be grayish. Of course, for nonlinear enlargement that preserves sharp edges, this does not apply.

aruzinsky wrote:

> On May 25, 10:16ýam, Harris <xgeorg...@xxxxxxxxxxxxx> wrote:
>
>> AE lover <aelove...@xxxxxxxxx> wrote in
>> news:2a03e221−19de−4e85−81d1−0b0e3ee4d7e3
>> @k37g2000hsf.googlegroups.com:
>>
>>> Hi all,
>>>
>>> I am considering the case of bilinear interpolation for a color
>>> image
>>> (say RGB image), to apply a bilinear interpolation, will we
>>> apply the
>>> formular of bilinear interpolation, which we use for a gray
>>> image, for
>>> each channel R, G, and B, separately?
>>> If so, why don't we take into account the interaction between
>>> three

channels?

Thanks

Since resizing is a spatial transformation, it should be invariant to intensity levels (mono or RGB).
Hence, it should not really matter if you do 3x (RGB) or 1x resizing (mono).
On the other hand,
interpolation errors on reparate RGB channels may produce a new (combined) pixel value slightly
different of the original one, leading to a color−distorted image (probably in the form of small color shifts
at pixel−level scales).

My advice would be: (1) store the RGB ratios for each pixel, i.e., keep the original image, (2) convert
the image into intensity matrix using a standard (linear) formula, (3) resize the intensity matrix, (4)
convert the intensity matrix back to RGB using the values/ratios for each pixel form the original image.
The single−channel approach cuts down your processing time almost to a factor of one−third (plus
RGB conversions). Of course, you also need a mapping from the original to the new pixels (1 pixel −>
N pixels), so it is much simpler if you use integer multipliers in resizing (e.g. x2, x3, etc).

−−
Harris

Let me paraphrase that:

1. Y = a*R + b*G + c*B + d, (d is small positive number to prevent division by zero)

2. ratioR = R/Y, ratioG = G/Y, ratioB = B/Y

3. Interpolate: Y' = bilinear(Y), ratioR' = box(ratioR), ratioG' = box(ratioG), ratioB' = box(ratioB)

4. R' = Y'*ratioR', G' = Y'*ratioG', B' = Y'*ratioB'

Bilinear interpolation on RGB is faster than these conversions.

I will digress here and say it that it is bad practice to put one's thoughts ahead of empiricism. This is an ethical issue because it is a major cause of chaos in society.

I know from experience that you can do fancy interpolation on the Y channel of YCbCr space and sloppy interpolation on th Cb and Cr channels, but "fancy" here is nonlinear and relatively slow.

.