

Re: interpolation for a color image?

Source: <http://sci.tech--archive.net/Archive/sci.image.processing/2008-05/msg00130.html>

- *From:* Harris <xgeorgiou@xxxxxxxxxxxxxxxx>
 - *Date:* Mon, 26 May 2008 10:31:15 +0000 (UTC)
-

aruzinsky <aruzinsky@xxxxxxxxxxxxxxxxxxxxxxxx> wrote in
news:0bf42d72-1f92-44df-aba2-687532433ede@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:

Let me paraphrase that:

1. $Y = a \cdot R + b \cdot G + c \cdot B + d$, (d is small positive number to prevent division by zero)
2. $\text{ratioR} = R/Y$, $\text{ratioG} = G/Y$, $\text{ratioB} = B/Y$
3. Interpolate: $Y' = \text{bilinear}(Y)$, $\text{ratioR}' = \text{box}(\text{ratioR})$, $\text{ratioG}' = \text{box}(\text{ratioG})$, $\text{ratioB}' = \text{box}(\text{ratioB})$
4. $R' = Y' \cdot \text{ratioR}'$, $G' = Y' \cdot \text{ratioG}'$, $B' = Y' \cdot \text{ratioB}'$

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 the Cb and Cr channels, but "fancy" here is nonlinear and relatively slow.

I really don't understand what your opinions (or mine) on social chaos has to do with image processing. If you mean to say that I said a stupid thing just to look smart, then just say it.

My answer is this: bilinear is not a good interpolation method but it is fast. Since it works in 2D, it requires 12 multiplications and 4 divisions per (new) pixel estimate. If you have any doubt, see: http://en.wikipedia.org/wiki/Bilinear_interpolation. Your "paraphrase" of what I initially proposed is too slow, because you don't really need all these conversions, only an intensity estimate (RGB -> mono) and the original image:

(1). $I(x,y) = a \cdot R(x,y) + b \cdot G(x,y) + c \cdot B(x,y)$

Re: interpolation for a color image?

- (2). apply bilinear interpolation on $I(x,y)$
- (3). convert back to RGB

Step 2 is the same as above, e.g. 12 multiplications and 4 divisions for bilinear per (new) pixel value. Step 3 can be implemented using the original channel value and the value range: if $I'(x,y)$ is the interpolated value, then new Red is: $R'(x,y) = I'(x,y)/I(x,y) * R(x,y)$.

As you can see, your proposal with full RGB interpolation requires $3 \times (12.mul + 4.div) = 36.mul + 12.div$ per pixel, probably with floating-point arithmetic due to the use of ratios. My proposal requires $1 \times (3.mul + bilinear + 1.mul + 1.div) = 16.mul + 5.div$ per pixel, which can be implemented easily with integer arithmetic in steps 1 and 3 (no explicit ratios used).

Please do the math before you criticise someone else's motives and knowledge here.

—

Harris

.