Re: Positive image interpolation

Source: http://sci.tech−archive.net/Archive/sci.image.processing/2006−10/msg00143.html

- *From*: davem@xxxxxxxxx (Dave Martindale)
- *Date*: Fri, 20 Oct 2006 05:25:50 +0000 (UTC)

dudesinmexico@xxxxxxxxx writes:

> if I understand your point correctly, you are saying that if I low−pass filter a discrete sequence of non−negative samples so that there are no frequency components above Nyquist, sinc interpolation will never yield negative values.

I was actually talking about the continuous case: the source scene has a continuous energy distribution, which is transferred to the sensor of whatever camera you are using with some loss (described by the MTF of the lens). The lens plus the anti−aliasing filter are supposed to reduce the amplitude of all frequency components above Fs/2 to zero to avoid aliasing. I argued that if that happened, then reconstructing the image with an interpolating filter would not produce overshoots.

This also applies when you've got a discrete sequence of samples that you're trying to downsample, if you get the details right.

But on second thought, I don't think that's true. See below.

> I thought about this some time ago, however I could not find an answer to this question:
> given N random samples {y_i} such that 0 <= y_i <= 1, i=0,...,N−1, and Ts (sampling interval) = 1, how to design such a filter?
> There seems to be a relationship between the fact that the values of the samples are bounded and the bandwidth of the signal.

If the signal is lowpass filtered to Fs/2 and limited to a particular amplitude range, then you know the maximum rate of change between two adjacent samples. So adjacent pixels can't be arbitrarily different.

However, suppose the original signal is a square wave − alternating white and black. Fourier analysis tells us that a square wave is composed of an infinite number of odd harmonics (frequencies F, 3F, 5F, ...) with amplitudes of (4/pi, 4/(3*pi), 4/(5*pi), ...). If you were to low−pass filter it with a filter that leaves only the fundamental, it would be a sine wave with amplitude 4/pi, almost 1.3 times that of the square wave. Thus, the sine wave overshoots the square wave's min and max intensity. If you keep fundamental and 3rd harmonic, the peak amplitude is about 1.2, and that doesn't reduce much further as you add back in the 5th, 7th, etc. harmonics. The integral of the overshoot gets progressively smaller, but the amplitude hardly changes. It's only when you include infinitely many harmonics that the overshoot disappears (by having its integral vanish).

So, if the original signal is a square wave that alternates between dark grey and light grey, and you bandlimit it, it's *correct* to end up with some overshoot beyond the original brightness range. But if the dark portions were already nearly true black (almost no light reflected), the overshoot would be blacker than black − how is this possible? I don't know.

I suspect that some of the answer lies in the fact that the math of Fourier transforms is based on real and complex numbers, while the "signal" we work with in image processing is all−positive. That in turn is because light sensors only measure power, not phase. In something like radar imaging (for example) the receiver can determine amplitude *and phase*. There, a negative value is perfectly meaningful − it means the received signal is out of phase with some reference signal.

To avoid overshoot of a square wave when low−pass filtering it, you need a filter with a more gradual rolloff, not a filter with "brick wall" response at the cutoff frequency. This often happens in practice. Oscilloscope probes have their frequency response adjusted until a square wave passed through them *looks* as square as possible with minimal overshoot. And real cameras and scanners often have a similar response.

So, with real image capture devices, an abrupt edge in the original scene turns into a transition that takes 2 or 3 pixels to complete in the digital image. Applying a conventional interpolation kernel to this generally doesn't produce overshoot. But, contrary to what I said before, that isn't guaranteed, because sometimes overshoot is the "right answer" to what happens to a low−pass−filtered square wave.

> Intuitively, I would say that if sinc interpolation on the {y_i} returns negative values it is because after sampling some frequency components above Nyquist of the original continuous−time signal have been lost. By enforcing a positive interpolating function (note that I'm not saying a positive interpolation kernel), these higher frequency components (or at least the components just below Nyquist) should be represented more accurately by the interpolating function.

I don't understand how "interpolating function" differs from "interpolation kernel", other than that one is continuous and the other discrete. Can you explain what you mean?

Dave
.