Pepper's Ghost Compensation

Gonzalo Ávila Alterach (http://gzalo.com)

January 25, 2015

One common drawback of using the Pepper's Ghost effect is that it requires very thin glass or other suitable material, because otherwise a double image is seen, product of the reflection on both surfaces of the used sheet. Hence more expensive and hard to find materials are needed, for instance dark acrylic or glasses with anti-reflection coating.

This project attempts to compensate that effect, by modifying the image or video to be used in the effect, using OpenGL for accelerated filtering.

1 Modeling the glass

In a 1D problem, the glass reflection may be modeled as a simple echo: the output consists of the input overlapped with the input shifted in space a certain amount τ (depending on the glass thickness). The second reflection, coming from the back surface of the glass is often attenuated by a certain value α.

Assuming that the glass works equally amongst its axis, and that the addition is linear, it can be thought as an LTI system with the following impulsive response:

$$h(x) = \delta(x) + \alpha\delta(x - \tau)$$

In reality, it can be seen that the addition isn't exactly linear (both due to the nonlinearities of the glass and the computer screen) and that the parameters depend mainly on the viewing angle and the position and angle of the glass.

2 Compensation

By applying the Fourier transform, it can be seen that the frequency response of the 1D LTI system is the following:

$$H(j\omega) = 1 + \alpha e^{-\tau j\omega}$$

Since a filter that inverts the glass effect is wanted, the frequency response must be the reciprocal of those of the glass:

$$H_{filter}(j\omega) = \frac{1}{1 + \alpha e^{-\tau j\omega}}$$

By applying the Fourier antitransform, we get that the response of the filter is:

$$h_{filter}(x) = \sum_{k=0}^{\infty}(-\alpha)^k\delta(x - k\tau) = \delta(x) - \alpha\delta(x - \tau) + \alpha^2\delta(x - 2\tau) - \cdots$$

It can be seen that each term acts by erasing the reflection of the previous one, so the final effect would be similar to using a glass with infinite thickness.

3 Implementation

To implement the convolution in real-time, an OpenGL shader was used. The impulsive response was cutoff after the 5th term, since there isn't a big difference after it. Since the system isn't exactly linear, fine tuning can be used

for each of the parameters. Also, to do gamma correction and work in a linear space inside the shader, each one of the texture lookups get's powered to a factor γ.

The rendering utilizes this equation:

$$OutColor = (image(x+\tau)^\gamma + \alpha_1 image(x+2\tau)^\gamma + \alpha_2 image(x+3\tau)^\gamma + \alpha_3 image(x+3\tau)^\gamma + \alpha_4 image(x+4\tau)^\gamma)^{1/\gamma}$$

In order to use the filter correctly, the parameters need to be adjusted in order to minimize the unwanted ghost images. An automated method with a feedback loop using a camera might be possible. To fine-tune the τ parameter, which changes amongst the main axis of the glass, it's possible to calibrate it on the top and on the bottom, and then use a linear fit to apply the effect correctly in the whole surface of the glass. Each parameter can tuned on each position by using a texture containing the value for each pixel.