Temporally-informed image denoise: supplementary material

Luke Goddard
luke@lukegoddard.info

FORWARD

These formulas are the supplementary equations for the *temporally-informed image denoise* which was presented as part of the Siggraph 2014 talk *Silencing the noise on Elysium*. For an overview of the algorithm and it's application, please read the short paper which can be found at *www.lukegoddard.info/siggraph2014/silencingTheNoise.pdf*. A C++ implementation of the algorithm can be found at *https://github.com/goddardl/temporalDenoise.git* along with test materials.

The algorithm

The algorithm is based on the sum of weighted differences between the target pixel's mean value and the samples in it's neighbourhood. The weighting of these offsets is seen here as the $Z(i)$ function. It is split into two main components: A smart blur $\frac{B(i,j)}{\sigma^2}$ that filters similar pixels together based on their similarity and an attenuation weight that selectively applies it in areas of noise. The weight is then normalized into the range of 0 to 1 using an inverse exponential:

$$F[v](i) = \mu_i + \sum_{j=1}^{N_i} \frac{1}{Z(i)}(v_i(j) - \mu_i)e^{-\frac{B(i,j)}{\sigma^2 G_i(j)((1.-\alpha)S_i(j)+\alpha)}}$$
$$Z(i) = \sum_{j=1}^{N_i} e^{-\frac{B(i,j)}{\sigma^2 G_i(j)((1.-\alpha)S_i(j)+\alpha)}} \tag{1}$$

Where N_i is the number of temporal samples in the neighbourhood of pixel i, $v_i(j)$ is the value of temporal sample j in the neighbourhood around pixel i, $G_i(j)$ is a Gaussian falloff based on the distance of sample j from pixel i, $S_i(j)$ is an attenuation function based on the similarity between pixel i and j, and $B(i,j)$ is a function that produces a blur weight. The control parameter α is a value in the range of 0-1 that allows the blur to be attenuated. A value of 0 means that it's application is entirely governed by the similarity function $S_i(j)$, and a value of 1 will apply it everywhere.

We use two different functions for producing the "smart blur". One produces a conservative blur:

$$B(i,j) = (\mu_j - \mu_i)^2 \tag{2}$$

And the other an aggressive blur:

$$B(i,j) = ((\mu_j - \mu_i)(C(j) - C(i)))^2 \tag{3}$$

Where $C(i)$ is the range of the temporal samples for pixel i.

The similarity function $S_i(j)$ returns a weight in the range of 0-1 that indicates the importance of the value of the sample $v_i(j)$ in the ith pixel's sample distribution. This is essentially a 1D convolution of one pixel's samples against another.

$$S_i(j) = f(v_i(j), \mu_i, \sigma_i(\beta+1))f(v_i(j), \mu_j, \sigma_j) \tag{4}$$

Where f is a standard distribution function that approximates the spread of the sample values at pixel i. In the ideal case, this function would accurately represent the distribution of the samples at pixel i, however, we have found that just using a standard distribution is good enough. The user control function β is used to scale the width of pixel i's distribution so that values outside of it's range can still contribute to it's filtered result if desired.

Acknowledgements

This work was developed at *Image Engine Vfx* during the production of the film *Elysium*. I would like to thank all of the bright sparks in the R&D department at Image Engine for their contributions to the project and Andrew Kaufman in particular for doing all of the leg work in it's submission to Siggraph '14.