A GENERALIZATION OF THE TEAGER ALGORITHM

Michael Moore and Sanjit Mitra

Dept. of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA 93106-9560

Reinhard Bernstein

Telecommunications Institute
University Erlangen-Nuremberg, Cauerstr. 7
D-91058 Erlangen, Germany

ABSTRACT

The 1-D Teager algorithm can be used to perform mean weighted highpass filtering with relatively few operations. We propose a generalization of the Teager algorithm. The modified algorithm allows us to adjust the dependence of the highpass output on the local mean. The derivation and interpretation of the modified algorithm is presented. Finally, the response of several implementations to a test input is presented.

1. INTRODUCTION

In its general form, a 1-D quadratic digital Volterra filter is given by the 2-D convolution of the 1-D sample products $x(n_1) \cdot x(n_2)$ with a 2-D kernel $h_2(n_1, n_2)$.

$$y(n) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h_2(k_1, k_2) \cdot \ x(n-k_1)x(n-k_2).$$

Teager's algorithm [1] is a 1-D Volterra filter defined by

$$y(n) = x^2(n) - x(n-1) \cdot x(n+1). \tag{1}$$

As discussed in [2], the output from Teager's algorithm is approximately equal to

$$y(n) \approx \mu[(x(n) - x(n-1)) + (x(n) - x(n+1))] \tag{2}$$

where $\mu = (x(n-1) + x(n) + x(n+1))/3$. Thus the output of a Teager filter is approximately equal to a highpass filter response weighted by the local mean.

In some applications, it is desireable to adjust the degree to which the highpass filter output depends on the average local intensity. This paper proposes a relatively simple generalization of the Teager filter to adjust the weighting of the local mean in the filter output. The next section introduces the generalized algorithm and derives the mean approximation. Section three discusses an amplitude mapping interpretation of the algorithm. The final section provides some results for various filter implementations.

2. GENERALIZED TEAGER ALGORITHM

The sensitivity of the 1-D Teager filter to the local mean can be adjusted by raising each term in eq. (1) to a fractional power. Specifically,

$$y(n) = [x(n)]^{\frac{2}{m}} - [x(n-1) \cdot x(n+1)]^{\frac{1}{m}}. \tag{3}$$

Like the original function, this equation can be approximated as a highpass function weighted by a local mean by applying the binomial theorem. Let μ be the average of the three input samples $x(n-1)$, $x(n)$, and $x(n+1)$ and δ_n equal $x(n) - \mu$. Then the first term on the right hand side of eq. (3) becomes

$$\begin{aligned} x^{\frac{2}{m}}(n) &= [\mu + \delta_n]^{\frac{2}{m}} \\ &= \mu^{\frac{2}{m}}[1 + \frac{1}{\mu}\delta_n]^{\frac{2}{m}} \\ &\approx \mu^{\frac{2}{m}}[1 + \frac{2}{m\mu}\delta_n], \end{aligned} \tag{4}$$

assuming that the variations around the mean are small compared to the mean value. The second term can be approximated as

$$\begin{aligned} &[x(n-1) \cdot x(n+1)]^{\frac{1}{m}} \\ &= [(\mu + \delta_{n-1}) \cdot (\mu + \delta_{n+1})]^{\frac{1}{m}} \\ &= [\mu^2 + \mu(\delta_{n-1} + \delta_{n+1}) + \delta_{n-1}\delta_{n+1}]^{\frac{1}{m}} \\ &= \mu^{\frac{2}{m}}[1 + \frac{1}{\mu}(\delta_{n-1} + \delta_{n+1}) + \frac{1}{\mu^2}\delta_{n-1}\delta_{n+1}]^{\frac{1}{m}} \\ &\approx \mu^{\frac{2}{m}}[1 + \frac{1}{m\mu}(\delta_{n-1} + \delta_{n+1})], \end{aligned} \tag{5}$$

ignoring second-order terms and once again assuming that variations are small relative to the mean. Subtracting eq. (5) from eq. (4) and replacing the δ terms by adding or subtracting μ as necessary, a final approximation can be found. Specifically,

$$y(n) \approx \frac{\mu^{\frac{2}{m}-1}}{m}[(x(n)-x(n-1))+(x(n)-x(n+1))], \tag{6}$$

where, as before, μ is the local mean. The weighting of the local mean can be adjusted by changing the parameter m.

For $m < 2$, the highpass output is multiplied by the local mean, raised to some power, and thus is weighted more heavily in high intensity regions. When $m > 2$, the highpass output is divided by the local mean, raised to some power, and thus is weighted more heavily in low intensity regions. For $m = 2$, the filter output does not depend on the local mean and approximates the output of a Laplacian filter.

The approximation is valid as long as two conditions are met. First, the input signal $x(n)$ must be strictly greater than zero. Second, the sum of the two surrounding values should be within the following range:

$$\frac{1}{2}x(n) < \frac{1}{2}(x(n-1) + x(n+1)) < \frac{3}{2}x(n) \qquad (7)$$

with the best results when $\frac{1}{2}(x(n-1)+x(n+1)) \approx x(n)$.

3. INPUT MAPPING INTERPRETATION

Note that eq. (6) can be rewritten as

$$y(n) = ([x(n)]^{\frac{1}{m}})^2 - [x(n-1)]^{\frac{1}{m}} \cdot [x(n+1)]^{\frac{1}{m}}. \qquad (8)$$

Therefore, the filter can be implemented by taking the mth root of each incoming sample and then filter using a conventional Teager filter to produce the output sample.

The premapped interpretation of the proposed algorithm is useful for explaining the effect of changing m on the output. Figure 1 is a plot of three mapping functions. The identity mapping corresponds to the Teager filter ($m = 1$). As m increases above one, the mapping function starts to bend out towards the upper left of the plot. The slope of the mapping function increases at low intensities and decreases at high intensities. Thus small changes at low intensities are mapped to larger changes before filtering. When $m = 2$, the mapping exactly compensates for the Teager filter. As m increases above two, low intensity changes start to dominate the mapped input signal.

Conversely, as m decreases below one, the mapping function starts to bend down towards the lower right of the plot. The slope of the mapping function increases at high intensities and decreases at low intensities. Thus small changes at high intensities are mapped to larger changes before filtering. Since the Teager algorithm already amplifies high average intensity differences more than lower intensity differences, the high intensity changes start to dominate the output very quickly as m decreases below one.

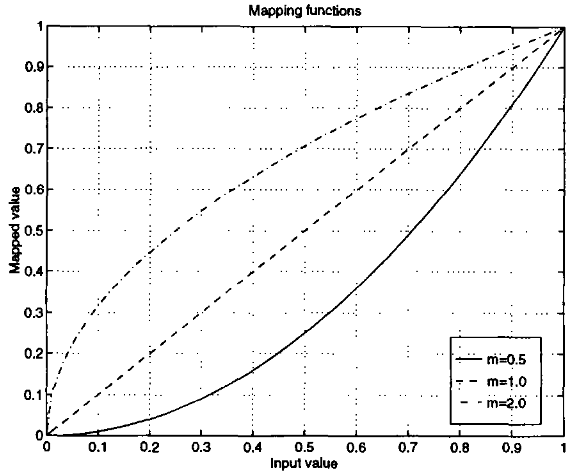

Figure 1: Mapping functions.

By using the premapped interpretation of the algorithm, arbitrary mapping functions can be created. The function can be designed to specifically enhance regions of certain average intensity by increasing the slope of the mapping function in those intensity bands. However, the use of an arbitrary mapping function instead of an mth root mapping function makes the determination of an approximation like eq. (6) difficult.

Figure 2 is a plot of a few arbitrary mapping functions. One function combines sections of two root plots to create a filter that enhances middle intensity regions more that either light or dark regions. The other two functions are linear approximations of the $m = 0.5$ and $m = 2$ root functions.

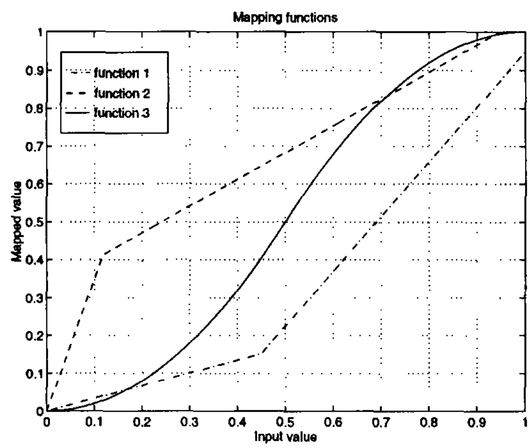

Figure 2: Mapping functions.

The premapping function can also be used to make other useful alterations in the input signal. For example, the function can map zero input samples to nonzero values. As can be seen from eqs. (3) and (7), zeroes violate the conditions under which the approximation is valid and result in large filter responses regardless of the value of the local mean. By removing zero values, this effect can be minimized.

As another example, the premapping function can be used to ensure that the algorithm output values are within a desired range. Let x_{max} be the maximum absolute value in the input range. The output of the Teager algorithm can vary within $\pm x_{max}{}^2$. The output of the generalized Teager algorithm can range over $\pm x_{max}{}^{\frac{2}{m}}$. However, by scaling the input a desired output range can be achieved. The scaling can be accomplished by the mapping function.

Premapping only depends on individual input values. Therefore, the input mapping can be accomplished simply by using a lookup table indexed by the possible input values. This implementation adds very little complexity to the normal Teager filter.

4. RESPONSE TO TEST INPUT

Figure 3 shows the output of the generalized Teager algorithm for three different values of m. The input signal consists of impulses of uniform height at three different base levels. The remaining plots compare the response of the algorithm to the input signal with $m = \frac{1}{2}$, $m = 2$, and $m = 3$.

Each output contains the response to both the impulses at each level and the steps between levels. Note that for $m = \frac{1}{2}$, the response to each impulse increases as the level increases. For $m = 2$, the impulse response does not depend on the base level. For $m = 3$, the impulse response decreases as the level increases.

Figure 4 shows the output of a normal Teager filter using premapped inputs. The same input signal as Figure 3 is used. The other plots correspond to the mapping functions shown in Figure 2.

Function 1 approximates the function x^2 using two line segments. Premapping the inputs using the square function is equivalent to filtering with the generalized Teager algorithm with $m = \frac{1}{2}$. Indeed, the response to the function 1 in Figure 4 approximates the response to $m = \frac{1}{2}$ in Figure 3.

Function 2 approximates the function $\sqrt{x}$ using two line segments. Premapping the inputs using the square root function is equivalent to filtering with the generalized Teager algorithm with $m = 2$. In Figure 4, the response to the inputs mapped by function 2 approximates the response to $m = 2$ in Figure 3, although there is a slight increase in response with intensity.

Function 3 combines scaled versions of the square function and a square of $x_{max} - x$. This mapping function has a steep slope for middle intensities and shallow slopes at high and low intensities. This mapping should result in greater response to changes at middle intensities. This response is seen for function 3 in Figure 4.

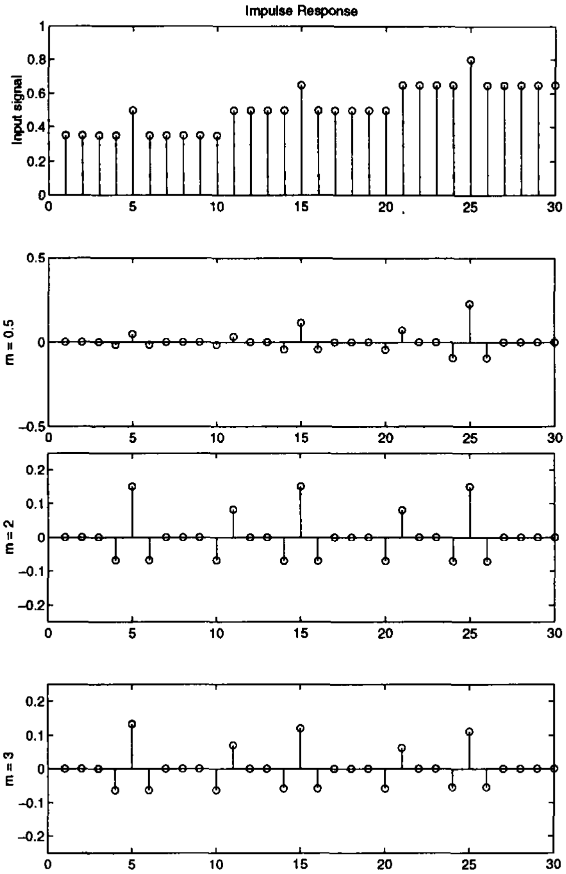

Figure 3: Generalized Teager filter test input response.

Note that the response for the highest intensity plateau using function 3 is larger than the response of the lowest intensity plateau. This response is a result of the fact that the mapping function was symmetric around an intensity of 0.5 but the basic Teager filter response was not. Therefore, although the high and low impulses were mapped to impulses of the same height, the Teager filter responded more strongly to the higher average intensity impulse. To equalize the high and low intensity responses, the premapping function would have to be flattened more at high intensities and steepened at low intensities.

5. FURTHER APPLICATIONS

The generalized Teager algorithm allows us to change the weighting of the local mean in the output by adjust-

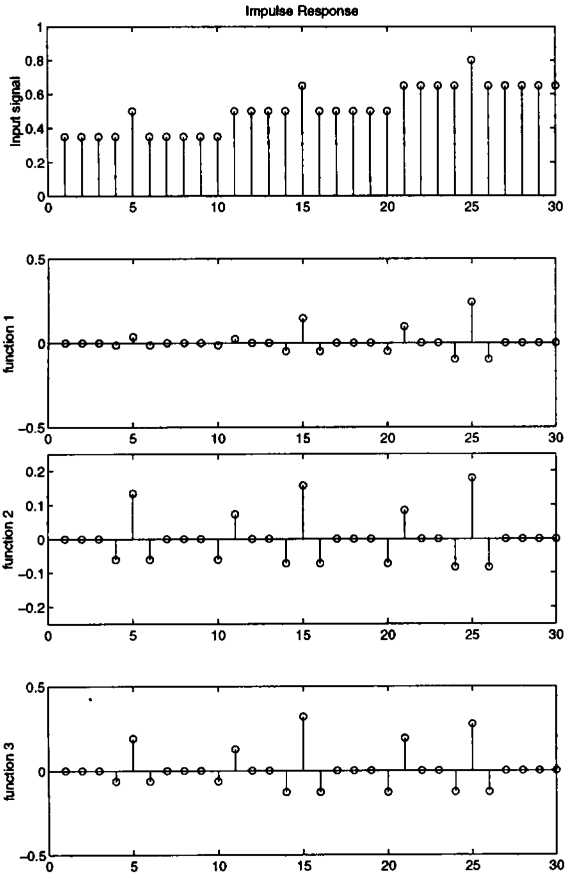

Figure 4: Premapped Teager filter test input response.

ing a single parameter. The modified filter can be simply implemented by premapping the input signal, for example by using a lookup table. Because the premapping is essentially independent of the Teager filter, the modified algorithm is applicable to other algorithms based on the Teager algorithm, such as 2-D Teager filters [3]. Therefore, this technique can be applied to applications such as unsharp masking of images. This will allow the unsharp results to be adjusted for greater or lesser sensitivity in areas of certain average intensity.

6. REFERENCES

[1] J. F. Kaiser, "On a simple algorithm to calculate the 'energy' of a signal," *Proc. of ICASSP*, vol. 1, pp. 381-384, 1990.

[2] S. K. Mitra, H. Li, I. S. Lin, T.-H. Yu, "A new class of nonlinear filters for image enhancement," *Proc. IEEE ICASSP*, Toronto, Canada, pp. 2525-2528, 1991.

[3] S. Thurnhofer, and S. K. Mitra, "A general framework for quadratic Volterra filters for edge enhancement," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 950-962, 1996.