ADAPTIVE SPATIO-TEMPORAL VIDEO NOISE FILTERING FOR HIGH QUALITY APPLICATIONS

Sitaram Bhagavathy and *Joan Llach*

Thomson Corporate Research
2 Independence way, Princeton, NJ 08540, USA.
{sitaram.bhagavathy, joan.llach}@thomson.net

ABSTRACT

High-quality digital video filtering applications pose interesting challenges. Noise reduction cannot come at the expense of losing picture detail or introducing post-filtering artifacts. We address these issues by proposing an adaptive spatio-temporal method for reducing noise in digital video. The novelty of this method is that it aims to achieve an "even" amount of noise reduction at each pixel while preserving picture detail. The emphasis on even noise reduction helps reduce post-filtering artifacts arising from uneven filtering. We further propose an algorithm for automatically estimating the key filtering parameters, given two intuitive high-level inputs, filtering strength and a tolerance value. The high-level inputs provide user-friendly "knobs" that could be adjusted to obtain the desired filtering performance.

Index Terms— digital video filtering, spatio-temporal filtering, film grain removal

1. INTRODUCTION

There is an increasing demand among consumers for high-quality digital video content, resulting in technologies such as high-definition (HD) broadcast channels, HD-DVD, and so on. Interesting challenges arise for content creators while processing video for high-quality applications. Some applications, such as pre-filtering for video compression (to reduce bit rates), require high noise reduction and are more tolerant toward losing picture detail. Others, such as movie post-production, desire noise reduction but do not tolerate loss of picture detail or post-filtering artifacts, i.e. artifacts introduced by the filtering process. These tend to use filters with conservative settings in an effort to preserve detail, which could result in uneven noise removal over the picture, which in turn could cause visual artifacts.

The challenge then is to design noise reduction filters that provide a graceful tradeoff between the amount of noise reduction and the resulting loss of picture quality. Our approach addresses this problem by adapting the filter characteristics on a pixel by pixel basis, attempting to enforce an "even" amount of noise reduction across all pixels in a video while preserving picture detail. The emphasis on even noise reduction helps reduce post-filtering artifacts which could arise from uneven filtering.

Most video noise reduction filters [1] exploit correlations between pixels in order to remove the uncorrelated noise. *Spatial* filters exploit pixel correlations in space, whereas *temporal* filters exploit correlations in time. Spatial filters are inexpensive but tend to introduce artifacts such as abrupt intensity changes and blurring of picture detail. Temporal filters avoid such artifacts. However, in order to account for temporal non-stationarity due to object and camera motion, temporal filtering commonly includes an expensive motion compensation step.

Spatio-temporal filters exploit correlations in both the spatial and the temporal directions. Previously proposed spatio-temporal filters [1] have married spatial and temporal filters in a variety of ways. Approaches that combine the two at the *frame-level* include applying temporal filtering after spatial filtering [2], computing a weighted combination of spatial and temporal estimates [3, 4], and performing temporal decorrelation followed by spatial filtering [5]. Combining spatial and temporal filtering at the *pixel-level* is expected to provide visually better results, although more expensive to implement. Such methods include adapting 3-D spatio-temporal filter coefficients at each pixel [6, 7], and nonlocal means image denoising [8].

In this paper, we propose a pixel-level adaptive spatio-temporal method. It differs from the previous methods thus: a) each pixel is filtered by averaging it with a *constant* number of temporally or spatially correlated pixels, and b) the correlated pixels are sought in a specific order. The former property aims to achieve an "even" amount of noise reduction in order to avoid artifacts. The latter aims at preserving picture detail since temporally correlated pixels are generally superior to spatially correlated ones in this regard. Complexity is kept in check by avoiding pixelwise estimation of filter coefficients. The filtering process is a simple average of pixels selected by some criteria.

Another contribution of this paper is a user-friendly method of controlling the quality of filtering. The quality of filtering

relies on many internal parameters of the filtering method. The "optimal" parameters depend on the video content. For inexperienced users, it often requires a time-consuming trial-and-error process to arrive at parameter values that result in good filtering quality. Therefore, we propose an algorithm that automatically estimates these parameters based on two intuitive high-level inputs, *strength* and *tolerance*. The high-level inputs provide user-friendly "knobs" that could be adjusted to obtain the desired filtering performance.

2. ADAPTIVE SPATIO-TEMPORAL FILTERING

Algorithm 1 outlines the proposed adaptive spatio-temporal filtering algorithm, applied to the luma component of the video. The same procedure applies to the chroma components with a small modification described later. Note that a pixel p is shorthand for $p(x, y, t)$ where x and y are the spatial coordinates and t is the frame index (time). Pixel p can be split into (p_Y, p_U, p_V), its luma (Y) and chroma (U, V) components.

The algorithm comprises three major parts, 1) selection of temporal candidates, 2) selection of spatial candidates, and 3) filtering by averaging the selected candidates. For filtering each pixel p in the video, the goal is to find N "good" candidates for averaging p with (including itself), where N is a constant through the filtering process. Let these "good" candidates be put in an averaging set $A_p = \{p, c_i; i = 1, \ldots, M\}$, where $M \leq N - 1$, and $M < N - 1$ only when enough "good" candidates are unavailable. Then the filtering process involves replacing p by the average of the elements of A_p.

The N "good" candidates are first sought in the temporal domain since temporal filtering is less likely to blur visual details. The temporal part first involves estimating the motion of pixel p from the current frame to n reference frames (usually the $n/2$ frames before and the $n/2$ frames after the current frame). Thereafter, each temporally predicted pixel q_j (from the j^{th} reference frame) is considered in turn as a candidate for A_p. If it is determined to be a "good" predictor, it is added to the set.

If we are unable to locate N "good" candidates in the temporal domain, we start looking for candidates in the spatial domain. One possibility is that we consider all pixels in a spatial r-neighborhood of p given by

$$N_r\left(p(x, y, t)\right) = \{p(x + i, y + j, t); i, j = \pm 1, \ldots, \pm r\} \quad (1)$$

where $r = r_Y$ for the luma component and $r = r_C$ for the chroma components. The value r is termed the *radius* of the neighborhood. The order in which we consider the spatial neighbors is determined by the proximity of the candidate pixel to p. If the candidate pixel is determined to be "good," it is added to A_p. Once N "good" candidates are obtained or all candidates have been scanned, we proceed to the filtering step. In this step, p is replaced by the average of the elements of A_p and we move on to the next pixel to be filtered.

One missing detail is a method of determining the "goodness" of candidates. In our implementation, a patch-based measure $d(p, q)$ is used for temporally predicted candidates and a pixel difference measure is used for spatial candidates. The patch-based measure is given by

$$d(p, q) = \left[\frac{\sum_{i,j=-w}^{w} |p(x + i, y + j) - q(x + i, y + j)|^2}{(2w + 1)^2} \right]^{\frac{1}{2}} \quad (2)$$

where w is the radius of the patch or window. The temporal predictor q is said to be "good" if $d(p, q)$ is less than a specified threshold. For spatial filtering, a candidate q is assumed to be "good" if $|p-q|$ is less than a specified threshold. Note that the thresholds for luma and chroma, T_Y and T_C, are specified separately.

The procedure for filtering the chroma components is similar to that for the luma component, except that the chroma components (U, V) are filtered together in one pass. For a pixel p, both p_U and p_V are replaced by the average of their respective candidates. The "goodness" criteria for choosing U and V candidates are the same. A temporal candidate q_j is "good" if $d(p_U, q_{j,U}) < T_C$ and $d(p_V, q_{j,V}) < T_C$. Similarly, a spatial candidate k is "good" if $|p_U - k_U| < T_C$ and $|p_V - k_V| < T_C$.

Algorithm 1 Adaptive Spatial-Temporal Filtering

```
1: Given N, the desired number of averaging candidates
2: Given the luma threshold, T_Y
3: Given the spatial neighborhood, N_{r_Y}(p)
4: for all pixels p do
5:    Initialize averaging set A_p = {p_Y}
6:    for all reference frames j = 1, ..., n do
7:       Get motion-based predictor q_j from frame j
8:       if d(p_Y, q_{j,Y}) < T_Y then
9:          Add q_{j,Y} to A_p
10:         if |A_p| = N then
11:            Skip to line 23
12:         end if
13:      end if
14:   end for
15:   for all spatial neighbors k ∈ N_{r_Y}(p) do
16:      if |p_Y − k_Y| < T_Y then
17:         Add k_Y to A_p
18:         if |A_p| = N then
19:            Skip to line 23
20:         end if
21:      end if
22:   end for
23:   Filtered pixel p_{filt,Y} ← average of all elements in A_p
24: end for
```

3. AUTOMATIC PARAMETER ESTIMATION

The automatic parameter estimation method estimates four parameters: the luma and chroma neighborhood radii (r_Y, r_C) of the spatial neighborhoods from which spatial candidates are obtained, and the luma and chroma thresholds (T_Y, T_C) used to verify the "goodness" of spatially or temporally predicted candidate pixels. The method takes two intuitive

Algorithm 2 Automatic Threshold Estimation

1: Given N, the filtering strength
2: Given α, the tolerance percentage value
3: Given the spatial neighborhood radius, r_Y
4: Detect homogeneous regions in the video. Let all pixels therein form a set H.
5: **for all** thresholds $u = 1, \ldots, 256$ **do**
6: $y \leftarrow 0$
7: **for all** pixels $p \in H$ **do**
8: $n_{p,u} \leftarrow 0$
9: **for all** spatial neighbors $k \in N_{r_Y}(p)$ **do**
10: **if** $|p_Y - k_Y| < u$ **then**
11: $n_{p,u} \leftarrow n_{p,u} + 1$
12: **end if**
13: **end for**
14: **if** $n_{p,u} < N$ **then**
15: $y \leftarrow y + 1$
16: **end if**
17: **end for**
18: $\beta_u = 100 \times y/|H|$
19: **if** $\beta_u \leq \alpha$ **then**
20: Skip to line 23
21: **end if**
22: **end for**
23: $T_Y \leftarrow u$

user inputs, *strength* and *tolerance*. "Strength" refers to the filtering strength, specifically the desired number of pixel candidates (N) to average over while filtering each pixel. The higher the N, the greater the expected noise reduction. "Tolerance" refers to the acceptable percentage, say α, of inadequately filtered pixels, i.e. pixels that do not have at least N candidates to average with. For example, α = 5% means that we shall try to choose thresholds such that no more than 5% of pixels are inadequately filtered. A lower tolerance drives the thresholds higher, thereby forcing more pixels to be adequately filtered at the expense of blurring detail. A higher tolerance goes easier on fine details by allowing more pixels to be less filtered.

Given the filtering strength N, the first task is to estimate the spatial neighborhood radius, r in (1). We choose the smallest positive integer r such that the neighborhood contains at least $2N$ pixels not including the center pixel, i.e. $r = \lceil \frac{\sqrt{2N-1}}{2} \rceil$ (where $\lceil . \rceil$ denotes the ceiling operator). This ensures that there are enough candidates in the neighborhood to choose N "good" candidates from. If the desired luma and chroma filtering strengths, N_Y and N_C, are different, their radii, r_Y and r_C, may also be different.

The next task is to estimate the candidate verification thresholds. Algorithm 2 outlines the automatic threshold estimation method. This method is applied separately to the luma and chroma components of the sequence. The first step of the procedure is to detect *homogeneous* regions in the video. Homogeneous regions are those that (except for a mean value) contain only the noise pattern that we wish to remove. Any homogeneous region detection method (e.g. [9]) can be used.

Let the set H contain all the pixels in the detected homogeneous regions. Now, all the possible thresholds, $u = 1, \ldots, 256$, are considered in order. Let the current value of the threshold be u. For each pixel $p \in H$, the number $n_{p,u}$ of neighbors $k \in N_{r_Y}(p)$ that satisfy the constraint $|p_Y - k_Y| < u$ are found. If $n_{p,u} \geq N$, then the pixel p is said to be "adequately" filtered. After scanning all pixels $p \in H$, the percentage of pixels in H that are *not* adequately filtered (say β_u) is computed. If $\beta_u \leq \alpha$, then the procedure is terminated and the luma threshold, T_Y, is set to u. The procedure for the chroma components is the same except for the evaluation of the threshold constraint. Here, a pixel p is said to be adequately filtered if at least N neighbors exist, such that $|p_U - k_U| < u$ and $|p_V - k_V| < u$.

4. EXPERIMENTAL RESULTS

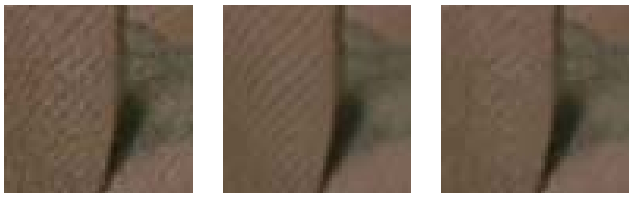
An important application of video filtering is the attenuation of "film grain" from high-resolution video content. Film grain is the grainy noise pattern in digital video introduced by scanning the grain of the original film source. It is often necessary to attenuate the film grain in digital video without losing picture detail or introducing post-filtering artifacts in the process. In this scenario, we compare the proposed algorithm with spatial filtering and temporal filtering, in order to appreciate the gain of combining the two approaches.

Three 480 × 480 video clips, which we name Clips 1-3, are cropped from different HD digital videos. These are filtered using a wide range of strengths (N = 5 to 17) and tolerances (α = 5% to 20%). Table 1 illustrates the variation of the noise reduction with strength and tolerance, considering the luma component of Clip 1. *Noise reduction* here is defined as the ratio of the empirically measured noise variance (in homogeneous regions) in the original video to that in the filtered video. Notice that the noise reduction is proportional to the strength and inversely proportional to the tolerance. The reason for the latter is that a higher tolerance leads to more inadequately filtered pixels (see Fig. 3) and thus is more likely to retain picture details.

The proposed method combines the advantages of spatial and temporal filtering to give a better visual result than either method applied by itself. Fig. 1 shows zoomed details after filtering Clip 1 using both the proposed method and spatial filtering. Note that details (better seen in the electronic pdf version) are better preserved as compared to spatial filtering. Fig. 2 shows zoomed details from Clips 2 and 3 after filtering them using both the proposed method and motion-compensated temporal filtering. The top row demonstrates the ability of the proposed method to filter noise even in high-

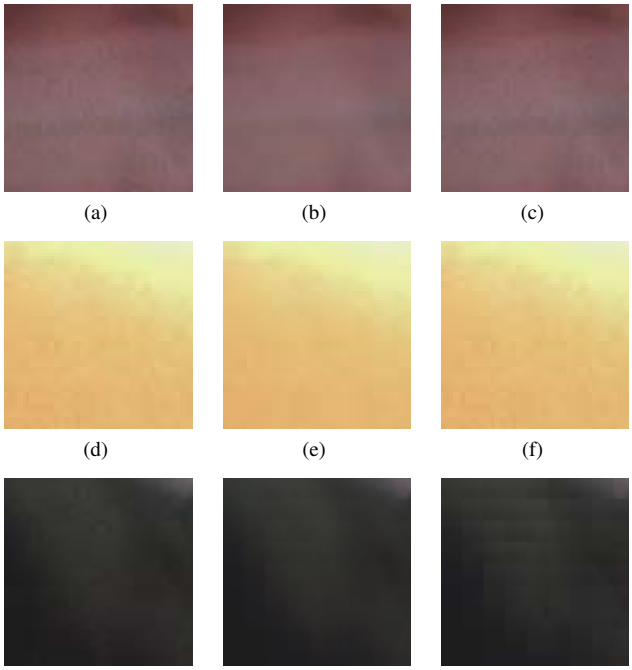
Table 1. Noise reduction in luma for different strengths (N) and tolerances (α).

	N = 5	N = 9	N = 17
α = 5%	2.3354	3.0346	3.2494
α = 20%	1.5997	2.4009	2.4142



(a) (b) (c)

Fig. 1. Comparison with spatial filtering: (a) Original; (b) proposed method; and (c) spatial filtering. Notice the blurring of details in (c).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 2. Comparison with temporal filtering: (a,d,g) Original; (b,e,h) proposed method; and (c,f,i) temporal filtering.

motion areas, wherein motion compensation (and hence temporal filtering in Fig. 2c) performs poorly. The same is observed in the case of sudden illumination changes (middle row), in the vicinity of a gunshot in the frame. The bottom row illustrates the reduced occurrence of blocking artifacts when using the proposed method. Note that the blockiness is more visible in the temporally filtered result (Fig. 2i). Blocking artifacts, a common side-effect of block-based motion compensation, are mostly smoothed away by the spatial component of the proposed approach.

5. CONCLUSION

We have proposed an adaptive spatio-temporal method for noise reduction in digital video. The proposed method provides a graceful tradeoff between the amount of noise reduction and the resulting loss of picture quality. Our approach attempts to enforce an "even" amount of noise reduction across all pixels in a video while preserving picture detail. Each

(a) (b) (c)

Fig. 3. Effect of the "tolerance" value: (a) Original; (b) Tolerance = 20%; and (c) Tolerance = 5%. Notice that (b) has more inadequately filtered pixels.

pixel is filtered by averaging it with a *constant* number of temporally or spatially correlated pixels, which are sought in a specific order. The emphasis on even noise reduction helps reduce post-filtering artifacts which could arise from uneven filtering.

Another contribution of this paper is an algorithm for automatically estimating the key filtering parameters, given two intuitive high-level inputs, filtering strength and a tolerance value. The high-level inputs provide user-friendly "knobs" that could be adjusted to obtain the desired filtering performance. This makes it easier for the user to achieve the aforementioned strength/quality tradeoff.

References

[1] J. C. Brailean, R. P. Kleihorst, S. Efstratiadis, A. K. Katsaggelos, and R. L. Lagendijk, "Noise reduction filters for dynamic image sequences: a review," *Proc. IEEE*, vol. 83, no. 9, pp. 1272–1292, Sep 1995.

[2] A. K. Katsaggelos, J. N. Driessen, S. Efstratiadis, and R. L. Lagendijk, "Temporal motion compensated noise filtering of image sequences," in *Proc. SPIE Vis. Comm. and Image Process.*, 1989, pp. 61–70.

[3] R. Dugad and N. Ahuja, "Video denoising by combining Kalman and Wiener estimates," in *Proc. IEEE Int. Conf. on Image Process.*, 1999, pp. 152–156.

[4] H. Y. Cheong, A. M. Tourapis, J. Llach, and J. Boyce, "Adaptive spatio-temporal filtering for video de-noising," in *Proc. IEEE Int. Conf. on Image Process.*, 2004, pp. 965–968.

[5] K. J. Boo and N. K. Bose, "A motion-compensated spatio-temporal filter for image sequences with signal-dependent noise," *IEEE Trans. CSVT*, vol. 8, no. 3, pp. 287–298, Jun 1998.

[6] M. K. Özkan, M. I. Sezan A. T. Erdem, and A. M. Tekalp, "Efficient multiframe Wiener restoration of blurred and noisy image sequences," *IEEE Trans. Image. Process.*, vol. 1, pp. 453–476, Oct 1992.

[7] M. El Hassouni, H. Cherifi, and D. Aboutajdine, "HOS-based image sequence noise removal," *IEEE Trans. Image. Process.*, vol. 15, no. 3, pp. 572–581, Mar 2006.

[8] A. Buades, B. Coll, and J. M. Morel, "Denoising image sequences does not require motion estimation," in *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, Sep 2005, pp. 70–74.

[9] P. Schallauer and R. Mörzinger, "Rapid and reliable detection of film grain noise," in *Proc. IEEE Int. Conf. on Image Process.*, 2006.