LOW COMPLEXITY TWO CLASSES GAUSS WEIGHTING FILTER FOR NOISE REDUCTION IN MOBILE RECEIVED ANALOG TV SIGNALS

Markus Friebe and André Kaup

Chair of Multimedia Communications and Signal Processing,
University Erlangen-Nuremberg, Cauerstr. 7, 91058 Erlangen
{friebe,kaup}@LNT.de

ABSTRACT

We introduce a low complexity method for noise reduction in mobile received analog TV signals. The two classes Gauss weighting filter is smoothing images while preserving image details and motion regions. This approach classifies neighboring image samples either as an edge or non-edge sample. Only image samples which belong to non-edge samples are used for noise reduction. Image samples which belong to edges are excluded from filtering.

1. INTRODUCTION

Noise reduction is an important task in the field of image processing. It is introduced into TV signals by the receiver and is very annoying for human perception. In Fig. 1 a region of a mobile analog received image is shown. The receivers Automatic Gain Control (AGC) unit has to amplify the video signal to a certain amplitude level. So, in bad receiving conditions more noise is introduced in TV signals as in good receiving conditions. The noise has been analysed in section 2.1 and is similar to additive Gaussian distributed noise. Therefore simulations are done with additive Gaussian noise. In Fig. 2 the measured noise and a reference Gaussian distribution is shown.

V. Zlokolica introduced the Best-Neighbor image sequence filter (BN) in [1]. This approach uses Best-Neighbors for image smoothing. Further, he introduced an adaptive version of a K-Nearest-Neighbor (Adapt KNN) filter in [2] for additive Gaussian noise. An adaptive window size is reducing motion artifacts. Additionally, image details are preserved by an adaptive filter length. An adaptive multiple class averaging filter (THRF) is introduced by V. Zlokolica in [3]. Neighboring image samples are classified regarding their amplitude values. For each class, different weighting factors exist for image filtering. These filters uses spatial and temporal image samples for filtering and are compared to other well know algorithms for noise reduction in [3]. C. Tomasi uses Bilateral filtering [4] for Gaussian noise reduction. This method prefers near values to distant values by using Gauss weights for both spatial distance and gray level difference. Bilateral filtering uses only spatial image samples for noise reduction. Methods [1] to [4] are used as reference to the proposed algorithm. All these methods blur image edges and motion artifacts are visible.

Using the two classes Gauss weighting (TCGW) approach edges can better be prevented from image smoothing and motion artifacts can further be reduced. The TCGW approach uses neighboring image samples in spatial and temporal direction for filtering. These samples are classified in two classes (edge or non-edge samples) regarding their intensity values. Edges occur in detailed images in spatial and in case of motion in temporal direction. If image samples are classified as edge samples, they are excluded from filtering. Otherwise they are used for filtering. For neighboring samples which are classified as non-edge samples, Gauss weighting coefficients are calculated. Using these coefficients, a Gauss weighted summation is done to compute the noise reduced image intensity value.

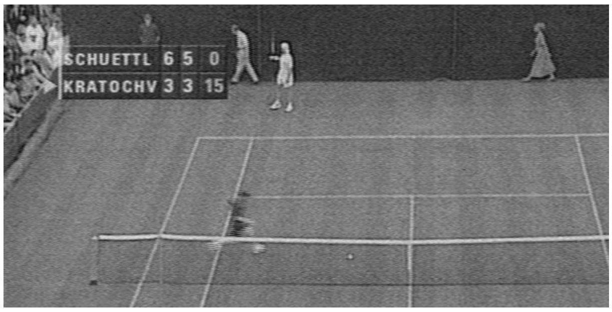

Figure 1: Mobile received image.

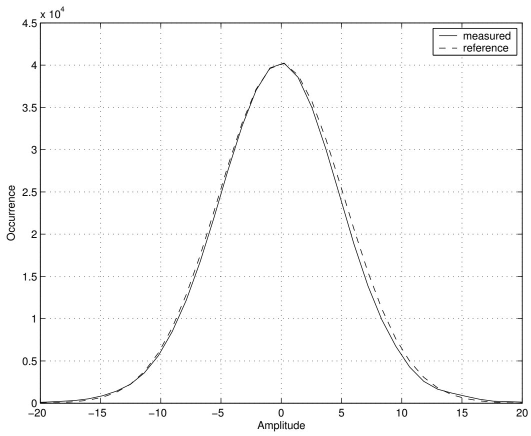

Figure 2: Noise distribution of the mobile received image.

In section 2 the proposed algorithm is described in detail. Simulation results are discussed in Section 3 before section 4 summerises the results of this approach.

2. ALGORITHM

In Fig. 3 the block diagram of the two classes Gauss weighting (TCGW) approach can be seen. The block “Noise Estimation” estimates the standard deviation of Gaussian noise in an image. The block “Structure and motion analysis” analyses details in spatial and temporal direction. The block

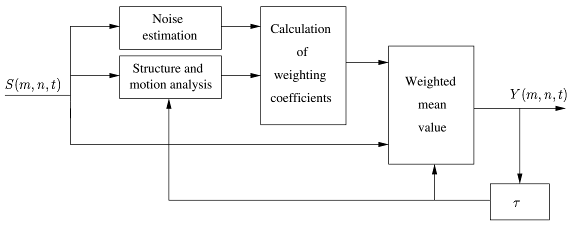

Figure 3: Block diagram of the two classes Gauss weighting filter.

"Calculation of weighting coefficients" computes a weighting factor for each non-detailed neighboring image sample. The output value is computed by a weighted mean of non-edge neighboring samples in the "Weighted mean value" block. In the following, S represents the input and Y the output image with frame height M and frame width N. m and n are indices for rows and columns. t represents the current and $t-1$ the previous video frame.

2.1 Noise Estimation

A fast method for noise variance estimation in introduced in [5]. The block "Noise Estimation" uses this method for estimating the standard deviation σ_n of additive Gaussian noise. The noise standard deviation σ_n is computed by using a separable Laplacian operator

$$H = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}. \tag{1}$$

The current input image samples $S(m,n,t)$ are convolved with the Laplacian operator H

$$S_H(m,n,t) = H * S(m,n,t) \tag{2}$$

and σ_n can be estimated by [5]

$$\sigma_n(t) \approx \sqrt{\frac{\pi}{2}} \frac{1}{6MN} \sum_m \sum_n |S_H(m,n,t)|. \tag{3}$$

The estimated standard deviation σ_n is used for computing weighting coefficients.

2.2 Structure and motion analysis

The block "Structure and motion analysis" considers image samples of the current input frame $S(m,n,t)$ and the previous output frame $Y(m,n,t-1)$. The image samples of a $3 \times 3 \times 2$ window are copied to an input vector

$$\mathrm{s} = (S(m-1,n-1,t),...,Y(m+1,n+1,t-1)). \tag{4}$$

The recursive behavior of this approach is achieved by using previous output samples $Y(m,n,t-1)$ in vector s. For detail analysing, the absolute differences of neighboring window samples

$$\boldsymbol{\Delta} = (|S(m,n,t) - s_1|, ..., |S(m,n,t) - s_L|) \tag{5}$$

with

$$L = 18 \tag{6}$$

to the center sample $S(m,n,t)$ are computed. To prevent edges from smoothing, the neighboring image samples are classified. For estimating image edges, the absolute differences Δ_l are compared to a threshold T_1.

$$C_l = \begin{cases} 0, & \Delta_l > T_1 \\ 1, & \Delta_l \leq T_1 \end{cases} \tag{7}$$

with

$$l = 0, ..., L \tag{8}$$

C_l represents the class for a neighboring sample with index l. If an absolute difference is greater than T_1, then this neighboring sample is classified as an edge sample ($C_l = 0$) regarding the center sample and is excluded from filtering. If an absolute difference is smaller or equal T_1, then this neighboring sample is classified as a non-edge sample ($C_l = 1$) and is included for filtering. The weighting coefficients will be computed as in the following.

2.3 Calculation of weighting coefficients

The weighting coefficients are computed for each neighboring sample in the "Calculation of weighting coefficients" block. These coefficients follow piecewise an Gauss function. In Fig. 4 a two classes weighting function for two different noise standard deviations $\sigma_n \approx 5$ (which represents the measured σ_n in Fig. 1) and $\sigma_n = 20$ is shown. If a neighboring sample with the absolute difference Δ_l is classified as an edge sample ($C_l = 0$), then this sample is excluded from filtering. The weighting coefficient w_l is set to zero. If a neighboring sample with the absolute difference Δ_l is classified as a non-edge sample ($C_l = 1$), then this sample is used for filtering. The weighting coefficient w_l is computed by an Gauss function. The parameter β is a constant value which spreads the Gauss function.

$$w_l = \begin{cases} 0, & C_l = 0 \\ e^{-\beta(\frac{\Delta_l}{\sigma_n})^2}, & C_l = 1 \end{cases} \tag{9}$$

After computing the weighting coefficients, the output value $Y(m,n,t)$ is computed in block "Weighted mean value" by the weighting vector $\mathbf{w}$ and the input vector s

$$Y(m,n,t) = \frac{\mathbf{w}^T \mathbf{s}}{\sum_l w_l}. \tag{10}$$

The computing time can be reduced by using pre-calculated weighting coefficients for each Δ_l and a lookup table.

3. SIMULATION RESULTS

The proposed algorithm is used in three different versions. Version (TCGW) is the algorithm proposed in section 2. Version (GW) is a pre-stage to the TCGW approach and uses a Gauss weighting function without classification in equation (9). Neighboring samples are always classified as non-edge samples ($C_l = 1$). (TCGW non-rec.) represents a non-recursive version of TCGW. The input vector s includes samples of the current and previous input images S. Equation (4) is changed to

$$\mathrm{s} = (S(m-1,n-1,t),...,S(m+1,n+1,t-1)). \tag{11}$$

Table 1: PSNR results for $\sigma = 10$ in dB

Video $\sigma = 10$	Coast-guard	Fore-man	Miss america	Sales-man
Noisy video	28.15	28.14	28.13	28.14
BN [1]	26.17	27.75	34.93	30.73
Adapt KNN [2]	29.14	31.88	35.43	31.97
THRF [3]	29.32	32.35	35.58	31.48
Bilateral [4]	30.83	33.57	35.66	32.28
GW	30.51	33.34	**37.04**	32.87
TCGW	30.60	33.38	37.01	32.95
TCGW non-rec.	**31.13**	**33.68**	36.11	**33.19**

Table 2: PSNR results for $\sigma = 20$ in dB

Video $\sigma = 20$	Coast-guard	Fore-man	Miss america	Sales-man
Noisy video	22.18	22.23	22.16	22.18
BN [1]	25.28	26.53	31.99	28.99
Adapt KNN [2]	26.74	28.63	31.09	28.90
THRF [3]	27.07	28.89	30.53	28.48
Bilateral [4]	26.36	27.41	28.05	27.04
GW	27.48	29.77	**33.20**	29.71
TCGW	27.50	**29.78**	33.15	**29.73**
TCGW non-rec.	**27.56**	29.40	31.18	29.25

The threshold T_1 for edge preserving and β the spread factor of the Gauss function has been analysed. With $T_1 = 4\sigma_n$ and $\beta = 0.125$ good noise reduction and edge preserving can be achieved.

In Table 1 mean PSNR results for frame (1-100) in four different videos with noise standard deviation $\sigma = 10$ are shown. The bold PSNR values represents the best match to the original undistorted video. The results show that for videos containing lower noise, image details, global and local motion (“Coastguard”, “Foreman” and “Salesman”), the non-recursive TCGW performs better than the recursive GW and TCGW. The Bilateral filter uses only spatial image samples for noise reduction and performs therefore also well. In Table 2 PSNR results for noise standard deviation $\sigma = 20$ are shown. For higher noise, the recursive GW and TCGW performs better than the non-recursive TCGW and Bilateral.

In Fig. 5 PSNR plots for frames (1-100) of the “Salesman” video with $\sigma = 10$ are shown. Bilateral and THRF filtering reaches very smooth PSNR curves comparing to the other methods. The highest PSNR variance is obtained by BN filtering. In case of local motion (frame 1-20) PSNR values changes up to 2 dB. The non-recursive TCGW achieves high and smooth PSNR results. TCGW and GW performs quite equal with a small offset. The small offset is introduced by using two classes Gauss weights.

Due to limited space only parts of images are shown (full images can be seen on [6]). In Fig. 6 on the left hand side an original undistorted and on the right hand side a corrupted “Salesman” frame 21 with $\sigma = 10$ can be seen. The result from BN filtering is shown in Fig. 7 on the left hand side and from THRF filtering on the right hand side. BN introduces edge blurring and motion artifacts, whereas THRF introduces only edge blurring. In Fig. 8 on the left hand side the

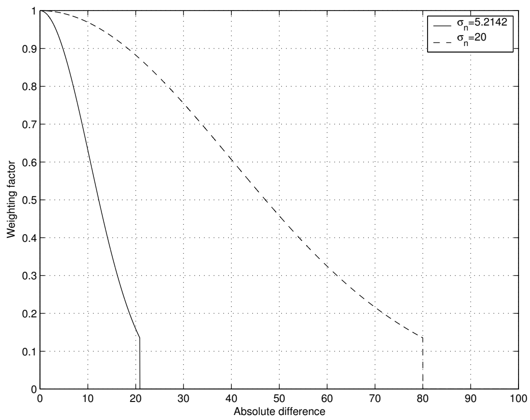

Figure 4: Two classes Gauss weighting function.

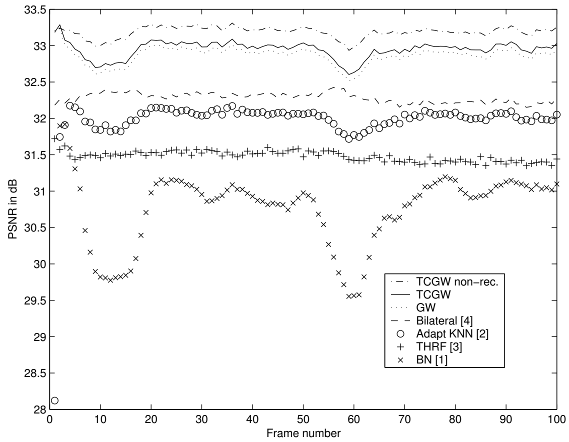

Figure 5: PSNR values for “Salesman” corrupted with additive Gaussian noise ($\sigma = 10$).

result from Bilateral and on the right hand side from TCGW filtering is shown. Bilateral filtering does not blur edges to much, but noise is still visible at the man’s forehead. Motion artifacts are not introduced, because this filter works only in spatial domain. The proposed TCGW algorithm reduces noise very well and edges are not blurred. Motion artifacts are further reduced by detecting edges in temporal direction.

PSNR results between GW and TCGW are quite equal. In Fig. 9 a difference image for frame 74 between the result from GW and TCGW is shown. In case of motion (left shoulder, the cube and the right wrist) and on the books edges, differences are visible. Through the classification of neighboring image samples in the TCGW approach, edges are detected and excluded from image smoothing. For these image samples differences are visible in Fig. 9. TCGW prevent such samples from image smoothing and videos appears without motion artifacts and with sharper edges. This is the main benefit of the TCGW algorithm.

In Fig. 10 on the left hand side a noisy mobile received image and on the right hand side the result from TCGW can be seen. In homogenous areas like the tennis field, subjectively well noise reduction is achieved. Edges seams not to be blurred and motion artifacts do not appear.

Figure 6: Left: Original image. Right: Image corrupted with additive Gaussian noise ($\sigma = 10$).

Figure 7: Left: Result from BN. Right: Result from THRF.

Figure 8: Left: Result from Bilateral. Right: Result from TCGW.

Figure 9: Difference between GW and TCGW for "Salesman" with $\sigma = 10$.

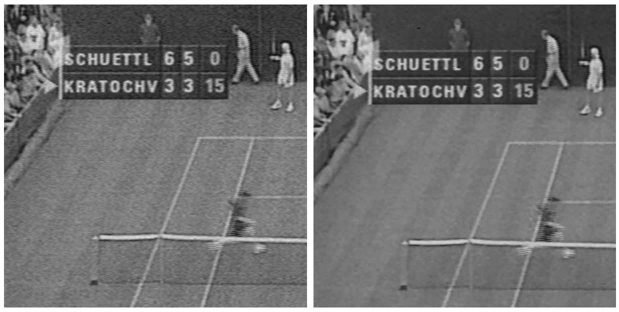

Figure 10: Left: Mobile received image. Right: Result from TCGW.

4. CONCLUSION

We presented a low complexity method for reducing additive Gaussian noise. In contrary to BN, Adapt KNN, THRF and Bilateral filter, our two classes Gauss weighting approach yields better PSNR results while preserving edges and reducing motion artifacts. For detailed videos with global and local motion containing lower noise ($\sigma = 10$), Bilateral filtering performs also well. PSNR results have shown that for lower noise non-recursive methods work better than recursive. For higher noise ($\sigma = 20$) the recursive methods perform better than non-recursive. Due to the human visual system, through filtering with recursive methods the videos appear smoother and less annoying. Also videos containing lower noise appear subjectively better for visual system by recursive filtering methods, like our two classes Gauss weighting approach.

In our complete system, the TCGW approach is used in combination with other streak noise reduction, intensity flicker, ghost cancel and deinterlacing methods for quality enhancement of mobile received analog TV signals.

REFERENCES

[1] V. Zlokolica, W. Philips, D. Van De Ville, "A New Nonlinear Filter for Video Processing," *IEEE Benelux Signal Processing Symposium*, pp. 02.1-02.4, March 2002.

[2] V. Zlokolica and W. Philips, "Motion and Detail Adaptive Denoising of Video," *Proceedings of SPIE*, Vol. 5298, pp. 403-412, May 2004.

[3] V. Zlokolica, A. Pizurica and W. Philips, "Video denoising using multiple class averaging with Multiresolution," *Visual Content Processing and Representation, 8th International Workshop*, pp. 172-179, Springer Verlag, September 2003.

[4] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proceedings of IEEE International Conference on Computer Vision*, pp. 839-846, January 1998.

[5] J. Immerkær, "Fast Noise Variance Estimation," *Proceedings of Computer Vision and Image Understanding*, Vol. 64, pp. 300-302, September 1996.

[6] http://www.lnt.de/lms/research/projects/Qu_en_mo_re_vi/tcgw_noise_reduction.html.