

A TEMPORAL MOSQUITO NOISE CORRECTOR

Claire Mantel^{1,2}, Patricia Ladret¹

Thomas Kunlin²

¹ GIPSA-Lab - Department of Signal and Images
Grenoble Institute of Technology - UMR CNRS 5216
961 rue de la Houille Blanche
Grenoble - France

² ST Microelectronics S.A.
12 Rue Jules Horowitz B.P. 217
Grenoble - France

ABSTRACT

Due to the required memory space and computing time, temporal processing is rarely used even in video processing. However, when the temporal aspect of a compression noise is major, it cannot be treated spatially only. The following article presents an example of such a noise: the mosquito noise. The equally spatial and temporal nature of the mosquito artifact is first explained through its definition. It is followed by the different ways already used to treat it. A spatio-temporal and compression-independent mosquito noise corrector is then introduced. Finally, the interest of a temporal corrector over purely spatial ones is shown through tests results.

Index Terms— Mosquito Noise, Video, Correction, Temporal Filtering, Temporal Artifact

1. INTRODUCTION

Temporal processing is seldom used for denoising a decompressed video because of the complexity involved. However, in [1] Anderson and Anderson show that images are not perceived by the Human Visual System (HVS) as what they are: a succession of static images, but that their temporal facet is interpreted as a real one. In other words, video motion goes through the same treatment as natural motion. As a result a video is not just series of images to our eyes, and considering its temporal aspect is simply emulating the HVS.

Temporal variations have been taken into account in video compression since its very beginning. To gain bandwidth temporal redundancy of a video was profited from by detecting still zones that did not need to be coded anew in H.120, the first video compression standard released in 1984, and by compensating motion since the MPEG1 standard in 1988.

As the main principles of the different video standards have not changed for a while (motion compensation and DCT-based transform are present in MPEG2, MPEG4 and H.264), the type of defects resulting from video compression remains the same. However, with the expansion of LCD displays, High Definition sources and monitors, the perception and expectations of video quality altered. Precisely, it is the respective annoyance of the different noises compared to one

another that has completely changed: for example high frequency noises, who were faded by the low pass answer of cathodic monitors, are emphasized by the increase in displays size and the sharpness of the pixel grid in LCD.

This paper deals with such an artifact: the mosquito noise. Whereas other purely spatial defects (such as blocking or ringing) have been thoroughly studied, this one has been less addressed in literature. The temporal aspect is especially major for this noise since it would not be that annoying in a static image but is mainly noticeable due to its variation through time. Our purpose was to build an efficient corrector that takes into account the temporal feature of this noise so it contains both a spatial and a temporal part. To extend the scope of our corrector as much as possible, we set it up to be totally independent from the compression parameters or information.

The interest of treating mosquito noise temporally will be explained in the first part of this article, describing its both spatial and temporal nature. A second paragraph will sum up the different denoising processes found in the literature review. Then we will describe our Temporal Variation-Inverse Filter and finally compare it with other algorithms.

2. MOSQUITO NOISE: A COMPLICATED DEFINITION

The first difficulty when studying mosquito noise is to establish what we actually deal with: there are as many definitions as there are publications on that topic. In its recommendation for assessing subjective video quality [2], the ITU-T defines mosquito noise as: "*Form of edge busyness distortion sometimes associated with movement, characterized by moving artifacts and/or blotchy noise pattern superimposed over the objects (resembling a mosquito flying around a person's head and shoulders).*" and the edge busyness by "*Distortion concentrated at the edge of objects, and further characterized by its temporal and spatial characteristics.*". Though those definitions should be a reference, they are somehow open and not sufficient.



2.1. Resemblance to other artifacts

The ITU characterizes the mosquito effect with a number of specificities which are unfortunately also valid to other kinds of artifacts. By extension, we can see that it is difficult to identify, and easy to mistake for other artifacts. Mosquito noise can typically be assimilated to ringing, since they are both located next to the edges. Some, as Segall et al. in [3], clearly integrate the temporal variations of ringing as mosquito artifact; while others, such as Coudoux et al. in [4], make even no difference between those two noises. There is actually no definition of "ringing" in the ITU documentation [2]. This artifact is in fact purely spatial and has common properties with the mosquito noise: they both appear on unified zones near the border of objects.

Another difficulty to consider with mosquito noise is that its cause is not clearly defined. As quantization is the only source of information loss in compression, it is inevitably linked to mosquito noise, but that is also true for every other compression noise.

2.2. Spatial aspects

Almost all papers on mosquito noise agree on the fact that it appears near the border of objects. Several articles put an additional indication concerning this kind of zone: it must be near a portion of intense activity. Some authors, such as Coudoux et al. in [4], specify that mosquito effects can commonly be seen at the border of uniform or nearly uniform zones (denoted as "flat"). Others, as Westen et al. in [5], also indicate that the blocks presenting mosquito artifact must contain both flat and textured areas. DelCorso et al. assert in [6] that mosquito noise is located near borders of objects in motion.

The second major feature of mosquito noise is its really low intensity. Every scientific paper concurs on this point: in a region where mosquito is present, the oscillations around the right gray level due to noise are quite small [4, 6, 7]. This explains why the artifact is invisible in portions including many details, or textured zones.

To sum up, if the various articles on the subject sometimes diverge about the surroundings of mosquito noise, a characteristic that is globally accepted is that the noise level is very low **compared** to the intensity variation of the image (edges or textures) neighboring the noise.

2.3. Temporal aspects

There are two different ways to consider the temporal aspect of mosquito noise. The first is a perceptual one: the reason this small-amplitude noise is a nuisance is that it varies through time as stated in [3]. The fact that the temporal oscillations are not a consistent motion and that their frequency is the video framerate, prevents the HVS from stabilizing on the noise. It is this "discontinuity along the temporal axis", as

named in [4], that is perceived as annoying. Since peripheral vision is very sensitive to movement, it can be noticed everywhere in the frame and not only in salient regions.

The second way is its relationship to the temporal part of video compression. In the most common video standards, some blocks are coded using inter-coding (or predictive coding). They are first motion-compensated regarding a reference frame and then only the remaining of the difference between the current block and its motion compensation are coded. That is what we call "temporal aspect" of video compression. Opinions are divided on whether mosquito effect can appear in immobile sequences, as asserted by Fenimore et al. in [8] whereas DelCorso et al. insist in [6] on the fact that the presence of a moving object and a motionless background are required for mosquito noise to occur. But it is clear from all papers on the subject that the coding of residuals after motion compensation increases the presence of mosquito noise. We can also deduce that there is a correlation between the fact that a sequence includes a wide number of movements and the presence of mosquito noise.

2.4. Spiral videos

A completely different approach has been set up by Fenimore et al. in [8], where they display a mosquito noise reference metric. When they confront themselves with the problem of separating mosquito noise from others to measure it exclusively, they adapt the videos they are using and not their metric. They describe video sequences in which compression noises is strictly mosquito noise. Their videos are made of black spirals moving on a white background.

Creating such videos allowed us to make sure that their compression produced mosquito noise as the most annoying artifact in MPEG2 (figure 1). For the H.264 standard results were less distinct, so we produced new sequences where the spirals were turning and shifted throughout the framework. The motion compensator of the encoder does not handle rotations, therefore it could not correctly estimate the current motion. It clearly appeared that by putting it in difficulty, we greatly increased the presence of mosquito noise (figure 1). As modifying motion in videos can only affect inter coded macroblocks, the link between mosquito noise and the type of encoding of a macroblock is thus proved by this experiment.

The most probable explanation of the presence of mosquito effects in image sequences is given by Abbas in [7]: the noise is not due to a unique cause, but the conjunction of several criteria. This approach allows understanding the variety of interpretations and more specifically to clarify the spatial/temporal ambivalence: the origins and the perception of the mosquito noise are both spatial and temporal.

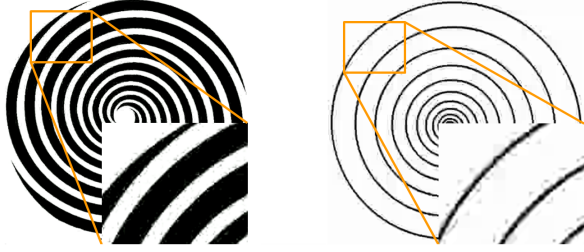


Fig. 1. Mosquito noise on spiral videos created by compression using MPEG2 (left) and H-264 (right)

3. MOSQUITO DENOISING METHODS

Since the focus of this paper is correcting mosquito noise, the state-of-the-art will not address pre-processing methods (as [5]) but only post-processing ones that are totally independent from compression.

Practically, proximity of mosquito noise to contours means that correctors must be precise enough to suppress it without damaging edges. To meet this constraint, various strategies have been adopted. In [4] Coudoux et al. first mark as edge pixels those which gradient is higher than 80 and afterward use that to tag each 8x8 block as including or not borders. They filter only the non-contour pixels in the blocks labeled as "containing borders", using a 2-D Wiener filter to remove the noise. We agree with Abbas and Karam ([7]) that it raises two issues: the non-correlation hypothesis of the Wiener filter is not respected since mosquito noise is not decorrelated from the signal (2.2) and a pixel can end-up being filtered with pixels belonging to another object, thus blurring the contours.

In [7] Abbas and Karam employ a Canny edge detector to find out where contours are precisely and then apply their filter to the whole image except to edge pixels as Coudoux et al.. To meet with spatial characteristics of mosquito effect, they chose ϵ -filter: a simple averaging filter which restrains its support to pixels that are within an ϵ from the current pixel. They state their Recursive ϵ -filter (REF) in which an ϵ -filter is recursively applied to correct mosquito noise, using the standard deviation of a block to approximate the one of the noise. Same as Coudoux et al., the edge pixels are not altered but may serve to filter pixels belonging to other objects. Indeed, in some cases the standard deviation might be high enough to introduce pixel of another object in the filter support. That might happen if a block is cut in half by the frontier between two objects, or when the edges are not sharp. It results in introducing blur.

As it has been said in 2.3, DelCorso et al. focus their corrector on moving boundaries only ([6]). Thus, their solution is applied on the entire image and uses a motion detector, skipping the edge detector part. But the surprising part is that, in order not to degrade moving zones, they do not filter them. They consider that mosquito noise is more visible on the background close to moving objects than on the ob-

jects themselves. Anyway since they partition the frame into 8x8 blocks, they miss potentially noisy pixels contained in moving blocks. They utilize the small amplitude of mosquito noise by asserting that correcting the average value of an 8x8 block is enough. To do so, they first apply a median filter on DC values of the current, the upper and lower blocks. The result is once again filtered by a median filter with the DC coefficient of the equivalent block in the previous and following frames.

In the end, the temporal feature of mosquito noise has been either not addressed at all, as in [7] or [4] or only for motionless areas through a motion-adapted filter in [6].

4. THE TEMPORAL VARIATION-INVERSE FILTER

The corrector we are to present here, the Temporal Variation-Inverse Filter (T-VIF), covers all three aspects of the mosquito noise definition. It is composed of a filter, the Variation-Inverse Filter (VIF) that will be described in 4.1, which support will be adjusted according to the two other major features of mosquito noise: its proximity to edges (4.2) and its temporal variation (4.3). Finally a fast motion-adapted version of our filter, especially accommodated for mosquito, is explained in 4.4.

4.1. Small Gray Level Fluctuation

The Variation Inverse Filter is a 1-D filter set up by Cr  t  roffet in [9] to suppress ringing artifact. We will describe here the vertical filtering process. It is an answer to one of the main issues in treating ringing artifact: to distinguish correctly what is noise and what is an edge. As noise amplitude depends directly on the edge strength, the dynamic of a ringing effect due to a strong edge can be higher than the one of a weak edge present in the same image. The first step of the VIF is thus to fit the strength of the filter accordingly to the one of the edge. As it was designed for MPEG2, the normalization *Norm* is made according to the following formulas, for (i,j) covering a whole 8x8 DCT block in the current image *Im*:

$$D_{i,j} = |Im_{i,j+1} - Im_{i,j}|$$

$$Max = \max(D_{i,j}) \text{ and } Min = \min(D_{i,j})$$

$$\text{If } Max \neq Min, \text{ Norm}_{i,j} = Thresh_{max} \cdot \frac{D_{i,j} - Min}{Max - Min}$$

If *Max* and *Min* are equal then variations are constant and the noise cannot be separated from the edge so we will not filter. *Thresh_{max}* represents the number of possible levels.

The VIF applies then an adaptive filtering. It relies on the idea that the more a neighboring pixel is distinct from the pixel being filtered, the more likely they do not belong to the same object. The weight given to each pixel is therefore based on the inverse of its difference with the current pixel. The coefficients $W_{i,j\pm 1}$ of pixels (i,j-1) and (i,j+1), which are used

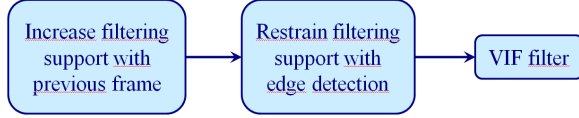


Fig. 2. The T-VIF structure.

to filter vertically the pixel (i,j) , are calculated according to the subsequent equations, where $Filter_Norm$ is the norm of the filter:

$$W_{i,j\pm 1} = \frac{1}{Filter_Norm} \cdot \frac{1}{Norm_{i,j\pm 1} + 1}$$

As explained in 2.2, the amplitude of mosquito noise is low regarding the neighboring edge amplitude. This is why the VIF is not only adapted for ringing artifact but also suited for the spatial characteristics of mosquito noise. Indeed, it was set up to correct a noise of small-amplitude measured up to an edge variation. Anyhow, the VIF needed to be adapted to 3-D filtering (2-D plus temporal) and to other partitions of image than MPEG2's 8x8 blocks. As gradient calculation is a stage of border detection we already have them available. We used the highest gradient value in the 15x15 neighboring of the currently filtered pixel as Max and the lowest as Min . The result $F_{i,j}$ is then calculated the following way:

$$F_{i,j} = \frac{1}{Filter_Norm} \cdot \left(1 + \sum_{(k,l) \in \{-1,1\}^2} \frac{1}{Norm_{i+k,j+l} + 1} \right)$$

The paragraphs 4.2, 4.3 and 4.4 modify the support of the VIF filter. Figure 2 shows the global structure of the T-VIF.

4.2. Close to Contours

As stated in 2.2, mosquito noise is present beside object edges. Thus, we decided to use an edge detector for two distinct reasons: the first is to determine for each pixel whether it presents a risk of mosquito. If a pixel could be within the same quantization block as a contour pixel during encoding, practically if it is in an 8x8 neighborhood of a contour pixel, it has to be filtered, otherwise no. Filtering uniquely where there may be ringing or mosquito noise allows time saving because the filter is only applied on "potentially noisy" pixels and also eliminates risk of degrading healthy zones that do not need correcting.

The other advantage to detect edges is the possibility to adapt the filter support to the frame content. To avoid degrading or blurring edges, we decided to restrain the filter support to pixels with the same tag from the edge detector (i.e. border/no border). Our aim is not to filter edges with edges and objects with objects. Indeed, it would not match reality because, except for synthetic videos, there is never a one-pixel border surrounding each object in videos: the "border" is part of an

object.

The benefit of restricting the filter support that way is to make sure we only use pixels belonging to the same object to filter. We chose Canny's edge detector because it provides exclusively one-pixel width edges. As we filter on a 3x3 neighborhood, the only case in which two pixels of the neighborhood can be part of different items is if there is an edge pixel between them. Since edges obtained through Canny's detector are connected, it means that the central pixel is an edge one. And in our algorithm it will be filtered using only other edge pixels.

If the edge detector was to make a mistake, the VIF also offers a security: if by mistake an outsider pixel was to be added to the filter support, its influence would be minimal due to its difference from the filtered pixel.

4.3. Temporal Variation

The major specificity of the T-VIF is its temporal aspect since, as shown in 3, there is currently few temporal corrector for mosquito noise and none addressing both the edge issue and the temporal one. As mosquito noise varies from one frame to another, our idea is that using the previous frame in the support of the filter will help converging toward the right value. Another gain is that using the time-preceding pixels to filter will diminish the temporal discontinuities. Thus we decided to modify our VIF into a Motion-Compensated spatio-temporal filter to improve the temporal smoothness of the processed sequence.

We used the Robust Optic Flow estimation from Black and Anandan stated in [10] and whose code is available on their personal webpage. Applied between two images, it provides motion vectors that optimize the match between the two images with pixel granularity. A motion-compensated frame is then created. The 3x3 motion-compensated neighborhood with identical coordinates as the current pixel is then adjoined to the filter support. This add-on follows the same sorting (by membership to border or non-border class) as the original neighboring pixels.

4.4. Fast Temporal Filtering

Motion compensation is quite time and resource consuming. We studied another way to add pixels from a prior image in the filter support: turning our T-VIF into a Motion-Adaptive Filter. This version will be called Fast-T-VIF in the paper. The advantage with mosquito effect is that its small amplitude and its alteration from frame to frame enable using a simple but efficient approach. The concept is to classify pixels not accordingly to their stillness but to their belonging to the same object or not. To guarantee that, the highest gradient amplitude in the 8x8 neighboring, which is identified as a contour one, serves once again as a standard besides which other pixels will be evaluated.

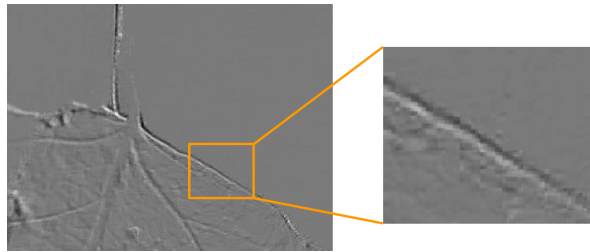


Fig. 3. Differences between image 17 and 16 of Aspen.

A difference frame is calculated between the current and preceding images (figure 3). Each neighboring pixel in the previous frame is then appended to the filter support if this difference is relatively smaller than the gradient maximum (i.e. if they are part of the same object). Like for other VIF methods, pixels are then sorted by their belonging to a contour or not before being used.

5. SIMULATION RESULTS

5.1. Video used

In this section the versions of VIF will be compared to the REF. The video sequences used are Aspen, InToTree (from the VQEG website: www.its.bldrdoc.gov/vqeg/) and Life (from www.hdgreetings.com/other/ecards-video/video-1080p.aspx). Aspen and IntoTree are natural videos with respectively wide and small motion, while Life is a synthetic video with average motion.

All video sequences were retrieved in "perfect" quality, cropped to SD and then encoded using the JM13 from the JVT, which favorably allows canceling the deblocking filter through its parameters. The chosen bitrates were respectively 1Mbps for Aspen and 600Kbps for the others. The three sequences were cropped to 500x500 format.

5.2. Spatial Analysis

The spacial results are evaluated through SSIM and PSNR computing between the corrected frames and the original ones. The corresponding graph is shown for Life in figure 4. For this video, the results are clearly in favor of the T-VIF and Fast-T-VIF.

The results for Aspen are slightly better for the T-VIF and Fast-T-VIF than for REF. Regarding the InToTree video, the SSIM values are equivalent, but they are also similar to those of the compressed video although the corrected sequences are visually better. In fact on figure 5, the corrections due to T-VIF, Fast-T-VIF or REF are conspicuous. It is also visible that the temporal versions of VIF suppress as much noise as the REF while keeping a better dynamic for gray levels (in the windows for example).

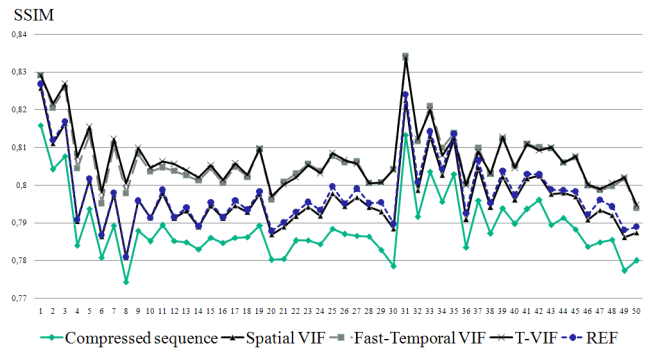


Fig. 4. SSIM results for Life.

5.3. Temporal Analysis

The main advantage of the T-VIF is its temporal aspect. The idea is that the correction will gain in smoothness on a temporal level. To demonstrate its effect, we calculated PSNR and SSIM on the difference between two successive images. Thus we study the effects of the different filters to the variation through time.

For the three tested videos, the variation from one frame to another in the T-VIF and Fast-T-VIF are better. For Life and Aspen this enhancement is significant, while for InToTree it is too slight to be meaningful. The PSNR values for the Aspen sequence are shown in figure 6.

6. CONCLUSION

On a spatial level, our Fast-T-VIF produces a significant enhancement for one of the test videos, a slight one for another and none for the third in terms of fidelity metric. On the temporal point of view the results indicate a moderate improvement in the sequence continuity. The tests should be extended to a wider video database for more significant tendencies. But the main issue while assessing the efficiency of mosquito noise correctors is how to evaluate the quality of the correction. Indeed, the precise localization and small amplitude of this noise both tend to level the usual global metrics and make its temporal aspect difficult to deal with. The next step for mosquito correction is thus the making of a dedicated metric, which exactly what we are currently working.

7. REFERENCES

- [1] J. Anderson and B. Anderson, "The myth of persistence of vision revisited," *Journal of Film and Video*, vol. 45, no. 1, pp. 3–12, 1993.
- [2] Recommendation P.930 ITU-T, "Principles of a reference impairment system for video," Tech. Rep., August 1996.



(a) Compressed video



(b) REF corrector



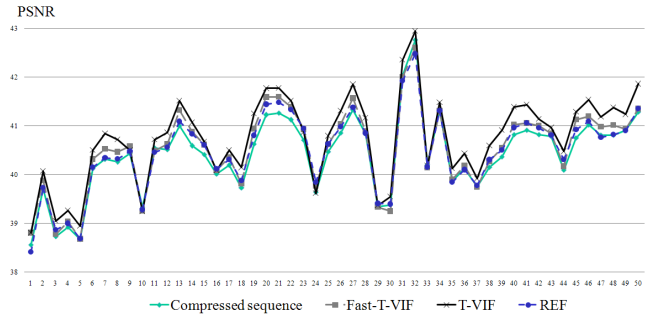
(c) T-VIF corrector



(d) Fast-T-VIF corrector

Fig. 5. Image 19 of Aspen.

- [3] C.A. Segall and A.K. Katsaggelos, "Pre- and post-processing algorithms for compressed video enhancement," *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference*

**Fig. 6.** PSNR results on variation video for Aspen.

on, vol. 2, pp. 1369–1373, 2000.

- [4] F.X. Coudoux, M.G. Gzalet, and P. Corlay, "A post-processor for reducing temporal busyness in low-bit-rate video applications," *Signal Processing: Image Communication*, vol. 18, pp. 455–463, July 2003.
- [5] S.J.P. Westen, R.L. Lagendijk, and J. Biemond, "Adaptive spatial noise shaping for dct based image compression," *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 4, pp. 2124–2127, 7–10 May 1996.
- [6] S. Del Corso, C. Miro, and J. Jung, "Mnr: a novel approach to correct mpeg temporal distortions," *Consumer Electronics, IEEE Transactions on*, vol. 49, no. 1, pp. 229–236, Feb. 2003.
- [7] H. Abbas and L.J. Karam, "Suppression of mosquito noise by recursive epsilon-filters," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, pp. I-773–I-776, 15–20 April 2007.
- [8] C. P. Fenimore, J. M. Libert, and P. Roitman, "Mosquito noise in mpeg-compressed video: test patterns and metrics," *Human Vision and Electronic Imaging V*, vol. 3959, no. 1, pp. 604–612, 2000.
- [9] F. Cr  t  -Roffet, *Estimer, mesurer et corriger les artefacts de compression pour la t  l  vision num  rique*, Ph.D. thesis, GIPSA-Lab - Universit   Joseph Fourier, 2007.
- [10] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer Vision and Image Understanding*, vol. 63, no. 1, pp. 75–104, January 1996.