Signal Processing for Multimedia
J.S. Byrnes (Ed.)
IOS Press, 1999

Restoration and Storage of Film and Video Archive Material

P.M.B. van Roosmalen and J. Biemond and R.L. Lagendijk

Information and Communication Theory Group
Faculty of Information Technology and Systems
Delft University of Technology
The Netherlands

ABSTRACT. Many unique records of archived motion pictures are in a fragile state and contain many artifacts. Preservation of these pictures can be achieved by copying them onto new digital media in compressed format, using, for instance, the MPEG standard. Restoration of the old film sequences prior to renewed storage is often beneficial both in terms of visual quality and in terms of coding efficiency. Restoring old image sequences manually is a tedious and costly process and therefore the use of an automated system for image restoration capable of dealing with common and less common artifacts is desirable. This chapter presents the AURORA project, whose goal it is to create such an automated system. Three algorithms developed within the AURORA project, for dealing with noise, blotches and intensity flicker, are described here. The success of these algorithms is demonstrated using both common measures of performance and using a new objective measure based on coding efficiency.

1. Introduction

AURORA[1] is an acronym for *AU*tomated *R*estoration of *OR*iginal video and film *A*rchives. AURORA consortium members that are actively involved in research include the BBC, Snell & Wilcox, and Cambridge University in the U.K., Tampere University in Finland, INA in France, and the Delft University of Technology in the Netherlands.

The goal of AURORA is to create new algorithms and real-time hardware for the restoration of old video and film sequences. Many unique records of historic, artistic, and cultural developments of every aspect of the 20th century are stored in huge stocks of archived moving pictures and many of these historically significant items are in a fragile state. Preservation of these pictures can be achieved by copying them onto new digital media in compressed format, using, for instance, the MPEG standard. There are a number of reasons why old motion pictures should be restored before they are stored on digital media. First, restoration improves the visual quality of the film sequence and thereby the commercial value increases. Second, restoration generally speaking leads to more efficient compression, *i.e.*, to higher quality at identical bit rates and conversely to lower bit rates at identical quality. The latter is especially important in digital broadcasting and storage environments where the price of broadcasting/storage is directly related to the number of bits being broadcast/stored.

The first part of the AURORA acronym, *AU*tomated, should be stressed because of the absolutely huge amounts of old film and video that need to be processed and because of economical constraints. Automated, real-time image restoration systems allow for bulk processing of data and reduce the high cost of labor required for manual restoration. In order to make bulk processing possible, the AURORA system takes digital image sequences as its starting point instead of physical reels of film. Rather than having a system capable of

[1] This work was funded by the European Union under contract AC 072.

handling a large number of film formats and systems that have been used in one period or another over the last century, it is assumed that these films have been digitized by skilled engineers who know best how to digitize the various types of film. Digital image sequences are obtained by digitizing the output of the film-to-video *telecine*. It must be kept in mind that the earlier telecines have their limitations in terms of noise characteristics and resolution. Sometimes a copy on video obtained from an earlier telecine is all that remains of a film.

Areas of interest within AURORA are typical artifacts in old film sequences such as noise, line-scratches, blotches (dirt and sparkle), film unsteadiness, line jitter and intensity flicker. In the following sections we focus on restoration techniques for a selection of these artifacts. In Section 2 we discuss the general principles of noise reduction and we present a new algorithm for this purpose. We devote Section 3 to the detection (and correction) of blotches. In Section 4 we present a novel and very successful approach to correct intensity flicker. In section 5 we investigate the improvement in quality of the restored sequences using a new objective quality measure. Here it is also verified that image restoration and coding efficiency are closely related. We conclude this chapter with a discussion in Section 6.

2. Noise reduction

Noise is a common problem in old film and video sequences and many methods for noise reduction can be found in the literature [15], [16], [22], [2], [7], [10]. In this section we consider image sequences $I(x, y, t)$ corrupted by noise $\eta(x, y, t)$ that is signal independent, white and additive (x, y indicate discrete spatial co-ordinates, t indicates the frame number). The observed images $Y(x, y, t)$ are thus given by:

$$Y(x, y, t) = I(x, y, t) + \eta(x, y, t). \tag{1}$$

For many practical purposes (1) is an adequate model. Using more accurate signal-dependent models (*e.g.,* for taking into account the gamma correction applied by television cameras that makes the noise signal dependent and the signal dependent film grain noise [7], [3]) often leads to little extra gain compared to the added complexity.

The goal of noise reduction is to obtain an estimate of the original signal $\hat{I}(x, y, t)$ given the noisy observed signal $Y(x, y, t)$. A class of very successful nonlinear filters is based on scale-space representations of the data to be filtered [22], [10]. Having representations of the data with various levels of detail allows local and global signal characteristics to be preserved better than purely spatial methods, *e.g.,* Wiener filters. Noise reduction is achieved by *thresholding* or *coring* the transform coefficients computed at the various scales. Taking the inverse transform of the adjusted coefficients gives the noise reduced result.

In the following sections we present a *scale-space* based noise reduction filter which tries to get an optimal separation between signal and noise by including both directional and temporal information into the transformation. In Section 2.1 we motivate the approach we take to obtaining a scale-space representation of the image data. After presenting details of this approach in Section 2.2, we extend the decomposition to 3D using wavelets in Section 2.3. We introduce the actual noise reduction operation in Section 2.4. In Section 2.5 we present some experiments and results. We conclude the topic of noise reduction with a brief discussion in Section 2.6.

2.1. *Methods for obtaining scale-space representations of data*

The DWT is a popular tool for obtaining scale-space representations of data. A problem with this transform, however, is that, due to the aliasing caused by the critical subsampling, a spatial shift of the input image may lead to totally different distribution of the signal energy over the transform coefficients. Therefore, shifting the input image can lead to significantly different filtering results. The DWT is not shift invariant. Shift invariance can be obtained by applying nondecimated DWT using bi-orthogonal wavelets [10]. However, this leads to a massive increase in the number of transform coefficients.

Instead of using the DWT, we follow the approach of Simoncelli et al. [20], who proposed a pyramid decomposition that is shift invariant. The shift invariance is accomplished avoiding aliasing effects by ensuring that no components with frequencies larger than $\pi/2$ are present before 2:1 subsampling. Furthermore, the Simoncelli pyramid has the advantage that it is based on directionally sensitive filters. This means that the distribution of signal energy over frequency bands depends on the orientation of structures within the image. This gives improved filtering results for diagonal components in the image compared to a separable DWT. Also, in the case that, say, 4 orientations are used, the noise energy is distributed over 4 orientations, whereas the energy of image structures such as straight lines is distributed over 1 or 2 orientations. This leads to better local separation between noise and signal.

The Simoncelli decomposition is significantly overcomplete, the number of transform coefficients is much larger than the number of pixels in the original image. For example, a five-level pyramid decomposition (four sets of high-pass coefficients and one set of low-pass coefficients) with 4 orientations of an $N \times N$ image results in about $9.3N^2$ coefficients. The undecimated DWT gives $10N^2$ for a four-level decomposition.

2.2. *The Simoncelli pyramid*

Figure 1 shows the 2D Simoncelli pyramid decomposition and reconstruction scheme. The filters L_i, H_i and F_j are the 2D low-pass, high-pass and directional (fan) filters, respectively. Figure 2 represents a decomposition in the frequency domain. The filters $L_0(\omega)$, $H_0(\omega)$ and $H_1(\omega)$ ideally are linear phase, self inverting and satisfy the following constraints: the aliasing in the low-frequency (subsampled) bands is minimized (eq. (2)), the overall system has unity response (eq. (3)), and, all radial bands have a bandwidth of one octave (eq. (4)):

$$L_1(\omega) \to 0 \qquad \text{for} \quad \omega > \pi/2 \tag{2}$$

$$|L_i(\omega)|^2 + |H_i(\omega)|^2 = 1 \tag{3}$$

$$L_0(\omega) = L_1(2\omega). \tag{4}$$

The 2D filters can be obtained from 1D linear phase FIR filters using the McClellan transform [12]. Using (4), the two-dimensional filter $L_0(\omega)$ can be obtained from $L_1(\omega)$. We used a conjugate gradient algorithm to find the filters $H_0(\omega)$ and $H_1(\omega)$ under the constraints set by (3) [17].

For practical purposes the high-pass filters $H_0(\omega)$ and $H_1(\omega)$ are directly combined with the fan filters $F_1(\omega)$ $F_2(\omega)$, $F_3(\omega)$, and $F_4(\omega)$. This can be done by multiplying the Fourier transform of the filters $H_0(\omega)$ and $H_1(\omega)$ with the angular term $f(\theta - \theta_0)$ in (5), where θ_m is the center of orientation of the filter. The inverse Fourier transform of the product gives the required 2D filter coefficients.

$$f(\theta - \theta_m) = \begin{cases} 1 & \text{if} \quad |\theta - \theta_m| < \pi/16 \\ cos(|\theta - \theta_m|) & \text{if} \quad \pi/16 \le |\theta - \theta_m| < 3\pi/16 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

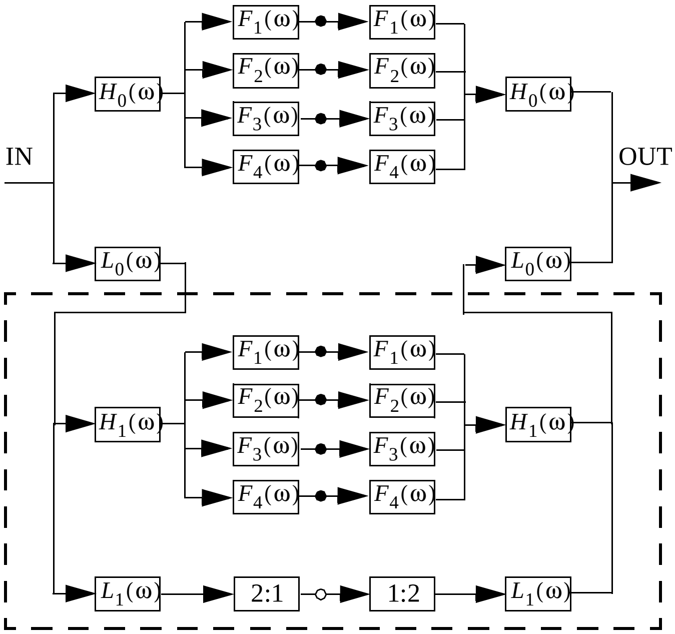


FIGURE 1. The Simoncelli analysis/synthesis filter bank. The total decomposition is obtained by recursively inserting the contents of the dashed box into the white spot.

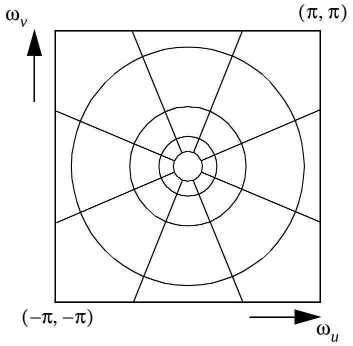


FIGURE 2. The pyramid decomposition in the frequency domain.

2.3. *An extension to 3D using wavelets*

The 2D decorrelating pyramid transform separates signal and noise. We now include motion compensated temporal information to improve on this separation. If the signals are stationary in a temporal direction, the motion compensated frames from $t - n, ..., t + m$ should all be identical to frame t , except for the noise term. The pyramid decomposition of these images should also be identical, except for the noise term. If we view a set of coefficients at corresponding scale-space locations in a temporal sense, we have a 1D DC signal plus noise. These can be separated into a low-pass and a high-pass signal; we propose applying the DWT to this effect.

Note that ideally speaking one would use a long filter to obtain good separation between signal and noise in the spatio-temporal decomposition step. The inaccuracies of the motion estimator and the fact that areas become occluded or uncovered form a limiting factor to the length of the wavelet used. The inaccuracies of the motion estimator are also the reason why we do not apply straightforward averaging in the temporal direction to obtain the low-pass signal: we want to avoid blur.

The 3D spatial-temporal pyramid decomposition is given by the following steps:

1a Calculate the motion compensated images for images $t - n, ..., t + m$ (n, m depend on the size of wavelet),

2a Calculate the Simoncelli pyramid decomposition for each (motion compensated) image (Fig. 1),

3a Apply the DWT in the temporal direction to each set of coefficients at corresponding scale-space locations, with the coefficient belonging to frame t being the center coefficient.

The reconstruction phase is as follows:

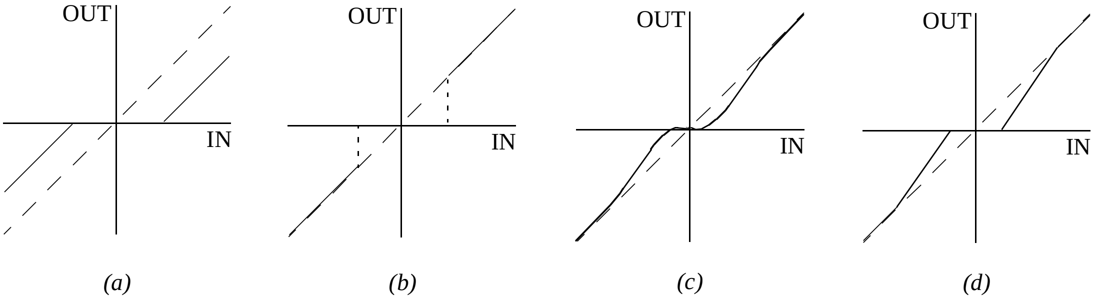


FIGURE 3. Coring functions. (a) Soft thresholding. (b) Hard thresholding. (c) Bayesian thresholding. (d) Piecewise linear approximation to Bayesian thresholding.

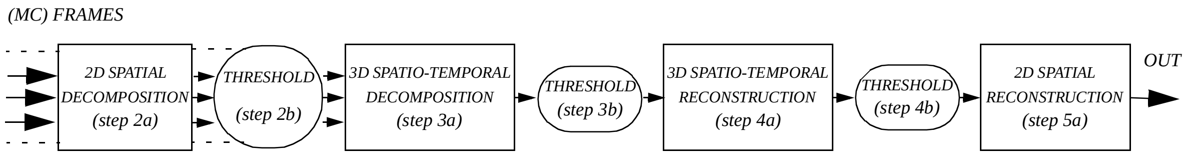

(MC) FRAMES
2D SPATIAL DECOMPOSITION (step 2a)
THRESHOLD (step 2b)
3D SPATIO-TEMPORAL DECOMPOSITION (step 3a)
THRESHOLD (step 3b)
3D SPATIO-TEMPORAL RECONSTRUCTION (step 4a)
THRESHOLD (step 4b)
2D SPATIAL RECONSTRUCTION (step 5a)
OUT

FIGURE 4. Schematic representation of the 3D pyramid noise reduction system

4a Apply the inverse DWT to each set of wavelet coefficients, this leaves us with the spatially decomposed frame at t,
5a Apply the synthesis stage of the Simoncelli scheme (Fig. 1).

2.4. *Noise reduction by coring*

Coring or *thresholding* [10], [19] the transform domain coefficients by either soft or hard thresholding are popular filtering operations. Alternatively, coring functions which are optimal in a mean-squared-error sense can be computed or approximated using a Bayesian framework. Figure 3 shows these coring functions. Soft thresholding leads to a slight loss in contrast of the reconstructed image. Hard thresholding introduces disturbing ringing artifacts near edges.

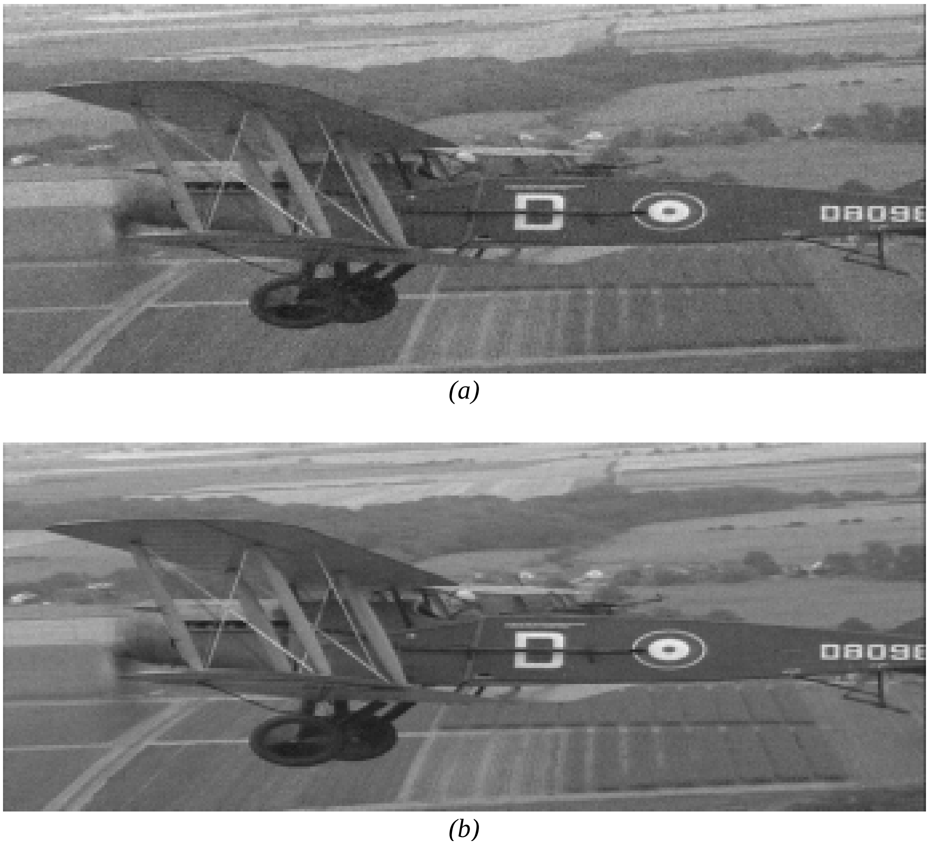
The structure of the proposed decomposition/reconstruction algorithm offers several possibilities for applying coring by introducing the following steps (see Fig. 4):

2b For all spatially decomposed (motion compensated) frames, threshold all the spatial transform coefficients (except for those in the DC band),
3b Threshold all the high-pass spatio-temporal coefficients,
4b Threshold the coefficients in all spatial frequency bands of the reconstructed frame t (except for those in the DC band).

Note that the spatio-temporal decomposition makes temporal filtering of the DC band from the Simoncelli pyramid possible without introducing visible blur or other artifacts. 2D scale-space noise reduction filters have no way of filtering the DC bands.

2.5. *Experiments and results*

For our experiments we used a three-level pyramid with four orientations. To significantly reduce the computational load we omitted the filters $L_0(\omega)$ and $H_0(\omega)$, *i.e.*, we omitted everything outside the dashed box in Fig. 1. This yields $5.3N^2$ transform coefficients for an $N \times N$ image. As a result, the highest frequency bands cover a larger relative bandwidth than the lower frequency bands. $L_1(\omega)$ was designed to have an attenuation of 40 dB at $\pi/2$. A 6-tap bi-orthogonal Daubechies wavelet [1] was used for the temporal extension. For the


(a)

(b)

FIGURE 5. (a) Noisy field from Plane sequence. (b) Corrected field.

sake of simplicity we applied soft-thresholding with values 1.0, 10.0 and 1.0 in steps 2b, 3b and 4b respectively (see Section 2.4).

The test sequence (100 frames) of a plane flying over a landscape was processed. It contains fine detail, sharp edges, uniform regions and significant motion. Eight levels of white Gaussian noise were added to this scene, leading to an average PSNR of 25.2 dB. The motion vectors from one field to the next were computed using a hierarchical block-matcher with additional constraints on the smoothness of the motion vectors. Motion vectors to and from fields further apart were obtained by vector tracing (adding the motion vectors of consecutive fields).

Figure 5 shows a noisy field and a noise reduced field. Clearly, much noise has been reduced whilst edges have not been blurred. However, we do note a slight reduction in contrast in some lightly textured regions. A considerable increase in PSNR of 5.2 dB results from this new filter.

2.6. *Discussion*

Even though the proposed filter gives very good results, it is not a practical filter due to its high complexity. The low-pass and high-pass filters used consisted of 23×23 taps and therefore, to ensure a reasonable processing time, the convolutions have to be performed in the Fourier domain. Clearly, this filter is too expensive to be built in *real-time* hardware. However, its function is more to serve as a benchmark to see what amount of noise reduction is attainable and to see how close the noise filters that are being implemented in hardware for AURORA come to that optimum.

3. Blotch detection

Blotches present a common type of artifact in old film sequences that manifests itself as disturbing bright or dark spots caused by dirt and by the loss of the gelatin covering the film due to aging effects and bad film quality. Characteristics of blotches are that they seldom appear at the same spatial location in consecutive frames, they tend to be smooth (little texture), and they usually have intensity values that are very different from the original contents they cover (see, *e.g.*, Fig. 9a). Films corrupted by blotches are often restored using a two-step approach. In the first step blotches are detected and detection masks are generated that indicate for each pixel whether or not it is part of a blotch. In the second step, corrupted pixels are corrected by means of spatio-temporal interpolation [8], [9], [6], [14].

Blotch detectors are either object based or pixel based. Pixel based detectors determine for each pixel whether or not it is part of a blotch independently from whether or not its neighboring pixels are considered to be part of a blotch. Object based detectors exploit the spatial coherence within blotches via, *e.g.*, Markov random fields. So far, pixel based detectors have shown to achieve similar detection results as object based detectors at a fraction of the computational cost [8].

We present three post-processing operations that can be applied on the candidate blotches output by a blotch detector to improve the quality of the detection masks. The key is that we use a pixel based detector and that in the post-processing we consider blotches as objects. This allows us to exploit the spatial coherency within blotches while maintaining low complexity and low computational effort. The first post-processing operation detects and removes possible false alarm by taking into account the probability that the detector wrongly detects a blotch (an object) of a certain size due to noise. The second post-processing operation finds missing pieces of blotches that would otherwise be only partially detected by applying a technique called hysteresis thresholding [4]. The final post processing operation consists of a constrained dilation operator that fills small *holes* in and on edges of the candidate blotches.

Section 3.1 describes the blotch detector we will be using. Section 3.2 describes the post-processing operations. Section 3.3 describes the results and concludes this topic.

3.1. *The simplified ranked ordered difference (S-ROD) detector*

Blotches are characterized by the fact that they seldom appear at the same location in a pair of consecutive frames and that they have intensity values different from the original image contents. Therefore, blotches can be detected by detecting temporal discontinuities in image intensity. The additional use of motion compensation significantly reduces the number of false alarms. The ROD detector [14] is based on these principles. We present a *simplified* version of ROD, which we call S-ROD.

Let $I_n(z)$ denote the intensity of a pixel at a spatial location $z^T = (x, y)$ in frame n. Let $p_{n,i}(z)$ form a set of six reference pixels, ordered by magnitude, obtained from spatially co-sited pixels and their vertical neighbors in motion compensated previous and next frames (see Fig. 6). The output of S-ROD is then defined by:

$$(6) \qquad d_n(z) = \begin{cases} min(p_{n,i}(z)) - I_n(z) & \text{if} \quad min(p_{n,i}(z)) - I_n(z) > 0 \\ I_n(z) - max(p_{n,i}(z)) & \text{if} \quad I_n(z) - max(p_{n,i}(z)) > 0, \\ 0 & \text{otherwise} \end{cases}$$

and a blotch is detected when:

$$(7) \qquad d_n(z) > T_1 \quad \text{with} \quad T_1 \geq 0.$$

What the S-ROD basically does compute the range of the reference intensities from motion compensated frames and it compares the pixel intensity under investigation to this range

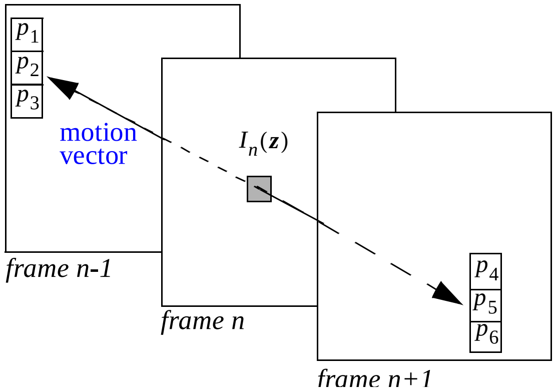


FIGURE 6. Selection of reference pixels $p_{n,i}(z)$ from previous and next frames using motion compensation

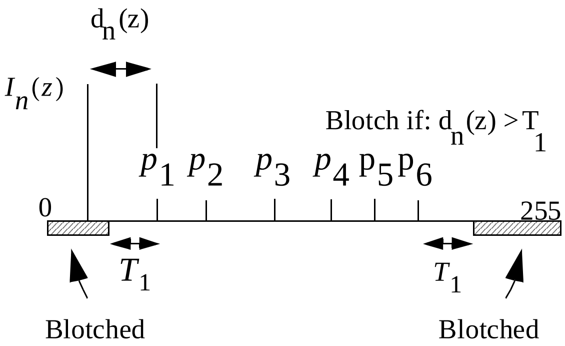

$d_n(z)$
$I_n(z)$
Blotch if: $d_n(z) > T_1$
p_1 p_2 p_3 p_4 p_5 p_6
0
255
T_1
T_1
Blotched
Blotched

FIGURE 7. Schematic overview of the functioning of the Simplified ROD detector. (The reference pixels p_i have been ordered by their magnitude)

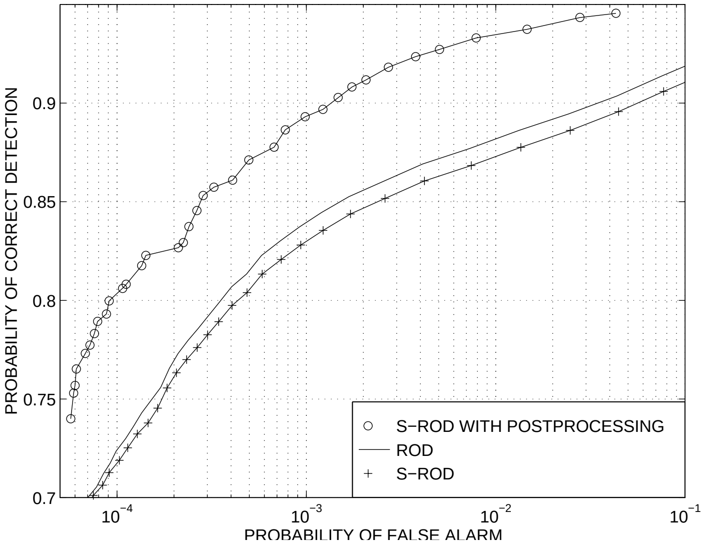


FIGURE 8. ROC-curves resulting from ROD, S-ROD, and S-ROD with post-processing applied to the test sequence

(Fig. 7). A blotch is detected if the intensity of the current pixel is not included in that range and lies far enough outside that range. What is considered *"far enough"* is determined by T_1. If T_1 is small many blotches will be detected correctly but many false alarms will occur. As T_1 becomes larger, fewer blotches are detected and the number of false alarms drops.

By means of a *receiver operator characteristic* (ROC) curve, which depicts the probability of false alarm vs. the probability of correct detection, the numerical performance of a detector can be visualized graphically. Figure 8 shows the ROC curves obtained from a test sequence using ROD, S-ROD and S-ROD with the post-processing proposed in the next section. The test sequence, which was also used in [8], [14], is the *Western* sequence (64 frames) to which artificial blotches have been added. Each artificial blotch had a fixed gray value which was drawn uniformly between 0 and 255. We observe that the performance of S-ROD is slightly below that of ROD. We also see that much is gained after applying the post-processing operations we describe next.

3.2. *Improving the detection results by post-processing*

We propose a number of post-processing operations in which the goal is to maximize the ratio of correct detections to false alarms. The key to the post-processing operations is that the candidate blotches are viewed not as individual pixels but as objects. Section 3.2.1 defines what we consider to be an object. Section 3.2.2 presents the first post-processing technique that detects and removes possible false alarms due to noise given a specific detector. Often blotches are detected only partially. Section 3.2.3 presents our second post-processing technique that finds more complete blotches in those cases. Section 3.2.4 presents the third post-processing technique, which is a constrained dilation technique that includes small holes in the candidate blotches that are missed by the detector.

3.2.1. *Object definition.* We want to manipulate candidate blotches as objects rather than as individual pixels. Because we are particularly interested in blotches it is reasonable to use characteristics of blotches in the object definition. The characteristics we use are that blotches are spatially coherent and that they tend to be smooth, *i.e.*, that adjacent pixels have similar intensities. We consider a pair of pixels to be similar if their difference is smaller than twice the standard deviation of the noise. Other characteristics of blotches are taken into account implicitly due to the fact that we are only interested in pixels flagged by the blotch detector.

Therefore, adjacent pixels that have similar intensities and that are flagged by the blotch detector are considered to be part of the same candidate blotch. To differentiate between the various candidate blotches a unique label is assigned to each candidate blotch (and to each pixel that is part of that blotch).

3.2.2. *Removing false alarms due to noise.* High correct detection rates are achieved by setting the blotch detector to a high degree of sensitivity. However, the detector is then not only sensitive to blotches but also to noise and many false alarms result. This is undesirable because it increases the probability of introducing visible errors during the correction stage. To reduce the influence of noise, we propose computing the probability that the detector gives a specific response due to noise under the assumption that no blotches are present. This allows us to compute the probability that a blotch of given size is wrongly detected. If that probability exceeds a certain risk R, all candidate blotches of that size and with the same corresponding detector response are removed from the detection mask.

We demonstrate this approach for the S-ROD detector. After labelling the candidate blotches we compute the size N and the mean value of the detector output $d_n(z)$ for each blotch. We assume that in the absence of noise $I_n(z) = p_{n,i}(z)$ for at least one i, *i.e.*, that no false alarms occur in the absence of noise. We also assume that the noise is *independent and identically distributed* (i.i.d.). The probability that S-ROD, with $T_1 \geq 0$, generates a single false alarm due to noise is then given by:

$$\begin{aligned} & P[d_n(z) > T_1] \\ & = P[I_n(z) - max(p_{n,i}(z)) > T_1, \quad I_n(z) - max(p_{n,i}(Z)) > 0] + \\ & \quad P[min(p_{n,i}(z)) - I_n(z) > T_1, \quad min(p_{n,i}(z)) - 0 I_n(z) > 0] \\ & = P[I_n(z) - max(p_{n,i}(z)) > T_1] + P[min(p_{n,i}(z)) - I_n(z) > T_1] \\ & = P[\text{all} \quad I_n(z) - p_{n,i}(z) > T_1] + P[\text{all} \quad p_{n,i}(z) - I_n(z) > T_1] \\ & = P^6[I_n(z) - p_{n,i}(z) > T_1] + P^6[p_{n,i}(z) - I_n(z) > T_1]. \end{aligned} \tag{8}$$

Because we operate in the digital domain, it is easy to compute the probability mass function $P[d_n(z) = X]$ once (8) has been determined, *i.e.*, the probability that S-ROD gives a specific response X for a single pixel due to noise.

After the labeling procedure, a candidate blotch is an object with spatial support S that consists of N pixels, and the mean value of the output of the blotch detector equals $\overline{d_n(z)}$ for that object. Let H_0 denote the hypothesis that this object is purely the result of false alarms

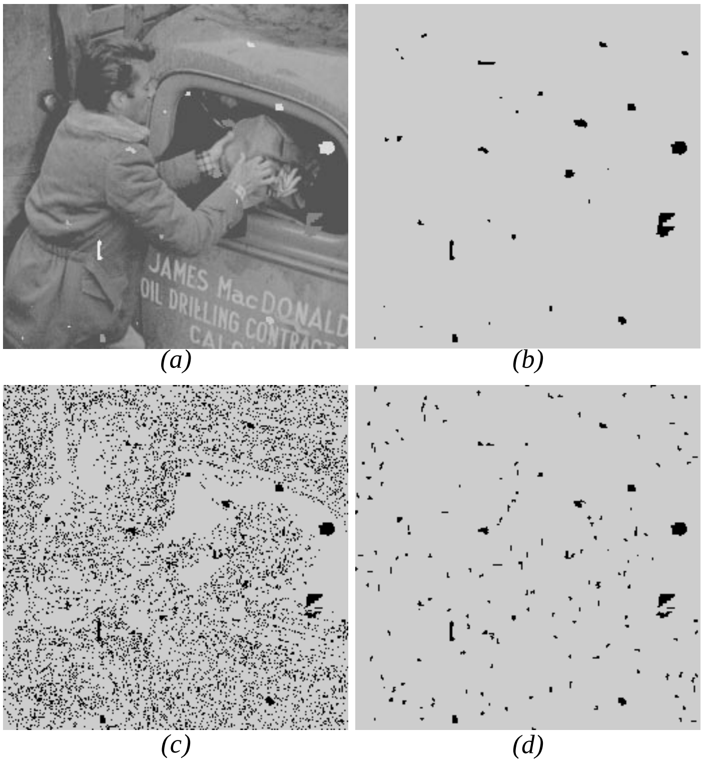


FIGURE 9. (a) Blotched frame from test sequence. (b) Mask of artificial blotches. (c) Initial detection mask using S-ROD with $T_1 = 0$. (d) Detection mask after removing possible false alarms due to noise.

$d_n(\mathbf{z})$	probability of false alarm	Remove blotches <
1	0.091921	5 pixels
2	0.060310	5 pixels
3	0.036622	4 pixels
4	0.020488	3 pixels
5	0.010353	3 pixels
6	0.004854	3 pixels
7	0.002095	2 pixels
8	0.000820	2 pixels
9	0.000301	2 pixels
10	0.000105	2 pixels
11	0.000028	2 pixels

Table 1. See text for explanation.

due to noise. $P[H_0]$ is then the probability that a collection of N individual pixels are flagged by the S-ROD independently of their location and of their neighbors. By approximation:

(9)
$$\begin{aligned} P[H_0] &= P[\tfrac{1}{N}\textstyle\sum_{z\in S} d_n(z) = \overline{d_n(z)}, size = N|\text{no blotch present}] \\ &\approx P^N[d_n(z) = \overline{d_n(z)}] \end{aligned}$$

We now remove those candidate blotches for which the probability that they are solely the result of noise exceeds a certain risk R:

(10)
$$P[H_0] > R.$$

The result of this approach is illustrated in Figure 9, which shows frame 8 of the *Western* sequence together with the artificial blotch mask and the detection masks before and after post-processing. The initial detection mask was obtained using S-ROD where we chose $T_1 = 0$. The noise was assumed to be i.i.d. Gaussian and the noise variance was estimated to be 9 using the method described in [11]. Table 1 shows the probability of a false alarm due to noise and the sizes below which blotches are removed when the average detector response value for that blotch equals $d_n(z)$. These sizes were computed by setting the risk to $R = 10^{-5}$.

In frame 8 84.8% of the blotches were detected correctly and 13.1% of the uncorrupted pixels were mistakenly flagged as being part of a blotch before post-processing. After post-processing 83.4% of the blotches were detected correctly and only 1.0% of the clean pixels were mistakenly flagged as being part of a blotch. Clearly this is a great improvement.

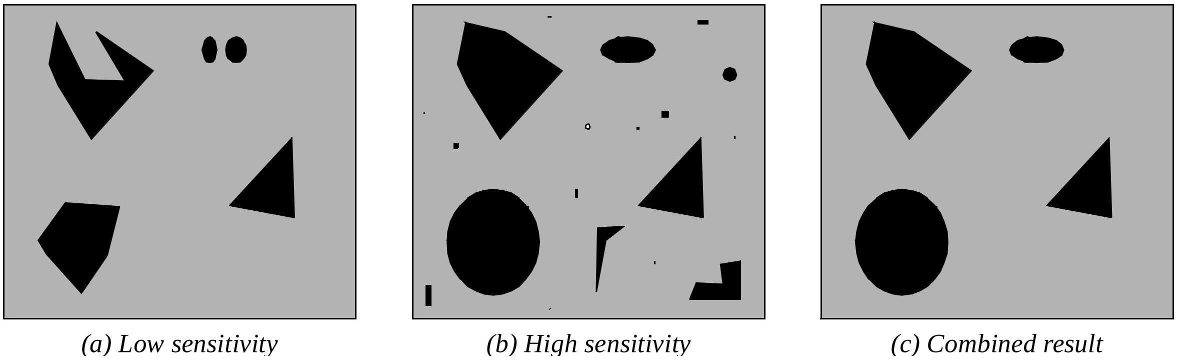

(a) Low sensitivity (b) High sensitivity (c) Combined result

FIGURE 10. Schematic overview of hysteresis thresholding. (a) Detection mask from detector set to low sensitivity with partially detected blotches. (b) Detection mask from detector set to high sensitivity with many false alarms. (c) Result after validation (propagation): the partially detected blotches are completed

3.2.3. *Completing partially detected blotches.* We note that when the detector is set to low detection rates, many blotches are not detected at all and other blotches are detected only partially. We now want to make those blotches that are detected only partially more complete. We achieve this by noting from Fig. 8 that as T_1 is lowered the probability of false alarms decreases faster than the probability of correct detections. This means that detections resulting from a blotch detector set to a low detection rate are more likely to be correct and can thus be used to validate the detections from that detector when set to a high detection rate.

This can be implemented by applying hysteresis thresholding [4] (see Fig. 10). The first stage computes and labels the set of candidate blotches using the operator settings of the blotch detector (in the case of S-ROD this is the operator setting of T_1). Possible false alarms due to noise are removed as described before. The second stage sets the blotch detector to a very high detection rate (*i.e.*, $T_1 = 0$ for S-ROD) and again a set of candidate blotches is computed and labeled. Candidate blotches from the second set can now be validated; they are preserved if corresponding candidate blotches in the first set exist. The other candidate blotches in the second set, which are more likely to have resulted from false alarms, are discarded. Effectively we have preserved the candidate blotches detected using the operator settings and we have made them more complete.

3.2.4. *Constrained dilation for missing details.* Even though a blotch detector may be very refined, there is always a probability that it fails to detect elements of a blotch. This is illustrated by Fig. 9 where it can be seen that, even though the S-ROD detector has been set to its most sensitive setting, not all the blotches have been detected completely. In this final post-processing step we refine the candidate blotches by removing small holes in and on the edges of the candidate blotches.

We propose using a constrained dilation operation for filling in the holes. It applies the following rule: if a pixel's neighbor is flagged as being blotched and its intensity difference with that neighbor is small (*e.g.*, less than twice the standard deviation of the noise) then that pixel should also be flagged as being part of that blotch. Because of the constraint on the differences in intensity, the probability that uncorrupted pixels surrounding a blotch will mistakenly become flagged as ”blotched” is reduced because blotches tend to have gray values that are significantly different from their surroundings.

It is important not to apply too many iterations of this constrained dilation operation because it is always possible that the contrast between a candidate blotch and its surroundings is low. The result would be that the candidate blotch would grow completely out of its bounds and many false alarms would occur. In practice, we found that applying two iterations leads to good results.

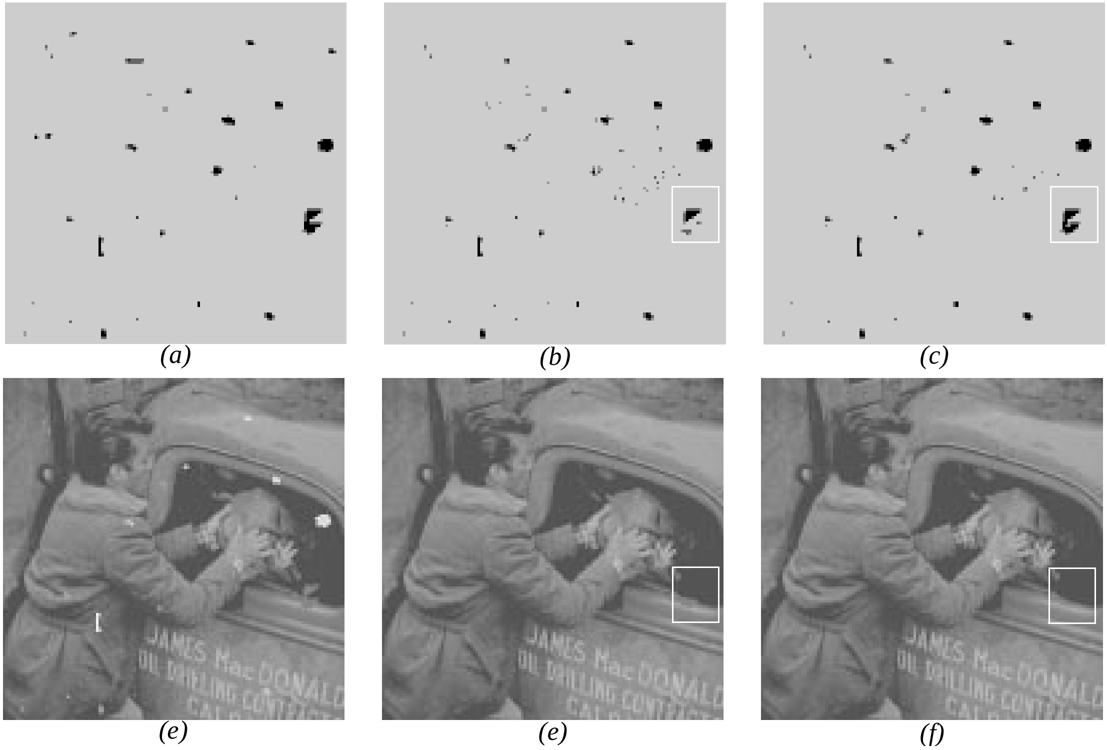


FIGURE 11. (a) Mask of artificial blotches. (b) Blotches detected using ROD (c) Blotches detected using S-ROD with post-processing. (d) Blotched frame. (e) corrected frame after ROD. (f) Corrected frame after S-ROD with post-processing. Note the differences in the boxed regions.

3.3. *Results and conclusions*

We already observed in Figure 8 that S-ROD in combination with post-processing gives a significant improvement over plain ROD. For example, the number of false alarms resulting from S-ROD with post-processing is a factor 6 lower than that which results from ROD at a correct detection rate of 85%.

Figure 11 shows the detection masks and corrected versions of frame 8 of the test sequence that result from ROD and S-ROD with post-processing. The ml3dex interpolation method as described in [9], a three-dimensional multi-stage median filter, was used for interpolating the blotched data. ROD and S-ROD with post-processing were set to an overall false alarm rate of 0.001 for the whole test sequence. Note the differences in the number of small blotches and the differences in detection and correction results in the boxed regions.

In conclusion, the methodology described here can also be applied to other blotch detectors. Alternatively, the rules and constraints posed by the post-processing could well be defined implicitly in a new detector, *e.g.* based on Markov random fields. This would, however, significantly increase the complexity and computational effort.

4. Correction of intensity flicker

A common artifact in old black-and-white film sequences is intensity flicker. We define intensity flicker as unnatural temporal fluctuations in perceived image intensity that do not originate from the original scene. Intensity flicker has a great number of causes, *e.g.*, aging of film, dust, chemical processing, copying, aliasing, and, in the case of earlier film cameras, variations in shutter time. Neither equalizing the intensity histograms nor equalizing the mean frame values of consecutive frames, as suggested in [13], [18] give general solutions to the problem. These methods do not take changes in scene contents into account, and they do not appreciate the fact that intensity flicker can be a spatially localized effect. We propose equalizing local intensity means and variances in a temporal sense to reduce the undesirable temporal fluctuations in image intensities.

In Section 4.1 we model the effects of intensity flicker, and we derive a solution to this problem for stationary sequences. Here we also define a measure of reliability for the model parameters. In section 4.2 we extend the applicability of our method to include nonstationary sequences by incorporating motion. In the presence of intensity flicker, it is difficult to compensate for motion of local objects in order to satisfy the requirement of stationarity. We therefore describe a method for compensating global motion (camera pan) and a method for detecting the remaining local object motion. The model parameters are interpolated where local motion is detected. We show, in Section 4.3, the overall system of intensity-flicker correction and discuss some practical aspects. Experiments and results form the topics of Section 4.4. Finally, we conclude with a discussion in Section 4.5.

4.1. *Estimating and correcting intensity flicker in stationary sequences*

We develop a method for correcting intensity flicker that is robust to the wide range of causes of this artifact. First, in Section 4.1.1 we model the effects of intensity flicker. We find a solution to this problem that is optimal in a linear mean square error sense. In Section 4.1.2 we concentrate on how the model parameters can be estimated for stationary image sequences, and we define a measure of reliability of those estimated parameters.

4.1.1. *Estimating and correcting intensity flicker in stationary sequences.* It is not practical to find explicit physical models for each of the mechanisms mentioned that cause intensity flicker. Instead, our model of the effects of this phenomenon is based on the observation that it causes temporal fluctuations in local intensity mean and variance. Since noise is unavoidable in the various phases of digital image formation, we also include a noise term in our model:

$$Y(x,y,t) = \alpha(x,y,t) \cdot I(x,y,t) + \beta(x,y,t) + \eta(x,y,t) \tag{11}$$

Here x, y are discrete spatial coordinates and t indicates the frame number. $Y(x,y,t)$ and $I(x,y,t)$ indicate the observed and original image intensities. Note that by $I(x,y,t)$ we do not necessarily mean the original scene intensities, but a signal that, prior to the introduction of intensity flicker, may already have been distorted. The distortion could be due to signal-dependent additive granular noise that is characteristic of film [3], [16], for example. The multiplicative and additive intensity-flicker parameters are denoted by $\alpha(x,y,t)$ and $\beta(x,y,t)$. In the ideal case, when no intensity flicker is present, $\alpha(x,y,t) = 1$ and $\beta(x,y,t) = 0$ for all x, y, t. We assume that $\alpha(x,y,t)$ and $\beta(x,y,t)$ are spatially smooth functions.

The intensity-flicker-independent noise, denoted by $\eta(x,y,t)$, models noise that has been added to the signal after the introduction of intensity flicker. We assume that this noise term is uncorrelated with the original image intensities. We also assume that $\eta(x,y,t)$ is a zero-mean signal with known variance. Examples are quantization noise and thermal noise originating from electronic studio equipment (VCR, amplifiers, etc.).

To correct intensity flicker, we must estimate the original intensity for each pixel from the observed intensities. We propose using the following linear estimator for estimating $I(x,y,t)$:

$$\hat{I}(x,y,t) = a(x,y,t) \cdot Y(x,y,t) + b(x,y,t). \tag{12}$$

If we define the error between the original image intensity and the estimated original image intensity as:

$$\epsilon(x,y,t) = I(x,y,t) - \hat{I}(x,y,t), \tag{13}$$

then we can easily determine that, given $\alpha(x,y,t)$ and $\beta(x,y,t)$, the optimal values for $a(x,y,t)$ and $b(x,y,t)$ in a *linear minimum mean square error* (LMMSE) sense are given by:

$$a(x,y,t) = \frac{var[Y(x,y,t)] - var[\eta(x,y,t)]}{var[Y(x,y,t)]} \cdot \frac{1}{\alpha(x,y,t)} \tag{14}$$

$$b(x,y,t) = -\frac{\beta(x,y,t)}{\alpha(x,y,t)} + \frac{var[\eta(x,y,t)]}{var[Y(x,y,t)]} \cdot \frac{E[Y(x,y,t)]}{\alpha(x,y,t)} \tag{15}$$

where $E[]$ stands for the expectation operator and $var[]$ indicates the variance. It is interesting to note that from equations (12), (14), and (15) it follows that, in the absence of noise, $a(x,y,t) = 1/\alpha(x,y,t)$, $b(x,y,t) = -\beta(x,y,t)/\alpha(x,y,t)$ and that $\hat{I}(x,y,t) = I(x,y,t)$. That is to say, the estimated intensities are exactly equal to the original intensities. In the extreme case that the observed signal variance equals the noise variance, we find that $a(x,y,t) = 0$ and $\hat{I}(x,y,t) = b(x,y,t) = E[I(x,y,t)]$; the estimated intensities equal the expected values of the original intensities.

4.1.2. *Estimating intensity-flicker parameters in stationary scenes.* In the previous section we derived a LMMSE solution to intensity flicker, assuming that the intensity-flicker parameters $\alpha(x,y,t)$ and $\beta(x,y,t)$ are known. This is not the case in most practical situations, and these parameters will have to be estimated from the observed data. In this section we determine how the intensity-flicker parameters can be estimated from stationary image sequences. We already assumed that $\alpha(x,y,t)$ and $\beta(x,y,t)$ are spatially smooth functions. For practical purposes we now also assume that the intensity-flicker parameters are constant locally:

$$\begin{cases} \alpha(x,y,t) = \alpha_{m,n}(t) \\ \beta(x,y,t) = \beta_{m,n}(t) \end{cases} \quad \text{for} x,y \in \Omega_{m,n}, \tag{16}$$

where $\Omega_{m,n}$ indicates a small image region. The image regions $\Omega_{m,n}$ can, in principle, have any shape, but they are rectangular blocks in practice, and m,n indicate their horizontal and vertical spatial locations. The $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ corresponding to $\Omega_{m,n}$ are considered as frame-dependent matrix entries at m,n. The size $M \times N$ of the matrix depends on the total number of blocks in the horizontal and vertical directions.

Keeping in mind the assumption that the zero-mean noise $\eta(x,y,t)$ is signal independent, we compute from (11) the expected value and variance of $Y(x,y,t)$ in a spatial sense for $x,y \in \Omega_{m,n}$:

$$E[Y(x,y,t)] = \alpha_{m,n}(t) \cdot E[I(x,y,t)] + \beta_{m,n}(t)m \tag{17}$$

$$var[Y(x,y,t)] = \alpha^2_{m,n}(t) \cdot var[I(x,y,t)] + var[\eta(x,y,t)]. \tag{18}$$

Rewriting (17) and (18) gives for $x,y \in \Omega_{m,n}$:

$$\beta_{m,n}(t) = E[Y(x,y,t)] - \alpha_{m,n}(t) \cdot E[I(x,y,t)], \tag{19}$$

$$\alpha_{m,n}(t) = \sqrt{\frac{var[Y(x,y,t)] - var[\eta(x,y,t)]}{var[I(x,y,t)]}}. \tag{20}$$

We now wish to solve (19) and (20) in a practical situation. The means and variances of $Y(x,yt)$ can be estimated directly from the regions $\Omega_{m,n}$ from the observed data. We assumed the noise variance to be known. What remains to be estimated are the expected values and variances of $I(x,y,t)$. For $x,y \in \Omega_{m,n}$, these estimates can be obtained by using the frame corrected previously as a reference:

$$E[I(x,y,t)] = E[\hat{I}(x,y,t-1)], \tag{21}$$

$$var[I(x,y,t)] = var[\hat{I}(x,y,t-1)]. \tag{22}$$

Thus, for $x,y \in \Omega_{m,n}$, the estimated intensity-flicker parameters are given by:

$$\hat{\beta}_{m,n}(t) = E[Y(x,y,t)] - \hat{\alpha}_{m,n}(t) \cdot E[\hat{I}(x,y,t-1)], \tag{23}$$

$$\hat{\alpha}_{m,n}(t) = \sqrt{\frac{var[Y(x,y,t)] - var[\eta(x,y,t)]}{var[\hat{I}(x,y,t-1)]}}. \tag{24}$$

We need a measure of reliability for $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$, to be able to avoid introducing significant errors in the corrected sequence. There are some cases in which the $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ are not very reliable. The first case is that of uniform image intensities. For any original image intensity in a uniform region, there are an infinite number of combinations of $\alpha_{m,n}(t)$ and $\beta_{m,n}(t)$ that lead to the observed intensity. Another case in which $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ are potentially unreliable is caused by the fact that (21) and (22) discard the noise in $I(x,y,t)$ originating from $\eta(x,y,t)$. Considerable errors result in regions $\Omega_{m,n}$ in which the signal variance is small compared to the noise variance (low signal-to-noise ratio). It is clear from these examples that the accuracy of the estimated parameters decreases with decreasing signal variances. Therefore we define the following measure of reliability, for $x,y \in \Omega_{m,n}$:

$$W_{m,n,t} = \begin{cases} 0 & \text{if} \quad var[Y(x,y,t)] < T_n \\ \sqrt{\frac{var[Y(x,y,t)] - T_n}{T_n}} & \text{otherwise} \end{cases} \tag{25}$$

where T_n is a threshold depending on the variance of $\eta(x,y,t)$. Large values for $W_{m,n,t}$ indicate reliable estimates, small values indicate unreliable estimates.

4.2. *Incorporating motion*

We have modeled the effects of intensity flicker, and we derived a solution for stationary sequences. Real sequences, of course, are seldom stationary. Measures will have to be taken to avoid estimates of $\alpha(x,y,t)$ and $\beta(x,y,t)$ that are incorrect due to motion. Compensating motion between $Y(x,y,t)$ and $\hat{I}(x,y,t-1)$ helps satisfy the assumption of stationarity. This requires motion estimation.

Robust methods for estimating global motion (camera pan) that are relatively insensitive to fluctuations in image intensities exist. Unfortunately, the presence of intensity flicker hampers the estimation of local motion (motion in small image regions) because local motion estimators usually have a *constant luminance* constraint, *i.e.,* pel-recursive methods and all motion estimators that make use of block matching in one stage or another [21]. Even if motion can be well compensated, a strategy is required for correcting flicker in previously occluded regions that have become uncovered.

For these reasons, our strategy for estimating intensity-flicker parameters in nonstationary scenes is based on local motion detection. First, we register a pair of frames to compensate for global motion (Section 4.2.1). Then we estimate the intensity-flicker parameters as outlined in Section 4.1.2. Using these parameters, we detect the remaining local motions (Section 4.2.2). Finally, we interpolate the missing parameters of the nonstationary regions (Section 4.2.3).

4.2.1. *Estimating global motion with phase correlation.* In sequences with camera pan, applying global-motion compensation only helps satisfy the requirement of stationarity if the global motion vectors (one vector to each frame) are accurate: *i.e.*, if the global motion estimator is robust against intensity flicker. A global motion estimator that suits our purpose is the phase correlation method [21] applied to high-pass-filtered versions of the images. Phase correlation determines motion based on phase shifts in the Fourier domain. Because it uses Fourier coefficients normalized by their magnitude, this method is relatively insensitive to fluctuations in image intensity. As we assumed that the amount of intensity flicker varies smoothly in a spatial sense, the direction of changes in intensity over edges and textured regions is preserved in the presence of intensity flicker. This means that the phases of the higher-frequency components will not be affected by intensity flicker. Local mean intensities can show considerable variations from frame to frame. This gives rise to random variations in the phase of the low-frequency components. These random variations are disturbing factors in the motion estimation process that can be avoided by removing the low-pass frequency components from the input images.

Phase correlation determines phase shift in the Fourier domain as follows:

$$C_{t,t-1}(\omega_1,\omega_2) = \frac{S_t(\omega_1,\omega_2)S^*_{t-1}(\omega_1,\omega_2)}{||S_t(\omega_1,\omega_2)S^*_{t-1}(\omega_1,\omega_2)||} \tag{26}$$

where $S_t(\omega_1,\omega_2)$ stands for the 2D Fourier transform of $Y(x,y,t)$ and * denotes the complex conjugate. If $Y(x,y,t)$ and $Y(x,y,t-1)$ are spatially shifted but otherwise identical images, the inverse transform of (26) leads to a delta pulse in the 2D correlation function. Its location yields the global displacement vector $(d_x,d_y)^T$. We can now compensate for global motion in estimating the model parameters by replacing (23) and (24) with:

$$\hat{\beta}_{m,n}(t) = E[Y(x,y,t)] - \hat{\alpha}_{m,n}(t)\cdot E[\hat{I}(x-d_x,y-d_y,t-1)], \tag{27}$$

$$\hat{\alpha}_{m,n}(t) = \sqrt{\frac{var[Y(x,y,t)] - var[\eta(x,y,t)]}{var[\hat{I}(x-d_x,y-d_y,t-1)]}}. \tag{28}$$

4.2.2. *Detecting remaining local motion.* It is important to detect the remaining local motion after we have compensated for global motion. Local motion causes changes in image statistics that are not due to intensity flicker. This leads to incorrect estimates of $\alpha(x,y,t)$ and $\beta(x,y,t)$: *i.e.*, to visible artifacts in the corrected image sequence.

We have developed a robust motion detection system that relies on the current frame only. The underlying assumption of this method is that motion should only be detected if visible artifacts would otherwise be introduced. First, the observed image is subdivided into blocks $\Omega_{m,n}$ that overlap their neighbors both horizontally and vertically (The overlapping boundary regions form sets of reference intensities. The intensity-flicker parameters are estimated for

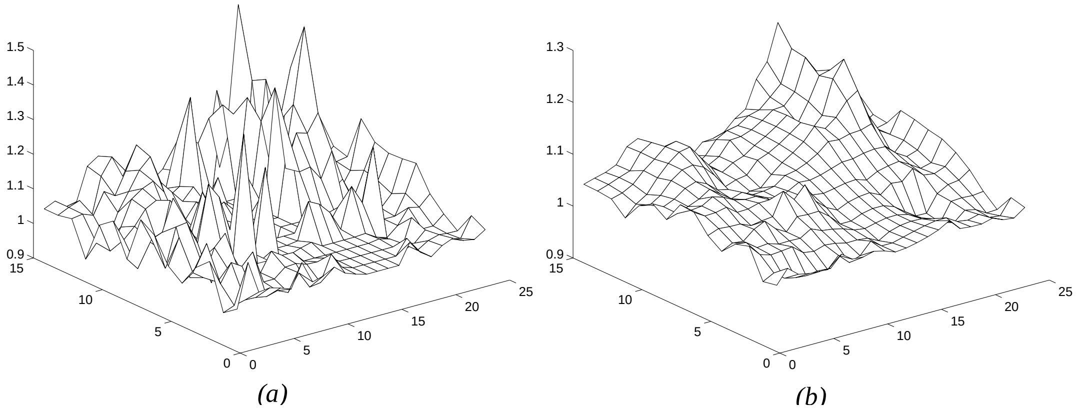


FIGURE 12. (a) Set of original measurements that have variable accuracy, the missing measurements have been set to 1. (b) Smoothed and interpolated parameters using SOR.

each block by (27) and (28). These parameters are used with (12), (14), and (15) for correcting the intensities in the boundary regions. Then, for each pair of overlapping blocks, the common pixels that are assigned significantly different values are counted. Corrected pixels are considered to be significantly different when their absolute difference exceeds a threshold T_d. Finally, motion is flagged if the number of significantly different pixels exceeds a constant D_{max} , which depends on the number of pixels compared.

4.2.3. *Interpolation of missing parameters using successive overrelaxation (SOR).* Due to noise and local motion, the estimated intensity-flicker parameters are unreliable in some cases. We refer to these parameters as *missing*. The other parameters are referred to as *known*. We want to find estimates of the *missing* parameters by means of interpolation. We also want to smooth the *known* parameters. The interpolation and smoothing functions should meet the following requirements. First, the system of intensity-flicker correction should switch itself off when the correctness of the interpolated values is less certain. This means that the interpolator should incorporate biases for $\hat{\alpha}_{m,n}(t)$ and $\hat{\beta}_{m,n}(t)$ towards unity and zero, respectively, that grow as the smallest distance to a region with *known* parameters becomes larger. Second, the reliability of the *known* parameters should be taken into account, *i.e.*, the weights $W_{m,n,t}$ defined in (eq. 25).

An interpolation method that meets our requirements is SOR, which is a well-known iterative method based on repeated low-pass filtering [17]. Figure 12 shows an example of this method.

4.3. *Practical issues*

Figure 13 shows the overall structure of the system of intensity-flicker correction. We have added some operations to this figure that we have not mentioned before and that improve the system's behavior. First, the current input and the previous system output (with global motion compensation) are low-pass filtered with a 5×5 Gaussian kernel. Prefiltering suppresses the influence of high-frequency noise and the effects of small motion. Then, local means μ and variances σ^2 are computed to be used for estimating the intensity-flicker parameters. These and the current input are used to detect local motions. Then, the missing parameters are interpolated and the known parameters are smoothed. Bilinear interpolation is used for upsampling the estimated parameters to full spatial resolution. This avoids the introduction of blocking artifacts in the correction stage that follows.

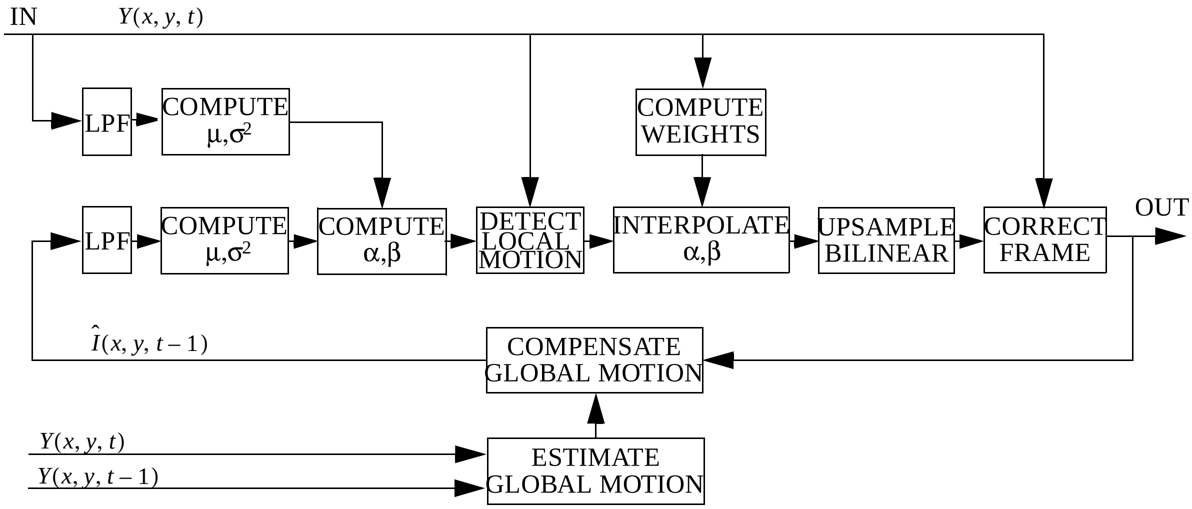


FIGURE 13. Global structure of intensity-flicker correction system

To avoid possible drift due to error accumulation (resulting from the need to approximate the expectation operator and from model mismatches), we bias the corrected intensities towards the contents of the current frame. Equation (12) is therefore replaced by:

$$\hat{I}(x,y,t) = \kappa \cdot (\alpha(x,y,t) \cdot Y(x,y,t) + b(x,y,t) + (1-\kappa) \cdot Y(x,y,t) \quad (29)$$

where κ is the forgetting factor. A practical value for κ is 0.85.

4.4. *Experiments and results*

We applied the system of intensity-flicker correction both to sequences containing artificially added intensity flicker and to sequences with real (non synthetic) intensity flicker. The first set of experiments takes place in a controlled environment and allows us to evaluate the correction system under extreme conditions. The second set of experiments verifies the practical effectiveness of our system and forms a verification of the underlying assumptions of our approach.

If our algorithm functions well and the image content does not change significantly, then the equalized frame means and variances, in a temporal sense, should have a low variance. Indeed, the converse need not be true, but visual inspection helps us to verify the results. Therefore, we propose to use the decreases in variation in frame means and in frame variances to measure the effectiveness of our system.

4.4.1. *Experiment on artificial intensity flicker.*

For our first experiment we used the Mobile sequence (40 frames) which contains moving objects and camera panning (0.8 pixels/frame). The sequence was degraded in a manner that simulates noise and intensity flicker as one can expect to find in a practical situation. First, we added film-grain noise to the original sequence. This noise was generated by the method mentioned in [16] with a variance of 33. Then we added artificial intensity flicker. The intensity-flicker parameters were artificially created from second-order 2D polynomial surfaces. The coefficients for the surfaces were drawn from the normal distribution $N(0, 0.1)$ (from $N(1, 0.1)$) for the zero-th order (DC) term) to generate the $\alpha(x, y, t)$ and from $N(0, 10)$ to generate the $\beta(x, y, t)$. Visually speaking, this leads to severe amounts of intensity flicker. Finally, we simulated thermal noise introduced by, for instance, a *telecine* or VCR. For this purpose we added zero-mean signal-independent white Gaussian noise with a variance of 20.

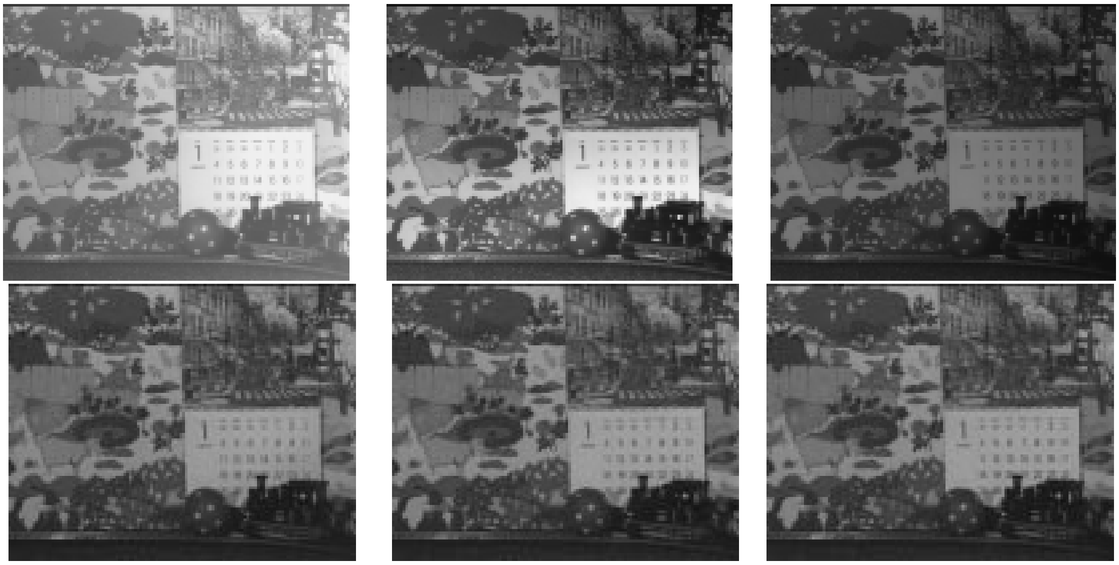


FIGURE 14. Top: frames 16, 17 and 18 of Mobile sequence with synthetic intensity-flicker. Bottom: corrected frames

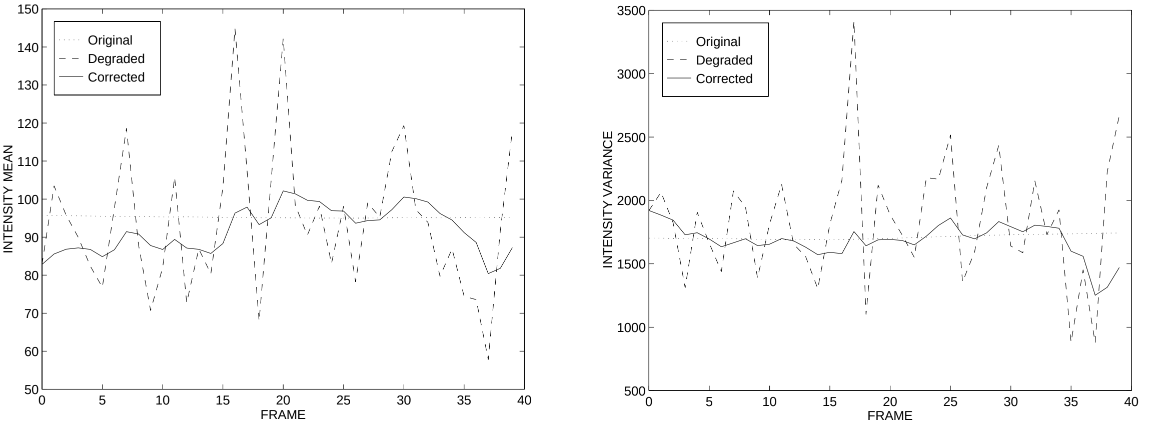


FIGURE 15. Means (left) and variances (right) of original Mobile sequence, Mobile sequence degraded with artificial intensity-flicker and noise, and corrected Mobile sequence.

Global motion was compensated for during the sequence correction.

Figure 14 shows some frames from the degraded and the corrected sequence. Figure 15 shows the frame means and variances of the original, degraded and corrected test sequence. From these graphs we note that the variation in mean and in variance has been significantly reduced. Visual inspection confirms that the amount of intensity flicker has been reduced significantly.

4.4.2. *Experiment on real intensity flicker*. For our second experiment we used a sequence, which we call *Tunnel*, that originates from the film archives of INA. Tunnel is 226 frames long and it consists of a man entering the scene through a tunnel. There is some camera unsteadiness during the first 120 frames, then the camera pans to the right and up. There is film-grain noise and considerable intensity flicker in this sequence. We used the method described in [11] to estimate the total noise variance, which was 8.9. Global motion compensation was applied during the correction. Figure 16 shows three frames from this sequence, original and corrected. Figure 17 shows that the fluctuation in frame means and variances

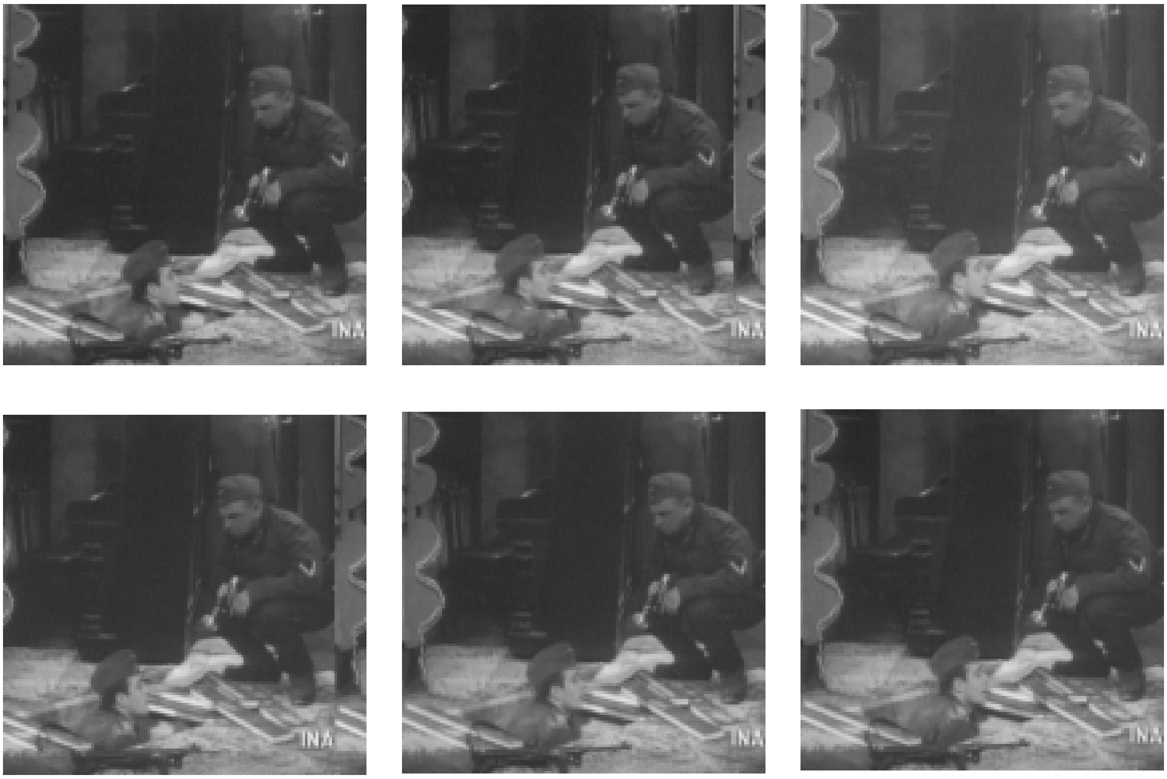


FIGURE 16. Top: frames 13, 14 and 15 of the Tunnel sequence. Bottom: corrected frames

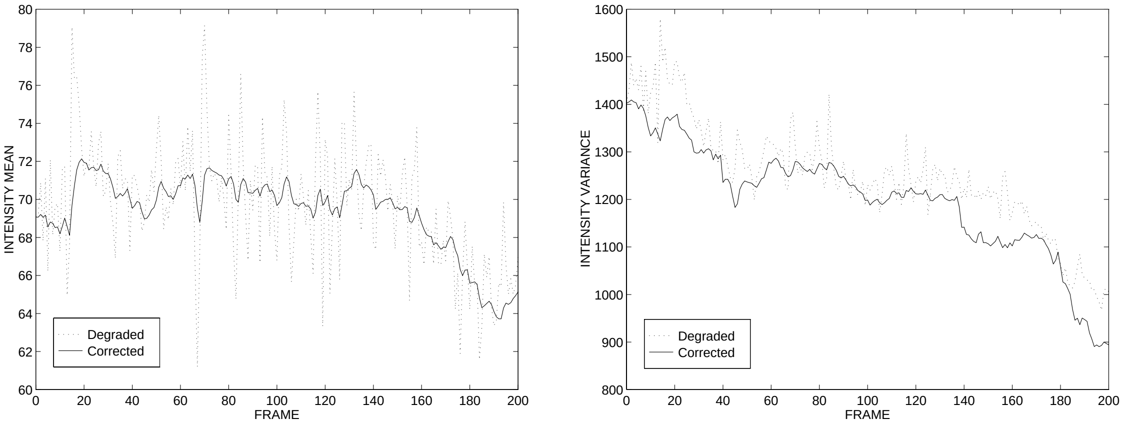


FIGURE 17. Means (left) and variances (right) of degraded and corrected Tunnel sequence.

have significantly been reduced. Visual inspection shows that the intensity flicker has been reduced significantly without introducing visible new artifacts.

4.5. *Conclusions*

We introduced a system for correcting intensity flicker that performs well on artificially and naturally degraded sequences. In broadcasting and in film restoration environments, real-time implementation of our system is required. From an implementation point of view, the required computations are simple and can easily be performed by hardware.

5. Objective evaluation of the improvement in quality of restored image sequences

Once an integrated image restoration system that is capable of simultaneously correcting multiple artifacts has been designed, as the one in AURORA, one wants to know how effective

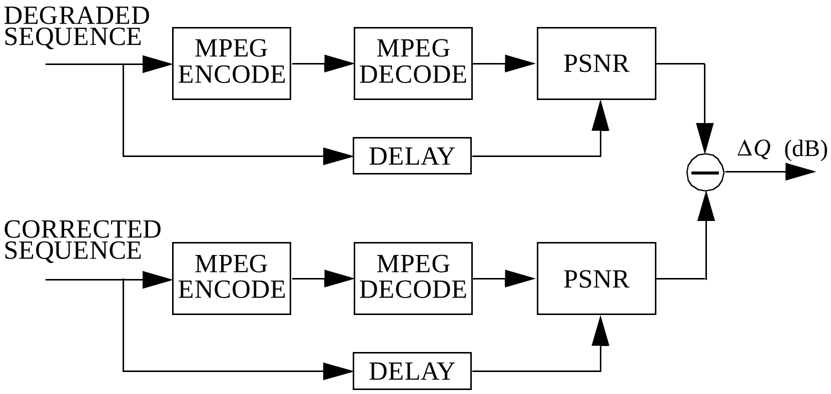


FIGURE 18. Overview of the OMIQ system.

the system is, how well it does its job. Because the goal of restoration is to improve the visual (and audio) quality of degraded film sequences, the appropriate evaluation method is by subjective evaluation. However, subjective evaluation is time consuming, expensive, and does not allow large amounts of information to be evaluated due to practical constraints. This means that some objective measure of quality must be defined.

In the previous sections we proposed three algorithms for image restoration and we used three different measures of performance. Though these measures give good objective indications of the performance of the individual restoration processes, they do not directly give a measure of quality of the overall result. In fact, some of the measures used require information that is not available in a practical situation, such as the original noise-free images and the real (not estimated) blotch masks.

In the next section, Section 5.1, we propose an objective measure for the improvement in image quality based on coding efficiency. Then in Section 5.2 we apply this measure to the experimental results from the previous sections. We conclude with a discussion in Section 5.3.

5.1. *An Objective Measure of Improvement in image Quality (OMIQ)*

It should be noted that, as no original unimpaired reference sequences are available, any measure for the quality of a restored sequence can be at most a relative measure. In order to come to an objective relative measure we make the following two assumptions:

1 Image restoration improves the objective quality by removing artifacts,
2 Removing artifacts increases the coding efficiency.

The second assumption is likely to be true for artifacts such as noise, blotches and intensity flicker because removing these reduces the magnitude of the prediction errors both in a temporal and a spatial sense. Having smaller prediction errors means that less bits are required to obtain the same quality and, conversely, that higher quality can be obtained at the identical bit rate. In [5] there is evidence that image stabilization also increases the coding efficiency. Note that there are some artifacts for which the second assumption is not true, *e.g.*, deblurred images require more bits for coding than the out of focus originals. For that reason we exclude unsharpness artifacts from the discussion.

We now define a measure of quality improvement ΔQ as the difference between the coding efficiency of the restored image sequence and the coding efficiency of the impaired sequence:

$$\Delta Q = E(\text{corrected}) - E(\text{impaired}). \tag{30}$$

The unit of the coding efficiency function ΔQ can be either *bits* or *dB*. Figure 18 shows the scheme that allows us to measure coding efficiency when dB is taken as the unit of efficiency. First, the restored image sequence is encoded for a constant bitrate. Then E(corrected) is given by the PSNR computed between the input and decoded output. Next, the impaired image sequence is encoded at the same bitrate and $E(impaired)$ is given by the PSNR computed between the input and decoded output. ΔQ can now be computed and its interpretation, in a digital broadcasting or storage environment, is that of the lower bound on the improvement in image quality as observed by the (home) viewer. It is a lower bound because, according to assumption 1, the quality of the corrected sequence is higher than that of the impaired original, but we do not how much higher. It measures an *improvement* because, according to assumption 2, the coding error is smaller and thus the visual quality of the decoded sequence is higher.

When we use *bit* as the unit of coding efficiency, we set the coder's bitrate so that the PSNR between coded/decoded corrected sequences equals the PSNR between coded/decoded impaired sequences. $E(\text{corrected})$ and $E(\text{impaired})$ are given by the sizes in bits of the compressed sequence. The interpretation of ΔQ , which is negative in this case, is that of how many bits of irrelevant information were removed by the restoration process.

5.2. *Objective evaluation of the experimental results*

We applied OMIQ to the impaired and corrected sequences from the previous sections. We used the standard TM5 MPEG2 encoder. Initially we set the bitrate to 5 Mb/s for the Plane, Mobile and Tunnel sequences, which all have a frame size of 720×576 pixels. Because the frame size of the Western sequence is smaller (256×256 pixels) we selected a lower bitrate for this sequence, namely 0.8 Mb/s. Figure 19 shows the PSNR between the input sequences and the coded/decoded results. From these graphs it is clear that the coding errors for corrected sequences are smaller than those for the original impaired sequences. This is in accordance with assumption 2 in the previous section.

Table 2 shows the mean of the PSNR of the coded sequences and the objective improvement in image quality in dB. These numbers confirm that image restoration does indeed lead to a higher quality of the coded sequence at identical bitrates for the artifacts under consideration. The gains achieved by noise reduction are especially noteworthy. Table 2 also gives the objective improvement in image quality in bits. These numbers were obtained by coding the impaired sequences at higher bit rates than before such that the PSNR of coded/decoded impaired sequences equals the PSNR of coded/decoded corrected sequences (at 5 Mb/s and 0.8 Mb/s respectively). We see that applying restoration can lead to great savings, ranging from 30 to 270 percent in bits, whilst maintaining the same coding quality.

5.3. *Discussion*

A question is how well this objective measure corresponds to subjective evaluation. We stated that OMIQ gives a lower bound on the improvement in image quality as observed by the (home) viewer. This, however, does not mean that the viewer experiences this improvement in quality as such. And, in fact, he or she does not. The reason is that it is difficult for an observer to grade the improvement in image quality in terms of dB or bits. It is natural to say, for instance, that removing intensity flicker, noise and blotches has increased the visual quality, but it is difficult to assign a figure of merit to this increase.

The conclusion, therefore, must be that OMIQ indicates what savings can be achieved in terms of bandwidth and storage capacity by applying image restoration prior to digital broadcast and storage. One can assume (for the artifacts under consideration) that when taking advantage of these savings the visual quality of coded/decoded corrected sequences is at least equal, but probably better, than that of coded/decoded impaired sequences. The achievable savings are a measure of the overall performance of the restoration system.

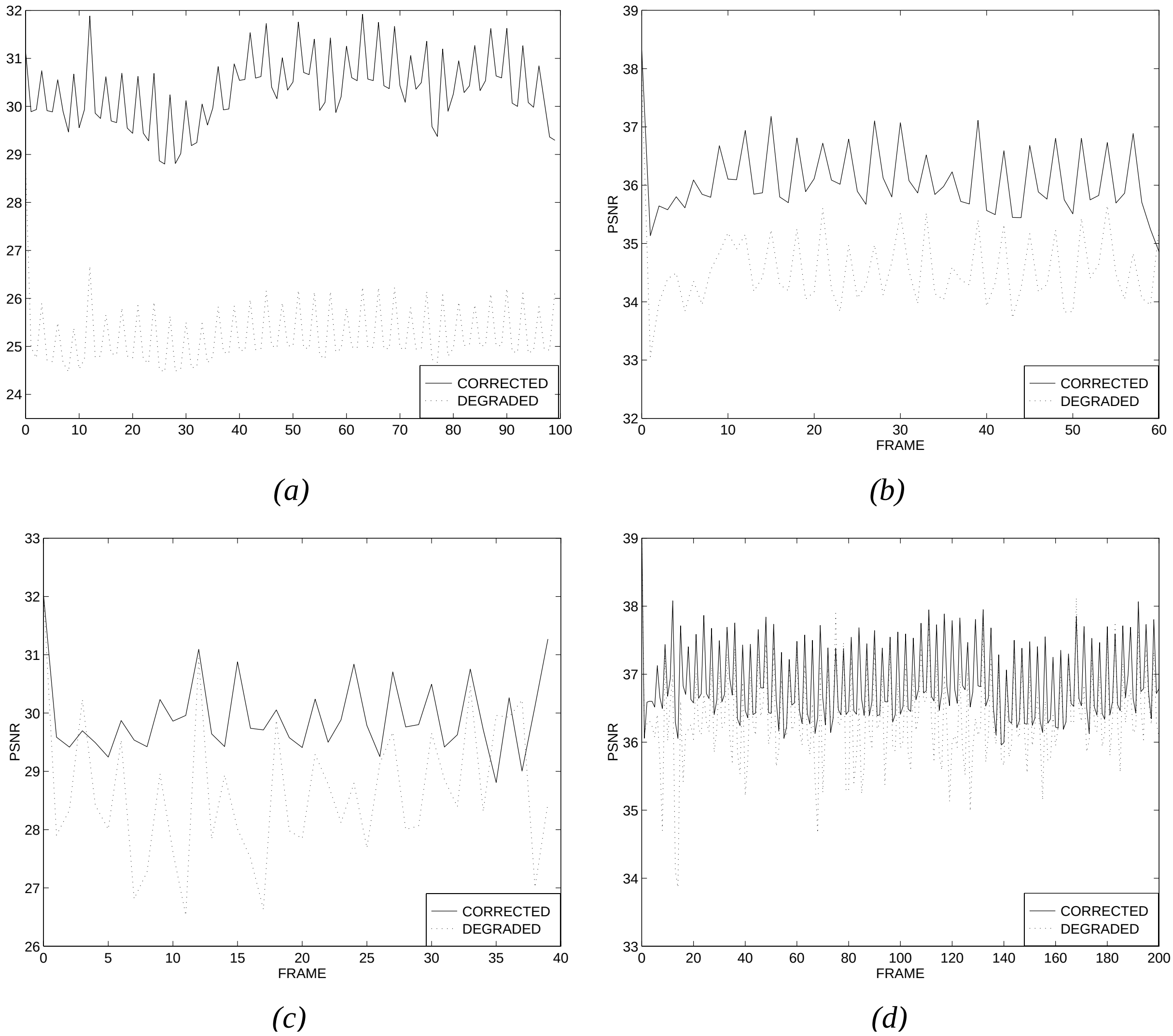


FIGURE 19. PSNR of sequences after MPEG2 encoding. (a) Impaired (noise) and corrected Plane sequence. (b) Impaired (blotches) and corrected Western sequence. (c) Impaired (intensity-flicker) and corrected Mobile sequence. (d) Impaired (intensity-flicker) and corrected Tunnel sequence.

SEQUENCE	IMPAIRMENT	MEAN PSNR (dB) IMPAIRED	MEAN PSNR (dB) CORRECTED	ΔQ (dB)	ΔQ (Mbit)	%
Plane	Noise	25.2	30.4	5.2	-13.5	270
Western	Blotches	34.6	36.1	1.5	-0.35	44
Mobile	Flicker	28.7	29.9	1.2	-2.7	54
Tunnel	Flicker	36.4	36.9	0.5	-1.5	30

Table 2. Sequence mean in PSNR between input and coded/decoded output at 5 Mbit/s. Also given is the improvement in image quality in dB and in bits. he last column gives the relative increase in the number of bits. See text for explanation.

6. Discussion

Applying image restoration prior to storing or broadcasting an image sequence in a compressed format can lead to significant savings in the number of bits required for coding.

This is important to note in an era in which broadcasters are transforming their broadcasts from analog to digital. Due to compression and digital broadcasting, the number of channels available to the home viewer will increase dramatically. However, these channels require programming. Nowadays, production costs for new programming are tremendous. An alternative to making new programming is recycling the many films, soaps and quiz shows that have been made over the last 50 or 60 years. This alternative can only be viable if the visual and audio quality meets the standards expected by the modern viewer and if the costs for broadcasting old films do not exceed the costs for broadcasting contemporary films. We suggest that image restoration can fulfill both of these requirements.

The Plane sequence was made available by courtesy of the BBC. The Western sequence was made available by courtesy of Dr. A.C. Kokaram of the University of Cambridge. The Tunnel sequence was made available by courtesy of INA.

References

[1] M. Antonini, T. Gaidon, P. Mathieu, and M. Barlaud. Wavelet transforms and image coding. In M. Barlaud, editor, *Wavelets in Image Communication*, volume 5, pages 66–78. Elsevier, 1994.

[2] G. R. Arce. Multistage order statistic filters for image sequence processing. *IEEE Trans. on Signal Processing*, 39(5):1146–1163, 1991.

[3] F.C. Bilingsley. Noise considerations in digital image processing hardware. In T.S. Huang, editor, *Topics in Applied Physics*, volume 6. Springer Verlag, Berlin, 1975.

[4] J. Canny. A computational approach to edge detection. *IEEE PAMI*, 8(6), 1986.

[5] A. T. Erdem and C. Eroglu. The effect of image stabilization on the performance of the mpeg-2 video coding algorithm. In *Proc. of VCIP-98*, volume 1, pages 272–277, 1998.

[6] S. Karla, M.N. Chong, and D. Krishnan. A new auto regressive (ar) model-based algorithm for motion picture restoration. In *Proc. ICASSP97*, volume 4, pages 2557–2560, Munich, 1997. ICASSP97.

[7] R.P. Kleihorst. *Noise Filtering of Image Sequences*. PhD thesis, Delft University of Technology, October 1994.

[8] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner. Detection of missing data in image sequences. *IEEE Trans. on Image Processing*, 4(11):1496–1508, 1995.

[9] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner. Interpolation of missing data in image sequences. *IEEE Trans. on Image Processing*, 4(11):1509–1519, 1995.

[10] M. Lang, H. Guo, J.E. Odegard, C.S. Burrus, and R.O. Wells. Nonlinear processing of a shift-invariant dwt for noise reduction. volume 2491. Orlando, Florida, 1995.

[11] J.B. Martens. Adaptive contrast enhancement through residue-image processing. *Signal Processing*, (44):1–18, 1995.

[12] J.H. McClellan. The design of two-dimensional filters by transformations. In *Proc. 7th Annual Princeton conference of Sciences and Systems*, pages 247–251, 1973.

[13] H. Muller-Seelich, W. Plaschzug, P. Schallauer, S. Potzman, and W. Haas. Digital restoration of 35mm film. In *Proc. of ECMAST 96*, volume 1, pages 255–265, Louvain-la-Neuve, Belgium, 1996. ECMAST96.

[14] M.J. Nadenau and S.K. Mitra. Blotch and scratch detection in image sequences based on rank ordered differences. volume 5 of *Time-Varying Image Processing and Moving Object Recognition*. Elsevier, 1997.

[15] M.K. Özkan, A.T. Erdem, M.I. Sezan, and A.M. Tekalp. Efficient multiframe wiener restoration of blurred and noisy image sequences. *IEEE Trans. on Image Processing*, 1(4):453–476, 1992.

[16] M.K. Özkan, M.I. Sezan, and M. Tekalp. Adaptive motion compensated filtering of noisy image sequences. *IEEE Trans. on Circuits and Systems for Video Technology*, 3(4), 1993.

[17] Press, Teukolsky, Vetterling, and Flannery. *Numerical Recipes in C*. Cambridge University Press, 2nd edition, 1992.

[18] P. Richardson and D. Suter. Restoration of historic film for digital compression: A case study. In *Proc. of ICIP-95*, volume 2, pages 49–52. ICIP95, 1995.

[19] E.P. Simoncelli and E.H. Adelson. Noise removal via bayesian wavelet coring. In *Proc. of ICIP-96*, volume 1, pages 379–372. ICIP96, 1996.

[20] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multiscale transforms. *IEEE Trans. on Information Theory*, 38(2):587–607, 1992.

[21] A. M. Tekalp. *Digital Video Processing*. Prentice Hall, 1995.

[22] N. Weyrich and G. T. Warhola. Wavelet shrinkage and generalized cross validation for image denoising. *IEEE Trans. on Image Processing*, 7(1):82–90, 1998.