

3.11

Video Enhancement and Restoration

Reginald L. Lagendijk,
Peter M.B. van Roosmalen,
Jan Biemond, Andrei Rares,
and Marcel J.T. Reinders
*Delft University of Technology,
The Netherlands*

1	Introduction.....	275
2	Spatio-Temporal Noise Filtering	277
2.1	Linear Filters • 2.2 Order-Statistic Filters • 2.3 Multi-Resolution Filters	
3	Blotch Detection and Removal.....	282
3.1	Blotch Detection • 3.2 Motion Vector Repair and Interpolating Corrupted Intensities • 3.3 Restoration in Conditions of Difficult Object Motion	
4	Vinegar Syndrome Removal	287
5	Intensity Flicker Correction	289
5.1	Flicker Parameter Estimation • 5.2 Estimation on Sequences with Motion	
6	Kinescope Moiré Removal	292
7	Concluding Remarks.....	293
	References	294

1 Introduction

Even with the advancing camera and digital recording technology, there are many situations in which recorded image sequences — or *video* for short — may suffer from severe degradations. The poor quality of recorded image sequences may be due to, for instance, the imperfect or uncontrollable recording conditions such as one encounters in astronomy, forensic sciences, and medical imaging. Video enhancement and restoration has always been important in these application areas not only to improve the visual quality, but also to increase the performance of subsequent tasks such as analysis and interpretation.

Another important application of video enhancement and restoration is that of preserving motion pictures and video tapes recorded over the last century. These unique records of historic, artistic, and cultural developments are deteriorating rapidly due to aging effects of the physical reels of film and magnetic tapes that carry the information. The preservation of these fragile archives is of interest not only to professional archivists, but also to broadcasters as a cheap alternative to fill the many television channels that have come available with digital broadcasting. Re-using old film and video material is,

however, only feasible if the visual quality meets the standards of today. First, the archived film and video is transferred from the original film reels or magnetic tape to digital media. Then, all kinds of degradations are removed from the digitized image sequences, in this way increasing the visual quality and commercial value. Because the objective of restoration is to remove irrelevant information such as noise and blotches, it restores the original spatial and temporal correlation structure of digital image sequences. Consequently, restoration may also improve the efficiency of the subsequent MPEG compression of image sequences.

An important difference between the enhancement and restoration of 2D images and of video is the amount of data to be processed. Whereas for the quality improvement of important images elaborate processing is still feasible, this is no longer true for the absolutely huge amounts of pictorial information encountered in medical sequences and film/video archives. Consequently, enhancement and restoration methods for image sequences should have a manageable complexity, and should be semi-automatic. The term semi-automatic indicates that in the end professional operators control the visual quality of the restored image sequences by selecting values for some of the critical restoration parameters.

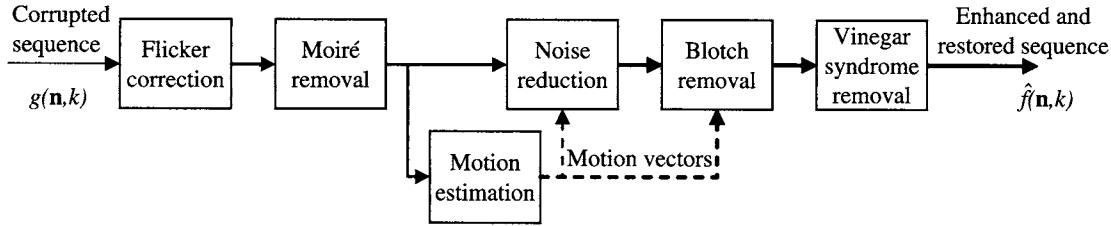


FIGURE 1 Some processing steps in the removal of various video artifacts.

The most common artifact encountered in the above-mentioned applications is noise. Over the last two decades an enormous amount of research has focused on the problem of enhancing and restoring 2D images. Clearly, the resulting spatial methods are also applicable to image sequences, but such an approach implicitly assumes that the individual pictures of the image sequence, or *frames*, are temporally independent. By ignoring the temporal correlation that exists, suboptimal results may be obtained and the spatial *intra-frame* filters tend to introduce temporal artifacts in the restored image sequences. In this chapter we focus our attention specifically on exploiting temporal dependencies, yielding *inter-frame* methods. In this respect the material offered in this chapter is complementary to that on image enhancement in the Chapters 3.1 to 3.4 of this *Handbook*. The resulting enhancement and restoration techniques operate in the temporal dimension by definition, but often have a spatial filtering component as well. For this reason, video enhancement and restoration techniques are sometimes referred to as *spatio-temporal* filters or 3D filters. Section 2 of this chapter presents three important classes of noise filters for video frames, namely *linear temporal filters*, *order-statistic filters*, and *multi-resolution filters*.

In forensic sciences and in film and video archives a large variety of artifacts are encountered. Besides noise, we discuss the removal of other important impairments that rely on spatial or temporal processing algorithms, namely blotches (Section 3), vinegar syndrome (Section 4), intensity flicker (Section 5), and kinescope moiré (Section 6). Blotches are dark and bright spots that are often visible in damaged film image sequences. The removal of blotches is essentially a temporal detection and interpolation problem. In certain circumstances, their removal needs to be done by means of spatial algorithms, as will be shown later in this chapter. Vinegar syndrome represents a special type of impairment related to film and it may have various appearances (e.g., partial loss of color, blur, etc.). In some cases, blotch removal algorithms can be applied for its removal. In general, however, the particular properties of their appearance need to be taken into account. Intensity flicker refers to variations in intensity in time, caused by aging of film, by copying and format conversion (for instance from film to video), and — in case of earlier film — by variations in shutter time. Whereas blotches are spatially

highly localized artifacts in video frames, intensity flicker is usually a spatially global, but not stationary, artifact. The kinescope moiré phenomenon appears during film-to-video transfer using Telecine devices. It is caused by the superposition of (semi-)periodical signals from the film contents and the scan pattern used by the Telecine device. Because of its (semi-)periodical nature, spectrum analysis is generally used for its removal.

It becomes apparent that image sequences may be degraded by multiple artifacts. For practical reasons, restoration systems follow a sequential procedure, where artifacts are removed one by one. As an example, Fig. 1 illustrates the order in which the removal of flicker, moiré, noise, blotches, and vinegar syndrome takes place. The reasons for this modular approach are the necessity to judge the success of the individual steps (for instance by an operator), and the algorithmic and implementation complexity.

As already suggested in Fig. 1, most temporal filtering techniques require an estimate of the motion in the image sequence. Motion estimation has been discussed in detail in Chapters 3.7 and 6.1 of this *Handbook*. The estimation of motion from degraded image sequences is, however, problematic. We are faced with the problem that the impairments of the video disturb the motion estimator, but that at the same time correct motion estimates are assumed in developing enhancement and restoration algorithms. In this chapter we will not discuss the design of new motion estimators [24, 36, 40] that are robust to the various artifacts, but we will assume that existing motion estimators can be modified appropriately such that sufficiently correct and smooth motion fields are obtained. The reason for this approach is that even under ideal conditions motion estimates are never perfect. Usually, incorrect or unreliable motion vectors are dealt with in a few special ways. In the first place, clearly incorrect or unreliable motion vectors can be repaired. Secondly, the enhancement and restoration algorithms should be robust against a limited amount of incorrect or unreliable motion vectors. Thirdly, areas with wrong motion vectors which are impossible to repair can be protected against temporal restoration, in order to avoid an outcome which is visually more objectionable than the input sequence itself. In such a case, the unavailability of temporal information makes the spatial-only restoration more suitable than the temporal one.

2 Spatio-Temporal Noise Filtering

Any recorded signal is affected by noise, no matter how precise the recording equipment. The sources of noise that can corrupt an image sequence are numerous (see Chapter 4.4 of this *Handbook*). Examples of the more prevalent ones include camera noise, shot noise originating in electronic hardware and the storage on magnetic tape, thermal noise, and granular noise on film. Most recorded and digitized image sequences contain a mixture of noise contributions, and often the (combined) effects of the noise are non-linear in nature. In practice, however, the aggregated effect of noise is modeled as an additive white (sometimes Gaussian) process with zero mean and variance σ_w^2 that is independent from the ideal uncorrupted image sequence $f(\mathbf{n}, k)$. The recorded image sequence $g(\mathbf{n}, k)$ corrupted by noise $w(\mathbf{n}, k)$ is then given by

$$g(\mathbf{n}, k) = f(\mathbf{n}, k) + w(\mathbf{n}, k) \quad (1)$$

where $\mathbf{n} = (n_1, n_2)$ refers to the spatial coordinates and k to the frame number in the image sequence. More accurate models are often much more complex but lead to little gain compared to the added complexity.

The objective of noise reduction is to make an estimate $\hat{f}(\mathbf{n}, k)$ of the original image sequence given only the observed noisy image sequence $g(\mathbf{n}, k)$. Many different approaches toward noise reduction are known, including optimal linear filtering, non-linear filtering, scale-space processing, and Bayesian techniques. In this section we discuss successively the class of linear image sequence filters, order-statistic filters, and multi-resolution filters. In all cases the emphasis is on the temporal filtering aspects. More rigorous reviews of noise filtering for image sequences are given in [9, 11, 41].

2.1 Linear Filters

Temporally Averaging Filters

The simplest temporal filter carries out a weighted averaging of successive frames. That is, the restored image sequence is obtained by [18, 24]:

$$\hat{f}(\mathbf{n}, k) = \sum_{l=-K}^{K} h(l)g(\mathbf{n}, k-l) \quad (2)$$

Here $h(l)$ are the temporal filter coefficients used to weight $2K+1$ consecutive frames. In case the frames are considered equally important we have $h(l) = 1/(2K+1)$. Alternatively, the filter coefficients can be optimized in a minimum mean-squared error fashion

$$h(l) \leftarrow \min_{h(l)} E[(f(\mathbf{n}, k) - \hat{f}(\mathbf{n}, k))^2] \quad (3)$$

yielding the well-known temporal Wiener filtering solution:

$$\begin{aligned} & \begin{pmatrix} h(-K) \\ \vdots \\ h(0) \\ h(1) \\ \vdots \\ h(K) \end{pmatrix} \\ = & \begin{pmatrix} R_{gg}(0) & \cdots & R_{gg}(-K) & \cdots & \cdots & R_{gg}(-2K) \\ \vdots & \ddots & & & & \vdots \\ R_{gg}(K) & & R_{gg}(0) & & & \vdots \\ \vdots & & & R_{gg}(0) & & \vdots \\ \vdots & & & & \ddots & \vdots \\ R_{gg}(2K) & \cdots & \cdots & \cdots & \cdots & R_{gg}(0) \end{pmatrix}^{-1} \\ & \times \begin{pmatrix} R_{fg}(-K) \\ \vdots \\ R_{fg}(0) \\ R_{fg}(1) \\ \vdots \\ R_{fg}(K) \end{pmatrix} \end{aligned} \quad (4)$$

where $R_{gg}(m)$ is the *temporal auto-correlation function* defined as $R_{gg}(m) = E[g(\mathbf{n}, k)g(\mathbf{n}, k-m)]$, and $R_{fg}(m)$ the *temporal cross-correlation function* defined as $R_{fg}(m) = E[f(\mathbf{n}, k)g(\mathbf{n}, k-m)]$. The temporal window length, i.e., the parameter K , determines the maximum degree by which the noise power can be reduced. The larger the window the greater the reduction of the noise, at the same time, however, the more visually noticeable the artifacts resulting from motion between the video frames. A dominant artifact is blur of moving objects due to the averaging of object and background information.

The motion artifacts can greatly be reduced by operating the filter (2) along the picture elements (pixels) that lie on the same motion trajectory [14, 25]. Equation (2) then becomes a *motion-compensated* temporal filter (see Fig. 2):

$$\begin{aligned} \hat{f}(\mathbf{n}, k) = & \sum_{l=-K}^{K} h(l)g(n_1 - d_x(n_1, n_2; k, l), n_2 \\ & - d_y(n_1, n_2; k, l), k-l) \end{aligned} \quad (5)$$

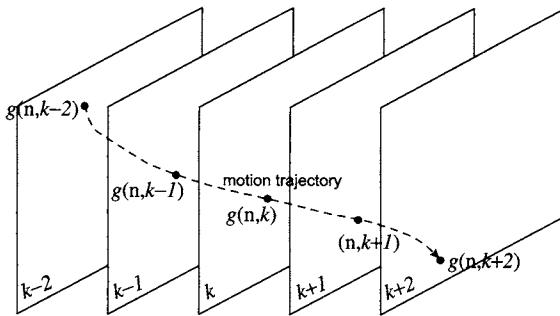


FIGURE 2 Noise filter operating along the motion trajectory of the picture element (\mathbf{n}, k) .

Here $\mathbf{d}(\mathbf{n}; k, l) = (d_x(n_1, n_2; k, l), d_y(n_1, n_2; k, l))$ is the motion vector for spatial coordinate (n_1, n_2) estimated between the frames k and l . It is pointed out here that the problems of noise reduction and motion estimation are inversely related as far as the temporal window length K is concerned. That is, as the length of the filter is increased temporally, the noise reduction potential increases but so are the artifacts due to incorrectly estimated motion between frames that are temporally far apart.

In order to avoid the explicit estimation of motion, which might be problematic at high noise levels, two alternatives are available that turn (2) into a motion-adaptive filter. In the first place, in areas where motion is detected (but not explicitly estimated) the averaging of frames should be kept to a minimum. Different ways exist to realize this. For instance, the temporal filter (2) can locally be switched off entirely, or can locally be limited to using only future or past frames, depending on the temporal direction in which motion was detected. Basically the filter coefficients $h(l)$ are spatially adapted as a function of detected motion between frames. Secondly, the filter (2) can be operated along M *a priori* selected motion directions at each spatial coordinate. The finally estimated value $\hat{f}(\mathbf{n}, k)$ is subsequently chosen from the M partial results according to some selection criterion, for instance as the median [18, 24]:

$$\begin{aligned}\hat{f}_{i,j}(\mathbf{n}, k) &= \frac{1}{3}(g(n_1 - i, n_2 - j, k - 1) + g(n_1, n_2, k) \\ &\quad + g(n_1 + i, n_2 + j, k + 1))\end{aligned}\quad (6a)$$

$$\begin{aligned}\hat{f}(\mathbf{n}, k) &= \text{median}(\hat{f}_{-1,-1}(\mathbf{n}, k), \hat{f}_{1,-1}(\mathbf{n}, k), \hat{f}_{-1,1}(\mathbf{n}, k), \\ &\quad \hat{f}_{1,1}(\mathbf{n}, k), \hat{f}_{0,0}(\mathbf{n}, k), g(\mathbf{n}, k))\end{aligned}\quad (6b)$$

Clearly cascading (6a) and (6b) turns the overall estimation procedure into a non-linear one, but the partial estimation results are still obtained by the linear filter operation (6a).

It is easy to see that the filter (2) can be extended with a spatial filtering part. There exist many variations to this concept, basically as many as there are spatial restoration techniques for noise reduction. The most straightforward extension of (2) is the following 3D weighted averaging filter [41]:

$$\hat{f}(\mathbf{n}, k) = \sum_{(\mathbf{m}, l) \in S} h(\mathbf{m}, l)g(\mathbf{n} - \mathbf{m}, k - l) \quad (7)$$

Here S is the spatio-temporal support or *window* of the 3D filter (see Fig. 3). The filter coefficients $h(\mathbf{m}, l)$ can be chosen to be all equal, but a performance improvement is obtained if they are adapted to the image sequence being filtered, for instance by optimizing them in the mean-squared error sense (3). In the latter case (7) becomes the theoretically optimal 3D Wiener filter.

There are, however, two disadvantages with the 3D Wiener filter. The first is the requirement that the 3D autocorrelation function for the original image sequence is known *a priori*. The second is the 3D wide-sense stationarity assumptions, which are virtually never true because of moving objects and scene changes. These requirements are detrimental to the performance of the 3D Wiener filter in practical situations of interest. For these reasons, simpler ways of choosing the 3D filter coefficients are usually preferred, provided that they allow for adapting the filter coefficients. One such choice for adaptive filter coefficients is the following [29]:

$$h(\mathbf{m}, l; \mathbf{n}, k) = \frac{c}{1 + \max(\alpha, (g(\mathbf{n}, k) - g(\mathbf{n} - \mathbf{m}, k - l))^2)} \quad (8)$$

Here $h(\mathbf{m}, l; \mathbf{n}, k)$ weights the intensity at spatial location $\mathbf{n} - \mathbf{m}$ in frame $k - l$ for the estimation of the intensity $\hat{f}(\mathbf{n}, k)$. The adaptive nature of the resulting filter can immediately be seen from (8). If the difference between the pixel intensity $g(\mathbf{n}, k)$ being filtered and the intensity $g(\mathbf{n} - \mathbf{m}, k - l)$ for which the filter coefficient is calculated, is less than α , this pixel is included in the filtering with weight $c/(1 + \alpha)$, otherwise it is weighted with a much smaller factor. In this way, pixel intensities that seem to deviate too much from $g(\mathbf{n}, k)$ — for instance due to moving objects within the spatio-temporal window S — are excluded from (7). As with the temporal filter (2) the spatio-temporal filter (7) can be carried out in a motion-compensated way by arranging the window S along the estimated motion trajectory.

Temporally Recursive Filters

A disadvantage of the temporal filter (2) and spatio-temporal filter (7) is that they need to buffer several frames of an image sequence. Alternatively, a recursive filter structure can be used that generally needs to buffer fewer (usually only one) frames. Furthermore, these filters are easier to adapt since there are

fewer parameters to control. The general form of a recursive temporal filter is as follows:

$$\hat{f}(\mathbf{n}, k) = \hat{f}_b(\mathbf{n}, k) + \alpha(\mathbf{n}, k)[g(\mathbf{n}, k) - \hat{f}_b(\mathbf{n}, k)] \quad (9)$$

Here $\hat{f}_b(\mathbf{n}, k)$ is the prediction of the original k -th frame on the basis of previously filtered frames, and $\alpha(\mathbf{n}, k)$ is the filter gain for updating this prediction with the observed k -th frame. Observe that for $\alpha(\mathbf{n}, k) = 1$ the filter is switched off, i.e., $\hat{f}(\mathbf{n}, k) = g(\mathbf{n}, k)$. Clearly, a number of different algorithms can be derived from (9) depending on the way the predicted frame $\hat{f}_b(\mathbf{n}, k)$ is obtained and the gain $\alpha(\mathbf{n}, k)$ is computed. A popular choice for the prediction $\hat{f}_b(\mathbf{n}, k)$ is the previously restored frame, either in direct form

$$\hat{f}_b(\mathbf{n}, k) = \hat{f}(\mathbf{n}, k - 1) \quad (10a)$$

or in motion-compensated form:

$$\hat{f}_b(\mathbf{n}, k) = \hat{f}(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k - 1), k - 1) \quad (10b)$$

More elaborate variations of (10) make use of a local estimate of the signal's mean within a spatio-temporal neighborhood. Furthermore, Equation (9) can also be cast into a formal 3D motion-compensated Kalman estimator structure [42, 20]. In this case the prediction $\hat{f}_b(\mathbf{n}, k)$ depends directly on the dynamic spatio-temporal state-space equations used for modeling the image sequence.

The simplest case for selecting $\alpha(\mathbf{n}, k)$ is by using a globally fixed value. As with the filter structures (2) and (7) it is generally necessary to adapt $\alpha(\mathbf{n}, k)$ to the presence or correctness of the motion in order to avoid filtering artifacts. Typical artifacts of recursive filters are "comet-tails" that moving objects leave behind.

A *switching filter* is obtained if the gain takes on the values α and 1 depending on the difference between the prediction $\hat{f}_b(\mathbf{n}, k)$ and the actually observed signal value $g(\mathbf{n}, k)$:

$$\alpha(\mathbf{n}, k) = \begin{cases} 1 & \text{if } |g(\mathbf{n}, k) - \hat{f}_b(\mathbf{n}, k)| > \varepsilon \\ \alpha & \text{if } |g(\mathbf{n}, k) - \hat{f}_b(\mathbf{n}, k)| \leq \varepsilon \end{cases} \quad (11)$$

For areas that have a lot of motion (if the prediction (10a) is used) or for which the motion has been estimated incorrectly (if the prediction (10b) is used), the difference between the predicted intensity value and the noisy intensity value is large, causing the filter to switch off. For areas that are stationary or for which the motion has been estimated correctly, the prediction differences are small yielding the value α for the filter coefficient.

A finer adaptation is obtained if the prediction gain is optimized to minimize the mean-squared restoration error (3), yielding:

$$\alpha(\mathbf{n}, k) = \max\left(1 - \frac{\sigma_w^2}{\sigma_g^2(\mathbf{n}, k)}, 0\right) \quad (12)$$

Here $\sigma_g^2(\mathbf{n}, k)$ is an estimate of the image sequence variance in a local spatio-temporal neighborhood of (\mathbf{n}, k) . If this variance is high, it indicates large motion or incorrectly estimated motion, causing the noise filter to switch off, i.e., $\alpha(\mathbf{n}, k) = 1$. If $\sigma_g^2(\mathbf{n}, k)$ is in the same order of magnitude as the noise variance σ_w^2 , the observed noisy image sequence is obviously very unreliable so that the predicted intensities are used without updating it, i.e., $\alpha(\mathbf{n}, k) = 0$. The resulting estimator is known as the local linear minimum mean-squared error (LLMMSE) estimator. A drawback of (12), as with any noise filter that requires the calculation of $\sigma_g^2(\mathbf{n}, k)$, is that outliers in the windows used to calculate this variance may cause the filter to switch-off. Order-statistic filters are more suitable for handling data in which outliers are likely to occur.

2.2 Order-Statistic Filters

Order-statistic (OS) filters are non-linear variants of weighted-averaging filters. The distinction is that in OS filters the observed noisy data — usually taken from a small spatio-temporal window — are ordered before being used. Because of the ordering operation, correlation information is ignored in favor of magnitude information. Examples of simple OS filters are the minimum operator, maximum operator, and median operator. OS-filters are often applied in *directional filtering*. In directional filtering different filter directions are considered corresponding to different spatio-temporal edge orientations. Effectively this means that the filtering operation takes place along the spatio-temporal edges, avoiding the blurring of moving objects. The directional filtering approach may be superior to adaptive or switching filters since noise around spatio-temporal edges can effectively be eliminated by filtering along those edges, as opposed to turning off the filter in the vicinity of edges [41].

The general structure of an OS restoration filter is as follows:

$$\hat{f}(\mathbf{n}, k) = \sum_{r=1}^{|S|} h_{(r)}(\mathbf{n}, k) g_{(r)}(\mathbf{n}, k) \quad (13)$$

Here $g_{(r)}(\mathbf{n}, k)$ are the ordered intensities, or *ranks*, of the corrupted image sequence, taken from a spatio-temporal window S with finite extent centered around (\mathbf{n}, k) (see Fig. 3). The number of intensities in this window is denoted by $|S|$. As with linear filters, the objective is to choose appropriate filter coefficients $h_{(r)}(\mathbf{n}, k)$ for the ranks.

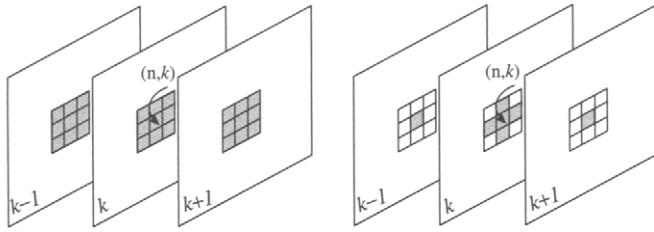


FIGURE 3 Examples of spatio-temporal windows to collect data for noise filtering of the picture element (\mathbf{n}, k) .

The most simple order-statistic filter is a straightforward temporal median, for instance taken over three frames:

$$\hat{f}(\mathbf{n}, k) = \text{median}(g(\mathbf{n}, k - 1), g(\mathbf{n}, k), g(\mathbf{n}, k + 1)) \quad (14)$$

Filters of this type are very suitable for removing shot noise. In order to avoid artifacts at the edges of moving objects, Eq. (14) is normally applied in a motion-compensated way. A more elaborate OS-filter is the multi-stage median filter (MMF) [3, 4]. In the MMF the outputs of basic median filters with different spatio-temporal support are combined. An example of the spatio-temporal supports is shown in Fig. 4. The outputs of these intermediate median filter results are then combined as follows:

$$\begin{aligned} \hat{f}(\mathbf{n}, k) &= \text{median}\left(g(\mathbf{n}, k), \max(\hat{f}_1(\mathbf{n}, k), \dots, \hat{f}_9(\mathbf{n}, k)), \right. \\ &\quad \left. \min(\hat{f}_1(\mathbf{n}, k), \dots, \hat{f}_9(\mathbf{n}, k))\right) \end{aligned} \quad (15)$$

The advantage of this class of filters is that although it does not incorporate motion estimation explicitly, artifacts on edges of moving objects are significantly reduced. Nevertheless, the intermediate medians can also be computed in a motion compensated way by positioning the spatio-temporal windows in Fig. 4 along motion trajectories.

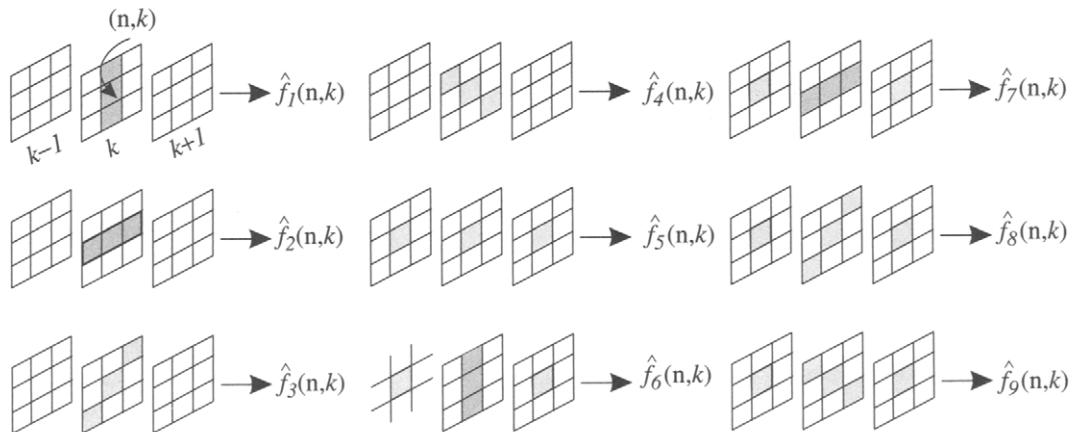


FIGURE 4 Spatio-temporal windows used in the multi-stage median filter.

The filter coefficients $h_{(r)}(\mathbf{n}, k)$ in (13) can also be statistically designed, as described in Chapter 4.4 of this *Handbook*. If the coefficients are optimized in the mean-squared error sense, the following general solution for the restored image sequence is obtained [21]:

$$\begin{aligned} \begin{pmatrix} \hat{f}(\mathbf{n}, k) \\ \hat{\sigma}_w^2(\mathbf{n}, k) \end{pmatrix} &= \begin{pmatrix} h_{(1)}(\mathbf{n}, k) & h_{(2)}(\mathbf{n}, k) \cdots h_{(|S|)}(\mathbf{n}, k) \\ n_{(1)}(\mathbf{n}, k) & n_{(2)}(\mathbf{n}, k) \cdots n_{(|S|)}(\mathbf{n}, k) \end{pmatrix} \begin{pmatrix} g_{(1)}(\mathbf{n}, k) \\ \vdots \\ g_{(|S|)}(\mathbf{n}, k) \end{pmatrix} \\ &= (\mathbf{A}' \mathbf{C}_{(w)}^{-1} \mathbf{A})^{-1} \mathbf{A}' \mathbf{C}_{(w)}^{-1} \begin{pmatrix} g_{(1)}(\mathbf{n}, k) \\ \vdots \\ g_{(|S|)}(\mathbf{n}, k) \end{pmatrix} \end{aligned} \quad (16a)$$

This expression formulates the optimal filter coefficients $h_{(r)}(\mathbf{n}, k)$ in terms of a matrix product involving the $|S| \times |S|$ auto-covariance matrix of the ranks of the noise, denoted by $\mathbf{C}_{(w)}$, and a matrix \mathbf{A} defined as

$$\mathbf{A} = \begin{pmatrix} 1 & E[w_{(1)}(\mathbf{n}, k)] \\ 1 & E[w_{(2)}(\mathbf{n}, k)] \\ \vdots & \vdots \\ 1 & E[w_{(|S|)}(\mathbf{n}, k)] \end{pmatrix} \quad (16b)$$

Here $E[w_{(r)}(\mathbf{n}, k)]$ denotes the expectation of the ranks of the noise. The result in (16a) not only gives an estimate of the filtered image sequence, but also for the local noise variance. This quantity is of use by itself in various noise filters to regulate the noise reduction strength. In order to calculate $E[w_{(r)}(\mathbf{n}, k)]$ and $\mathbf{C}_{(w)}$ the probability density function of the noise has to be assumed known. In case the noise $w(\mathbf{n}, k)$ is uniformly distributed, (16a) becomes the average of the minimum and maximum observed intensity. For Gaussian

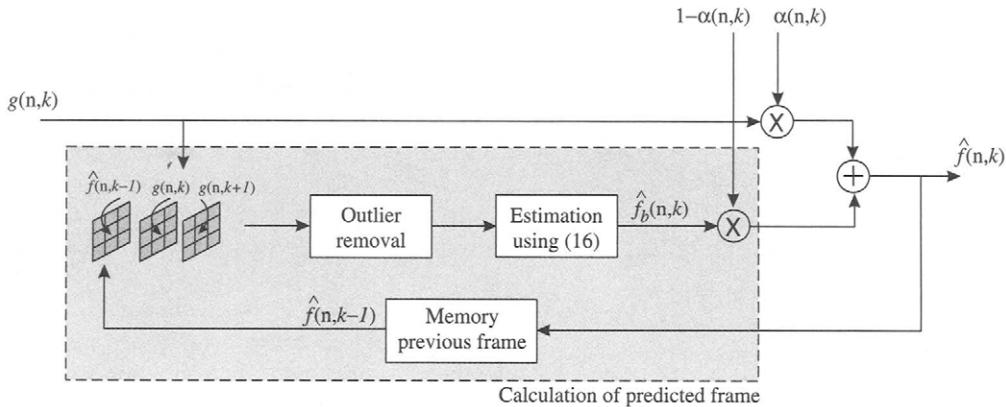


FIGURE 5 Overall filtering structure combining (9), (16), and an outlier removing rank order test.

distributed noise (16a) degenerates to (2) with equal weighting coefficients.

An additional advantage of ordering the noisy observation prior to filtering is that *outliers* can easily be detected. For instance, with a statistical test — such as the rank order test [21] — the observed noisy values within the spatio-temporal window S that are significantly different from the intensity $g(\mathbf{n}, k)$ can be detected. These significantly different values originate usually from different objects in the image sequence, for instance due to motion. By letting the statistical test reject these values, the filters (13) and (16) use locally only data from the observed noisy image sequence that is close — in intensity — to $g(\mathbf{n}, k)$. This further reduces the sensitivity of the noise filter (13) to outliers due to motion or incorrectly compensated motion.

The estimator (16) can also be used in a recursive structure such as the one in Equation (9). Essentially (16) is then interpreted as an estimate for the local mean of the image sequence, and the filtered value resulting from (16) is used as the predicted value $\hat{f}_b(\mathbf{n}, k)$ in (9). Furthermore, instead of using only noisy observations in the estimator, previously filtered frames can be used by extending the spatio-temporal window S over the current noisy frame $g(\mathbf{n}, k)$ and the previously filtered frame $\hat{f}(\mathbf{n}, k - 1)$. The overall filter structure is shown in Fig. 5.

2.3 Multi-Resolution Filters

The multi-resolution representation of 2D images has become quite popular for analysis and compression purposes [39]. This signal representation is also useful for image sequence restoration. The fundamental idea is that if an appropriate decomposition into bands of different spatial and temporal resolutions and orientations is carried out, the energy of the structured signal will locally be concentrated in selected bands whereas the noise is spread out over all bands. The noise can therefore effectively be removed by mapping all small (noise) components in all bands to zero, while leaving the remaining

larger components relatively unaffected. Such an operation on signals is also known as *coring* [13, 35]. Figure 6 shows two coring functions, namely soft- and hard-thresholding. Chapter 3.4 of this *Handbook* discusses 2D wavelet-based thresholding methods for image enhancement.

The discrete wavelet transform has been widely used for decomposing one- and multidimensional signals into bands. A problem with this transform for image sequence restoration is, however, that the decomposition is not shift-invariant. Slightly shifting the input image sequence in spatial or temporal sense can cause significantly different decomposition results. For this reason, in [38] a shift-invariant, but over-complete, decomposition was proposed known as the Simoncelli pyramid. Figure 7a shows the 2D Simoncelli pyramid decomposition scheme. The filters $L_i(\omega)$ and $H_i(\omega)$ are linear phase low- and high-pass filters, respectively. The filters $F_i(\omega)$ are fan-filters that decompose the signal into four directional bands. The resulting spectral decomposition is shown in Fig. 7b. From this spectral tessellation, the different resolutions and orientations of the spatial bands obtained by Fig. 7a can be inferred. The radial bands have a bandwidth of 1 octave.

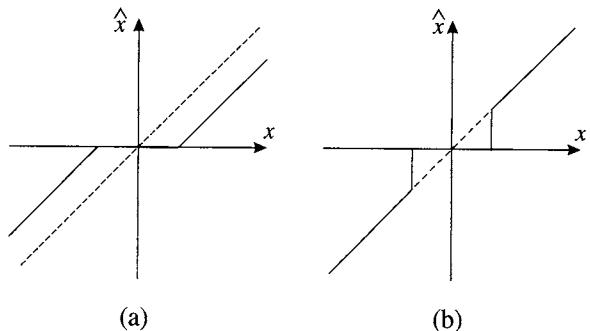


FIGURE 6 Coring functions: (a) Soft-thresholding; (b) hard-thresholding. Here x is a signal amplitude taken from one of the spatio-temporal bands (which carry different resolution and orientation information), and \hat{x} is the resulting signal amplitude after coring.

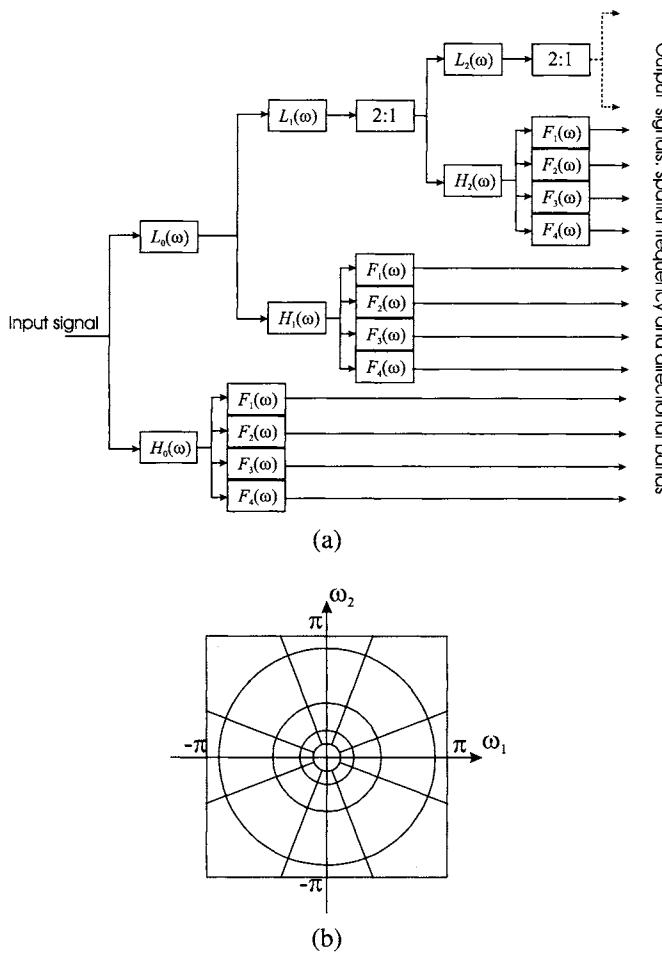


FIGURE 7 (a) Simoncelli pyramid decomposition scheme; (b) resulting spectral decomposition, illustrating the spectral contents carried by the different resolution and directional bands.

The Simoncelli pyramid gives a spatial decomposition of each frame into bands of different resolution and orientation. The extension to temporal dimension is obtained by temporally decomposing each of the spatial resolution and orientation bands using a regular wavelet transform. The low-pass

and high-pass filters are operated along the motion trajectory in order to avoid blurring of moving objects. The resulting motion-compensated spatio-temporal wavelet coefficients are filtered by one of the coring functions, followed by the reconstruction of the video frame by an inverse wavelet transformation and Simoncelli pyramid reconstruction. Figure 8 shows the overall scheme.

Though multi-resolution approaches have been shown to outperform the filtering techniques described in Section 2.1 and 2.2 for some types of noise, they generally require much more processing power due to the spatial and temporal decomposition, and — depending on the temporal wavelet decomposition — they require a significant number of frame stores.

3 Blotch Detection and Removal

Blotches are artifacts that are typically related to film. Dirt particles covering film introduce bright or dark spots on the frames, and the mishandling or aging of film causes loss of gelatin covering the film. Figure 11a shows a film frame containing dark and bright spots: the blotches. A model for this artifact is the following [24, 33]:

$$g(\mathbf{n}, k) = (1 - b(\mathbf{n}, k))f(\mathbf{n}, k) + b(\mathbf{n}, k)c(\mathbf{n}, k) + w(\mathbf{n}, k) \quad (17)$$

Here $b(\mathbf{n}, k)$ is a binary mask that indicates for each spatial location in each frame whether or not it is part of a blotch. The (more or less constant) intensity values at the corrupted spatial locations are given by $c(\mathbf{n}, k)$. Though noise is not considered to be the dominant degrading factor in the section, it is still included in (17) as the term $w(\mathbf{n}, k)$. The removal of blotches is a two-step procedure. In the first step, the blotches need to be detected, i.e., an estimate for the mask $b(\mathbf{n}, k)$ is made [22]. In the second step, the incorrect intensities $c(\mathbf{n}, k)$ at the corrupted locations are spatio-temporally interpolated [23]. In case a motion-compensated interpolation is carried

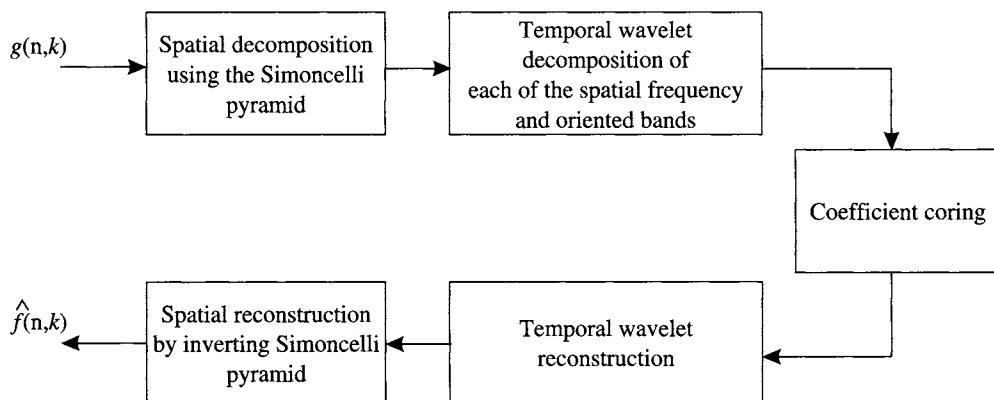


FIGURE 8 Overall spatio-temporal multi-resolution filtering using coring.

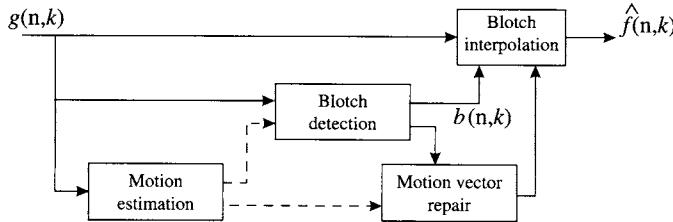


FIGURE 9 Blotch detection and removal system.

out, the second step also involves the local repair of motion vectors estimated from the blotched frames. The overall blotch detection and removal scheme is shown in Fig. 9.

3.1 Blotch Detection

Blotches have three characteristic properties that are exploited by blotch detection algorithms. In the first place, blotches are temporally independent and therefore hardly ever occur at the same spatial location in successive frames. Secondly, the intensity of a blotch is significantly different from its neighboring uncorrupted intensities. Finally, blotches form coherent regions in a frame, as opposed to, for instance, spatio-temporal shot noise.

There are various blotch detectors that exploit these characteristics. The first is a pixel-based blotch detector known as the spike-detector index (SDI). This method detects temporal discontinuities by comparing pixel intensities in the current frame with motion-compensated reference intensities in the previous and following frame:

$$\text{SDI}(\mathbf{n}, k) = \min \left((g(\mathbf{n}, k) - g(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k-1), k-1))^2, (g(\mathbf{n}, k) - g(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k+1), k+1))^2 \right) \quad (18)$$

Since blotch detectors are pixel-oriented, the motion field $\mathbf{d}(\mathbf{n}; k; l)$ should have a motion vector per pixel, i.e., the motion field is dense. Observe that any motion-compensation procedure must be robust against the presence of intensity spikes: this will be discussed later in this Section. A blotch pixel is detected if $\text{SDI}(\mathbf{n}, k)$ exceeds a threshold:

$$b(\mathbf{n}, k) = \begin{cases} 1 & \text{if } \text{SDI}(\mathbf{n}, k) > T \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

Since blotch detectors are essentially searching for outliers, order-statistic based detectors usually perform better. The rank-order difference (ROD) detector is one such method. It takes $|S|$ reference pixel intensities: $\mathbf{r} = \{r_i \mid i = 1, 2, \dots, |S|\}$ from a motion-compensated spatio-temporal window S (see for instance the greyed pixels in Fig. 10), and ranks them by

intensity value: $r_1 \leq r_2 \leq \dots \leq r_{|S|}$. It then finds the deviation between the pixel intensity $g(\mathbf{n}, k)$ and the reference pixels ranked by intensity value as follows:

$$\text{ROD}_i(\mathbf{n}, k) = \begin{cases} r_i - g(\mathbf{n}, k) & \text{if } g(\mathbf{n}, k) \leq \text{median}(\mathbf{r}) \\ g(\mathbf{n}, k) - r_{|S|-i} & \text{if } g(\mathbf{n}, k) > \text{median}(\mathbf{r}) \end{cases} \quad \text{for } i = 1, 2, \dots, \frac{|S|}{2} \quad (20)$$

A blotch pixel is detected if any of the rank order differences exceeds a specific threshold T_i :

$$b(\mathbf{n}, k) = \begin{cases} 1 & \text{if } \exists i \text{ such that } \text{ROD}_i(\mathbf{n}, k) > T_i \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

More complicated blotch detectors explicitly incorporate a model for the uncorrupted frames, such as a two- or three-dimensional autoregressive model or a Markov Random Field to develop the maximum *a posteriori* detector for the blotch mask $b(\mathbf{n}, k)$. Figure 11b illustrates the detection probability versus the false detection probability of three different detectors on a sequence of which a representative blotched frame is shown in Fig. 11a. These results indicate that for reasonable detection probabilities the false detection probability is fairly high. False detections are detrimental to the restoration process because the interpolation process itself is fallible and may introduce disturbing artifacts that were not present in the blotched image sequence.

The blotch detectors described so far are essentially pixel-based detectors. They do not incorporate the spatial coherency of the detected blotches. The effect is illustrated by Fig. 11c which shows the detected blotch mask $b(\mathbf{n}, k)$ using a simplified version of the ROD detector (with $T_i \rightarrow \infty$, $i \geq 2$). The simplified ROD (sROD) is given by the following

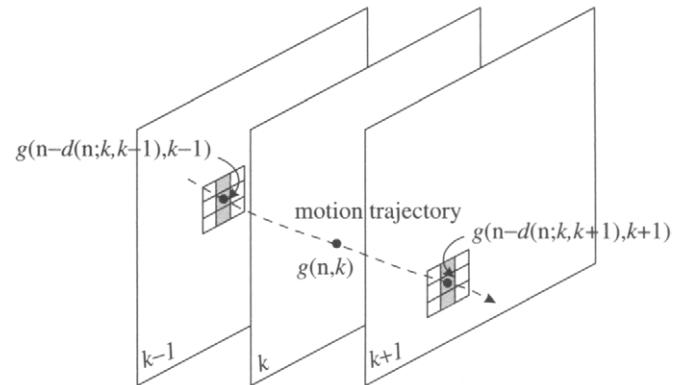


FIGURE 10 Example of motion-compensated spatio-temporal window for obtaining reference intensities in the ROD detector.

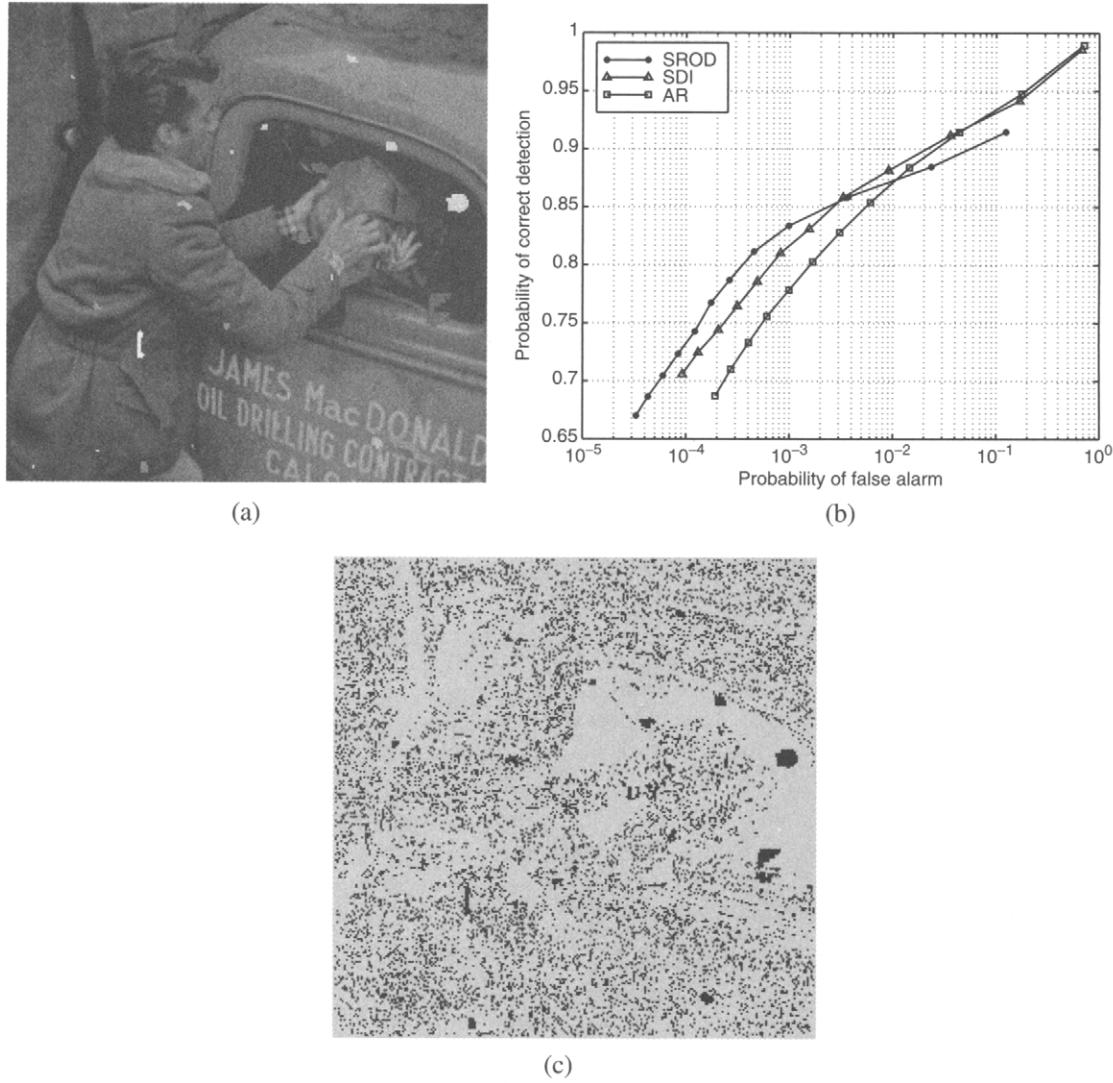


FIGURE 11 (a) Video frame with blotches; (b) correct detection versus false detection for three different blotch detectors; (c) blotch detection mask using the sROD ($T=0$).

relations [33]:

$$\text{sROD}(\mathbf{n}, k) = \begin{cases} \min(\mathbf{r}) - g(\mathbf{n}, k) & \text{if } g(\mathbf{n}, k) < \min(\mathbf{r}) \\ g(\mathbf{n}, k) - \max(\mathbf{r}) & \text{if } g(\mathbf{n}, k) > \max(\mathbf{r}) \\ 0 & \text{elsewhere} \end{cases} \quad (22a)$$

$$b(\mathbf{n}, k) = \begin{cases} 1 & \text{if } \text{sROD}(\mathbf{n}, k) > T \\ 0 & \text{otherwise} \end{cases} \quad (22b)$$

The sROD basically looks at the range of the reference pixel intensities obtained from the motion-compensated window,

and compares it with the pixel intensity under investigation. A blotch pixel is detected if the intensity of the current pixel $g(\mathbf{n}, k)$ lies far enough outside that range.

The performance of even this simple pixel-based blotch detector can be improved significantly by exploiting the spatial coherence of blotches. This is done by postprocessing the blotch mask in Fig. 11c in two ways, namely by removing small blotches, and by completing partially detected blotches. We first discuss the removal of small blotches. The detector output (22a) is not only sensitive to intensity changes due to blotches corrupting the image sequence, but also to noise. If the probability density function of the noise — denoted by $f_W(w)$ — is known, the probability of false detection for a single pixel can be calculated. Namely, if the sROD uses $|S|$ reference intensities in evaluating (22a), the probability that

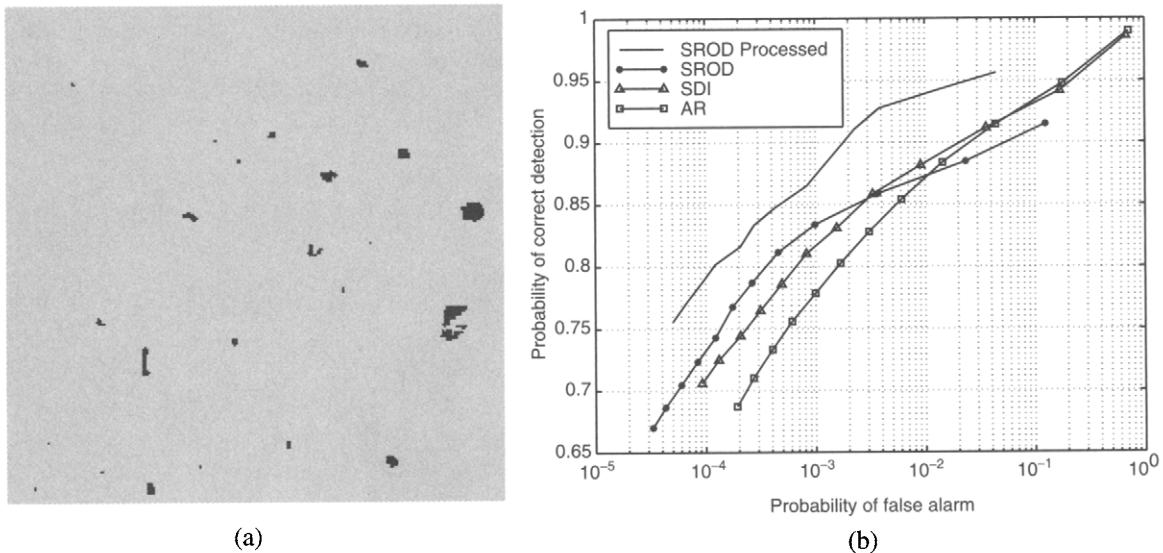


FIGURE 12 (a) Blotch detection mask after postprocessing; (b) correct detection versus false detections obtained for sROD with postprocessing (top curve), compared to results from Fig. 11b.

$s\text{ROD}(\mathbf{n}, k)$ for a single pixel is larger than T due to noise only is [33]:

$$\begin{aligned} P(s\text{ROD}(\mathbf{n}, k) > T | \text{no blotch}) \\ &= P(g(\mathbf{n}, k) - \max(\mathbf{r}) > T | \text{no bl.}) \\ &\quad + P(\min(\mathbf{r}) - g(\mathbf{n}, k) > T | \text{no bl.}) \\ &= 2 \int_{-\infty}^{\infty} \left[\int_{-\infty}^{u-T} f_w(w) dw \right]^S f_w(u) du \end{aligned} \quad (23)$$

In the detection mask $b(\mathbf{n}, k)$ blotches may consist of single pixels or of multiple connected pixels. A set of connected pixels that are all detected as (being part of a) blotch, is called a spatially coherent blotch. If a coherent blotch consists of N connected pixels, the probability that this blotch is due to noise only is

$$\begin{aligned} P(s\text{ROD}(\mathbf{n}, k) > T \text{ for } N \text{ connected pixels} | \text{no blotch}) \\ &= (P(s\text{ROD}(\mathbf{n}, k) > T | \text{no blotch}))^N \end{aligned} \quad (24)$$

By bounding this false detection probability to a certain maximum, the minimum number of pixels identified by the sROD detector as being part of a blotch can be computed. Consequently, coherent blotches consisting of fewer pixels than this minimum are removed from the blotch mask $b(\mathbf{n}, k)$.

A second postprocessing technique for improving the detector performance is hysteresis thresholding. First a blotch mask is computed using a very low detection threshold T , for instance $T = 0$. From the detection mask the small blotches

are removed as described above, yielding the mask $b_0(\mathbf{n}, k)$. Nevertheless, due to the low detection threshold this mask still contains many false detections. Then a second detection mask $b_1(\mathbf{n}, k)$ is obtained by using a much higher detection threshold. This mask contains fewer detected blotches and the false detection rate in this mask is small. The second detection mask is now used to validate the detected blotches in the first mask: only those spatially coherent blotches in $b_0(\mathbf{n}, k)$ that have a corresponding blotch in $b_1(\mathbf{n}, k)$ are preserved, all others are removed. The result of the above two postprocessing techniques on the frame shown in Fig. 11a is shown in Fig. 12a. In Fig. 12b the detection and false detection probabilities are shown.

3.2 Motion Vector Repair and Interpolating Corrupted Intensities

Block-based motion estimators will generally find the correct motion vectors even in the presence of blotches, provided that the blotches are small enough. The disturbing effect of blotches is usually confined to small areas of the frames. Hierarchical motion estimators will experience little influence of the blotches at the lower resolution levels. At higher resolution levels blotches covering larger parts of (at those levels) small blocks will significantly influence the motion estimation result. If the blotch mask $b(\mathbf{n}, k)$ has been estimated, it is also known which estimated motion vectors are unreliable.

There are two strategies in recovering motion vectors that are known to be unreliable. The first approach is to take an average of surrounding motion vectors. This process — known as *motion vector interpolation* or *motion vector repair* — can be realized using, for instance, the median or average

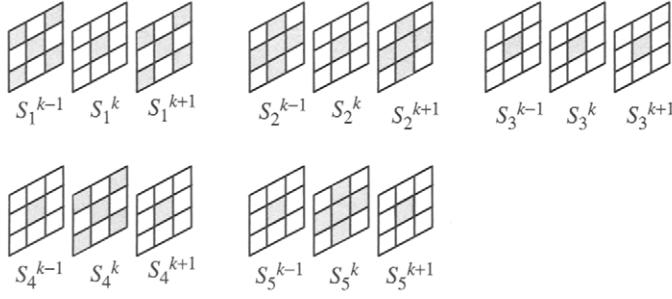


FIGURE 13 Five spatio-temporal windows used to compute the partial results in Eq. (25).

of the motion vectors of uncorrupted regions adjacent to the corrupted blotch. Though simple, the disadvantages of averaging are that motion vectors may be created that are not present in the uncorrupted part of the image and that no validation of the selected motion vector on the actual frame intensities takes place.

The second — more elaborate — approach circumvents this disadvantage by validating the corrected motion vectors using intensity information directly neighboring the blotched area. As a validation criterion the motion-compensated mean squared intensity difference can be used [11]. Candidates for the corrected motion vector can be obtained either from motion vectors taken from adjacent regions or by motion re-estimation using a spatial window containing only uncorrupted data such as the pixels directly bordering the blotch.

The estimation of the frame intensities labeled by the mask as being part of a blotch can be done either by a spatial or temporal interpolation, or a combination of both. We concentrate on spatio-temporal interpolation. Once the motion vector for a blotched area has been repaired, the correct temporally neighboring intensities can be obtained. In a multi-stage median interpolation filter, five interpolated results are computed using the (motion-compensated) spatio-temporal neighborhoods shown in Fig. 13. Each of the five interpolated results is computed as the median over the corresponding neighborhood \$S_i\$:

$$\hat{f}_i(\mathbf{n}, k) = \text{median}\left(\{f(\mathbf{n}, k-1) | \mathbf{n} \in S_i^{k-1}\} \cup \{f(\mathbf{n}, k) | \mathbf{n} \in S_i^k\} \cup \{f(\mathbf{n}, k+1) | \mathbf{n} \in S_i^{k+1}\}\right) \quad (25)$$

The final result is computed as the median over the five intermediate results:

$$\hat{f}(\mathbf{n}, k) = \text{median}(\hat{f}_1(\mathbf{n}, k), \hat{f}_2(\mathbf{n}, k), \hat{f}_3(\mathbf{n}, k), \hat{f}_4(\mathbf{n}, k), \hat{f}_5(\mathbf{n}, k)) \quad (26)$$

The multi-stage median filter does not rely on any model for the image sequence. Though simple, this is at the same time a drawback of median filters. If a model for the original image

sequence can be assumed, it is possible to find statistically optimal values for the missing intensities. For the sake of completeness we mention here that if one assumes the popular Markov Random Field, the following complicated expression needs to be optimized:

$$\begin{aligned} P(\hat{f}(\mathbf{n}, k) | f(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k-1), k-1), f(\mathbf{n}, k), \\ f(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k+1), k+1)) \propto \exp\left(-\sum_{\mathbf{m}: d(\mathbf{m}, k)=1} \gamma(\mathbf{m})\right), \\ \gamma(\mathbf{m}) = \sum_{\mathbf{s} \in S^k} (\hat{f}(\mathbf{m}, k) - \hat{f}(\mathbf{s}, k))^2 \\ + \lambda \left(\sum_{\mathbf{s} \in S^{k-1}} (\hat{f}(\mathbf{m}, k) - f(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k-1), k-1))^2 \right. \\ \left. + \sum_{\mathbf{s} \in S^{k+1}} (\hat{f}(\mathbf{m}, k) - f(\mathbf{n} - \mathbf{d}(\mathbf{n}; k, k+1), k+1))^2 \right) \end{aligned} \quad (27)$$

The first term of \$\gamma\$ on the right hand side of (27) forces the interpolated intensities to be spatially smooth, while the second and third term enforce temporal smoothness. The sets \$S^{k-1}\$, \$S^k\$ and \$S^{k+1}\$ denote appropriately chosen spatial windows in the frames \$k-1\$, \$k\$ and \$k+1\$. The temporal smoothness is calculated along the motion trajectory using the repaired motion vectors. The optimization of (27) requires an *iterative optimization technique*. If a simpler 3D autoregressive model for the image sequence is assumed, the interpolated values can be calculated by solving a set of linear equations.

Instead of interpolating the corrupted intensities, it is also possible to directly copy and paste intensities from past or future frames. The simple copy-and-paste operation instead of a full spatio-temporal data regeneration is motivated by the observation that — at least on local and motion-compensated basis — image sequences are heavily correlated. Furthermore, straightforward interpolation is not desirable in situations where part of the information in the past and future frames itself is unreliable, for instance if it was part of a blotch itself or if it is situated in an occluded area. The objective is now to determine — for each pixel being part of a detected blotch — if intensity information from the previous or next frame should be used. This decision procedure can again be cast into a statistical framework [27, 33]. As an illustration, Fig. 14 shows the interpolated result of the blotched frame in Fig. 11a.

3.3 Restoration in Conditions of Difficult Object Motion

Due to various types of complicated object movements [31], wrong motion vectors are sometimes extracted from the



FIGURE 14 Blotch-corrected frame resulting from Fig. 11a.

sequence. As a result, the spatio-temporal restoration process that follows may introduce unnecessary errors that are visually more disturbing than the blotches themselves. The extracted temporal information becomes unreliable, and a source of errors by itself. This triggered research on situations in which the motion vectors cannot be used.

An initial solution to the problem was to detect areas of “pathological” motion and protect them against any restoration [31]. This solution, however, preserves the artifacts which happen to lie in those areas. To restore these artifacts, too, several solutions have been proposed which discard the temporal information and use only spatial information coming from the same frame. Spatial restoration is currently the subject of intensive research [6, 8, 15, 19, 32] and has proven to be a powerful alternative to temporal or spatio-temporal restoration.

Spatial restoration algorithms can be classified mainly in two categories: smoothness-preserving ones, and texture synthesis ones. The first category comprises various inpainting approaches which usually propagate isophotes (i.e., level lines), gradients or curvatures inside the artifact by means of variational models [5, 6, 10, 28], or approaches which reconstruct the explicit structure of objects based on edge reconnection [32]. In the second category, parametric and non-parametric approaches are employed for texture synthesis based on Markov Random Fields [8, 12, 15], Bayesian auto-regressive models [26], projection onto convex sets [17] and other models [1].

As their naming suggests, texture synthesis methods do a better reconstruction of the textural content. However, they do not preserve object edges properly. This is done better with smoothness-preserving methods, which, on their turn, do not reproduce textural content. In order to combine the advantages of both methods, a third category of algorithms

has appeared comprising hybrid methods that combine structure and texture reconstruction [7, 19, 30, 32].

Figure 15 presents an example of an artificially degraded image which was restored by means of a hybrid method. The restoration consisted of a texture synthesis constrained by edge-based structure reconstruction [32]. The main steps are: (1) edge detection and feature extraction; (2) image structure reconstruction; and (3) texture synthesis. In the first step, edges are detected around the artifact, based on the contours that result from a segmentation procedure. Features are extracted for each edge, such as: histograms of the luminance and gradient angles on both sides of the edge. In the second step, the missing object edges within the artifact area are recovered by pairing the edges extracted around the artifact. The pairing procedure assesses the similarity of the paired edges based on the features extracted in the previous step; on an estimate of how well one edge continuous into another one (assuming, for example, local circularity); and on the spatial order of the edges with respect to each other. All these factors contribute to a global consistency score of the resulting structure, and the best scoring configuration is chosen to represent the final image structure. The edges which do not belong to any pair are considered to be T-junctions (coming from edge occlusions). Finally, in the third step, the artifact is restored by texture synthesis, confined by the structure reconstructed in the previous step.

In image sequences, this restoration method can be automatically switched on to repair single frames spatially when the extracted temporal information is not reliable. This is realized by means of a detector of complex events that supervises the output of the motion estimator. The task of the complex event detector is actually twofold: besides detecting the frame areas where the motion estimator fails, it has to discover artifacts which may lie in such areas. The latter task is performed by means of spatial algorithms, since the common artifact detectors would fail due to the unreliability of the temporal information [31, 32].

4 Vinegar Syndrome Removal

Content stored on acetate-based film rolls faces deterioration at a progressive pace. These rolls can be affected by dozens of types of film artifacts, each of them having its own properties and showing up in particular circumstances. One of the major problems of all film archives is the vinegar syndrome [32]. This syndrome appears when, in the course of their chemical breakdown, the acetate-based film bases start to release acetic-acid, giving a characteristic vinegar smell. It is an irreversible process, and from a certain moment on, it becomes auto-catalytic, progressively fuelling itself in the course of time.

The vinegar syndrome has various appearances. It may show up as a partial loss of color, bright or dark tree-like

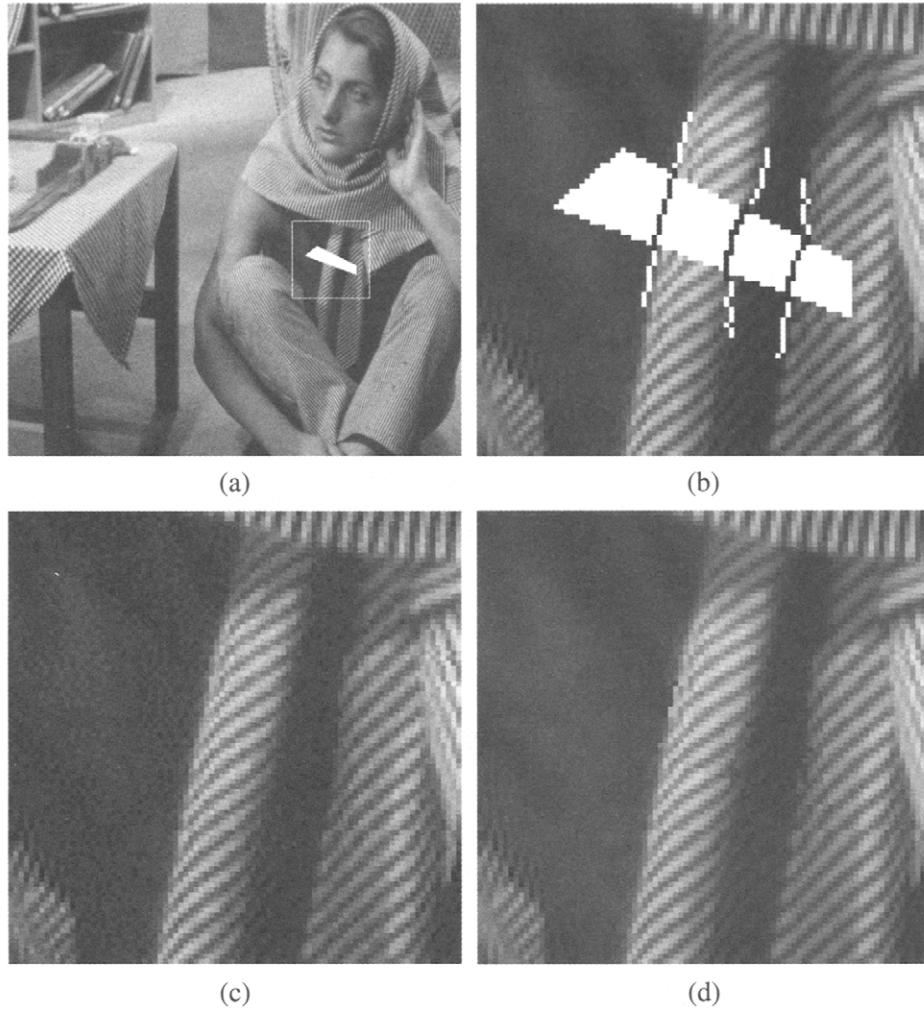


FIGURE 15 Constrained texture synthesis. (a) Original image, with artificial artifact surrounded by a white box; (b) structure; (c) original content of the artifact; (d) restoration result.

branches, non-uniformly blurred images, etc [32]. Here, we only focus on one type of vinegar syndrome, namely the partial loss of color. This manifests itself as a localized total loss of information in some of the film dye layers (e.g., *green* and *blue*). Thanks to the sandwiched structure of the film (sketched in Fig. 16), the inner layer (*red*) is more protected and still preserves some of the original information. This type of vinegar syndrome may be accompanied by emulsion melting (a dissolution of the dye layers). Figure 18a shows an example of vinegar syndrome combined with emulsion melting.

The detection and correction of this type of vinegar syndrome can be performed with the normal spatio-temporal techniques, as long as it has a temporally impulsive appearance, like normal blotches. However, there are cases when the vinegar syndrome happens in areas of pathological motion, or when it appears in the same place in consecutive frames. In both cases, the information from the surrounding frames is of little use and one has to rely only on the

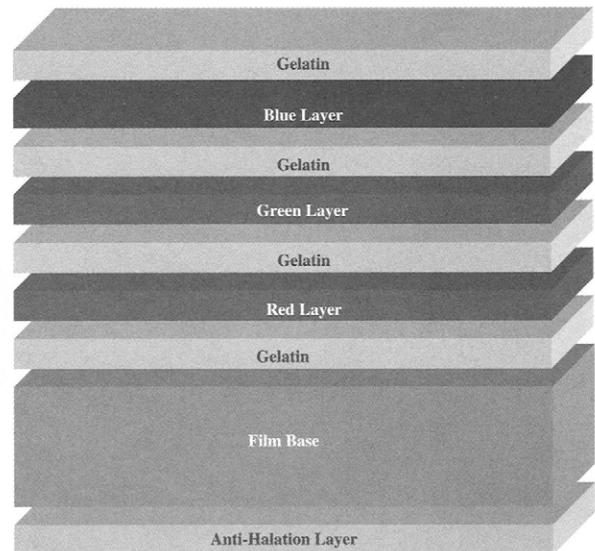


FIGURE 16 The layered structure of a film. (See color insert.)

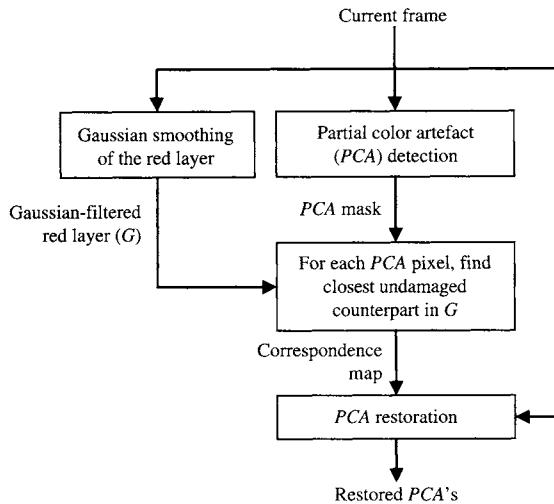


FIGURE 17 Restoration scheme for the vinegar syndrome.

information from the current frame which lies in and around the artifact. Moreover, the blotch removal algorithms do not exploit the information that remains inside the artifact, in the red layer.

For the detection step, one may use the fact that the loss of color makes the artifact look very bright mainly in the green and blue layers [32]. In order to avoid the influence of irregularities inside the artifact, a hysteresis thresholding is performed as described in section 3.1 of this chapter, with different thresholds for each color layer and in each of the two thresholding phases. In the end, a conditional dilation makes sure that the dark border surrounding the artifact is also added to the artifact mask.

As opposed to conventional spatial restoration techniques, in the restoration step under consideration one may use the information that still exists inside the artifact in the red layer [32]. This information may also be partially affected by the artifact, so a gaussian smoothing is first performed on the red layer. The isophotes that result from this smoothing are presumed to follow the shapes of the the isophotes from the original, undegraded image. These isophotes are then used to “propagate” information inside the artifact from its surrounding area. The general scheme is presented in Fig. 17.

An artifact pixel p^i with smoothed red value r_s^i will be overwritten with the non-smoothed RGB values of a pixel $p^k = (r^k, g^k, b^k)$ lying in the neighborhood of the artifact and representing the “closest” pixel to p^i , (in terms of physical distance and color resemblance). Pixel p^k is selected from a set of pixels representing the connected neighborhood of pixel p^i and having values in $[r_s^i - \Delta_s \dots r_s^i + \Delta_s]$. p^k is found using Eq. (28):

$$k = \arg \min_j \sqrt{\left(r_s^i - r_s^j\right)^2 + \left(\frac{d(p^i, p^j)}{d(p^i, p^j) + 1}\right)^2} \quad (28)$$

where $d(p^i, p^j)$ is the physical distance between pixels p^i and p^j , and is normalized to values in $[0 \dots 1]$ in order to have the same range as the color distance (all pixel values are in the $[0 \dots 1]$ interval).

Figure 18 presents the recovery of portions of a film sequence that were presumed to have been completely lost. A value $\Delta_s \approx 4\%$ of the overall greyvalue range was used for the connected neighborhood calculation, and a standard deviation $\sigma = 7$ for the gaussian smoothing step. The result largely reflects the image structure that has been recovered in the smoothed red layer.

5 Intensity Flicker Correction

Intensity flicker is defined as unnatural temporal fluctuations of frame intensities that do not originate from the original scene. Intensity flicker is a spatially localized effect that occurs in regions of substantial size. Figure 19 shows three successive frames from a sequence containing flicker. A model describing the intensity flicker is the following

$$g(\mathbf{n}, k) = \alpha(\mathbf{n}, k)f(\mathbf{n}, k) + \beta(\mathbf{n}, k) + w(\mathbf{n}, k) \quad (29)$$

Here, $\alpha(\mathbf{n}, k)$ and $\beta(\mathbf{n}, k)$ are the multiplicative and additive *unknown* flicker parameters, that locally scale the intensities of the original frame. The model includes a noise term $w(\mathbf{n}, k)$ that is assumed to be flicker-independent. In the absence of flicker we have $\alpha(\mathbf{n}, k) = 1$ and $\beta(\mathbf{n}, k) = 0$. The objective of flicker correction is the estimation of the flicker parameters, followed by the inversion of Equation (29). Since flicker always affects fairly large areas of a frame in the same way, the flicker parameters $\alpha(\mathbf{n}, k)$ and $\beta(\mathbf{n}, k)$ are assumed to be spatially smooth functions. Temporally the flicker parameters in one frame may not be correlated at all with those in a subsequent frame.

The earliest attempts to remove flicker from image sequences applied intensity histogram equalization or mean-equalization on frames. These methods do not form a general solution to the problem of intensity flicker correction because they ignore changes in scene contents, and do not appreciate that intensity flicker is a localized effect. In Section 5.1 we show how the flicker parameters can be estimated on stationary image sequences. Section 5.2 addresses the more realistic case of parameter estimation on image sequences with motion [34].

5.1 Flicker Parameter Estimation

When removing intensity flicker from an image sequence, we essentially make an estimate of the original intensities, given the observed image sequence. Note that the undoing of intensity flicker is only relevant for image sequences, since flicker is a temporal effect by definition. From a single frame intensity flicker cannot be observed nor be corrected.

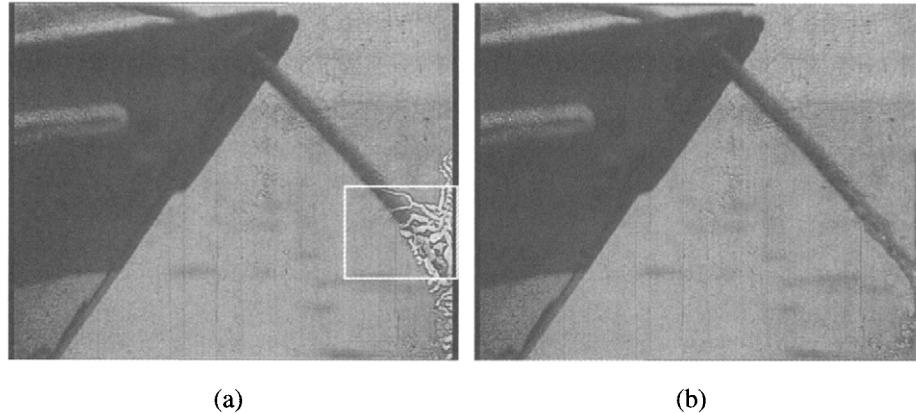


FIGURE 18 Restoration example (sequence courtesy of RTP — Radiotelevisão Portuguesa). (a) Original frame, with artifact surrounded by a white box; (b) restored frame. (See color insert.)

If the flicker parameters were known, then one can form an estimate of the original intensity from a corrupted intensity using the following straightforward linear estimator:

$$\hat{f}(\mathbf{n}, k) = h_1(\mathbf{n}, k)g(\mathbf{n}, k) + h_0(\mathbf{n}, k) \quad (30)$$

In order to obtain estimates for the coefficients $h_i(\mathbf{n}, k)$, the mean-squared error between $f(\mathbf{n}, k)$ and $\hat{f}(\mathbf{n}, k)$ is minimized, yielding the following optimal solution:

$$h_0(\mathbf{n}, k) = -\frac{1}{\alpha(\mathbf{n}, k)} \left(\beta(\mathbf{n}, k) + \frac{\sigma_w^2(\mathbf{n}, k)}{\sigma_g^2(\mathbf{n}, k)} E[g(\mathbf{n}, k)] \right) \quad (31a)$$

$$h_1(\mathbf{n}, k) = \frac{1}{\alpha(\mathbf{n}, k)} \frac{\sigma_g^2(\mathbf{n}, k) - \sigma_w^2(\mathbf{n}, k)}{\sigma_g^2(\mathbf{n}, k)} \quad (31b)$$

If the observed image sequence does not contain any noise, then (31) degenerates to the obvious solution

$$h_0(\mathbf{n}, k) = -\frac{\beta(\mathbf{n}, k)}{\alpha(\mathbf{n}, k)} \quad h_1(\mathbf{n}, k) = \frac{1}{\alpha(\mathbf{n}, k)} \quad (32)$$

In the extreme situation that the variance of the corrupted image sequence is equal to the noise variance, the combination

of (30) and (31) shows that the estimated intensity is equal to the expected value of the original intensities $E[f(\mathbf{n}, k)]$.

In practice, the true values for the intensity-flicker parameters $\alpha(\mathbf{n}, k)$ and $\beta(\mathbf{n}, k)$ are unknown and need to be estimated from the corrupted image sequence itself. Since the flicker parameters are spatially smooth functions, we assume that they are locally constant:

$$\begin{cases} \alpha(\mathbf{n}, k) = \alpha_m(k) \\ \beta(\mathbf{n}, k) = \beta_m(k) \end{cases} \quad \forall \mathbf{n} \in S_m \quad (33)$$

where S_m indicates a small frame region. This region can, in principle, be arbitrarily shaped, but in practice rectangular blocks are chosen. By computing the averages and variances of both sides of Equation (29), the following analytical expressions for the estimates of $\alpha_m(k)$ and $\beta_m(k)$ can be obtained:

$$\hat{\alpha}_m(k) = \sqrt{\frac{\sigma_g^2(\mathbf{n}, k) - \sigma_w^2(\mathbf{n}, k)}{\sigma_f^2(\mathbf{n}, k)}} \quad (34)$$

$$\hat{\beta}_m(k) = E[g(\mathbf{n}, k)] - \hat{\alpha}_m(k)E[f(\mathbf{n}, k)]$$

To solve (34) in a practical situation, the mean and variance of $g(\mathbf{n}, k)$ are estimated within the region S_m . The only

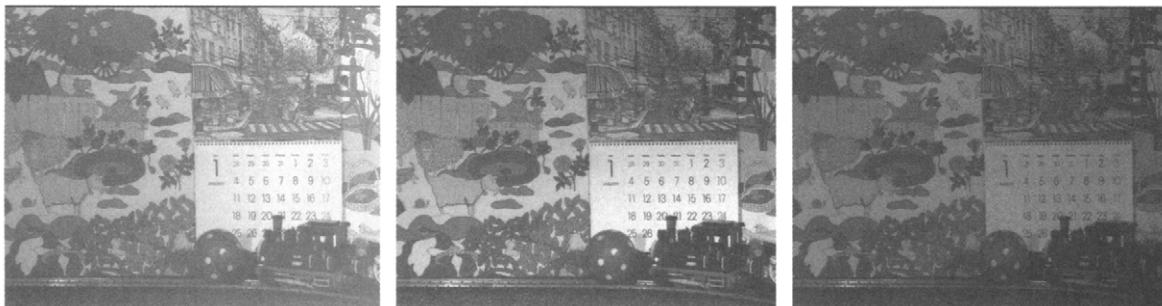


FIGURE 19 Three successive frames that contain intensity flicker.

quantities that remain to be estimated are the mean and variance of the original image sequence $f(\mathbf{n}, k)$. If we assume that the flicker correction is done frame-by-frame, we can estimate these values from the previous corrected frame $k-1$ in the temporally corresponding frame region S_m :

$$\begin{aligned} E[f(\mathbf{n}, k)] &\approx \frac{1}{|S_m|} \sum_{\mathbf{m} \in S_m} \hat{f}(\mathbf{m}, k-1) \\ \sigma_f^2(\mathbf{n}, k) &\approx \frac{1}{|S_m|} \sum_{\mathbf{m} \in S_m} (\hat{f}(\mathbf{m}, k-1) - E[f(\mathbf{n}, k)])^2 \end{aligned} \quad (35)$$

There are situations in which the above estimates are unreliable. The first case is that of uniform intensity areas. For any original image intensity in a uniform region, there are an infinite number of combinations of $\alpha_m(k)$ and $\beta_m(k)$ that lead to the observed intensity. The estimated flicker parameters are also potentially unreliable because of ignoring the noise $w(\mathbf{n}, k)$ in (34) and (35). The reliability of the estimated flicker parameters can be assessed by the following measure:

$$W_m(k) = \begin{cases} 0 & \text{if } \sigma_g^2(\mathbf{m}, k) < T \\ \sqrt{\frac{\sigma_g^2(\mathbf{m}, k) - T}{T}} & \text{otherwise} \end{cases} \quad (36)$$

The threshold T depends on the noise variance. Large values of $W_m(k)$ indicate reliable estimates, while for the most unreliable estimates $W_m(k) = 0$.

5.2 Estimation on Sequences with Motion

The results (34) and (35) assume that the image sequence intensities do not change significantly over time. Clearly this is an incorrect assumption if motion occurs. The estimation of motion on image sequences that contain flicker is, however, problematic because virtually all motion estimators are based on the constant luminance constraint. Because of the intensity flicker this assumption is violated heavily. The only motion that can be estimated with sufficient reliability is global motion such as camera panning or zooming. In the following we assume that in the evaluation of (35) and (36) possible global motion is compensated for. At that point we still need to detect areas with any remaining — and uncompensated — motion, and areas that were previously occluded. For both of these cases the approximation in (35) leads to incorrect estimates, which in turn lead to visible artifacts in the corrected frames.

There are various approaches for detecting local motion. One possibility is the detection of large differences between the current and previously (corrected) frame. If local motion occurs the frame differences will be large. Another possibility to detect local motion is to compare the estimated intensity-flicker parameters to threshold values.

If disagreeing temporal information has been used for computing (35), we will locally find flicker parameters that do not correspond with its spatial neighbors or with the *a priori* expectations of the range of the flicker parameters. An outlier detector can be used to localize these incorrectly estimated parameters.

For frame regions S_m where the flicker parameters could not be estimated reliably from the observed image sequence, the parameters are estimated on the basis of the results in spatially neighboring regions. At the same time, for the regions in which the flicker parameters could be estimated, a smoothing post-processing step needs to be applied to avoid sudden parameter changes that lead to visible artifacts in the corrected image sequence. Such an interpolation and smoothing post-processing step may exploit the reliability of the estimated parameters, as for instance given by Equation (36). Furthermore, in those frame regions where insufficient information was available for reliably estimating the flicker parameters, the flicker correction should switch-off itself. Therefore, smoothed and interpolated parameters are biased toward $\alpha_m(k) = 1$ and $\beta_m(k) = 0$.

In Figure 20, an example of smoothing and interpolating the estimated flicker parameter for $\alpha_m(k)$ is shown as a 2D matrix [34]. Each entry in this matrix corresponds to a 30×30 pixels region Ω_m in the frame shown in Fig. 19. The interpolation technique used is successive over-relaxation (SOR). Successive over-relaxation is a well-known iterative interpolation technique based on repeated low-pass filtering. Starting off with an initial estimate $\alpha_m^0(k)$ found by solving (34), at each iteration a new estimate is formed as follows:

$$\begin{aligned} r_m^{i+1}(k) &= W_m(k)(\alpha_m^i(k) - \alpha_m^0(k)) + \lambda C(\alpha_m^i(k)) \\ \alpha_m^{i+1}(k) &= \alpha_m^i(k) + \omega \frac{r_m^{i+1}(k)}{W_m(k) + \lambda} \end{aligned} \quad (37)$$

Here $W_m(k)$ is the reliability measure, computed by (36), and $C(\alpha_m(k))$ is a function that measures the spatial smoothness of the solution $\alpha_m(k)$. The convergence of the iteration (37) is determined by the parameter ω , while the smoothness is determined by the parameter λ . For those estimates that have a high reliability, the initial estimates $\alpha_m^0(k)$ are emphasized, while for the initial estimates that are deemed less reliable, i.e., $\lambda \gg W_m(k)$, emphasis is on achieving a smooth solution. Other smoothing and interpolation techniques include dilation and 2D polynomial interpolation. The smoothing and interpolation needs to be applied not only to multiplicative parameter $\alpha_m(k)$, but also to the additive parameter $\beta_m(k)$.

As an example, Fig. 21 shows the mean and variance as a function of the frame index k of the corrupted and corrected image sequence “Tunnel”. Clearly the temporal fluctuations of

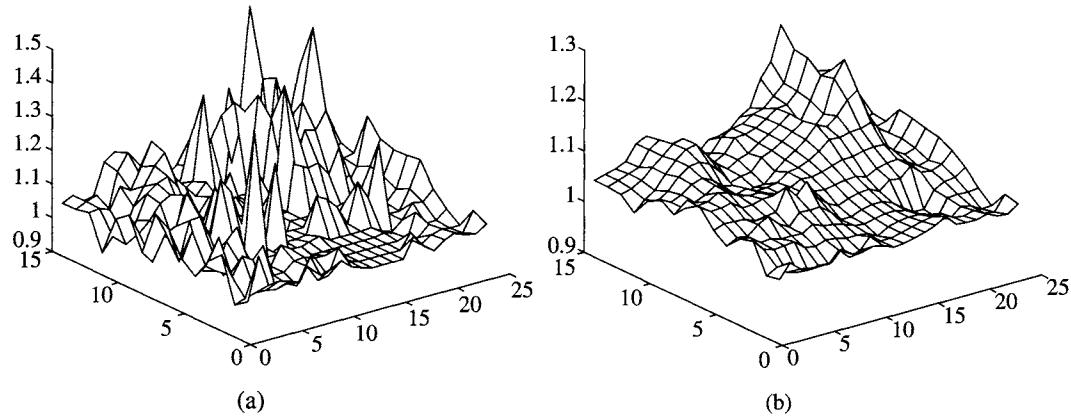


FIGURE 20 (a) Estimated intensity flicker parameter $\alpha_m(k)$ using (34) and local motion detection; (b) smoothed and interpolated $\alpha_m(k)$ using SOR.

the mean and variance have been greatly reduced, indicating the suppression of flicker artifacts. An assessment of the resulting visual quality, as with most results of video processing algorithms, has been done by actually viewing the corrected image sequences. Although the original sequence cannot be recovered, the flicker-corrected sequences have a much higher visual quality and they are virtually without any remaining visible flicker.

6 Kinescope Moiré Removal

The moiré phenomenon represents an interference of (semi-)periodical structures which gives rise to patterns that did not exist in the original signal. It appears in scanned or printed images [2], in video sequences [16, 37] etc. Moiré phenomena are caused in general by the sampling process (due to the aliasing phenomenon), or a superposition of structures. Kinescope moiré is caused by the process of transferring old TV archives to magnetic tapes. In the early times of the television, broadcasted programs were stored on films that

were recorded directly from a TV monitor. The scan lines of the monitor were thus recorded as well. The transfer from film to magnetic tapes took place years later by means of a Telecine device. This machine scans the film with a flying spot that moves in horizontal lines. These scan lines and those of the initial TV monitor do not always coincide, giving rise to an aliasing phenomenon. The resulting moiré shows up as a beating pattern of horizontal lines (see Fig. 23a). Because the initial TV monitors were not always flat, some curvilinear distortion also takes place in the recorded signal, which further complicates the modeling and removal of the moiré.

Currently, the most successful approaches work in the spectral domain. The algorithm presented in [37] replaces peaks of vertical frequency from the Fourier domain with a noise floor, as explained below. The algorithm steps are:

1. After applying the 2D Fourier transform, a range of vertical frequencies around the origin in the spectral domain: $[-\Omega \dots +\Omega]$ is selected. This horizontal

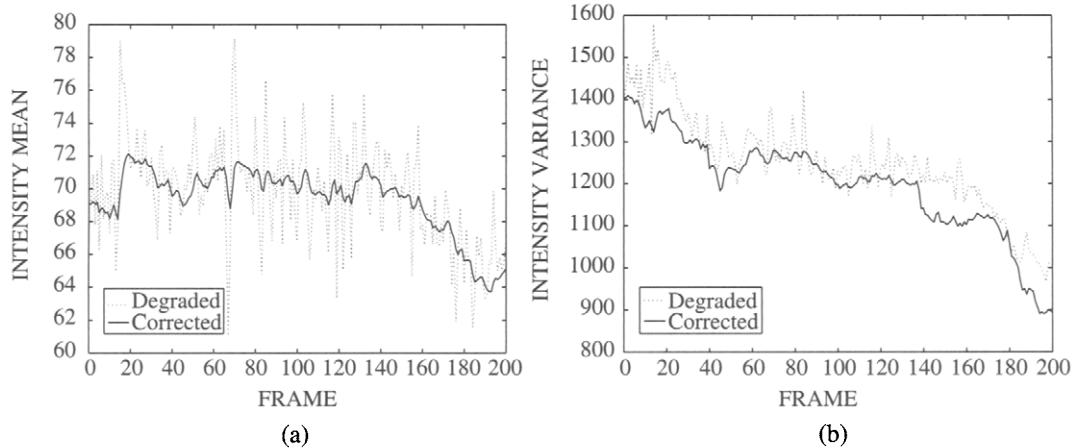


FIGURE 21 (a) Mean of the corrupted and corrected image sequence; (b) variance of the corrupted and corrected image sequence.

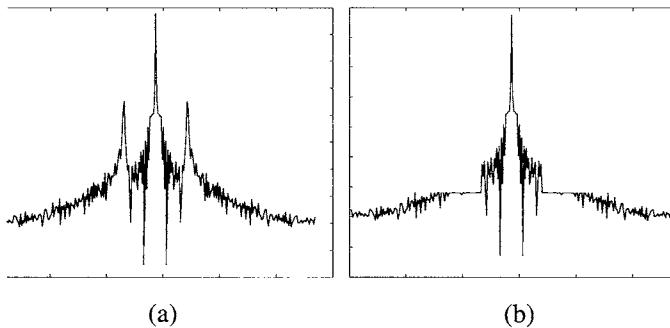


FIGURE 22 An example of spectrum magnitude at zero horizontal frequency.¹ (a) Degraded image; (b) restored image.

band from the spectrum will be protected against any changes.

2. A binary mask \mathbf{B} is created that will indicate which frequencies will be changed:

$$\mathbf{B}(\omega_h, \omega_v) = \begin{cases} 1 & \text{if } (|F(\omega_h, \omega_v)| > \varepsilon\Delta) \text{ AND } (|\omega_v| > \Omega) \\ 0 & \text{Otherwise} \end{cases} \quad (38)$$

where $|F(\omega_h, \omega_v)|$ represents the spectrum magnitude at frequency (ω_h, ω_v) , and $\Delta = \frac{\max|F(\omega_h, \omega_v)| - \min|F(\omega_h, \omega_v)|}{2} + \min|F(\omega_h, \omega_v)|$.

3. The magnitudes of the spectral components selected with mask \mathbf{B} are replaced with a noise floor value, as shown in Fig. 22. This value represents the median magnitude of the frequencies lying outside the band defined in step 1.

4. The resulting magnitude is combined with the original phase extracted in step 1, and the inverse 2D Fourier transform is applied in order to obtain the restored image.

Figure 23 presents the results of the moiré removal algorithm. The parameter values used in this case were: $\varepsilon = 0.1$, $\Delta = 94.4$ and $\Omega = 25$. The moiré pattern was largely removed. Some ringing effect still persists due to the discontinuities inserted in the Fourier domain by the magnitude replacement operation shown in Fig. 22b.

The kinescope moiré phenomenon remains an open research track. The nonlinear distortions associated with it (in particular, the curved geometry) are difficult to model. Until now, the algorithms working in the frequency domain were more successful than those working in the spatial domain. More insight into moiré phenomena is expected to be gained from its temporal analysis.

7 Concluding Remarks

This chapter has described methods for enhancing and restoring corrupted video and film sequences. The material that was offered in this chapter, is complementary to the spatial enhancement and restoration techniques described in other chapters of the *Handbook*. For this reason the algorithmic details concentrated mostly on the temporal processing aspects of image sequences, or on solutions for cases where the temporal information is not usable. Although the focus has been on the detection and removal of a limited number of impairments, namely noise, blotches, vinegar syndrome, flicker and moiré, the approaches and tools described in this chapter are of a more general nature and they

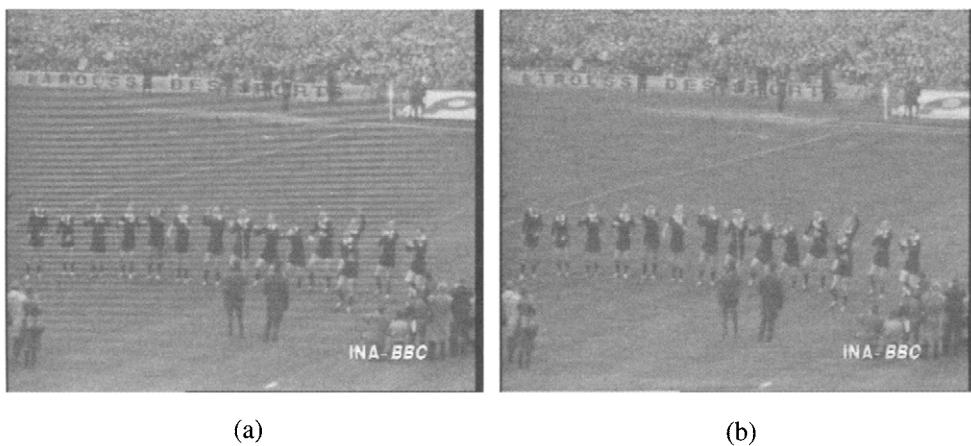


FIGURE 23 (a) Image affected by moiré²; (b) restored image.

¹Images used with permission from the authors of [37].

²Images used with permission from the authors of [37]. Courtesy of INA (Institut de l'Audiovisuel) and BBC (British Broadcast Corporation).

can be used for developing enhancement and restoration methods for other types of degradation.

References

- [1] S.T. Acton, D.P. Mukherjee, J.P. Havlicek, A.C. Bovik, "Oriented texture completion by AM-FM reaction-diffusion", *IEEE Transactions on Image Processing*, 10(6), 2001.
- [2] I. Amidror, *The Theory of the Moiré Phenomenon*, Kluwer Academic, Boston, 2000.
- [3] G.R. Arce, "Multistage order statistic filters for image sequence processing", *IEEE Transactions on Signal Processing*, 39(5), 1147–1163, 1991.
- [4] N. Balakrishnan, C.R. Rao, (eds) *Handbook of Statistics 16: Order Statistics: Theory & Methods*, North-Holland, 1998.
- [5] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels", *IEEE Transactions on Image Processing*, 10(8), August 2001.
- [6] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, "Image inpainting", *Proceedings of SIGGRAPH 2000*, New Orleans, USA, July 2000.
- [7] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, "Simultaneous structure and texture image inpainting", *IEEE Transactions on Image Processing*, 12(8), August 2003.
- [8] R. Bornard, *Probabilistic Approaches for the Digital Restoration of Television Archives*, Ph.D. Thesis, École Centrale Paris, France, 2002.
- [9] J.C. Brailean, R.P. Kleihorst, S. Efstratiadis, A.K. Katsaggelos, R.L. Lagendijk, "Noise reduction filters for dynamic image sequences: a review", *Proceedings of the IEEE*, 83(9), 1272–1291, September 1995.
- [10] T. Chan, J. Shen, "Mathematical models for local non-texture inpainting", *SIAM Journal on Applied Mathematics*, 62(3), 1019–1043, 2001.
- [11] M.J. Chen, L.G. Chen, R. Weng, "Error concealment of lost motion vectors with overlapped motion compensation", *IEEE Trans. on Circuits and Systems for Video Technology*, 7(3), 1997.
- [12] A. Criminisi, P. Pérez, K. Toyama, "Object removal by exemplar-based inpainting", *Proceedings of CVPR 2003*.
- [13] D.L. Donoho, I.M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage", *Biometrika*, 81, 425–455, 1994.
- [14] E. Dubois, S. Sabri, "Noise reduction using motion compensated temporal filtering", *IEEE Transactions on Communications*, 32, 826–831, 1984.
- [15] A.A. Efros, T.K. Leung, "Texture synthesis by non-parametric sampling", *Proceedings of ICCV 1999*, Corfu, Greece, September 1999.
- [16] C. Hentschel, "Video moiré cancellation filter for high-resolution CRTs", *IEEE Transactions on Consumer Electronics*, 47(1), 16–24, 2001.
- [17] A.N. Hirani, T. Totsuka, "Combining frequency and spatial domain information for fast interactive image noise removal", *ACM SIGGRAPH*, 269–276, 1996.
- [18] T.S. Huang (ed.), *Image Sequence Analysis*, Springer Verlag, Berlin, 1991.
- [19] J. Jia, C.-K. Tang, "Inference of segmented color and texture description by tensor voting", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6), 771–786, June 2004.
- [20] J. Kim, J.W. Woods, "Spatio-temporal adaptive 3-D kalman filter for video", *IEEE Transactions on Image Processing*, 6(3), 414–424, March 1997.
- [21] R.P. Kleihorst, R.L. Lagendijk, J. Biemond, "Noise reduction of image sequences using motion compensation and signal decomposition", *IEEE Transactions on Image Processing*, 4(3), 274–284, 1995.
- [22] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, P.J.W. Rayner, "Detection of missing data in image sequences", *IEEE Transactions on Image Processing*, 4(11), 1995.
- [23] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, P.J.W. Rayner, "Interpolation of missing data in image sequences", *IEEE Transactions on Image Processing*, 4(11), 1995.
- [24] A.C. Kokaram, *Motion Picture Restoration: Digital Algorithms for Artifact Suppression in Degraded Motion Picture Film and Video*, Springer Verlag, 1998.
- [25] A.C. Kokaram, S. Godsill, "Joint noise reduction, motion estimation, missing data reconstruction, and model parameter estimation for degraded motion pictures", *Proc. of SPIE*, San Diego, 1998.
- [26] A.C. Kokaram, "Parametric texture synthesis using stochastic sampling", *Proceedings of ICIP 2002* (IEEE), New York, USA, September 22–25, 2002.
- [27] A.C. Kokaram, "On missing data treatment for degraded video and film archives: a survey and a new bayesian approach", *IEEE Transactions on Image Processing*, 13(3), 397–415, March 2004.
- [28] S. Masnou, "Disocclusion: A variational approach using level lines", *IEEE Transactions on Image Processing*, 11(2), February 2002.
- [29] M.K. Ozkan, M.I. Sezan, A.M. Tekalp, "Adaptive motion-compensated filtering of noisy image sequences", *IEEE Transactions on Circuit and Systems for Video Technology*, 3(4), 277–290, 1993.
- [30] S. Rane, M. Bertalmio, G. Sapiro, "Structure and texture filling-in of missing image blocks for wireless transmission and compression applications", *IEEE Transactions on Image Processing*, 2002.
- [31] A. Rareş, M.J.T. Reinders, J. Biemond, "Statistical Analysis of Pathological Motion Areas", *IEE Seminar on Digital Restoration of Film and Video Archives*, London, UK, January 16, 2001.
- [32] A. Rareş, *Archived Film Analysis and Restoration*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 2004.
- [33] P.M.B. Roosmalen, *Restoration of Archived Film and Video*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 1999.
- [34] P.M.B. van Roosmalen, R.L. Lagendijk, J. Biemond, "Correction of intensity flicker in old film sequences", *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7), 1013–1019, 1999.
- [35] P.M.B. van Roosmalen, R.L. Lagendijk, J. Biemond, "Embedded coring in mpeg video compression", *IEEE Transactions on Circuits and Systems for Video Technology*, 3, 205–211, 2002.

- [36] M.I. Sezan, R.L. Lagendijk (eds.), *Motion Analysis and Image Sequence Processing*, Kluwer Academic, Boston, 1993.
- [37] D.N. Sidorov, A.C. Kokaram, "Suppression of moiré patterns via spectral analysis", in *Visual Communications and Image Processing 2002*, Editor C.-C. Jay Kuo, *Proceedings of SPIE*, 4671, 895–906.
- [38] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger, "Shiftable Multiscale transform", *IEEE Transactions on Information Theory*, 38(2), 587–607, 1992.
- [39] E.P. Simoncelli, "Bayesian denoising of visual images in the wavelet domain", in *Bayesian Inference in Wavelet Based Models*, P. Müller and B. Vidakovic, (eds.) Springer-Verlag, Lecture Notes in Statistics 141, 1999.
- [40] C. Stiller, J. Konrad, "Estimating motion in image sequences", *IEEE Signal Processing Magazine*, 16(4), 70–91, July 1999.
- [41] A.M. Tekalp, *Digital Video Processing*, Prentice Hall, Upper Saddle River, NJ, 1995.
- [42] J.W. Woods, J. Kim, "Motion-compensated Spatio-temporal Kalman Filter", in *Motion Analysis and Image Sequence Processing*, M.I. Sezan, and R.L. Lagendijk, eds., Kluwer Academic Publishers, Boston, 1993.

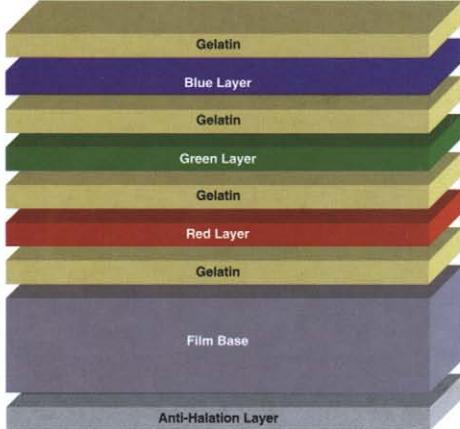


FIGURE 3.11.16 The layered structure of a film.

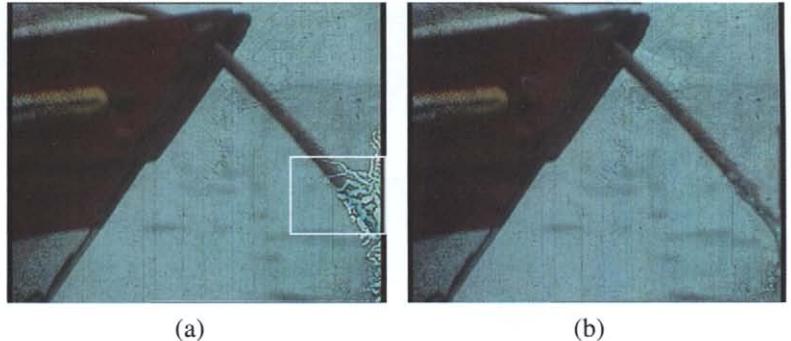


FIGURE 3.11.18 Restoration example (sequence courtesy of RTP — Radiotelevisão Portuguesa). (a) Original frame, with artifact surrounded by a white box; (b) restored frame.



FIGURE 3.13.8 This example demonstrates that mosaics can be used to demonstrate dynamic information. This is a 2,200-frame mosaic obtained from the predator imagery. Red marks show the path of moving vehicles. The mosaic was created using an iterative variant [41] of the described optical flow-based algorithm. The paths were obtained by applying KLT tracker [42] on the stabilized sequence.