Flicker stabilization in image sequences

Julie Delon, Agnès Desolneux

FLICKER STABILIZATION IN IMAGES SEQUENCES*

JULIE DELON † AND AGNÈS DESOLNEUX ‡

Abstract. Flicker is an artefact appearing in old movies or in some videos, which is characterized by intensity changes from a frame to another. Some of these contrast changes are not global and may be localized only on a part of the image. In this paper, a new method to stabilize flicker in image sequences is proposed. This stabilization is local and relies on image patches. Local contrast changes are performed in order to equalize the grey level distribution of an image patch with the grey level distributions of all its corresponding patches in the previous and following frames. The correspondences of a patch are computed thanks to a similarity measure that is built to be robust to contrast changes.

1. Introduction. Flicker is an artefact that can be observed in image sequences and which can be described as fast and unnatural intensity fluctuations from one frame to the other. This artefact is very common in old movies, especially in movies shot before the twenties. The main causes of flicker are support ageing (the negative of the film contains many chemical products which get older and damaged with time) and a non constant exposure time from one frame to the other (the film was driven manually at this time). Flicker also appears in more recent movies or videos, especially sequences with a low time sampling, such as biological sequences of evolving cells or video surveillance sequences. Removing or at least reducing flicker in movies is important in order to improve the subjective quality of films, but it can also be a crucial step before any other post-processing (like compression, segmentation, or image comparison for instance).

One of the characteristics of flicker is that it does not create salient geometrical structures. In practice, we can even say that it is almost invisible on a single frame pulled out of the film, whereas it can be striking in motion. A consequence of this "transparency" property is that flicker reduction requires the use of successive frames in order to work properly: flicker cannot be removed just by looking at each frame independently of the others (which is not necessarily the case for denoising algorithms for instance). Flicker can have global and local aspects: depending on the film, it can act as a global contrast change or like several local contrast changes which affect differently the different parts of the current frame. In consequence, papers dealing with flicker either consider it as a global phenomenon [6, 11, 14, 16, 7], or as local phenomenon [15, 12, 13, 14, 17, 16, 19], acting non-uniformly on images. Global methods provide satisfactory results in the sense that they are robust to motion and to the presence of small defects (dirt, scratches) in the frames. However, these methods are unable to cope with a strongly localized flicker. Now, one of the main difficulties of local flicker reduction is the necessity to take motion into account. Indeed, local methods must decide if a local illumination change in the film is due to flicker or to a moving object. This can get worse if the local flicker has a structured motion. These considerations are illustrated by the different examples of Figure 1.1. The corresponding sequences are available at `http://www.tsi.enst.fr/~delon/Demos/Flicker_stabilization/`. Now the main difference between a contrast change and motion is that a contrast change does not modify the local geometry of an image: in other words, it affects the dynamic of the grey levels, but not their local ordering. In this paper, we will introduce a similarity measure between image patches relying mostly on the geometry of patches (*i.e.* robust to contrast changes). This similarity measure will allow us to estimate motion in the presence of flicker and as a consequence to correctly stabilize local flicker.

The outline of the paper is the following. In Section 2, we present the different flicker models proposed in the literature and the state of the art of global and local flicker reduction. We will see that most of these methods consider motion as an outlier and use either robust estimation

*This work was supported by the French Agence Nationale de la Recherche (ANR), under the grant FREEDOM (ANR07-JCJC-0048-01), "Films, REstauration Et DOnnées Manquantes", and by the GDR ISIS. The authors thank Marc Sandberg, which gave his permission to reproduce here some images from the film *Les Aventures des Pieds Nickelés*, Emile Cohl/Eclair, 1917-1918.

†LTCI, TÉLÉCOM PARIS (CNRS UMR 5141) 46, RUE BARRAULT, 75634 PARIS CEDEX 13 (`JULIE.DELON 'AT' ENST.FR`).

‡MAP5 (CNRS UMR 8145), UNIVERSITÉ PARIS DESCARTES, 45 RUE DES SAINTS-PÈRES, 75006 PARIS (`AGNES.DESOLNEUX 'AT' MI.PARISDESCARTES.FR`).

(a) Images from *The Cure*, Charlie Chaplin, 1917

(b) Images from *Les Aventures des Pieds Nickelés*, Emile Cohl/Eclair, 1917-1918 (copyright: Marc Sandberg)

(c) Taxi sequence with some synthetic local flicker

FIG. 1.1. *Three sequences containing (a), (b) real flicker or (c) synthetic flicker.*

methods or rejection criteria to detect motion zones and discard them from the flicker estimation, relying implicitly on the assumption that there are few motions in the film. Then, in Section 3, we introduce a similarity measure between image patches robust to contrast changes. This will allow us to obtain a new method for local flicker stabilization that will take motion into account. In Section 4, the results of this method are presented on several image sequences: sequences with synthetic local flickers and extracts from old movies with strong real flickers.

2. Flicker model and state of the art.

2.1. Some notations. We introduce here the notations and vocabulary that will be used throughout this paper: t and s denote time variables and x and y spatial variables. The damaged film is denoted by $u = (u_t(.))$, where u_t is the frame at time t. The image domain is denoted by Ω. It is generally a rectangle of $\mathbb{R}^2$ or of $\mathbb{Z}^2$ (in the case of discrete images). We only consider "black and white" movies, which means that each image u_t is a function from Ω to $\mathbb{R}$. The original, non-observed and unknown film, is denoted by u^0.

Applying a *contrast change* to an image v means changing v into $g(v)$, where g is an increasing function from $\mathbb{R}$ into itself. We will call *affine contrast change* any contrast change of the form $g(\lambda) = a\lambda + b$ with $a, b \in \mathbb{R}$ and $a > 0$.
The *upper and lower level sets* of an image $v : \Omega \mapsto \mathbb{R}$ are respectively defined as the sets

$$\chi_\lambda(v) = \{x \in \Omega;\ v(x) > \lambda\} \text{ and } \chi^\lambda(v) = \{x \in \Omega;\ v(x) \leq \lambda\},$$

where $\lambda \in \mathbb{R}$. If g is a contrast change, then $\chi_{g(\lambda)}(g(v)) = \chi_\lambda(v)$, which means that the set of all upper level sets (resp. the set of all lower level sets) remains the same under contrast changes. As

a consequence, the *topographic map* of an image [3, 5], *i.e.* the set of connected components of the topological boundaries of its level sets, remains unchanged under contrast changes. This means that the geometrical content of an image is not modified by contrast changes, as we can see on Figure 2.1.

FIG. 2.1. *On the left, two images containing the same geometric content. Each image can be obtained by applying a well chosen contrast change to the other. On the right, some level lines of these images. The small differences that can be observed between level lines are due to quantization artifacts.*

An image u_t of the film being given, we write $\chi_\lambda(t)$ and $\chi^\lambda(t)$ for the upper and lower level sets of u_t. The histogram of u_t, *i.e.* the distribution of its grey levels, is denoted by h_t. The corresponding cumulative distribution function (or cumulative histogram) is denoted by H_t. If u_t is defined on a discrete grid Ω with $|\Omega| = N^2$ and takes its values in $\{0, \dots, L\}$, the cumulative histogram $H_t : \{0, \dots, L\} \to \{0, \dots, N^2\}$ is given by:

$$H_t(\lambda) = |\{x \in \Omega | u_t(x) \leq \lambda\}| = \sum_{\mu=0}^{\lambda} h_t(\mu),$$

where $|\cdot|$ denotes the cardinality of a set.
This definition leads to the following interpretations:

- For any $x \in \Omega$, $H_t(u_t(x))$ is the *rank* of x, when the grey levels of u_t are ordered increasingly.
- Let k be an integer, we define $H_t^{-1}(k) = \min\{\lambda \in \{0, \dots, L\} \ ; \ H_t(\lambda) \geq k\}$. If x is a pixel of rank k in u_t, then $H_t^{-1}(k)$ is the grey level of x.
- Let G be an increasing discrete function on $\{0, \dots, L\}$, taking its values in $\{0, \dots, N^2\}$. Let us define $G^{-1}(k) = \min\{\lambda \in \{0, \dots, L\} \ ; \ G(\lambda) \geq k\}$. If we assume that H_t is onto, then $v_t = G^{-1} \circ H_t(u_t)$ and u_t share the same geometry, and the cumulative histogram of v_t is given by the function G. In practice, the hypothesis that H_t is onto is almost never satisfied, and the cumulative histogram of $G^{-1} \circ H_t(u_t)$ will only be very close to G. Roughly speaking, this means that we can map the grey level distribution of an image onto the grey level distribution of any other image by an adequate contrast change.

2.2. Modeling the flicker as global. Papers dealing with flicker can be roughly classified in two categories: those which consider flicker as a global phenomenon, and those which consider it as a local phenomenon, acting non-uniformly on images. This section begins with a review of global methods and then focus specifically on one of them, which will be later extended to the case of local flicker.

One of the simplest way to model the flicker is to consider that it acts as a global affine change of contrast, *i.e.*

$$u_t(x) = \alpha_t u_t^0(x) + \beta_t, \quad \forall x \in \Omega,$$

where α_t and β_t are two unknown constants, depending only on t. To correct this affine flicker, the authors of [14, 16] apply an affine grey level transform to each frame, so that it fills a fixed dynamic range. If this global correction is simple, it is emphasized in [6] that it is not really satisfactory. Decencière [6] proposes instead to apply to u_t an affine contrast change which fits the mean and range of u_t onto those of u_{t-1}. The image u_{t+1} is then corrected thanks to u_t, etc. Unfortunately, this model is often too far from the true flicker damage, and the recursive aspect of the method makes it oversensitive to initialization and unstable (subject to errors accumulation).

In order to avoid these shortcomings, any flicker reduction method should meet two requirements. First, the method should not depend of the time origin of the film, which means that the flicker correction should commute with any time shift $t \to t + t_0$. Second, the result should be the same if we reverse the time direction, which means that the correction should commute with the operation $t \to -t$. In order to take into account these considerations and to extend the flicker correction to general (not only affine) contrast changes, the authors of [11] propose to fit the contrast of u_t onto a target grey level distribution, computed as the average grey level distribution of all images of the film in a temporal neighborhood of t. However, it can be shown that averaging directly a set of histograms is not always satisfactory and can lead to severe artifacts [8]. The next paragraph gives some indications on the "good way" to average a set of histograms.

2.2.1. Midway equalization between two images. Let us begin with the case of two images. Let u_1 and u_2 be two discrete images defined on a grid Ω containing $|\Omega| = N^2$ pixels and taking their values in $\{0, \dots, L\}$. Let h_1 and h_2 denote their respective grey level histograms. In order to average the contrast between u_1 and u_2 (*i.e.* to equalize their respective grey level histograms on a common one), one could just apply a contrast change to each image such that the grey level distribution of both images would fit $(h_1 + h_2)/2$. This solution is generally not satisfactory. Indeed, if u_1 (resp. u_2) has a unimodal grey level histogram, centered at m_1 (resp. $m_2 \neq m_1$), the average histogram $(h_1 + h_2)/2$ contains two modes: one at m_1 and the other at m_2. It would be much more natural to define a midway histogram between them as a unimodal histogram centered at $(m_1 + m_2)/2$.

(a) Before midway equalization

(b) After midway equalization

FIG. 2.2. *Top: two images and their respective cumulative histograms. Bottom: the images and their cumulative histograms after the midway equalization.*

As it was shown in [7], the "good" midway histogram between h_1 and h_2 can be defined in the following way: if H_i is the discrete cumulative histogram corresponding to h_i, we call *midway* histogram between H_1 and H_2 the cumulative histogram

$$H_{1/2} = \left(\frac{H_1^{-1} + H_2^{-1}}{2} \right)^{-1}.$$

The properties of this midway distribution are the following:

- the images $H_{1/2}^{-1} \circ H_1(u_1)$ and $H_{1/2}^{-1} \circ H_2(u_2)$ have the same cumulative histogram $H_{1/2}$.
- if there exists an image u_0 and two contrast changes f and g such that $u_1 = f(u_0)$ and $u_2 = g(u_0)$, then $H_{1/2}$ is the cumulative histogram of $(u_1 + u_2)/2$.
- action on level sets: grey levels having the same rank in both images are averaged together.

Remark 1: In practice, in order to equalize the histograms of two images, one can for instance create two vectors of length N^2 containing respectively the ordered grey levels of u_1 and u_2, then compute the mean of these two vectors, and finally assign to each pixel of each image the grey level of the element in the averaged vector which has the same rank as the considered pixel.

Remark 2: If we consider h_1 and h_2 as densities of two distributions ν_1 and ν_2, then in the theory of optimal transportation, the Wasserstein distance between ν_1 and ν_2 is defined as $\int |H_1^{-1} - H_2^{-1}|$ (see [18]). Now, consider the geodesic between ν_1 and ν_2 for the previous distance and define $\mu_{1/2}$ as the middle of this geodesic. Then $H_{1/2}$ is the cumulative distribution function of $\nu_{1/2}$.

2.2.2. A scale-time correction for global flicker. An extension of the midway equalization to image sequences, called Scale-Time Equalization (STE), can be used to stabilize the flicker globally. In [8], Delon has shown the following proposition, which gives the canonical form of a flicker reduction operator when some simple hypotheses are assumed.

PROPOSITION 2.1. *[8] Let (u_t) be a film and (H_t) the discrete cumulative histograms of the frames. Let STE be an operator acting on films and satisfying the following properties:*

- *STE acts like a contrast change on each frame,*
- *for any λ, the action of STE on the lower level sets $\chi_\lambda(t)$ of the frames does not depend on the action of STE on the upper level sets $\chi^\lambda(t)$,*
- *STE acts on spatially constant films (*i.e. *films such that each frame is a constant image, but the constant can vary in time) like a linear scale space (that is like a convolution with a Gaussian kernel, see [9]).*

Then, there is a scale parameter σ such that the operator STE fits the histogram of u_t on the "midway" histogram $\mathbb{H}_t$ whose inverse is defined by

$$\mathbb{H}_t^{-1}(k) = \int \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(t-s)^2/2\sigma^2} H_s^{-1}(k)\, ds. \tag{2.1}$$

The corresponding operator on u can be written

$$STE u_t(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int e^{-(t-s)^2/2\sigma^2} \underbrace{H_s^{-1} \circ \underbrace{H_t(u_t(x))}_{\text{rank of } x \text{ in } u_t}}_{\text{grey level of the pixel of same rank as } x \text{ in } u_s} ds. \tag{2.2}$$

Observe that this correction does not depend of the origin of the film: the operator commutes with translations and symmetry in time. This correction is easy to perform and gives good results as long as the flicker acts roughly globally on the film (see for instance Figure 2.3). However, such a global method is obviously unable to eliminate local flicker as the ones observed in the examples of Figure 1.1 (b) and (c).

2.3. Modeling the flicker as local. The methods of flicker reduction mentioned previously are well adapted when the flicker is global and affects the whole image equally. In practice, the flicker can vary from one part of the image to the other. Even worse, the zones which are the most affected by the flicker can change with time. A typical example can be described as dark and transparent shadows covering parts of the frame and moving from one frame to the other in the sequence.

2.3.1. Mathematical model. The most general model for local flicker relies on the assumption that each frame of the film is locally affected by a contrast change. This can be written as:

$$u_t(x) = f_t(x, u_t^0(x)), \tag{2.3}$$

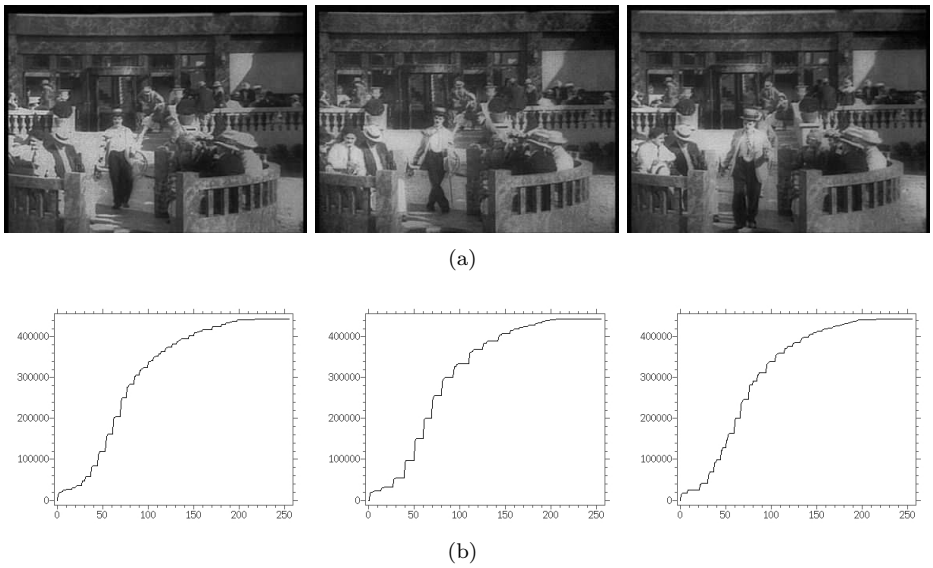


FIG. 2.3. *(a) Three images of the Chaplin's film* The Cure *(see Figure 1.1 for original images) after a global flicker correction using Equation (2.2). (b) Corresponding cumulative histograms.*

where $f_t(x, \cdot)$ is a contrast change ($\lambda \to f_t(x, \lambda)$ is increasing). Obviously, this model is not restrictive enough: if u_1 and v_1 are two images with nothing in common, we can always write $v_1(x) = (v_1(x)/u_1(x))u_1(x)$, which means that we can transform u_1 into v_1 by Formula (2.3) using $f(x, \lambda) = (v_1(x)/u_1(x))\lambda$. In order to reduce the field of possible f_t, it is generally assumed that f_t is smooth in the spatial variable x (but not in the time variable t).

Some authors, in a different framework, have already given a mathematical definition of a local change of contrast. For instance, in [3] and [4], *local contrast changes* are defined in a strong mathematical sense: those are changes which preserve *exactly* the topographic map of the image. More precisely, these changes are defined by an equation like (2.3), but with the additional constraint on f that if $u(x) = u(y) = \lambda$ and if x and y are on the same connected component of a level line, then $f(x, \lambda) = f(y, \lambda)$. Such a definition cannot be adopted for flicker, mainly because of two points. First, there is no reason for the flicker to be correlated with the topographic map of the images. Second, the topographic map of an image affected by some flicker is not exactly preserved (see the experimental section at the end of the paper for an example of this). As already mentioned, the flicker is transparent and preserve the salient geometric structures but not the whole topographic map of the images.

In the literature, the model given in Equation (2.3) is often reduced to locally affine contrast changes:

$$u_t(x) = a_t(x)u_t^0(x) + b_t(x), \tag{2.4}$$

where a_t and b_t are smooth functions of x.

2.3.2. State of the art of local flicker reduction. Several methods have been proposed to correct locally the flicker. In most of these methods [17, 15, 12, 13], the image domain is divided into a fixed grid of blocks (the blocks can overlap). The correction is then applied locally on each block independently from the others, either recursively [17, 15], or via a "filtering" using blocks with the same spatial position in consecutive frames of the film [12, 13]. In [17], a recursive affine correction is applied on each block. A rejection criterion is introduced to avoid the flicker estimation in blocks containing motion or blotches. However, the proposed rejection criterion cannot tell apart between the motion of a low-contrasted object and a local flicker. In order to avoid this shortcoming, the authors of [19] replace the rejection criterion by an explicit block-matching motion estimation. The authors of [15] also propose a recursive correction of flicker,

where the motion is seen as an outlier: a scale parameter is used to decide if a local change is due to flicker or not. For the reasons developed at the beginning of Section 2.2, the recursive aspect of these methods makes them unstable and difficult to initialize. In [12], robust estimators are used to estimate the flicker and discard what they consider as outliers in this estimation (*i.e.* blotches and motion). In [13], a real parameter $\gamma(x)$, varying slowly in space, controls the intensity of flicker around each pixel x. The estimation of γ should theoretically involve motion estimation. However, considering that "the motion estimation is deteriorated by the presence of flicker" [13], the authors replace this estimation by a temporal filtering of γ.

2.3.3. Does flicker reduction need motion estimation?. All the previous methods, with the exception of [19], rely on the implicit assumption that there are few motions in the film. In these methods, motion is never explicitly estimated. Instead, these approaches use either robust estimation methods or rejection criteria to detect motion zones and discard them in the flicker estimation. In both cases, motion is treated as an outlier, and assumed to be minor enough to not disturb the local flicker estimation. If the dominant motion (or background motion) of the film is non-zero, or if the film contains several objects moving at the same time, most of these methods might fail. Even in the case of low motion, deciding if a localized contrast change is due to flicker or to the motion of a low contrasted object is not easy. In the next sections, we will see that some kind of motion estimation is necessary for flicker reduction, even if this estimation doesn't have to be precise. Motion can be estimated in the presence of flicker thanks to specific similarity measures, designed to be robust to contrast changes.

3. Motion based flicker reduction without prior estimation.

3.1. A similarity measure robust to contrast changes. Usually, when one wants to estimate the motion in a movie, one computes the correspondences of the pixels based on the assumption that grey levels are preserved. Unfortunately, when the movie is damaged by some flicker, the grey level of a given object in the movie can vary in time. Thus, to recover the motion, one needs to decide if two pixels correspond to each other even when there has been some contrast change. To this aim, we propose to use a simple block-matching approach, relying on a similarity measure D between patches (small squares of pixels) built to be robust to contrast changes. In order to minimize the risk of errors, this similarity measure should nevertheless be discriminant (the similarity measure between two similar patches should be much smaller than the similarity measure between two very different patches). In short, we are looking for a similarity measure D between images patches I and J having the following properties:

(i) Symmetry : $D(I, J) = D(J, I)$,
(ii) Robustness to affine and increasing contrast changes: $\forall a > 0$ and $\forall b$, $D(I, aI + b)$ should be equal to 0,
(iii) Ability to discriminate: we want $D(I, J)$ to be "large" when J doesn't look like any $aI+b$.

We will consider three different similarity measures satisfying all the first and second properties above, and will see that the third property will help choosing the best one. The similarity measures that we will consider are the following ones:

- D_{angle} : the *angular measure* is the L^2 distance between gradient orientations. For an image I, let $x \to \theta_I(x) = \text{Arg}(\nabla I(x)) \in [0, 2\pi)$ denote the orientation of the gradient of I at any pixel x. It is invariant to any change of contrast (indeed, if $g : \mathbb{R} \to \mathbb{R}$ is increasing, then $\nabla g(I) = g'(I)\nabla I$, and thus $\theta_{g(I)} = \theta_I$). We then define the "angular measure" between two image patches I and J (defined on the same domain Λ) by

$$D_{\text{angle}}(I, J)^2 = \| e^{i\theta_I} - e^{i\theta_J} \|_2^2 = \frac{1}{|\Lambda|} \sum_{x \in \Lambda} \| e^{i\theta_I(x)} - e^{i\theta_J(x)} \|^2 .$$

  Such a distance has been used for instance by Lisani and Morel in [10] to detect changes in satellite images. This distance has the property of being invariant under any increasing change of contrast on I or J.
- D_{corr} : the *correlation measure* is the L^2 distance between image patches normalized by their empirical mean and variance. Let $\overline{I} = \sum_{x \in \Lambda} I(x)/|\Lambda|$ (resp. $\overline{J}$) denote the empirical

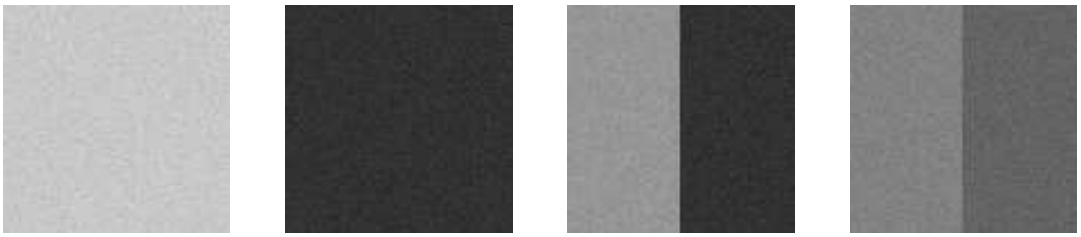


FIG. 3.1. *Four patches, from left to right: I_1, I_2, J_1 and J_2. The first two patches I_1 and I_2 both represent a constant zone up to a contrast change and an additive noise. The last two patches J_1 and J_2 both represent the same edge zone up to a contrast change and an additive noise.*

mean grey level of I (resp. of J) on the domain Λ, $S_I^2 = \sum_{x\in\Lambda}(I(x)-\overline{I})^2/|\Lambda|$ (resp. S_J^2) denote the empirical variance and $\mathrm{Cov}(I,J) = \sum_{x\in\Lambda}(I(x)-\overline{I})(J(x)-\overline{J})/|\Lambda|$ denote the empirical covariance between I and J. Then the correlation measure is given by

$$D_{\mathrm{corr}}(I,J)^2 = \frac{1}{2}\left\|\frac{I-\overline{I}}{S_I} - \frac{J-\overline{J}}{S_J}\right\|_2^2 = 1-\mathrm{Corr}(I,J) = 1-\frac{\mathrm{Cov}(I,J)}{S_I S_J},$$

- D_{aff} : the *"affine" similarity measure* computes the L^2 distance of each patch to all the increasing affine transforms of the other, and then takes the max of these two measures to achieve symmetry. This measure can be written as

$$D_{\mathrm{aff}}(I,J) = \max\left(\min_{a\geq 0,b} \parallel I-aJ-b\parallel_2, \min_{a\geq 0,b}\parallel J-aI-b\parallel_2\right).$$

  D_{aff} can be easily exactly computed, and we find that it is given by:

$$D_{\mathrm{aff}}(I,J)^2 = \max(S_I^2,S_J^2)(1-\mathrm{Corr}(I,J)^2),$$

  where S_I^2, S_J^2 and $\mathrm{Corr}(I,J)$ are defined as above in the case of the correlation measure.

Remark: We have used here the word "similarity measure" instead of "distance" because the usual properties of the mathematical definition of a distance are not satisfied (mainly the fact that we can have $D(I,J)=0$ without having $I=J$).

Remark 2: The measure D_{aff} could be generalized to

$$D_{monotone}(I,J) = \max\left(\min_{g \text{ increasing}} \|I-g(J)\|, \min_{g \text{ increasing}} \|g(I)-J\|\right),$$

which can be easily computed via the *pool adjacent violators* algorithm [1]. For the sake of simplicity, we stick in this paper with the affine formulation.

The similarity measure D will be used to compare image patches in different frames of the movie that could be affected by some flicker. A main requirement is thus that D is invariant to affine contrast changes: the similarity measure between two patches of the same geometry should be small. This property is satisfied by the three measures defined above but of course not by the L^2 distance. On the other hand, the similarity measure has to be discriminant: the similarity between two patches representing very different geometries (for instance between a patch containing an edge and an almost constant patch) should be large. These two goals (invariance and discrimination) are of course contradictory: the more invariance a similarity measure has, the less its discrimination power is. To find the optimal measure, a good trade-off between the two criteria has to be found.

To choose between $D_{\mathrm{angle}}, D_{\mathrm{corr}}$ and D_{aff}, we will first theoretically analyze how these similarity measures behave when they have to compare image patches with the same geometry (constant zone, or same edge zone with a change of contrast), and patches with dissimilar geometries (a flat zone and an edge zone). Such patches are shown on Figure 3.1.

PROPOSITION 3.1 (Choice of a similarity measure). *Let Λ denote the domain on which the image patches are defined: it is assumed to be a square of $n\times n$ pixels. Let I_1 (resp. I_2)*

be a random image patch, whose pixel grey levels are i.i.d. random variables with the normal distribution $\mathcal{N}(c_1, \sigma^2)$ (resp. $\mathcal{N}(c_2, \sigma^2)$), where c_1, c_2 and σ are constants. I_1 and I_2 can be seen as noisy constant patches. Now, let J_1 (resp. J_2) be a random image patch, whose pixel grey levels are independent, and such that for all $1 \leq k \leq n/2$, $J_1(k,l)$ (resp. $J_2(k,l)$) follows the $\mathcal{N}(\alpha_1, \sigma^2)$ (resp. $\mathcal{N}(\alpha_2, \sigma^2)$) distribution, and for all $n/2 < k \leq n$, $J_1(k,l)$ (resp. $J_2(k,l)$) follows the $\mathcal{N}(\beta_1, \sigma^2)$ (resp. $\mathcal{N}(\beta_2, \sigma^2)$) distribution, where α_1, α_2, β_1 and β_2 are constants. J_1 and J_2 can be seen as noisy edge patches. Then the asymptotic behavior (in the sense of the a.s. convergence when $n \to \infty$) of the three similarity measures defined above is summarized in the following table:

	$D^2(I_1, I_2)$	$D^2(I_1, J_1)$	$D^2(J_1, J_2)$
D^2_{angle}	$2\ (\pm \frac{\sqrt{2}}{n})$	$2\ (\pm \frac{\sqrt{2}}{n})$	$2 - \frac{2\mu(C_1/\sigma, C_2/\sigma)}{n}$
D^2_{corr}	1	1	$1 - \frac{C_1 C_2}{\sqrt{C_1^2+4\sigma^2}\sqrt{C_2^2+4\sigma^2}}$
D^2_{aff}	σ^2	$\frac{C_1^2}{4} + \sigma^2$	$\left(\sigma^2 + \max(\frac{C_1^2}{4}, \frac{C_2^2}{4})\right) \cdot (1 - \frac{C_1^2 C_2^2}{(4\sigma^2+C_1^2)(4\sigma^2+C_2^2)})$

where $C_1 = \beta_1 - \alpha_1$ (resp. $C_2 = \beta_2 - \alpha_2$) is the contrast of the edge in J_1 (resp. J_2), and $\mu(C_1/\sigma, C_2/\sigma)$ is a function of C_1/σ and C_2/σ which takes its values between -1 and 1.

Proof.

• We first start with the angular measure D_{angle}. We will assume here that the gradient of any discrete image I is computed by simple differences on a four pixels neighborhood, that is:

$$\nabla I(k,l) = \frac{1}{2} \begin{pmatrix} I(k+1,l) + I(k+1,l+1) - I(k,l) - I(k,l+1) \\ I(k,l+1) + I(k+1,l+1) - I(k,l) - I(k+1,l) \end{pmatrix}.$$

Using complex numbers notations, we write

$$2\nabla I(k,l) = (1+i)(I(k+1,l+1) - I(k,l) + i(I(k,l+1) - I(k+1,l))).$$

Now, observe that if X_1, X_2, X_3 and X_4 are four i.i.d. $\mathcal{N}(0,1)$ random variables, then $X_4 - X_1 + i(X_3 - X_2) = Re^{i\Theta}$, where Θ is uniform on $[0, 2\pi)$ and R has the probability density $r \to \frac{1}{2} r \exp(-r^2/4)$ on $\mathbb{R}_+$. Thus, if I is a noisy constant image patch (like I_1 or I_2), the orientations $\theta_I(k,l)$ are uniformly distributed on $[0, 2\pi)$. If I is a noisy edge patch (like J_1 or J_2), the orientations $\theta_I(k,l)$ are uniform on $[0, 2\pi)$ when the point (k,l) does not belong to the edge (i.e. when $k \neq n/2$). When the pixel (k,l) belongs to the edge (i.e. when $k = n/2$), then $\theta_I(k,l) = \text{Arg}(C + i\sigma R e^{i\Theta}) = \text{Arg}(\frac{C}{\sigma} + iRe^{i\Theta})$, where C is the contrast of the edge and where R and Θ follow the laws given above.

Let us now compute the angular measure D_{angle} between two generic image patches I and J: $D^2_{\text{angle}}(I,J) = \frac{1}{n^2} \sum_{k,l} |e^{i\theta_I(k,l)} - e^{i\theta_J(k,l)}|^2 = 2 - \frac{2}{n^2} \sum_{k,l} \cos(\theta_I(k,l) - \theta_J(k,l))$. If Θ is a random variable uniformly distributed on $[0, 2\pi)$, then $\mathbb{E}(\cos\Theta) = \frac{1}{2\pi} \int_0^{2\pi} \cos\theta \, d\theta = 0$ and $\text{Var}(\cos\Theta) = \frac{1}{2\pi} \int_0^{2\pi} \cos^2\theta \, d\theta = \frac{1}{2}$. Now, if I_1 and I_2 are two noisy constant image patches, $\theta_{I_1}(k,l) - \theta_{I_2}(k,l)$ is uniformly distributed on $[0, 2\pi)$. It follows that the expected value of their squared angular distance is $\mathbb{E}(D^2_{\text{angle}}(I_1, I_2)) = 2$, and its variance is of the order of $\frac{4}{n^4} n^2 \text{Var}(\cos\Theta) = 2/n^2$. If I_1 is a noisy constant patch and J_1 is a noisy edge patch, then $\theta_{I_1}(k,l) - \theta_{J_1}(k,l)$ is also uniformly distributed on $[0, 2\pi)$, thanks to the fact that θ_{I_1} is uniformly distributed on $[0, 2\pi)$ and independent of θ_{J_1}. Consequently, their angular similarity measure has exactly the same property as $D_{\text{angle}}(I_1, I_2)$. Finally, assume that J_1 and J_2 are two noisy edge patches, with respective contrasts C_1 and C_2 across their edges. Consider the two random variables $\Phi_1 = \text{Arg}(C_1/\sigma + iR_1 e^{i\Theta_1})$ and $\Phi_2 = \text{Arg}(C_2/\sigma + iR_2 e^{i\Theta_2})$, where R_1, R_2, Θ_1 and Θ_2 are independent random variables following the laws defined above. Let us denote $\mu(C_1/\sigma, C_2/\sigma) = \mathbb{E}(\cos(\Phi_1 - \Phi_2))$. The computation of this function μ is not straightforward: it involves special Bessel type functions. However, notice that when the normalized contrasts C_1/σ and C_2/σ increase, the laws of Φ_1 and Φ_2 get more concentrated around 0 and μ gets closer to 1. Finally, to compute the expected angular similarity measure between J_1 and J_2, we just write

$$\mathbb{E}(D^2_{\text{angle}}(J_1, J_2)) = 2 - \frac{2}{n} \mu(C_1/\sigma, C_2/\sigma)$$

• The computation of the two other similarity measures D_{corr} and D_{aff} is simpler. In the following, the $\rightarrow_{a.s.}$ sign denotes the almost sure convergence when $n \rightarrow \infty$. Observe that the image patch I_1 can be written $I_1(k,l) = c_1 + \sigma X_{k,l}$, where the $X_{k,l}$'s are i.i.d. with the $\mathcal{N}(0,1)$ distribution. In the same way, I_2 can be written $I_2(k,l) = c_2 + \sigma Y_{k,l}$, where the $Y_{k,l}$'s are i.i.d. with the $\mathcal{N}(0,1)$ distribution. It follows that

$$\overline{I_1} = \frac{1}{n^2}\sum_{k,l}(c_1 + \sigma X_{k,l}) = c_1 + \sigma\overline{X} \rightarrow_{a.s.} c_1, \quad S^2_{I_1} = \sigma^2 S^2_X \rightarrow_{a.s.} \sigma^2,$$

$$\text{and} \quad \text{Cov}(I_1, I_2) = \frac{\sigma^2}{n^2}\sum_{k,l}(X_{k,l} - \overline{X})(Y_{k,l} - \overline{Y}) \rightarrow_{a.s.} 0.$$

Thus, when $n \rightarrow \infty$, $D^2_{\text{corr}}(I_1, I_2) \rightarrow_{a.s.} 1$ and $D^2_{\text{aff}}(I_1, I_2) \rightarrow_{a.s.} \sigma^2$.

For the patch J_1, it can be written $J_1(k,l) = \alpha_1 + \sigma Z_{k,l}$ for $1 \leq k \leq n/2$ and $J_1(k,l) = \beta_1 + \sigma Z_{k,l}$ for $n/2 < k \leq n$, where the $Z_{k,l}$'s are i.i.d. with the $\mathcal{N}(0,1)$ distribution. As a consequence,

$$\overline{J_1} = \frac{1}{n^2}\sum_{(k,l);k\leq n/2}(\alpha_1 + \sigma Z_{k,l}) + \frac{1}{n^2}\sum_{(k,l);k>n/2}(\beta_1 + \sigma Z_{k,l}) = \frac{\alpha_1+\beta_1}{2} + \sigma\overline{Z} \rightarrow_{a.s.} \frac{\alpha_1+\beta_1}{2}, \text{ and}$$

$$\begin{aligned} S^2_{J_1} &= \frac{1}{n^2}\left(\sum_{(k,l);k\leq n/2}(\frac{\alpha_1-\beta_1}{2} + \sigma(Z_{k,l} - \overline{Z}))^2 + \sum_{(k,l);k>n/2}(\frac{\beta_1-\alpha_1}{2} + \sigma(Z_{k,l} - \overline{Z}))^2\right) \\ &= \frac{C_1^2}{4} + \sigma^2 - C_1\frac{\sigma}{n^2}\left(\sum_{(k,l);k\leq n/2} Z_{k,l} - \sum_{(k,l);k>n/2} Z_{k,l}\right) \rightarrow_{a.s.} \frac{C_1^2}{4} + \sigma^2. \end{aligned}$$

In the same way, it can be shown that

$$\text{Cov}(I_1, J_1) = -\frac{C_1}{2}\frac{\sigma}{n^2}\left(\sum_{(k,l);k\leq n/2} X_{k,l} - \sum_{(k,l);k>n/2} X_{k,l}\right) + \frac{\sigma^2}{n^2}\sum_{k,l}(X_{k,l} - \overline{X})(Z_{k,l} - \overline{Z}) \rightarrow_{a.s.} 0$$

$$\text{and} \quad \text{Cov}(J_1, J_2) \rightarrow_{a.s.} \frac{C_1 C_2}{4}.$$

□

If we look at the first two columns in the table of Proposition 3.1, we notice that both the angular similarity measure and the correlation measure are unable to discriminate between a constant image patch and an edge zone. Whereas for the affine similarity measure, we statistically have that $D^2_{\text{aff}}(I_1, J_1) > D^2_{\text{aff}}(I_1, I_2)$, which means that it is able to discriminate between constant zones and edge zones. Thus, the conclusion is the following:

COROLLARY 3.1. *The affine similarity measure D_{aff} is the only one of the three considered similarity measures which is able to discriminate between an edge patch and a constant patch.*

An experimental way to check this result is to compute similarity maps on real images. This is illustrated by Figures 3.2 and 3.3. For a given patch, we compute the similarity measures to all neighboring patches for four different measures: L^2, D_{angle}, D_{corr} and D_{aff}. The similarity maps thus obtained are an experimental check of Proposition 3.1 on more realistic patches. In particular, we notice on the figures that the measures D_{corr} and D_{angle} are not very discriminant: patches in flat zones are similar to almost all other patches, whatever their geometry. On the other hand, the similarity measure D_{aff} is able to select only the patches which have the same geometry as the considered patch I.

The conclusions of both the theoretical analysis and the experimental study are the same: among the considered similarity measures, the optimal one is D_{aff}. In all the following, we will use $D = D_{\text{aff}}$ to measure the similarity between image patches.

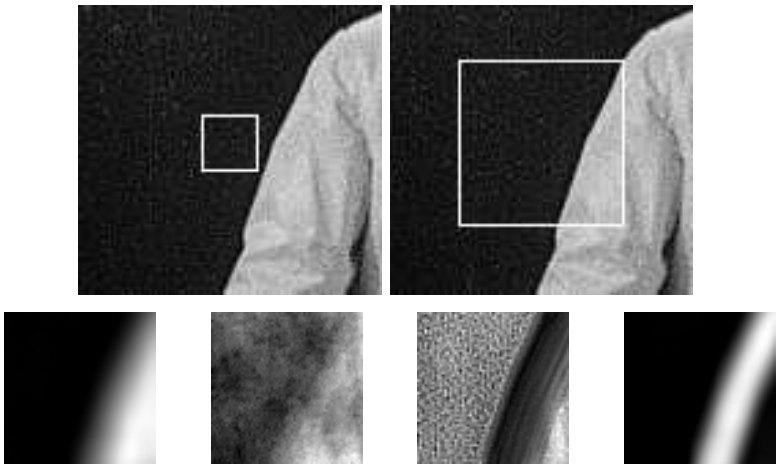


FIG. 3.2. *Similarity map for a constant patch. Top: two consecutive images of a movie. Top left: patch I delimited by the white frame. Top right: neighborhood to which belongs the center of the patch J. Bottom, from left to right: similarity map ($J \to D(I,J)$) for the similarity measures L^2, D_{angle}, D_{corr} and D_{aff}. For each map, the white grey level corresponds to the maximal value obtained on the neighborhood and the black grey level corresponds to the minimal one. We notice here that the similarity measure D_{aff} is the only one which is at the same time discriminant (the "distance" between the constant patch and an edge patch is large) and invariant to affine contrast changes.*

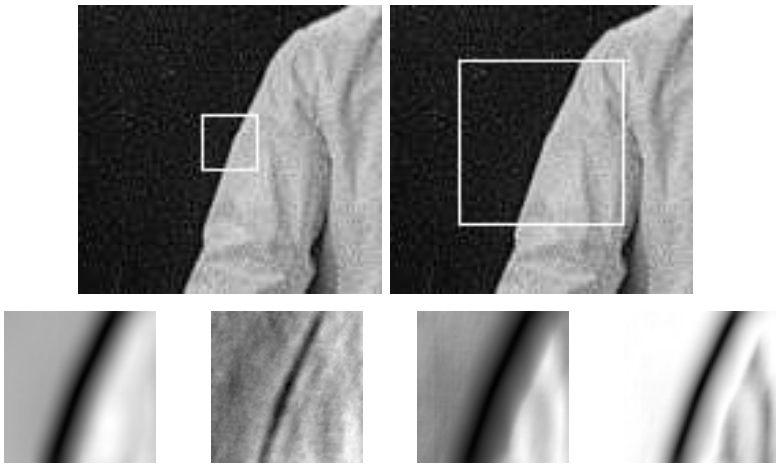


FIG. 3.3. *Similarity map for an edge patch. Top: two consecutive images of a movie. Top left: patch I delimited by the white frame. Top right: neighborhood to which belongs the center of the patch J. Bottom, from left to right: similarity map ($J \to D(I,J)$) for the similarity measures L^2, D_{angle}, D_{corr} and D_{aff}. For each map, the white grey level corresponds to the maximal value obtained on the neighborhood and the black grey level corresponds to the minimal one.*

3.2. Motion based flicker reduction. The idea of the method we propose for flicker reduction will be to change locally the contrast in the movie thanks to weighted means of inverse cumulative local histograms. The weights will be computed from the similarity between patches.

Let u be a discrete movie damaged by some flicker (local or not) and let u_t be the frame of the movie at time t. Let Λ be a square of pixels centered at 0. For $x \in \Omega$, we consider the patch $u_t(x+\Lambda)$ centered in x. Let Λ_x denote the set $\{x+\Lambda\}$. The similarity measure D_{aff} introduced in Section 3.1 is insensitive to contrast changes, and thus can be used to estimate the motion in the presence of flicker. For each pixel x in u_t, its matching point $\varphi_{t,s}(x)$ in u_s can be estimated by comparing $u_t(x+\Lambda)$ and $u_s(y+\Lambda)$ for all y in a suitable neighborhood:

$$\varphi_{t,s}(x) = \text{Argmin}_{y \in W_{t,s}(x)} D_{\text{aff}}(u_t(x+\Lambda), u_s(y+\Lambda)), \tag{3.1}$$

where $W_{t,s}(x)$ is a search window in u_s. Obviously, we get in particular $\varphi_{t,t}(x) = x$. In practice, the choice of $W_{t,s}(x)$ is crucial and will be discussed in details in Section 3.5 (in particular, we will see how the size of this window should be related to the size of the patches). Let us mention however that this window can either be static and written $x+W$, with W a neighborhood of 0, or be updated at each frame and written $W_{t,s}(x) = \varphi_{t,s-1}(x) + W$ if $s > t$ and $W_{t,s}(x) = \varphi_{t,s+1}(x) + W$ if $s < t$. In this last case, the estimation of $\varphi_{t,s}(x)$ depends on the estimation of $\varphi_{t,s-1}(x)$ (or $\varphi_{t,s+1}(x)$ if $s < t$).

Relying on this estimation, a first naive motion-based flicker stabilization operator M can be defined as

$$Mu_t(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_s e^{-(t-s)^2/2\sigma^2} H^{-1}_{s,\Lambda_{\varphi_{t,s}(x)}}(H_{t,\Lambda_x}(u_t(x)))\, ds, \quad \forall x \in \Omega \tag{3.2}$$

where H_{t,Λ_x} denotes the cumulative histogram of the patch $u_t(\Lambda_x)$. It may happen that the Argmin operator in (3.1) is ambiguous and returns several points. In this case, one can simply use all of them in Equation (3.2).

The previous stabilization uses the similarity measure D_{aff} between patches merely as a way to estimate the motion. Now, it may happen that the motion is badly estimated or difficult to estimate for some s (for instance because of a change of sequence) and that the patch $u_s(\varphi_{t,s}(x)+\Lambda)$, which achieves the min in (3.1), is not as similar to $u_t(x+\Lambda)$ as it should be. It thus could be interesting to take only into account patches such that $D_{\text{aff}}(u_t(x+\Lambda), u_s(\varphi_{t,s}(x)+\Lambda))$ is actually small. This can be done by weighting the mean by a decreasing function of the distance $D^2_{\text{aff}}(u_t(x+\Lambda), u_s(y+\Lambda))$. A weighted motion-based flicker stabilization operator WM can then be defined as:

$$WMu_t(x) = \frac{1}{\widetilde{Z}_{t,x}} \int_s e^{-(t-s)^2/2\sigma^2} w_{t,x}(s, \varphi_{t,s}(x)) H^{-1}_{s,\Lambda_{\varphi_{t,s}(x)}}(H_{t,\Lambda_x}(u_t(x)))\, ds, \tag{3.3}$$

where $w_{t,x}$ is a weight function given by $w_{t,x}(s,y) = g\left(D^2_{\text{aff}}(u_t(x+\Lambda), u_s(y+\Lambda))\right)$, with $g : \mathbb{R}^+ \to \mathbb{R}^+$ decreasing, and $\widetilde{Z}_{t,x}$ a normalizing constant which ensures that the total sum of all weights is equal to 1, *i.e.*

$$\widetilde{Z}_{t,x} = \int_s e^{-(t-s)^2/2\sigma^2} w_{t,x}(s, \varphi_{t,s}(x))\, ds.$$

In practice, we used in our experiments the decreasing function $g(x) = e^{-\frac{x^2}{h^2}}$, where h is a parameter that has to be tuned by the user. This parameter plays the same role as the value of the threshold in soft thresholding denoising methods: when h is "large", all weights become almost equal, and thus the mean will mix everything together. On the other hand when h is "small", only points which look very similar to x will have a non negligible weight in the sum. We will again discuss the way to set the value of h in the experimental section at the end of the paper. Observe that other decreasing functions could be used for g, such as a hard thresholding function for example.

We will see in the experimental section at the end of the paper that this flicker stabilization procedure provides good results. One could argue that the motion estimation performed by Equation (3.1) is not always accurate. Indeed, because of the redundancy of images, and because the similarity measure we use is robust to affine contrast changes, a pixel can have several "good" matches, and it is then uneasy to find the right one among them. Following the idea of the NL-means method [2] introduced by Buades, Coll and Morel for movie denoising, one could think of taking advantage of the redundancy of images and thus, when a patch in u_t has several strong correspondences in u_s, just use all of them to stabilize the flicker. This is the idea of the stabilization method presented in the next section.

3.3. Multi-patch based flicker stabilization. As explained above, we can define a new flicker stabilization operator WP (Weighted multi-Patch stabilization operator) which is similar in its form to the NL-means method for movie denoising with however two main differences: we take

the mean of inverse cumulative histograms (instead of grey-levels directly) and we use the affine similarity measure D_{aff} (instead of the L^2 distance). In this approach, the grey level distribution of a patch $u_t(x+\Lambda)$ is replaced by a weighted mean (in the sense of Equation (2.1)) of the distributions of all patches $u_s(y+\Lambda)$ which are similar to $u_t(x+\Lambda)$ for the distance D_{aff}. The mean is taken only for $y \in W_{t,s}(x)$ where $W_{t,s}(x)$ is defined as above.

More precisely, the weighted multi-patch stabilization operator WP can be written

$$WPu_t(x) = \frac{1}{Z_{t,x}} \int_s C_{t,x}(s) \int_{y \in W_{t,s}(x)} w_{t,x}(s,y) H^{-1}_{s,\Lambda_y}(H_{t,\Lambda_x}(u_t(x)))\, dy\, ds, \tag{3.4}$$

where

- $C_{t,x}(s)$ is a normalizing constant which controls the total weight of u_s. For instance, if we wish to have Gaussian weights (as it was the case in Equation (2.1)), we just set

$$C_{t,x}(s) = \frac{e^{-(t-s)^2/2\sigma^2}}{\int_y w_{t,x}(s,y)\, dy}.$$

  Another possibility is to normalize each frame in such a way that its total weight is $\max_y w_{t,x}(s,y)$: this can be useful whenever there is a change of sequence in the movie;
- $Z_{t,x}$ is the final renormalization constant: it ensures that the total sum of all weights is 1,

$$Z_{t,x} = \int_s C_{t,x}(s) \int_y w_{t,x}(s,y)\, dy\, ds.$$

One of the main interests of this method relying on weighted patches is that it does not rely as heavily on motion estimation as the weighted motion one. If the search window $W_{t,s}(x)$ is fixed, the method does not rely on any motion estimation at all. The motion being taken indirectly into account through the weights $w_{t,s}(x)$, this approach can appear as more elegant than the approach involving the operator WM (Equation (3.3)). Unfortunately, it provides results which are from far not as good as the ones provided by WM. As we will see in the experimental section, it can even create artefacts in the image sequences when the parameters are not perfectly well chosen.

3.4. Links between WM and WP. The operator WM can be seen as the limit of WP when we introduce a real parameter p that we let go to infinity. More precisely, if we define the operator $WP^{(p)}$ by

$$WP^{(p)}u_t(x) = \frac{1}{Z^{(p)}_{t,x}} \int_s \frac{e^{-(t-s)^2/2\sigma^2}}{\int_y w^p_{t,x}(s,y)\, dy} \int_{y \in \{W_{t,s}(x)\}} w^{p+1}_{t,x}(s,y) H^{-1}_{s,\Lambda_y}(H_{t,\Lambda_x}(u_t(x)))\, dy\, ds,$$

where

$$Z^{(p)}_{t,x} = \int_s \frac{e^{-(t-s)^2/2\sigma^2}}{\int_y w^p_{t,x}(s,y)\, dy} \int_{y \in \{W_{t,s}(x)\}} w^{p+1}_{t,x}(s,y)\, dy\, ds,$$

then $WP^{(0)} = WP$ and $WP^{(p)}u_t(x)$ goes to $WMu_t(x)$ as p goes to infinity. This amounts to change h^2 in two different ways depending on the considered sum: it is changed into $h^2/(p+1)$ in the weighted sum of grey levels and it is changed into h^2/p in the normalization per frame. This result is analogous to the fact that in some sense "the L^∞ norm is the limit of the L^p norm as p goes to infinity".

3.5. Setting the parameters. The different parameters (the parameter h in the decreasing function $g(x) = e^{-\frac{x^2}{h^2}}$, the size of Λ and the size of W) used in the previous stabilization procedures are important and should be chosen carefully.

On the size of the patches $|\Lambda|$. If $|\Lambda|$ is too large, the patches can contain different moving objects and the estimation of the similarity measure $D_{\text{aff}}(u_t(x+\Lambda), u_s(y+\Lambda))$ can be unreliable. However, if $|\Lambda|$ is too small, the information in the patches can be poor and the measure $D_{\text{aff}}(u_t(x+$

$\Lambda), u_s(y+\Lambda))$ can not be trusted either. A good choice for the size $|\Lambda|$ is clearly related to the resolution of the frames and the size of the objects moving in the sequences. In all the experiments presented in this paper, Λ was set as a 21×21 square window.

On the choice of the parameter h. The parameter h is related to the values taken by the similarity measure D_{aff} on the image. Since the weights are given by $\exp(-D_{\text{aff}}(\cdot,\cdot)^2/h^2)$, h can be seen as a soft threshold. It can also be seen as a regularization scale for the flicker. If h is large, then all weights in (3.4) will be close to 1. In this case, the flicker is efficiently stabilized, but some artefacts can appear since patches which can have very dissimilar geometries are mixed together. Conversely, when h is small, only patches which are very similar to the considered frame will have non negligible weights in the sum. But then, flicker is not as efficiently stabilized. In practice, we observed that values of h between 10 and 30 (the distance D_{aff} between patches being normalized by the size of the patches) generally yield good results on most sequences.

On the size of the search window $|W|$. The size of the search window W should be large enough to follow a pixel in the presence of a large motion. On the other hand, if W is too large, there is a risk that patches corresponding to different objects, with the same geometry but different contrast, will be mixed together. Now, a very simple and elementary requirement for the choice of W (and for the whole approach in general) is that if a movie u does not contain any motion nor any flicker (which means that all frames are the same - let v denote one of these frames), then it should be invariant under the operators WM and WP. To satisfy such a requirement, we need to ensure that for any pixel x, a patch $v(y+\Lambda)$ with its center y in the window $x+W$ and which is similar (according to the similarity measure D_{aff}) to the patch $v(x+\Lambda)$ will not create "new" grey level values at x when equalizing the grey level histograms. In other words, if we consider the operator $\mathcal{T}$ defined on a single image v by

$$\mathcal{T}v(x) = \frac{\int_{x+W} w(x,y) H^{-1}_{\Lambda_y}(H_{\Lambda_x}(v(x)))\, dy}{\int_{x+W} w(x,y)\, dy}, \tag{3.5}$$

where $w(x,y) = \exp(-D_{\text{aff}}2(v(x+V), v(y+V))/h^2)$, then we should have $\mathcal{T}v \simeq v$.

PROPOSITION 3.2. *In order to avoid any "ringing" effect in the neighborhood of edges, we have to set*

$$W \subset 2\Lambda.$$

Proof. Let v be the image of a noisy edge, as defined in Proposition 3.1. We assume that the image domain is $\{-N,\dots,-1,0,1,\dots,N\}^2$ and that v is given by: $v(x) = v(x_1,x_2) = \alpha + \sigma n(x)$ if $x_1 < 0$, and $v(x) = \beta + \sigma n(x)$ if $x_1 \geq 0$, where α and β are two grey levels, σ is the standard deviation of the noise, and the $n(x)$ are i.i.d. $\mathcal{N}(0,1)$.

We then consider a patch $v(x+\Lambda)$ where $\Lambda = \{-K,\dots,-1,0,1,\dots,K\}^2$. By symmetry, we will just compute what happens on the right side of the edge (*i.e.* in the domain $x_1 \geq 0$). Let $r(x) = H_{\Lambda_x}(v(x))$. By definition, it is the rank of $v(x)$ in the set of all $v(x+z)$ with $z \in \Lambda$. If $x = (x_1,x_2)$ with $x_1 \geq K$, then the patch $v(x+\Lambda)$ doesn't intersect the edge. It is thus an almost constant patch (up to the additive noise), and then $r(x)$ is uniformly distributed between 1 and $|\Lambda| = (2K+1)^2$.

The affine similarity measure between $v(x+\Lambda)$ and another patch $v(y+\Lambda)$ can be easily computed (in a way similar to what we did in the proof of Proposition 3.1), and we obtain that: $D^2_{\text{aff}}(v(x+\Lambda), v(y+\Lambda)) \simeq \sigma^2$ when $v(y+\Lambda)$ is another constant patch (that is when $y_1 \geq K$ or $y_1 < -K$). And $D^2_{\text{aff}}(v(x+\Lambda), v(y+\Lambda)) \simeq p_y(1-p_y)(b-a)^2 + \sigma^2$ when $v(y+\Lambda)$ is a patch containing the edge and where $p_y = (K-y_1)/(2K+1)$ is the proportion of $\simeq \alpha$ values in the patch.

On the other hand, we can also compute the expected values for $H^{-1}_{\Lambda_y}(H_{\Lambda_x}(v(x)))$ and get:

$$H^{-1}_{\Lambda_y}(H_{\Lambda_x}(v(x))) = H^{-1}_{\Lambda_y}(r(x)) = \begin{cases} \alpha & \text{if } y_1 < -K, \\ \beta & \text{if } y_1 \geq K, \\ \alpha p_y + \beta(1-p_y) & \text{otherwise.} \end{cases}$$

Let now W be the neighborhood used in the definition of the operator $\mathcal{T}$ in Equation (3.5). We will denote it by $W = \{-m, \dots, -1, 0, 1, \dots, m\}^2$.

If $m > 2K$, then there will exist points x such that $K \leq x_1 < m - K$. For all these points, since $x_1 - m < -K$ their neighborhood $x + W$ will contain points y such that $y_1 < -K$. Thus, since the similarity measure between $v(x + \Lambda)$ and all other patches $v(y + \Lambda)$ such that $y_1 < -K$ or $y_1 \geq K$ is the same and smaller than the similarity measure to edge patches, the grey level $\mathcal{T}v(x)$ will be a weighted mean of α and β values, and will not remain close to $v(x) \simeq \beta$. This effect can clearly be observed on Figure 3.5. Thus, in order to avoid creating such artefacts, we have to set $m \leq 2K$, in other words $W \subset 2\Lambda$.

On the other hand, points x which are such that the patch $v(x + \Lambda)$ contains the edge will have an almost unchanged grey level because there are no patches in the image which will have the same geometry but with very different grey level values.

The typical edge profiles that we obtain after applying Equation (3.5) is shown on Figure 3.4. □

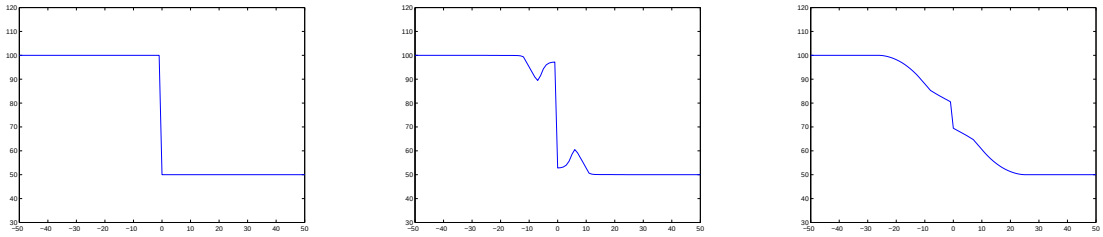


FIG. 3.4. *Result of Equation* (3.5) *when* W *is larger than* Λ. *In this experiment, the size of* Λ *is* 15×15 *pixels and the size of the search window* W *is* 37×37. *Left: profile of a pure edge* v. *Middle: profile of* $\mathcal{T}v$ *when* $h = 10$. *This profile shows some "ringing phenomenon". Right: profile of* $\mathcal{T}v$ *when* $h = 100$. *In this case, since* h *is large, all patches have an almost equal weight, whatever their geometry.*

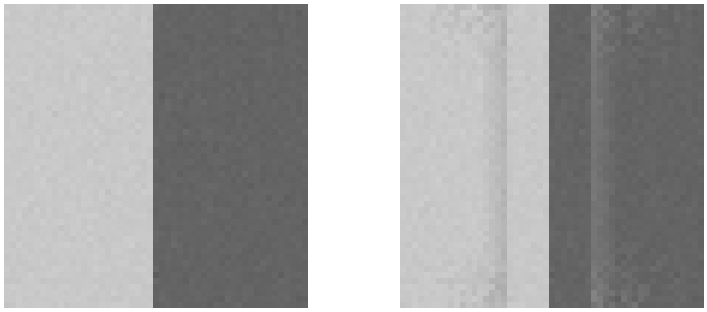


FIG. 3.5. *Left: image* v *of a noisy edge (size* 67×67 *pixels). Right: image* $\mathcal{T}v$ *when the size of* Λ *is* 15×15 *pixels and the size of the search window* W *is* 37×37. *The search window is too large compared to the size of the patches, and some artefacts appear: in particular, we can notice some kind of "ringing phenomenon" around the edge.*

4. Experiments and discussion. This section presents the results of the proposed stabilization procedures on several sequences, containing real or synthetic flicker. The sequences and the corresponding results are available at `http://www.tsi.enst.fr/~delon/Demos/Flicker_stabilization/`.

4.1. Implementation. Equations (3.3) and (3.4) can be implemented independently on each pixel x of u_t. However, the computation of the weighted motion for the neighborhood Λ_x of each pixel x in Ω can take time. In order to accelerate the process, the stabilization procedure can be implemented by blocks. The domain Ω is divided into a fixed grid of overlapping blocks Λ_i, $i = 1, \dots, N_B$. Equation (3.3) (or (3.4)) is then used on each of these blocks to stabilize the flicker of all pixels contained in the block (and not only the central pixel of the block). If a pixel x belongs

FIG. 4.1. *Left: a frame of a sequence containing no visible flicker. Center: same frame, after applying Equation* (3.3) *with* $h = 10$ *to the whole sequence. Right: difference between both frames, rescaled from* $[0, 15]$ *to* $[0, 255]$ *for better visualization.*

to several blocks $\Lambda_{i_1}, \ldots, \Lambda_{i_x}$, the restored grey levels obtained at x for each Λ_{i_j} are averaged to obtain the final restored value at x. This averaging regularization is a way to avoid block effects.

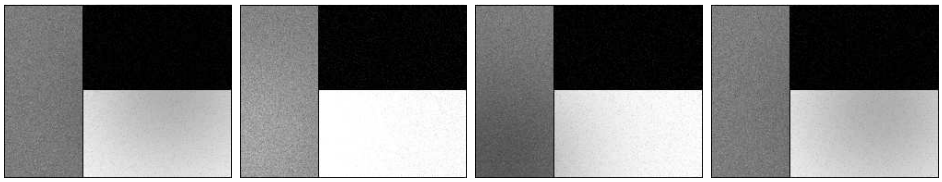
The following experiments are obtained using this block-based implementation. In these experiments, the block (or patch) Λ is chosen as a square of size 21×21. The overlap between these blocks is chosen as half the size of a block, *i.e.* as 10 here. The search window W is also chosen as 21×21, which is large enough for the motion observed in the sequences presented here and which permits to satisfy the condition of Proposition 3.2. The temporal standard deviation σ is set to 5, which in practice means that the temporal neighborhood taken into account in the restoration of u_t is approximately $[t - 2\sigma, t + 2\sigma] = [t - 10, t + 10]$ (10 images before t and 10 images after t).

4.2. Effects on a flicker-free sequence. A crucial property of a flicker stabilization procedure is its ability to leave intact a sequence which does not contain any flicker. Figure 4.1 illustrates this property. On the left, one can see a frame extracted from a sequence containing several artifacts but no visible flicker. After applying Equation (3.3) to the whole sequence with $h = 10$, the same frame in the modified sequence is shown in the center of the figure. The difference between both frames is shown on the right, rescaled from its range $[0, 15]$ to $[0, 255]$ for better visualization. The two images are almost identical and the mean grey level difference between both images is 0.79. If the sequence is stabilized with $h = 30$, the mean grey level difference becomes 0.91, which is still quite small. Although this mean difference remains moderate, a slight difference can be perceived at some very specific locations. These zones correspond to small blotches, caused by dirts on the film support. These blotches are present in this frame but do not correspond to any information in other frames of the movie. They tend to disappear under the stabilization procedure, since the grey level distribution of their neighborhood is mixed with neighborhoods in other frames which do not contain blotches. As a consequence, these blotches appear clearly when we compute the difference between the original sequence and the regularized one.

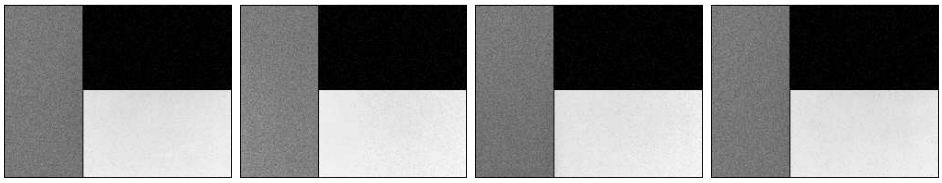
4.3. Synthetic flicker. One of the worst difficulties of old film restoration is that most of the films suffer from several defects (blotches, scratches, flicker) at the same time. As it can be noticed on the example of Figure 4.1, the restoration of a single defect is not completely independent of the presence of the others. For this reason, we begin our experiments with two sequences containing no such defects. The first sequence, called *flag* is build from a single 247×188 frame, to which we added independent samples of a white Gaussian noise random image. This sequence, on purpose, does not contain any motion. The second sequence is the classical sequence *taxi* [1], generally used in motion estimation benchmarks. In this sequence, the size of the frames is 256×191.

To each sequence, we added two kinds of synthetic flicker. The first one consists in a random multiplicative and local flicker, placed on random positions along the sequence. More precisely,

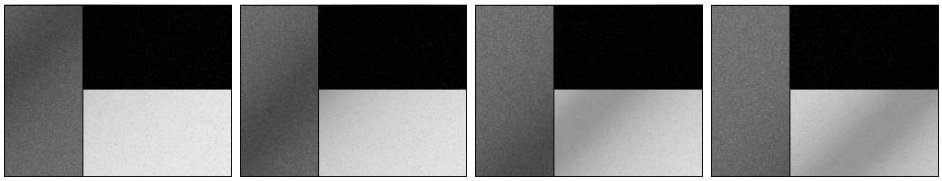
[1] This sequence is available at `http://i21www.ira.uka.de/image_sequences/`.



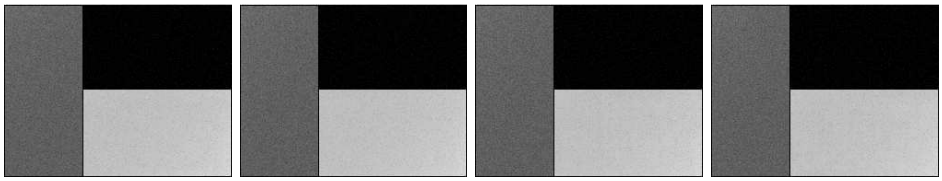
(a) Random local flicker



(b) Same frames after flicker stabilization with WM, $h = 30$



(c) Motion-coherent local flicker



(d) Same frames after flicker stabilization with WM, $h = 30$

FIG. 4.2. *(a) Four frames of the flag sequence, affected by the random flicker given by Equation* (4.1)*. (b) Same frames after using Equation (3.3) to remove the flicker. (c) Four frames of the flag sequence, affected by a motion-coherent flicker given by Equation* (4.2) *(a dark strip that goes through the images). (d) Same frames after using Equation (3.3) to remove the flicker. One can notice that in both cases, since we only perform changes of contrast, there is no spatial smoothing, and thus the noisy texture of the movie remains intact.*

each frame u_t of size $N_c \times N_r$ becomes

$$\forall (k,l) \in \Omega, \quad \tilde{u_t}(k,l) = u(k,l) \times \left(1 + S_t e^{-\frac{(k-X_t)^2+(l-Y_t)^2}{r^2}}\right), \tag{4.1}$$

where S_t, X_t and $_t$ are independent and $S_t \sim \mathcal{U}[-0.3, 0.3]$, $X_t \sim \mathcal{U}[0, N_c]$, $Y_t \sim \mathcal{U}[0, N_r]$; and where the radius r is fixed to 100. This flicker has no motion coherence (see Figure 4.2 (a) and 4.3 (a)). The frames become randomly locally darker or lighter. The second kind of flicker consists in a multiplicative dark transparent strip moving from the top left to the down right of the frames as the movie is played. More precisely, a frame u_t becomes

$$\forall (k,l) \in \Omega, \quad \tilde{u}_t(k,l) = u(k,l) \times \left(0.6 + 0.4 e^{-\frac{w}{0.5+k+l-(t+1)\gamma}}\right), \tag{4.2}$$

where w is the width of the strip and γ is the speed of the flicker motion. The width w is set to 60 for both sequences and the speed s is set to 40 for the *flag* sequence and to 15 for the *taxi* sequence. This flicker is motion-coherent (see Figures 4.2 (c) and 4.3 (c)).

The four resulting sequences and the results of the WM operator (Equation (3.3)) using the implementation described in Section 4.1 can be seen at the address `http://www.tsi.enst.fr/~delon/Demos/Flicker_stabilization/`. All the parameters are set as described in Section 4.1

(a) Random local flicker

(b) Same frames after flicker stabilization with WM, $h = 30$

(c) Motion-coherent local flicker

(d) Same frames after flicker stabilization with WM, $h = 30$

FIG. 4.3. *(a) Four frames of the* taxi *sequence with an additional random local flicker, given by Equation* (4.1). *(b)Same frames after applying Equation (3.3) to remove the flicker. (c) Four frames of the* taxi *sequence with an additional local flicker, looking like a dark transparent strip moving across the frames, given by Equation* (4.2). *(d) Same frames after applying Equation (3.3) to remove the flicker.*

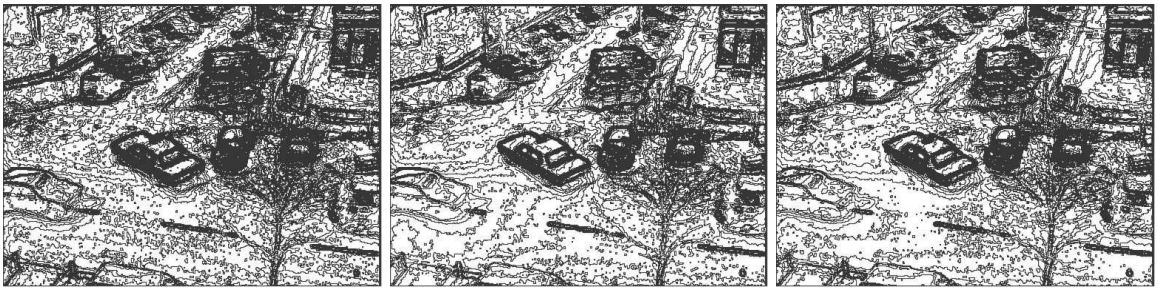


FIG. 4.4. *Left: level lines of a frame of the original taxi sequence. the figure shows all level lines at a sample rate of* 5 *grey levels. Center: level lines of the same frame in the sequence containing the motion-coherent flicker. Right: level lines of the same frame in the sequence restored thanks to Equation* (3.3).

and h is set to 30. Some frames extracted from the different sequences and the corresponding frames in their restored versions can also be seen on Figures 4.2 and 4.3. Observe that the restored sequences are not identical and depend strongly on the flicker that we try to eliminate. For instance, the contrast of the frames in Figure 4.2 (b) (resp. Figure 4.3 (b)) is larger than the contrast of the frames in 4.2 (d) (resp. Figure 4.3 (d)). Since the flicker has destroyed some contrast information, we cannot hope to recover completely the contrast of the original sequence,

but only to filter it enough to make it barely noticeable. Observe also on the *flag* sequence how the flicker disappears while the noisy texture remains intact. This is due to the fact that only changes of contrast are performed, without spatial smoothing.

Obtaining satisfactory results with the WP approach on these sequences is trickier. Indeed, since this formulation makes use of all patches of a neighborhood to stabilize the contrast of a given block, its results highly depend on the choice of the different parameters, and especially on the choice of the parameter h. Most often, a given patch has several "good correspondences" in each of the neighboring frames, which can all be reasonably used to stabilize the contrast of the patch. For these patches, the operators WM and WP yield similar results. However, some patches are singular, in the sense that they only have one (or zero) good corresponding patch per neighboring frame. For these singular patches, it should be ensured that the weights of all the non-similar patches in Equation (3.4) are negligible. This is illustrated by Figure 4.5, which shows the kind of artefacts that can appear when these weights are not well chosen. This implies that h should be set carefully. In practice, satisfactory results, however not as satisfactory as the ones obtained with WM, could be obtained with $h = 5$ on the previous sequences. However, it should be noted that for a really strong flicker, choosing $h = 5$ can be insufficient to remove the flicker completely. In these cases, we strongly recommend to use the WM formulation with a higher value of h.

FIG. 4.5. *Left: center of a frame of the flag sequence with the random local flicker, after a local stabilization by WP, with $h = 30$. For this value of h, some artefacts appear around the singular zones (here the area around the intersection between the three regions). Right: extract of a frame of the taxi sequence with the motion-coherent flicker, after a local stabilization by WP, with $h = 100$. Observe the artefacts around the zones of motion.*

It is generally not easy to evaluate the performance of a flicker stabilization method because in most cases there is no available ground truth. However, in the case of a synthetic flicker, we can compare a frame of the original sequence with the same frame in the flickering sequence and with the same frame after flicker stabilization. In order to measure the ability of the stabilization procedure to recover the geometrical information of the original sequence, we compare the topographic map of an original frame with the one of the corresponding restored frame. This is illustrated by Figure 4.4. On the center, we notice that the local flicker has modified some level lines. In particular, it has created level lines which are parallel to the dark transparent strip (see below the white car, on the right of the black car). We also see that the most contrasted level lines, which correspond to the salient objects in the image are not modified. On the right of the figure, we show the level lines after flicker stabilization by WM. The level lines created by the flicker have disappeared in this restored frame.

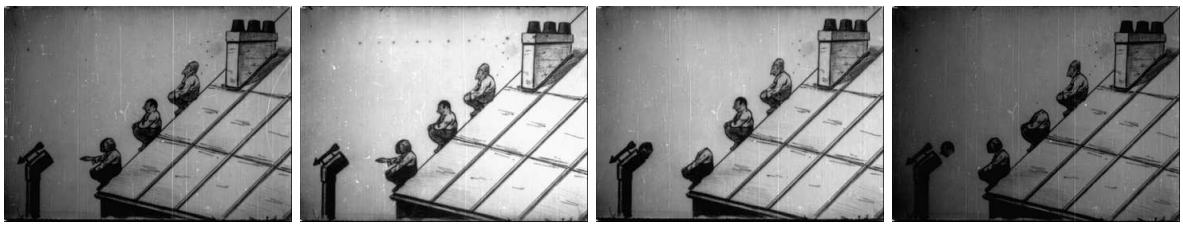
4.4. Sequences extracted from old movies. We finally show the results of the WM operator on two sequences extracted from old movies. These sequences can be also seen at `http://www.tsi.enst.fr/~delon/Demos/Flicker_stabilization/`. Some frames extracted from the original and stabilized sequences are shown on Figures 4.6 and 4.7. Here, the flicker has not been artificially added: it is naturally present in these old movies. In the first sequence, shown on Figure 4.6, the flicker is a mix of global contrast changes and of a dark transparent strip going through the images from right to left when the movie is being played (the strip is placed on the face of the man on the middle frame in the figure). This sequence can thus be compared to the

synthetic sequences created in the previous section with the motion-coherent flicker. After applying the WM operator, with $h = 30$, the other parameters being set as described in Section 4.1, the flicker is not visible anymore.

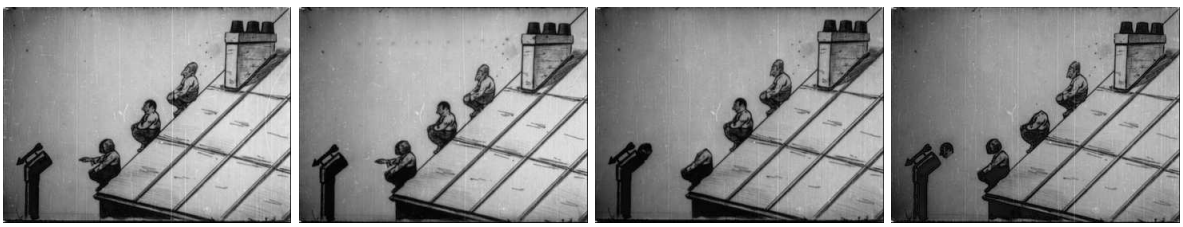
(a) Original sequence

(b) Sequence stabilized with WM, $h = 30$

FIG. 4.6. *(a) Three images of an old movie suffering from a strong local flicker. This flicker is visible as a dark transparent strip going through the images from right to left when the movie is being played. The strip is placed on the face of the man on the middle frame. (b) Same frames after flicker stabilization by Equation (3.3), with $h = 30$. Other parameters are set as described in Section 4.1.*



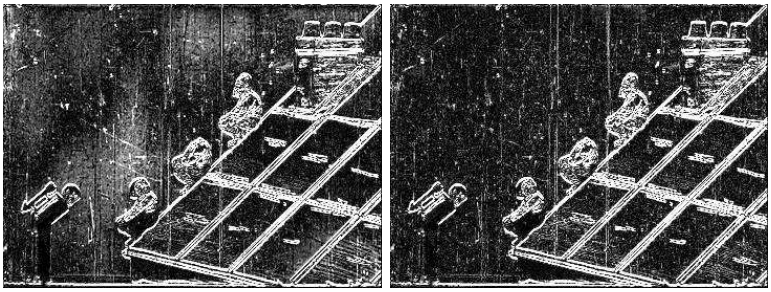
(a) Original sequence



(b) Sequence stabilized with $STE + WM$, $h = 30$

FIG. 4.7. *(a) Four images of an old movie Les Aventures des Pieds Nickelés, Emile Cohl/Eclair (1917-1918, copyright: Marc Sandberg) suffering from local flicker. (b) Same frames after a global correction by STE, followed by a local stabilization by WM, with $h = 30$. Other parameters are set as described in Section 4.1.*

The second sequence is as short extract from the movie *Les Aventures des Pieds Nickelés*, Emile Cohl/Eclair (1917-1918, copyright: Marc Sandberg). In this sequence, shown on Figure 4.7 (a), the flicker has a strong global component, although local variations can also be perceived along the sequence. As a consequence, we applied first the global operator STE (Equation (2.2)) to this sequence, with a large value of σ to get rid of these strong global contrast changes along the sequence. After this global stabilization, local and fast fluctuations remained visible, taking the shape of large bright spots at different locations on the frames. This is illustrated by Figure 4.8 (a), which shows the absolute difference between two successive frames of the sequence after the global correction. The visible differences are not only due to motion (or film shaking in this case) but appear also as large bright spots at different places. As shown in Figures 4.7 (b) and 4.8 (b),



(a) After STE (b) After $STE + WM$

FIG. 4.8. *Left: difference between two successive frames of Les Aventures des Pieds Nickelés after a global flicker stabilization thanks to the operator STE (Equation (2.2)). Right: difference between the same frames after an additional local stabilization by WM, with $h = 30$. In both differences, a ratio of 10% pixels with maximal values has been put to the maximal value for better visualization.*

applying WM with $h = 30$ to this sequence enabled us to remove these spots and the remaining flicker.

5. Conclusion. In this paper, a new procedure to stabilize the flicker in image sequences has been proposed. The method relies on a similarity measure between image patches that has been demonstrated to be at the same time robust to affine contrast changes and discriminant enough. This similarity measure permits to roughly estimate the motion in the movie despite the presence of flicker. The main stabilization operator introduced in this paper, called WM, makes use of this similarity measure to find the correspondences of a given patch and replaces the grey level distribution of the patch by a weighted mean of the grey level distributions of all these corresponding patches, the weights being chosen as a decreasing function of the similarity measures between patches. The ability of this operator to deal with strongly localized flicker has been demonstrated on several synthetic and real examples. In particular, since this operator only performs contrast changes on image patches, and does not involve spatial smoothing, it is able to preserve fine spatial features such as noise or textures in the frames.

The perspectives of this work are twofold. First, we believe that this approach should not be restricted to old movie restoration and could be of great interest for other types of sequences, such as biological sequences of evolving cells, in which the time sampling is very sparse and the frames contain large local intensity variations. Second, we plan to study in more detail the theoretical aspects of the method. In particular, the different operators introduced (WM and WP for movies, and $\mathcal{T}$ for a single image) should certainly find statistical interpretations, in the same vein as what has been done for the NL-means operator [2].

REFERENCES

[1] L. Birgé. The Grenander estimator: A nonasymptotic approach. *The Annals of Statistics*, 17(4):1532–1549, 1989.

[2] A. Buades, B. Coll, and J.-M. Morel. Nonlocal image and movie denoising. *Int. J. Comput. Vision*, 76(2):123–139, 2008.

[3] V. Caselles, B. Coll, and J.-M. Morel. Topographic maps and local contrast changes in natural images. *International Journal of Computer Vision*, 33(1):5–27, 1999.

[4] V. Caselles, J.-L. Lisani, J.-M. Morel, and G. Sapiro. Shape preserving local histogram modification. *IEEE Transactions on Image Processing*, 8(2):220–230, 1999.

[5] V. Caselles and P. Monasse. *Geometric Description of Topographic Maps and Applications to Image Processing*. Lecture Notes in Mathematics. Springer, 2009. To Appear.

[6] E. Decencière. *Restauration automatique de films anciens*. PhD thesis, École Nationale Supérieure des Mines de Paris, 1997.

[7] J. Delon. Midway image equalization. *Journal of Mathematical Imaging and Vision*, 21(2):119–134, 2004.

[8] J. Delon. Movie and video scale-time equalization. *IEEE Transactions on Image Processing*, 15(1):241–248, 2006.

[9] T. Lindeberg. On the axiomatic foundations of linear scale-space: Combining semi-group structure with

causality vs. scale invariance. In K. A. Publisher, editor, *Gaussian Scale-Space Theory*, pages 75–98, 1997.

[10] J.-L. Lisani and J.-M. Morel. Detection of major changes in satellite images. In *International Conference on Image Processing (ICIP)*, pages 941–944, 2003.

[11] V. Naranjo and A. Albiol. Flicker reduction in old films. In *International Conference on Image Processing (ICIP)*, volume 2, 2000.

[12] F. Pitié, R. Dahyot, F. Kelly, and A. Kokaram. A new robust technique for stabilizing brightness fluctuations in image sequences. In *2nd Workshop on Statistical Methods in Video Processing In conjunction with ECCV 2004*. Springer, 2004.

[13] F. Pitié, B. Kent, B. Collis, and A. Kokaram. Localised deflicker of moving images. In *Proceedings of the 3rd IEE European Conference on Visual Media Production (CVMP'06)*, pages 134–143, 2006.

[14] P. Richardson and D. Suter. Restoration of historic film for digital compression: A case study. In *International Conference on Image Processing (ICIP)*, volume 2, 1995.

[15] T. Saito, T. Komatsu, T. Ohuchi, and T. Seto. Image processing for restoration of heavily-corrupted old film sequences. In *15th International Conference on Pattern Recognition*, volume 3, pages 13–16 vol.3, 2000.

[16] D. Suter and P. Richardson. Historical film restoration and video coding. In *Proceedings of the Picture Coding Symposium*, pages 389–394, 1996.

[17] P. van Roosmalen, R. Lagendijk, and J. Biemond. Correction of intensity flicker in old film sequences. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7):1013–1019, 1999.

[18] C. Villani. *Topics in optimal transportation*. American Mathematical Society, 2003.

[19] K. Wong, A. Das, and M. Chong. Improved flicker removal through motion vectors compensation. In *Third International Conference on Image and Graphics*, pages 552–555, 2004.