A Variational Method for Flicker Removal in High Speed Video Sequences

Ali Kanj, Hugues Talbot, Jean-Christophe Pesquet, Raoul Rodriguez Luparello

A Variational Method for Flicker Removal in High Speed Video Sequences

Ali Kanj, Hugues Talbot, Jean-Christophe Pesquet, and Raoul Rodriguez Luparello

Abstract

Deflickering of color image sequences constitutes a fundamental task in many video applications. Recently, it has received increased attention with the popularity of consumer-level high speed video acquisition devices, for instance in newer smartphone generations or sports/action cameras. These are capable of acquiring videos at more than 100 frames per second. In this context, flicker corresponds to undesirable intensity and chroma variations due to the interaction between the acquisition frequency on the one hand, and the alternating current powering artificial light sources on the other. Non-uniform periodic flicker is not easy to correct since its estimation requires object tracking, and since most tracking techniques assume consistent illumination. In this paper, we adopt a new optimization approach to solve this problem. We propose causal and non-causal robust correction schemes using a block matching technique paired with color variation estimation. A post processing step is introduced to remove block artifacts in an efficient manner. A multi-resolution approach is also developed to reduce computation time without affecting the quality of our results. We demonstrate the efficacy of our method both on simulated and real data.

Index Terms

High speed imaging, illumination variation, color correction, motion compensation, mutliresolution, Majoration-Minimization.

I. Introduction

In video processing, flickering is generally defined as an undesirable appearance fluctuation in an image sequence. This effect can arise at any acquisition frequency and it can be caused by a wide variety of phenomena such as sensor artifacts, video transmission problems, illumination changes, and more. Illumination can change over time for desirable reasons, e.g. a light appliance being turned on or off in a movie. Flicker can be transient or periodic, it may affect the whole frame or just a small portion of it.

In this work, we investigate a specific form of flicker, termed periodic flicker, which is more prevalent in high-speed video. When the acquisition frame rate of an artificially lit sequence approaches or exceeds twice the frequency of the AC current (100 or 120 Hz in most countries), then illumination artifacts may become visible, typically over the whole frame. These include not only luminance but also chrominance changes. If the motion in the sequence is simple and the illumination comes from similar sources, then periodic flicker may be easy to estimate and remove. However, when the motion in the sequence is complex and the scene is illuminated from several sources with different properties (incandescent, fluorescent, etc), or itself contains lighting appliances, the problem of flicker removal becomes highly challenging. Saturation and other sensor effects can produce additional unsightly effects. This problem has become more critical with the advent of consumer-level high speed video acquisition devices, such as in newer smartphone generations or sports/action cameras.

Methods for video luminosity stabilization have been studied in the literature, for instance for underwater image sensing, surveillance systems, camcorder videos, archived videos, image-video compression and time lapse videos. Modeling the flicker effect differs from an application to the other. For instance, sunflicker effect, constitutes a big challenge in underwater image sensing. This non-periodic effect is created from refracted sunlight casting fast moving patterns on the seafloor. Shihavuddin et al. [1] proposed a removal method which consider sunflicker effect as a dynamic texture. They proposed to warp the previous illumination field to the current frame so as to predict the current one, and then to find the homography between the previous and current frames in order to finally remove sunflicker patterns from images. Flicker is also present in indoor smart surveillance cameras in the presence of fluorescent lamps, complicating feature tracking. Ozer et al. [2] proposed to eliminate the background which contains reflective surfaces and shadows, and to track the regions of interest only. Camcorded videos are flicker-affected when scrolling stripes are observed along the video. This problem is due the fact that the higher frequency backlight of the screen is sampled at a lower rate by the camcorder. In this case, flicker is defined by a periodic rectangular signal. Baudry et al. [3] used the temporal discrete Fourier transform (DFT) to estimate the flicker parameters.

We can classify existing methods for flicker removal in archived videos into two categories depending on whether they use a linear or a non-linear model. Decencière [4] proposed a linear model linking the observed image to the original image. Rosmalen et al. [5] used also an affine model taking into account spatial dependency. In [5], [6], flicker parameters are estimated by

A. Kanj and H. Talbot (corresponding author) are with the Laboratoire d'Informatique Gaspard Monge, UMR CNRS 8049, Université Paris-Est, 77454 Marne la Vallée Cedex 2, France. E-mail: first.last@univ-paris-est.fr.

J.-C. Pesquet is with the Center for Visual Computing, INRIA, CentraleSupélec, 91192 Gif sur Yvette, France. E-mail: jean-christophe@pesquet.eu.

R. Rodriguez Luparello is with SUBBLAB Production, France and Spain. subraoul@gmail.com.

interpolation, but these methods may fail to resolve block mismatching problems caused by occlusions or blotches due to outliers. Ohuchi et al. [7] used the same affine model together with an M-estimator to find flicker parameters. They considered objects in motion as outliers, which may lead to failures in the presence of large objects. Kokaram et al. [8] proposed a model to reduce the accumulating error taking the previous restored frame as a reference. They used also a linear model with a robust regression together with registration. Zhang et al. [9] offer a method to generate reference images followed by a proposal to correct flicker by a linear model involving an M-estimator. Non-linear models have also been studied. Among them, [10], [11] find flicker parameters by optimizing a non-linear model using histogram matching. Pitié et al. [12] improved on Naranjo's model by taking spatial variations into account. In [13], authors expanded these works by developing a method for finding parameters at the pixel level, allowing for non-linear flicker distortions. Separately, [14], [15], [16] computed differences of grey-scale histograms between degraded and reference images. In image sequence coding, flicker may appear when similar regions between two images are incoherently encoded. Ren et al. [17] proposed a block-matching, adaptive multiscale motion method to address this problem. This method may encounter problems in presence of thin objects or outliers and tends to smooth image details. A low pass filtering approach is used in [18] to compensate for flicker. It is also worth mentioning the work by Fuh et al. [19], where a model is developed for dealing with affine shape and intensity transformations. A pixel-recursive model was also developed in [20] to estimate simultaneously motion and illumination variation parameters. Note that most of these works consider grey-level sequences and none specifically deal with periodic flicker.

In [21], we proposed a rigid registration method using point descriptors that is robust to illumination variations and which can be combined with a method for stabilizing brightness variations in image sequences using a Least Square adjustment approach. This deflickering method was designed for high speed videos in the presence of a single source illuminating the scene, by assuming a global displacement between frames. However, in general, the source consists of high definition image quality (HD), high-speed acquisition (up to 2000 frames per second) while the scene is illuminated by various artificial sources and contains complex motion. In these contexts, the method in [21] is no longer suitable. This paper aims at studying periodic flickering effects in these high speed imaging scenarios, and introduces a novel removal approach. Our method is based on a robust regression model, which is optimized to find flicker parameters on the one hand, and to remove blocks artifacts on the other. We also pay attention to the computation cost of the proposed approach which needs to be compatible with (almost) real-time processing requirements and an accelerated multiresolution procedure is subsequently proposed to reach this goal.

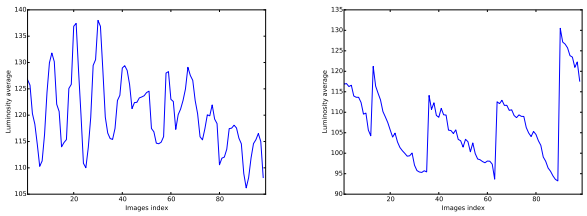
The rest of this paper is organized as follows. In Section II, we briefly formulate the flicker problem in high speed video acquisition. In Section III, we describe the assumptions we make in order to estimate the appropriate flicker correction model. In Section IV, we propose a causal method for video color correction based on a block matching approach and a novel block artifact removal scheme. Our approach leverages a novel Majorization-Minimization (MM) strategies. In Section V, we introduce a non-causal extension for video color correction allowing the use of two reference sources for each affected frame. In Section VI, an accelerated pyramidal approach is proposed. Section VII reports some experimental results, and finally, some conclusions are drawn in Section VIII.

II. Flicker analysis

A. Brightness variations

Flicker characteristics differ between applications. In archived movies, flicker is defined as unnatural temporal fluctuations in perceived image intensity that do not originate from the original scene, and which are often due to imprecisions in the shutter. These artifacts present global, sudden and *random* variations of luminance and contrast between consecutive frames of a sequence (see Figure 1(a)).

Time-lapse acquisitions are very slow and sometime irregular sequence acquisitions, that are used in animations, advertisement, or documentary films. They are typically used to accelerate rates of changes, for instance to show how a plant grows. Flicker may occurs when luminosity is varying depending on the time of day under natural lighting, or because of artificial lighting especially if the recording takes place at night and also according to the weather. This kind of flicker is defined as a correlated random luminosity variation (see Figure 1(b)), and usually it can be compensated in a causal manner refering to a past frame in each sequence portion which induces some luminosity distortions in the corrected sequence.



(a) Flicker in archived videos (b) Flicker in time-lapse videos

Fig. 1. Comparison between several types of flicker.

In high speed video acquisition, flicker corresponds to an undesirable *periodic* intensity variation due to the interaction between the lighting and acquisition frequencies (See Figure 5). Assuming negligible motion and a simple camera model, at the acquisition speed, each pixel value on a frame corresponds to the integration of a lighting function. Standard light sources such as incandescent or fluorescent lamps powered by an alternating current are subject to light flux variations at a frequency ν of about 100 or 120Hz[1]. At usual video rates (25-30 frames per second), these variations are integrated over a long enough period. Assuming approximately sinusoidal variations, we have at a given pixel $\mathbf{s}$ and time t:

$$\begin{aligned} f(\mathbf{s},t) &= \frac{f_0(\mathbf{s},t)}{T}\int_t^{t+T}\big(1+\Delta(\mathbf{s})\cos(2\pi\nu\tau+\varphi)\big)d\tau \\ &= f_0(\mathbf{s},t)\Big(1+\Delta(\mathbf{s})\frac{\sin(2\pi\nu(t+T)+\varphi)-\sin(2\pi\nu t+\varphi)}{2\pi\nu T}\Big) \\ &\sim f_0(\mathbf{s},t) \end{aligned} \tag{1}$$

where $T \gg 1/\nu$ is the exposure time, $\varphi \in [0, 2\pi[$ is a phase shift, $f_0(\mathbf{s},t)$ is the field intensity in the absence of illumination variations, and $\Delta(\mathbf{s})$ is the magnitude of these variations. Therefore, the intensity / chroma variations are usually not perceptible.

In contrast, at high acquisition frame rates, the integration of the lighting function is performed over a very short interval $T \ll 1/\nu$. Then, we get

$$f(\mathbf{s},t) \sim f_0\big(\mathbf{s},t\big)\big(1+\Delta(\mathbf{s})\big(\cos(2\pi\nu t+\varphi)-\pi\nu T\sin(2\pi\nu t+\varphi)\big)\big) \tag{2}$$

and the intensity variations are no longer negligible.

B. Chromatic variations

In the lighting field, the color temperature provides information on the general hue of the emitted light: from the "warm" hues - where the red is dominating - like sunrise or sunset colors, to the "cold" hues - where the blue color is dominating - like in the intense midday sun colors, or over a snow field. The color temperature is given usually in Kelvin (K), corresponding to the equivalent black-body temperature radiation. As the color temperature increases, so does the proportion of blue in light (see Figure 2).

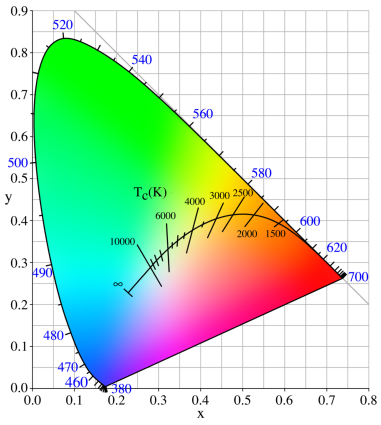


Fig. 2. Chroma alterations with light temperature variations.

With most light sources, power variations are accompanied with changes in lighting temperature, inducing visible periodic chroma alterations. For example, while using classical incandescent lamps (see Figure 3), the lamp filament temperature is proportionately varying to the power alterations, so with each AC period, the light color becomes first red and gradually transforms into blue.

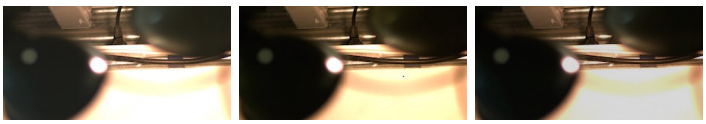


Fig. 3. Rapid illumination / chromatic changes in a high-speed video sequence under incandescent lighting.

Fluorescent and LED light sources, while based on a different light emitting principle, are also subject to high-speed chroma variations (see an example in Figure 4).

[1]Corresponding to twice the frequency of the alternating current.

Fig. 4. Illumination / chromatic changes in a high-speed video sequence under fluorescent lighting.

III. Assumptions for flicker correction

A. Vector model

To address this periodic illumination and color variation problem, we assume that we know the acquisition frame rate and the local mains AC frequency. We therefore can estimate the period of the flickering effect. We also assume that general intensity variations integrated over few flickering periods are slow. In Figure 5, the acquisition frame rate is 1000 images/second and the AC mains power frequency is 100Hz.

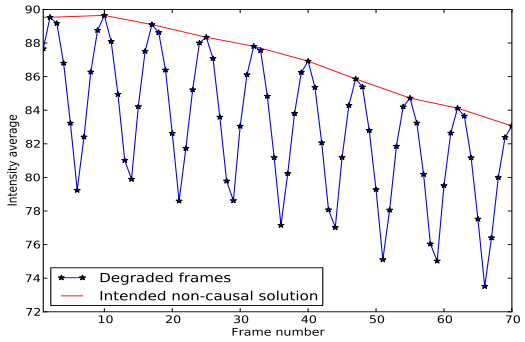


Fig. 5. The blue curve shows the luminosity variation in an affected image sequence. In red, we show the intensity variation in the absence of periodic flicker.

In order to check our assumption regarding the influence of periodic flicker on chroma variations, we acquired a sequence affected with periodic flicker due to a single source and no motion. In Figure 6, we plotted the cross-channel correlations between a pixel in a manually selected reference image and a subsequent frame in the same period. We observe that the vast majority of pixel colors transform linearly from the flicker-affected frame to the reference image, except at the extremal range values, due to sensor saturation effects.

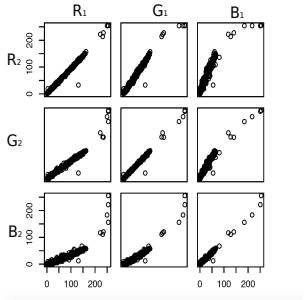


Fig. 6. Cross-channel correlation between a flicker-affected frame and a reference image.

In accordance with these observations, we subsequently assume a linear color transformation model between a reference and flicker-affected one. We model the color illumination transform by a 3×3 matrix $\boldsymbol{M}_{\mathbf{s},t}$ between two pixels $\mathbf{s}$ and $\mathbf{s}'$ referring to the same (possibly displaced) physical area in the scene at times t and t_{ref} respectively:

$$\boldsymbol{M}_{\mathbf{s},t} = \begin{bmatrix} r_1(\mathbf{s},t) & g_1(\mathbf{s},t) & b_1(\mathbf{s},t) \\ r_2(\mathbf{s},t) & g_2(\mathbf{s},t) & b_2(\mathbf{s},t) \\ r_3(\mathbf{s},t) & g_3(\mathbf{s},t) & b_3(\mathbf{s},t) \end{bmatrix}. \tag{3}$$

Using this matrix, we have

$$\mathbf{f}(\mathbf{s}', t_{\text{ref}}) = \boldsymbol{M}_{\mathbf{s},t}\, \mathbf{f}(\mathbf{s},t) + \mathbf{w}(\mathbf{s},t), \tag{4}$$

where $\mathbf{f}(\mathbf{s},t) \in \mathbb{R}^3$ is the vector of chrominance values of pixel $\mathbf{s}$ at time t and $\mathbf{w}(\mathbf{s},t) \in \mathbb{R}^3$ corresponds to some additive noise. This model plays a prominent role in the approaches developed in the remainder of the article.

B. Selecting reference frames

In theory, any frame in a flicker period could serve as a reference, but it is preferable for the reference frame to be as little affected as possible by the flicker effect. Practically, this means selecting the brightest frame over a period, at time t_{ref}, as a reference image. For this purpose, we use the average pixel intensity in each image along the sequence. As seen in Figure 5, changes in global illumination not related to flicker may only induce low frequency variations in the luminosity of the scene.

IV. Causal method for periodic flicker removal

In the case of multiple light sources illuminating the scene, the flicker effect is spatially variant, so we cannot model the illumination variation by a single matrix for all pixels in the frame. In addition, most videos contain background and moving objects, and so matching pixels between frames needs to be performed.

A. Tracking

Classical optical flow methods [22], [23] assume constancy of illumination and they are sensitive to the presence of illumination variations. In our approach, we propose a localized color transform estimation method. Since tracking single pixels is impractical, we use a block-matching method, assuming that each spatial block $\mathcal{B}_k$ indexed by k can be assigned its own model parameters, i.e. $(\forall \mathbf{s} \in \mathcal{B}_k)\ \boldsymbol{M}_{\mathbf{s},t} = \widetilde{\boldsymbol{M}}_{k,t}$, which means that all pixels indexed by $\mathbf{s}$ of the block $\mathcal{B}_k$ are processed using the same color transformation matrix $\widetilde{\boldsymbol{M}}_{k,t}$. More precisely, we propose to divide all frames into non-overlapping blocks of identical sizes, typically 64×64, and to perform a joint block-matching / color transform estimation between flicker-affected frames and the previous reference frame.

In accordance with our experimental model, for a given block $\mathcal{B}_k$ in frame t, the optimal transform matrix $\widehat{\boldsymbol{M}}_{k,t}$ and displacement vector $\widehat{\mathbf{d}}_{k,t} \in \mathbb{R}^2$ are those that minimize the following energy function:

$$J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) = \sum_{\mathbf{s} \in \mathcal{B}_k} \Phi\big(\widetilde{\boldsymbol{M}}_{k,t}\, \mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\text{ref}})\big), \tag{5}$$

where $\Phi \colon \mathbb{R}^3 \to [0, +\infty[$ is a cost function. In the following, an optimal value of $\widetilde{\boldsymbol{M}}_{k,t}$ in $\mathbb{R}^{3\times 3}$ will be sought while the optimal value of the candidate displacement vector $\mathbf{d}_{k,t}$ between two corresponding blocks will be searched within a finite set $\mathcal{W} \subset \mathbb{R}^2$ (see Figure 7).

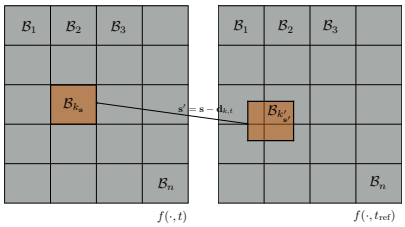


Fig. 7. Block matching scheme.

In the next section, we describe an iterative method allowing to solve this optimization problem for a wide class of cost functions Φ.

B. Iterative optimization strategy

In the optimization of a cost function $g(\mathbf{u})$ with respect to parameters $\mathbf{u}$, we seek an estimation of the unknown parameter $\hat{\mathbf{u}}$ minimizing (or maximizing) g. In our context, our cost function measures the error of matching blocks between the target image and the reference image, hence our problem is the minimization:

$$\text{Find} \quad \hat{\mathbf{u}} = \underset{\mathbf{u} \in \mathbb{R}^N}{\operatorname{argmin}}\ g(\mathbf{u}). \tag{6}$$

Most optimization algorithms operate within a general iterative approach consisting, at each iteration to get closer to the minimum by solving an easier minimization sub-problem.

Initialization: Set $u^{(0)} \in \mathbb{R}^N$.
for $n = 0, 1, \ldots$ **do**
  $u^{(n+1)} = \underset{u \in \mathbb{R}^N}{\operatorname{argmin}}\ q_{u^{(n)}}(u)$.
end

Algorithm 1: General Majorization-Minimization algorithm.

1) Majorization-Minimization approaches: The general MM algorithm was first proposed in [24]. It consists of finding an iterative solution to the optimization problem (6), where Φ is assumed to be differentiable. It relies upon the following concept:

Definition 1: Let $g\colon \mathbb{R}^N \to]-\infty, +\infty]$. Let $\upsilon \in \mathbb{R}^N$. The function $q_\upsilon\colon \mathbb{R}^N \to]-\infty, +\infty]$ is a tangent majorant function of g at υ if $(\forall u \in \mathbb{R}^N)\ f(u) \leqslant q_\upsilon(u)$ and $f(\upsilon) = q_\upsilon(\upsilon)$.

The MM algorithm minimizes the considered tangent majorant function of g at each iteration $n \in \mathbb{N}$. It can be summarized by the following algorithm and Figure 8.

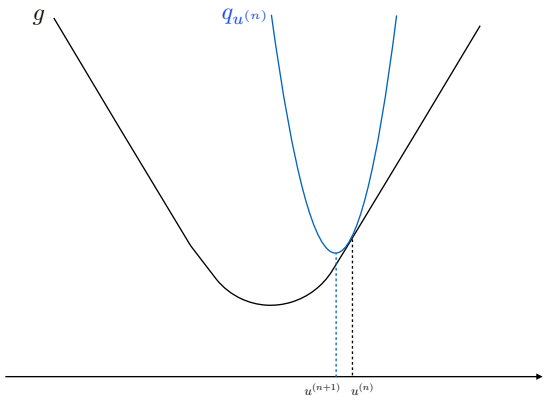


Fig. 8. Illustration of an MM algorithm for the minimization of function $g\colon \mathbb{R}^N \to]-\infty, +\infty]$. At iteration $n \in \mathbb{N}$, we use a majorant function $q_{u^{(n)}}$ of g at a point $u^{(n)}$, and then we define $u^{(n+1)}$ as a minimizer of $q_{u^{(n)}}$.

A typical choice of majorant function is a stricly convex quadratic function [25]. It ensures a unique solution to the sub-problem in Algorithm 1. The following property will be instrumental to build such a majorant function:

Property (Descent Lemma [26]) If g is β-Lipschitz differentiable on $\mathbb{R}^N$, a tangent majorant function of g at $\upsilon \in \mathbb{R}^N$ is defined as

$$(\forall u \in \mathbb{R}^N)\ \ q_\upsilon(u) = g(\upsilon) + \nabla g(\upsilon)^\top (u - \upsilon) + \frac{\mu}{2}\|u - \upsilon\|^2 \tag{7}$$

where $\mu \in [\beta, +\infty[$.

This property holds in our context, provided that Φ is a differentiable cost function with a Lipschitzian gradient. In our case, we have to minimize a function of two-variables $\widetilde{\boldsymbol{M}}_{k,t}$ and $\mathbf{d}_{k,t}$. For every $(\widetilde{\boldsymbol{M}}_{k,t}, \widehat{\boldsymbol{M}}_{k,t}) \in (\mathbb{R}^{3\times 3})^2$ and $(\mathbf{d}_{k,t}, \hat{\mathbf{d}}_{k,t}) \in \mathcal{W}^2$, our aim is thus to construct a surrogate function $Q_{\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}}(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t})$ of $J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t})$ satisfying the following tangent majorization property:

$$\begin{cases} J(\widetilde{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}) \leqslant Q_{\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}}(\widetilde{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}) \\ J(\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}) = Q_{\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}}(\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}). \end{cases} \tag{8}$$

The proposed algorithm is shown as Algorithm 2.

Note that this algorithm constitutes an original contribution of this work since it mixes a discrete optimization step with respect to $\mathbf{d}_{k,t}$ with a continuous optimization with respect to $\widetilde{\boldsymbol{M}}_{k,t}$. The first step is performed by an exact search, while the second one employs the surrogate function Q. Property (8) is instrumental for proving the convergence of our method as

Initialization: Set $\widehat{\boldsymbol{M}}_{k,t}^{(0)} \in \mathbb{R}^{3\times 3}$ and $\widehat{\mathbf{d}}_{k,t}^{(0)} \in \mathcal{W}$.
for $n = 0, 1, \ldots$ **do**
$\widehat{\mathbf{d}}_{k,t}^{(n+1)} = \underset{\mathbf{d}\in\mathcal{W}}{\operatorname{argmin}}\ J(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\mathbf{d}), \mathbf{d})$,
where $\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\mathbf{d}) = \underset{\boldsymbol{M}\in\mathbb{R}^{3\times 3}}{\operatorname{argmin}}\ Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\boldsymbol{M}, \mathbf{d})$
$\widehat{\boldsymbol{M}}_{k,t}^{(n+1)} = \widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n+1)})$.
end

Algorithm 2: Proposed MM algorithm.

follows:

$$\begin{aligned} J(\widehat{\boldsymbol{M}}_{k,t}^{(n+1)}, \widehat{\mathbf{d}}_{k,t}^{(n+1)}) &= J(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n+1)}), \widehat{\mathbf{d}}_{k,t}^{(n+1)}) \\ &\leqslant J(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n)}), \widehat{\mathbf{d}}_{k,t}^{(n)}) \\ &\leqslant Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\widetilde{\boldsymbol{M}}_{k,t}^{(n+1)}(\widehat{\mathbf{d}}_{k,t}^{(n)}), \widehat{\mathbf{d}}_{k,t}^{(n)}) \\ &\leqslant Q_{\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}}(\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}) \\ &= J(\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \widehat{\mathbf{d}}_{k,t}^{(n)}). \end{aligned} \tag{9}$$

These inequalities prove that J is a decreasing sequence. It is converging because it is bounded from below by zero.

In our particular case, we optimize (5). As already mentioned, Property (7) holds since Φ is a differentiable cost function with a Lipschitzian gradient. Without loss of generality, let the Lipschitz constant be normalized to 1. Let $(\widetilde{\boldsymbol{M}}_{k,t}, \widehat{\boldsymbol{M}}_{k,t}) \in (\mathbb{R}^{3\times 3})^2$ and $(\mathbf{d}_{k,t}, \widehat{\mathbf{d}}_{k,t}) \in \mathcal{W}^2$. Applying the Descent Lemma to Φ yields

$$\Phi(\varphi) \leqslant \Phi(\varphi_*) + \nabla\Phi(\varphi_*)^\top(\varphi - \varphi_*) + \frac{1}{2}\|\varphi - \varphi_*\|^2 \tag{10}$$

where

$$\begin{cases} \varphi = \widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\text{ref}}), \\ \varphi_* = \widehat{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \widehat{\mathbf{d}}_{k,t}, t_{\text{ref}}) = \mathbf{e}_{k,\mathbf{s},t}. \end{cases} \tag{11}$$

This leads to the following inequality:

$$\begin{aligned} J(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) &\leqslant \sum_{\mathbf{s}\in\mathcal{B}_k} \Phi(\mathbf{e}_{k,\mathbf{s},t}) \\ &+ \sum_{\mathbf{s}\in\mathcal{B}_k} \nabla\Phi(\mathbf{e}_{k,\mathbf{s},t})^\top(\widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\text{ref}}) - \mathbf{e}_{k,\mathbf{s},t}) \\ &+ \frac{1}{2}\sum_{\mathbf{s}\in\mathcal{B}_k} \|\widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\text{ref}}) - \mathbf{e}_{k,\mathbf{s},t}\|^2. \end{aligned} \tag{12}$$

The right-hand side term of the inequality defines the majorant function of J whose expression simplifies to

$$\begin{aligned} &Q_{\widehat{\boldsymbol{M}}_{k,t}, \widehat{\mathbf{d}}_{k,t}}(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) \\ &= C + \sum_{\mathbf{s}\in\mathcal{B}_k} \nabla\Phi(\mathbf{e}_{k,\mathbf{s},t})^\top \widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t) \\ &+ \frac{1}{2}\sum_{\mathbf{s}\in\mathcal{B}_k} \big(\|\widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t)\|^2 - 2(\mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\text{ref}}) + \mathbf{e}_{k,\mathbf{s},t})^\top \widetilde{\boldsymbol{M}}_{k,t}\,\mathbf{f}(\mathbf{s}, t)\big) \end{aligned} \tag{13}$$

where C is a constant independent from $\widetilde{\boldsymbol{M}}_{k,t}$. This can be rewritten as

$$\begin{aligned} &Q_{\widehat{\boldsymbol{M}}_{k,t}, \widehat{\mathbf{d}}_{k,t}}(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) \\ &= \frac{1}{2}\big\langle \widetilde{\boldsymbol{M}}_{k,t}, \widetilde{\boldsymbol{M}}_{k,t} \sum_{\mathbf{s}\in\mathcal{B}_k} \mathbf{f}(\mathbf{s}, t)\,\mathbf{f}(\mathbf{s}, t)^\top \big\rangle_{\text{F}} \\ &+ \big\langle \widetilde{\boldsymbol{M}}_{k,t}, \sum_{\mathbf{s}\in\mathcal{B}_k} \big(\nabla\Phi(\mathbf{e}_{k,\mathbf{s},t}) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\text{ref}}) - \mathbf{e}_{k,\mathbf{s},t}\big)\mathbf{f}(\mathbf{s}, t)^\top \big\rangle_{\text{F}} + C \end{aligned} \tag{14}$$

where $\langle \cdot, \cdot \rangle_{\mathrm{F}}$ is the Frobenius scalar product, defined by $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_{\mathrm{F}} = \mathrm{tr}(\boldsymbol{A}\,\boldsymbol{B}^\top)$. The function $Q_{\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}}(\cdot, \mathbf{d}_{k,t})$ is quadratic and convex, its gradient being given by

$$\begin{aligned}
&\nabla_{\widetilde{\boldsymbol{M}}_{k,t}} Q_{\widehat{\boldsymbol{M}}_{k,t}, \hat{\mathbf{d}}_{k,t}}(\widetilde{\boldsymbol{M}}_{k,t}, \mathbf{d}_{k,t}) \\
&= \widetilde{\boldsymbol{M}}_{k,t} \sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s}, t)\, \mathbf{f}(\mathbf{s}, t)^\top \\
&\quad + \sum_{\mathbf{s} \in \mathcal{B}_k} \big(\nabla \Phi(\mathbf{e}_{k,\mathbf{s},t}) - \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}}) - \mathbf{e}_{k,\mathbf{s},t} \big) \mathbf{f}(\mathbf{s}, t)^\top .
\end{aligned} \tag{15}$$

For every candidate vector $\mathbf{d}_{k,t} \in \mathcal{W}$, the majorant function thus admits as minimizer:

$$\begin{aligned}
&\widetilde{\boldsymbol{M}}_{k,t}(\mathbf{d}_{k,t}) \\
&= \Big(\sum_{\mathbf{s} \in \mathcal{B}_k} \big(\mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}}) + \mathbf{e}_{k,\mathbf{s},t} - \nabla \Phi(\mathbf{e}_{k,\mathbf{s},t}) \big) \mathbf{f}(\mathbf{s}, t)^\top \Big) \boldsymbol{R}_{k,t}^\dagger ,
\end{aligned} \tag{16}$$

where

$$\boldsymbol{R}_{k,t} = \sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s}, t) \mathbf{f}(\mathbf{s}, t)^\top \tag{17}$$

and $(\cdot)^\dagger$ denotes the pseudo-inverse operation. By plugging the above expression in Algorithm 2, we generate a sequence $(\widehat{\boldsymbol{M}}_{k,t}^{(n)}, \hat{\mathbf{d}}_{k,t}^{(n)})_{n \in \mathbb{N}}$ of estimates for the 11 scalar parameters for each block (9 for color correction and 2 for translation).

C. Choice for the cost function Φ

Typical choices for Φ are the squared Euclidean norm, yielding a standard least squares approach, the separable Huber function, which constitutes a smoothed version of the ℓ_1 norm and is useful for performing robust regression as well as smooth approximations of the ℓ_0 pseudo-norm.

1) Squared Euclidean norm: When $\Phi = \| \cdot \|^2/2$, we obtain $\nabla \Phi(\mathbf{e}_{k,\mathbf{s},t}) = \mathbf{e}_{k,\mathbf{s},t}$ and the standard least squares solution is recovered:

$$\widetilde{\boldsymbol{M}}_{k,t}(\mathbf{d}_{k,t}) = \Big(\sum_{\mathbf{s} \in \mathcal{B}_k} \mathbf{f}(\mathbf{s} - \mathbf{d}_{k,t}, t_{\mathrm{ref}}) \mathbf{f}(\mathbf{s}, t)^\top \Big) \boldsymbol{R}_{k,t}^\dagger . \tag{18}$$

and so the convergence of Algorithm 2 is ensured in one iteration.

2) Huber function: The Huber function is useful in robust regression. As a differentiable approximation of the ℓ_1 norm, it is well-known to be less sensitive to outliers than the squared Euclidean norm, which makes the estimation of displacement and color transformation parameters more accurate, and thus block artifacts are reduced in the corrected sequence.

We can write $\mathbf{e}_{k,\mathbf{s},t}$ in vector form as follows:

$$\mathbf{e}_{k,\mathbf{s},t} = \begin{bmatrix} e_{k,\mathbf{s},t}^{(\mathrm{R})} \\ e_{k,\mathbf{s},t}^{(\mathrm{G})} \\ e_{k,\mathbf{s},t}^{(\mathrm{B})} \end{bmatrix}, \tag{19}$$

and we have then

$$\Phi(\mathbf{e}_{k,\mathbf{s},t}) = \sum_{c \in \{\mathrm{R},\mathrm{G},\mathrm{B}\}} \phi_{\mathrm{h}}(e_{k,\mathbf{s},t}^{(c)}), \tag{20}$$

where, by using Huber's definition ([27]),

$$(\forall a \in \mathbb{R}) \quad \phi_{\mathrm{h}}(a) = \begin{cases} \frac{1}{2} a^2 & \text{if } |a| \leqslant \varepsilon, \\ \varepsilon(|a| - \frac{1}{2}\varepsilon), & \text{otherwise,} \end{cases} \tag{21}$$

with $\varepsilon > 0$. The gradient of our cost function is given by

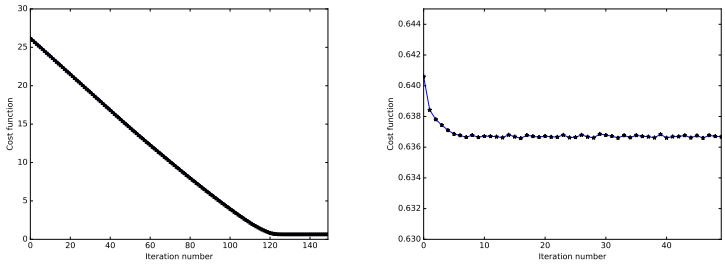
$$\nabla \Phi(\mathbf{e}_{k,\mathbf{s},t}) = \begin{bmatrix} \phi_{\mathrm{h}}'(\mathbf{e}_{k,\mathbf{s},t}^{(\mathrm{R})}) \\ \phi_{\mathrm{h}}'(\mathbf{e}_{k,\mathbf{s},t}^{(\mathrm{G})}) \\ \phi_{\mathrm{h}}'(\mathbf{e}_{k,\mathbf{s},t}^{(\mathrm{B})}) \end{bmatrix}, \tag{22}$$

where the derivative of ϕ_{h} is

$$(\forall a \in \mathbb{R}) \quad \phi_{\mathrm{h}}'(a) = \begin{cases} \varepsilon & \text{if } a > \varepsilon, \\ -\varepsilon & \text{if } a < -\varepsilon, \\ a, & \text{otherwise.} \end{cases} \tag{23}$$

Using this cost function, if the initialization of the parameters is performed arbitrarily, convergence can be somewhat slow (up to 100 iterations), therefore the initialization step is very important. We propose to first estimate parameters, in particular

the translation vectors, using the quadratic approach, and use them as initial values for the subsequent iterations. This procedure reduces the number of required iterations by a factor of more than 20.



(a) With arbitrary parameter initialization (b) Initialization with the quadratic solution

Fig. 9. The convergence of our iterative algorithm depends of the parameters initialization. This test was performed on an image of the bird sequence.

In qualitative terms, the smaller ε is, the more effective is the algorithm at reducing the outliers and the algorithm converges more slowly to an optimal best corrected solution. As illustrated in Figure 9 (and observed on several sequences), a suitable choice for ε appears to be around 0.2.

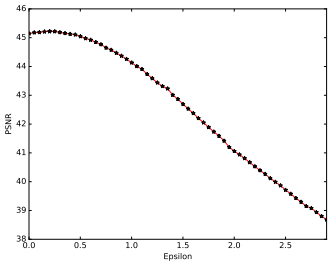


Fig. 10. Huber function: PSNR of the processed image for the bird sequence as a function of ε.

3) Other cost functions: We tested our iterative optimization strategy with other more complex criteria. We considered some non-convex functions for the majorant function, typically the Welsch and Geman-McClure functionals providing smooth approximations of the ℓ_0 pseudo-norm.

Welsch function. Using Welsch's definition [28], we can write:

$$\Phi(\mathbf{e}_{k,\mathbf{s},t}) = \sum_{c\in\{R,G,B\}} \phi_{\mathrm{w}}(e^{(c)}_{k,\mathbf{s},t}), \tag{24}$$

where, for every $a \in \mathbb{R}$,

$$\phi_{\mathrm{w}}(a) = \frac{\lambda_{\mathrm{w}}}{2}\Big(1 - \exp\big(-\frac{a^2}{C_{\mathrm{w}}^2}\big)\Big), \tag{25}$$

C_{w} and λ_{w} being positive constants. For a given value of C_{w}, the parameter λ_w is set in order to ensure that the Lipschitz constant L_w of the derivative (Hessian maximum value) is equal to 1:

$$L_{\mathrm{w}} = \phi''_{\mathrm{w}}(0) = \frac{\lambda_{\mathrm{w}}}{C_{\mathrm{w}}^2} = 1 \Rightarrow \lambda_{\mathrm{w}} = C_{\mathrm{w}}^2, \tag{26}$$

where ϕ''_{w} denotes the second-order derivative of ϕ_{w}. According to our experiments, a suitable choice for the parameter C_{w} is then 0.15.

Geman-McClure function The Geman-McClure M-estimator proposed by [29] can also be tested. It is defined as

$$\Phi(\mathbf{e}_{k,\mathbf{s},t}) = \sum_{c\in\{R,G,B\}} \phi_{\mathrm{g}}(e^{(c)}_{k,\mathbf{s},t}), \tag{27}$$

where, for every $a \in \mathbb{R}$,

$$\phi_{\mathrm{g}}(a) = \frac{\lambda_{\mathrm{g}}\, a^2}{2{C_{\mathrm{g}}}^2 + a^2}, \tag{28}$$

and C_{g} and λ_{g} are positive constants. For a given value of C_{g}, the parameter λ_{g} must also be chosen so as to ensure that the Lipschitz constant L_g is normalized to 1:

$$L_{\mathrm{g}} = \phi''_{\mathrm{g}}(0) = \frac{\lambda_g}{C_{\mathrm{g}}^2} = 1 \Rightarrow \lambda_{\mathrm{g}} = C_{\mathrm{g}}^2. \tag{29}$$

The choice of the parameter C_{g} which appears to provide the best results when running the MM algorithm is 0.11.

4) Comparison between different cost functions: In order to evaluate the performance of the color correction strategy using our iterative algorithm with different cost functions, a PSNR (Peak Signal-to-Noise Ratio) similarity measure is performed on the processing results of a synthetic flicker sequences produced from a flicker-free, naturally lit video. We constructed the majorant function Q using the different approximations of norms/pseudo-norms with convexity (quadratic and Huber functions) and non-convexity properties (Welsch and Geman-McClure functions).

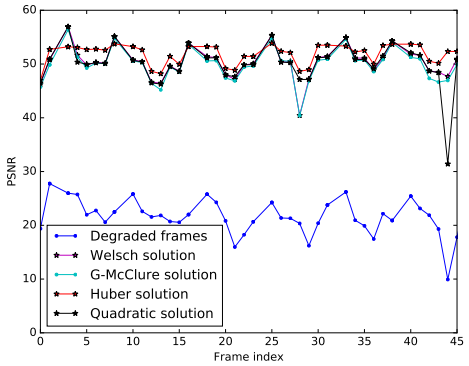


Fig. 11. Similarity comparison: PSNR average of the processed sequence using different cost functions. The sequence is acquired with a large local motion and a periodic artificial flicker has been added. The studied sequence images are illustrated in Figure 23.

We observe that the Huber function provides the best PSNR. This observation was confirmed on several other video sequences. It allows us to gain approximately 4 dB along the processed sequence (see Figure 11).

D. Flicker compensation

Once the optimal color transform between the reference and the current frames has been found for each block, it is applied on each. This yields a flicker-compensated frame. Reproducing the same procedure for each frame yields a flicker-compensated sequence. This procedure is quite simple, however there is no guarantee for the optimization procedure that we have just described to yield similar color-correction matrices for neighboring blocks. As a result, the boundary between blocks may become perceptible in the flicker-compensated results. This may happen for instance when the translation parameters are not estimated precisely enough due to the arbitrary process of image division into blocks, which implies that some blocks may include strong intensity edges or outliers. A variable-size block matching could be implemented to alleviate this issue, but at the expense of an increased complexity. To alleviate this issue, a post processing step is required.

E. Block artifact removal

Following the flicker compensation step, we propose to re-process each frame of the affected video to remove block artifacts in an efficient manner. We base our approach on the color transformation matrices and translation parameters that we have already found. The simple idea of interpolating the color matrices near the boundaries of the blocks does not yield accurate enough results in practice.

Instead, we re-estimate a new color correction matrix $\boldsymbol{M}_{\mathbf{s},t}$ for each pixel $\mathbf{s}$ in a frame, taking into account its neighborhood and its position in the block relative to neighboring blocks. For each pixel $\mathbf{s}$, we reuse the pair $(\widehat{\boldsymbol{M}}_{k_{\mathbf{s}},t}, \widehat{\mathbf{d}}_{k_{\mathbf{s}},t})$ already estimated for its corresponding block of index $k_{\mathbf{s}}$. In the optimization process, we now include a regularization term weighted by a positive constant λ. A new energy function is then defined as

$$\begin{aligned} K(\boldsymbol{M}_{\mathbf{s},t}) = & \sum_{\mathbf{s}' \in \mathcal{B}_{k_{\mathbf{s}}}} \Phi\big(\boldsymbol{M}_{\mathbf{s},t}\, \mathbf{f}(\mathbf{s}', t) - \mathbf{f}(\mathbf{s}' - \widehat{\mathbf{d}}_{k_{\mathbf{s}},t}, t_{\mathrm{ref}})\big) \\ & + \underbrace{\frac{\lambda}{2} \sum_{\mathbf{s}' \in \mathcal{V}_{\mathbf{s}}} \|\boldsymbol{M}_{\mathbf{s},t} - \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\|_{\mathrm{F}}^2}_{R_g(\boldsymbol{M}_{\mathbf{s},t})}, \end{aligned} \tag{30}$$

where $\mathcal{V}_{\mathbf{s}}$ is a neighbourhood of the pixel $\mathbf{s}$ to be processed, and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. The previous optimization problem can also be solved by an iterative Majorize-Minimize strategy, still grounded on the use of Descent Lemma 7. At iteration $n+1$, we construct a new majorant function called $P_{\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)}}$ as follows:

$$P_{\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)}} = Q_{\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)},\widehat{\mathbf{d}}_{k_{\mathbf{s}},t}}(\cdot,\widehat{\mathbf{d}}_{k_{\mathbf{s}},t}) + R_g. \tag{31}$$

where $\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)}$ is the estimate of the color correction matrix at the previous iteration. The minimizer of (31) is obtained by setting to zero the gradient of $P_{\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)}}$. The gradient of the regularization term is expressed as

$$\nabla R_g(\boldsymbol{M}_{\mathbf{s},t}) = \lambda\Big(|\mathcal{V}_{\mathbf{s}}|\boldsymbol{M}_{\mathbf{s},t} - \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\Big), \tag{32}$$

where $|\mathcal{V}_{\mathbf{s}}|$ is the number of pixels considered in the neighborhood of $\mathbf{s}$. Refering to (15) and (32), the minimizer of the surrogate function $P_{\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)}}$ is

$$\begin{aligned}\check{\boldsymbol{M}}_{\mathbf{s},t}^{(n+1)} = \Big(\sum_{\mathbf{s}'\in\mathcal{B}_{k_{\mathbf{s}}}} \big(\mathbf{f}(\mathbf{s}'-\widehat{\mathbf{d}}_{k_{\mathbf{s}},t},t_{\mathrm{ref}}) + \mathbf{e}_{\mathbf{s}',t}^{(n)} \\ - \nabla\Phi(\mathbf{e}_{\mathbf{s}',t}^{(n)})\big)\mathbf{f}(\mathbf{s}',t)^{\top} + \lambda \sum_{\mathbf{s}'\in\mathcal{V}_{\mathbf{s}}} \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\Big)\Big(\boldsymbol{R}_{k_{\mathbf{s}},t} + \lambda|\mathcal{V}_{\mathbf{s}}|\boldsymbol{I}_3\Big)^{-1},\end{aligned} \tag{33}$$

where $\boldsymbol{I}_3$ is the 3×3 identity matrix and

$$\mathbf{e}_{\mathbf{s}',t}^{(n)} = \check{\boldsymbol{M}}_{\mathbf{s},t}^{(n)}\,\mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}'-\widehat{\mathbf{d}}_{k_{\mathbf{s}},t},t_{\mathrm{ref}}). \tag{34}$$

Note that this formulation ensures that, if we set λ to zero, we recover the previous solution $\widehat{\boldsymbol{M}}_{k_{\mathbf{s}}}$ computed on block $\mathcal{B}_{k_{\mathbf{s}}}$. In addition, in terms of computational complexity, this removal procedure needs only to be applied to pixels $\mathbf{s}$ for which $\mathcal{V}_{\mathbf{s}}$ is not a subset of $\mathcal{B}_{k_{\mathbf{s}}}$.

F. An outline

As will be illustrated in Section VII, this causal method compensates the periodic flicker in an image with respect to a previous reference frame. When the flicker is not uniform in some image regions for some reason (different light source, higher noise level, etc), some local luminance distortions at the boundary between flicker periods may however remain in the corrected sequence. In the next section, we propose an improvement of our method which is also suitable for aperiodic flicker removal.

V. Non-causal method using two reference frames

Most state of the art deflickering methods use a single reference to process the current frame. Some consider the previous corrected frame as a reference to maintain constancy in luminosity variation, and especially when processing non-periodic flicker, this approach may accumulate restoration errors from one image to another. In this context, in order to avoid luminosity distortions between flicker periods in the corrected sequences, we propose a new strategy consisting of processing each frame with respect to a preceeding and a following reference frames, depending on its position between these two. This approach is called "non-causal" because it uses both past and future frames. Obviously it can only be used offline, so the moniker "non-causal" is only descriptive. As already explained in Section III, the previous and next reference frames are respectively the last and next maximum intensity frames in the current period, which are better in terms of signal-to-acquisition noise ratio and in terms of matching similar contents. Figure 12 shows the benefits which can be drawn from the proposed approach.

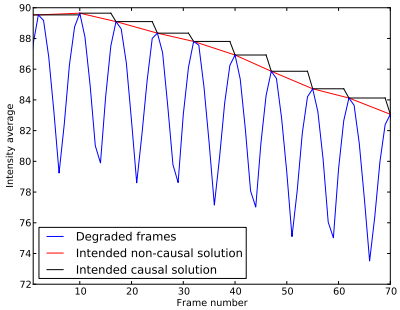


Fig. 12. The blue curve shows the brightness variation of images in a flicker-affected sequence. In black, we display the intensity averages of the corrected frames using a single reference image by flicker period (causal approach). The red curve shows the desired solution that better aligns brightness levels using two reference frames using our non-causal approach.

A. Tracking step

We continue to use the same joint block-matching / color transform estimation principle. More precisely, we find for each spatial block $\mathcal{B}_k$ indexed by k its two similar blocks in the previous and next reference frames respectively, and we assign to this block its own model parameters. The global color correction matrix then reads:

$$\widetilde{\boldsymbol{M}}_{k,t} = \alpha_t \widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref1}}} + (1-\alpha_t)\widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref2}}}, \tag{35}$$

where $\alpha_t = (t_{\mathrm{ref2}} - t)/(t_{\mathrm{ref2}} - t_{\mathrm{ref1}})$ is a linear interpolation factor, $\widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref1}}}$ and $\widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref2}}}$ are the same correction matrices for all pixels of the block $\mathcal{B}_k$, which are computed with respect to their similar blocks in the two reference sources at instants t_{ref1} et t_{ref2}.

Following the same methodology as described above, for a block $\mathcal{B}_k$ in a frame at an instant t, we minimise the following energy function:

$$\begin{aligned} &J(\widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref}i}}, \mathbf{d}_{k,t/t_{\mathrm{ref}i}}) \\ &= \sum_{\mathbf{s}'\in\mathcal{B}_k} \Phi\big(\widetilde{\boldsymbol{M}}_{k,t/t_{\mathrm{ref}i}}\, \mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \mathbf{d}_{k,t/t_{\mathrm{ref}i}}, t_{\mathrm{ref}i})\big), \end{aligned} \tag{36}$$

where $i \in \{1,2\}$, $\mathbf{d}_{k,t/t_{\mathrm{ref}i}}$ is a translation vector between two spatial blocs corresponding to instants t and $t_{\mathrm{ref}i}$ and $\Phi\colon \mathbb{R}^3 \to [0,+\infty[$ is a differentiable cost function with a Lipschitzian gradient. For every $i \in \{1,2\}$, the optimal pair $\big(\widehat{\boldsymbol{M}}_{k,t/t_{\mathrm{ref}i}}, \widehat{\mathbf{d}}_{k,t/t_{\mathrm{ref}i}}\big)$ can be computed by following the same optimization method as in Section IV-B. For each block in the current image, two color correction matrices (each comprising 9 parameters) and two translation vectors are estimated with respect to the two associated reference frames, the final color correction matrix $\widehat{\boldsymbol{M}}_{k,t}$ is deduced from the linear interpolation formula (35).

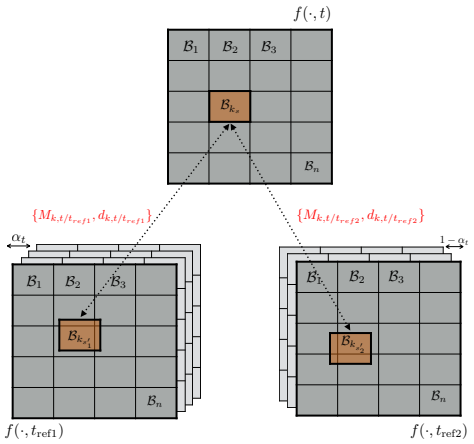


Fig. 13. Two reference sources tracking method

B. Block artifact removal step

For each pixel $\mathbf{s}$ in the block $\mathcal{B}_{k_\mathbf{s}}$, we re-use the two pairs $\big(\widehat{\boldsymbol{M}}_{k_\mathbf{s},t/t_{\mathrm{ref1}}}, \widehat{\mathbf{d}}_{k_\mathbf{s},t/t_{\mathrm{ref1}}}\big)$ and $\big(\widehat{\boldsymbol{M}}_{k_\mathbf{s},t/t_{\mathrm{ref2}}}, \widehat{\mathbf{d}}_{k_\mathbf{s},t/t_{\mathrm{ref2}}}\big)$ already found. As in the artifact removal step for the causal method, we add to the loss function a regularization term weighted by a positive multiplicative constant λ. The new energy function is defined as

$$\begin{aligned} &K(\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}}, \boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}}) = \\ &\sum_{\mathbf{s}'\in\mathcal{B}_{k_\mathbf{s}}} \Phi\big(\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}}\, \mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \widehat{\mathbf{d}}_{k_\mathbf{s},t/t_{\mathrm{ref1}}}, t_{\mathrm{ref1}})\big) \\ &+ \sum_{\mathbf{s}'\in\mathcal{B}_{k_\mathbf{s}}} \Phi\big(\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}}\, \mathbf{f}(\mathbf{s}',t) - \mathbf{f}(\mathbf{s}' - \widehat{\mathbf{d}}_{k_\mathbf{s},t/t_{\mathrm{ref2}}}, t_{\mathrm{ref2}})\big) \\ &+ \frac{\lambda}{2} \sum_{\mathbf{s}'\in\mathcal{V}_\mathbf{s}} \|\alpha_t \boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref1}}} + (1-\alpha_t)\boldsymbol{M}_{\mathbf{s},t/t_{\mathrm{ref2}}} - \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t}\|_{\mathrm{F}}^2, \end{aligned} \tag{37}$$

where $\widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t} = \alpha_t \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t/t_{\mathrm{ref1}}} + (1-\alpha_t)\widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t/t_{\mathrm{ref2}}}$ is the color correction matrix estimate for the block containing pixel $\mathbf{s}'$. Although various choices for Φ are possible, for simplicity, we choose $\Phi = \|\cdot\|^2/2$, leading to a least squares approach. The solution is thus readily given by

$$\begin{bmatrix} \check{\boldsymbol{M}}_{\mathbf{s},t/t_{\mathrm{ref1}}} \\ \check{\boldsymbol{M}}_{\mathbf{s},t/t_{\mathrm{ref2}}} \end{bmatrix} = \begin{bmatrix} \sum\limits_{\mathbf{s}'\in\mathcal{B}_{k_\mathbf{s}}} \mathbf{f}(\mathbf{s}'-\hat{\mathbf{d}}_{k_\mathbf{s},t/t_{\mathrm{ref1}}},t_{\mathrm{ref1}})\mathbf{f}(\mathbf{s}',t)^\top + \lambda\alpha_t \sum\limits_{\mathbf{s}'\in\mathcal{V}_\mathbf{s}} \widehat{\boldsymbol{M}}_{k_{\mathbf{s}'},t} \\ \sum\limits_{\mathbf{s}'\in\mathcal{B}_{k_\mathbf{s}}} \mathbf{f}(\mathbf{s}'-\hat{\mathbf{d}}_{k_\mathbf{s},t/t_{\mathrm{ref2}}},t_{\mathrm{ref2}})\mathbf{f}(k_{\mathbf{s}'},t)^\top + \lambda(1-\alpha_t) \sum\limits_{\mathbf{s}'\in\mathcal{V}_\mathbf{s}} \widehat{\boldsymbol{M}}_{\mathbf{s}',t} \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{k_\mathbf{s},t} + \lambda\alpha_t^2|\mathcal{V}_s|\boldsymbol{I}_3 & \lambda\alpha_t(1-\alpha_t)|\mathcal{V}_s|\boldsymbol{I}_3 \\ \lambda\alpha_t(1-\alpha_t)|\mathcal{V}_s|\boldsymbol{I}_3 & \boldsymbol{R}_{k_\mathbf{s},t} + \lambda(1-\alpha_t)^2|\mathcal{V}_s|\boldsymbol{I}_3 \end{bmatrix}^{-1}. \tag{38}$$

The comparison results with the causal approach are shown and discussed in the final section of this article.

VI. Pyramidal accelerated approach

Processing time is particularly important in video processing, and thus for high speed imaging, it is a challenge to meet real time constraints during the processing of long image sequences. At each iteration of Algorithm 2, we have to carry out the search of an optimal displacement vector for each block over the window $\mathcal{W}$. This block matching process requires a significant amount of processing time, especially when processing HD images (1920×1080 pixels). Subsequently, we propose a multi-resolution algorithm to accelerate this step.

A. Algorithm

In order to process a current image with respect to a previous reference, (without loss of generality, we consider here only the causal approach), for each block $\mathcal{B}_k$, we estimate first the translation parameters represented by $\widehat{\mathbf{d}}_k^{S_N} = (\widehat{d}_{k_x}^{S_N}, \widehat{d}_{k_y}^{S_N})$ between the two images at the most reduced scale ($S_N = \frac{1}{2^N}$), using a small search grid for the displacement vector, say for instance $(\pm 2, \pm 2)$ At the next higher resolution $S_{N-1} = \frac{1}{2^{N-1}}$, these estimated parameters become the new centers of the search grid $(2\widehat{d}_{k_x}^{S_N} \pm 2, 2\widehat{d}_{k_y}^{S_N} \pm 2)$, which are used to improve the estimate of the displacement vector. We repeat this process at all the scale levels up to scale $S_0 = \frac{1}{2^0} = 1$, to find the final estimation of displacement vector and color transformation parameters $(\widehat{\mathbf{d}}_k^{S_0}, \widehat{\boldsymbol{M}}_k^{S_0})$ in the current original frame. Figure 14 illustrates the proposed approach.

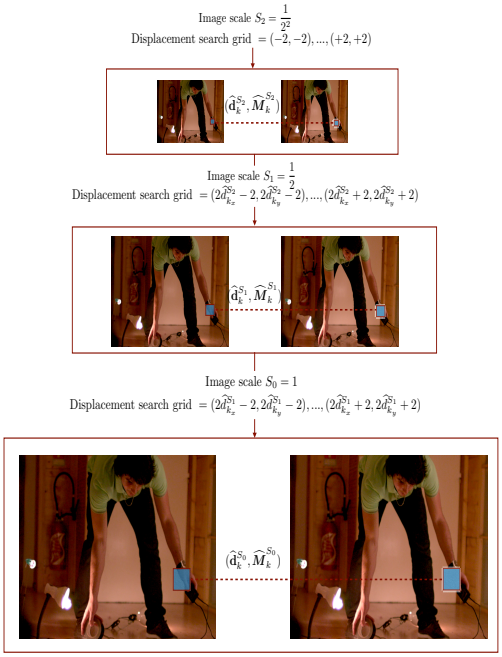


Fig. 14. Multiresolution scheme with the most reduced scale is $S_N = \frac{1}{4}$.

B. Complexity

To perform the block matching-based method, the complexity of the used method depends on the search grid area around the target block and the size of this block. In order to deduce the computation time gain, we can compare the complexity of the pyramidal approach with respect to our standard causal block matching approach.

Considering a block $\mathcal{B}_k$ in the current original frame (at the full resolution), the complexity of applying the causal method on this block can be expressed as

$$C_o \propto (m_1+1)(m_2+1)w_1 w_2 \tag{39}$$

where m_1, m_2 are the horizontal and vertical search pixels interval respectively (i.e. $\mathcal{W} = \{-\frac{m_1}{2}, \ldots, \frac{m_1}{2}\} \times \{-\frac{m_2}{2}, \ldots, \frac{m_2}{2}\}$), and w_1, w_2 are the width and height sizes (in pixel) of the processed block

In the pyramidal approach, we assume that horizontal and vertical search intervals are constant at all resolution levels but reduced by a factor 2^c with respect to the full resolution approach. The complexity of the pyramidal approach is thus

$$\begin{aligned} C_p \propto & \sum_{i=1}^{N} \Big(\frac{m_1}{2^c}+1\Big)\Big(\frac{m_2}{2^c}+1\Big)\frac{w_1}{2^i}\frac{w_2}{2^i} \\ & = \frac{(m_1+2^c)(m_2+2^c)w_1 w_2}{2^{2c}}\frac{4}{3}\Big(1-\frac{1}{4^{N+1}}\Big). \end{aligned} \tag{40}$$

Then the gain in terms of processing time of the pyramidal approach with respect to the standard algorithm is given by

$$\frac{C_o}{C_p} = \frac{3\times 4^{c-1}}{1-\frac{1}{4^{N+1}}}\frac{(m_1+1)(m_2+1)}{(m_1+2^c)(m_2+2^c)}. \tag{41}$$

In our context where we often process HD images, we found it sufficient to reduce the resolution twice ($N=2$), i.e. up to $S_N = \frac{1}{2^2}$. For simplicity we chose to reduce the search interval by the same factor by choosing $c=2$, which is consistent with the case already discussed in Figure 14 where $m_1 = m_2 = 16$ and the constant search interval is in a range of five pixels (i.e. ± 2). This yields about an 8 times improvement in speed.

The pyramidal approach was tested on the causal method using the Huber solution to verify whether or not the processing results keep the same quality while reducing the resolution of images and decreasing the computation time. Figure 15 shows that the PSNR average is almost unchanged with or without applying the pyramidal approach.

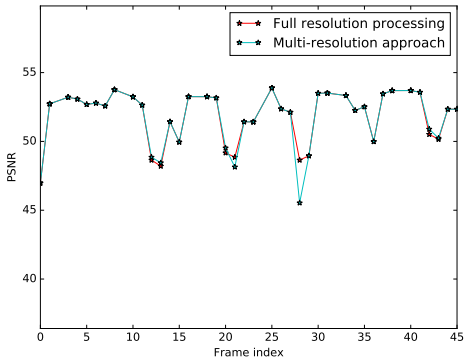
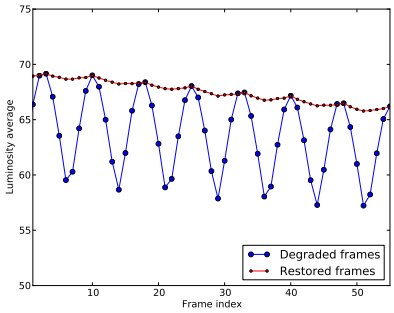


Fig. 15. PSNR measure for the pyramidal approach and the causal approach using Huber solution on the Bird sequence.

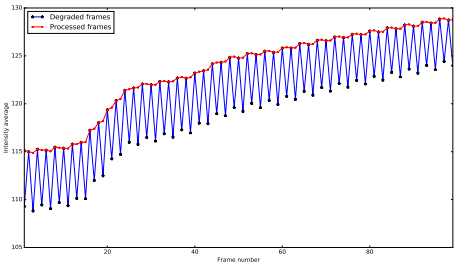
VII. Results and Discussion

In this section, we present some experimental results to demonstrate the efficacy of the proposed approaches.

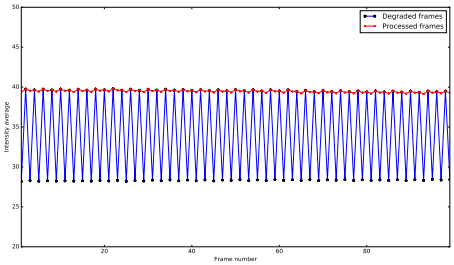
First, our color correction method was tested on synthetic periodic/non-periodic flicker sequences produced from a flicker-free, naturally lit video and also on a real, studio-lit video affected with periodic flicker and featuring multiple light sources and complex motion. The artificial non-periodic flicker was created in such a manner to exhibit large differences in brightness between the different periods. Plotting the global luminosity versus the image sequence index allows us to evaluate the performance of the methods. Figure 16 shows the result of processing the real degraded image sequences using the causal method. The luminosity variation graphs in Figures 17-19 show that brightness levels are well restored in the color-corrected videos, and even in the presence of different acquisition and lighting conditions, different geometrical motions and also at varying frame rates.



(a) Video sequence 1: including complex motions at 1000 fps



(b) Video sequence 2: including rotation, translations, scale changes, focusing and defocusing effects

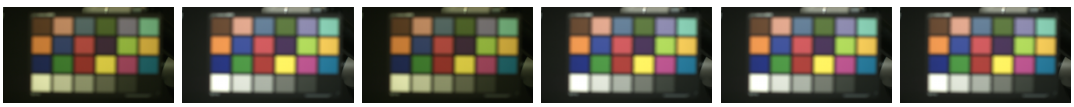


(c) Video sequence 3: including fast rotations at 240 fps

Fig. 16. Luminosity fluctuation and restoration in some real sequences with different acquisition properties: complex motions, different lighting conditions and acquisition frame rates.

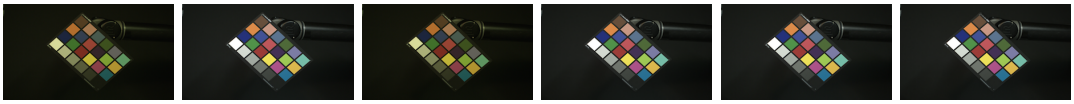
(a) Degraded frames (b) Processed frames

Fig. 17. Video sequence 1 acquired at 1000 fps, illuminated by three light sources, it includes complex motions, saturation effects, noise at the background and some outliers.



(a) Degraded frames (b) Processed frames

Fig. 18. Video sequence 2 acquired at 240 fps, and so we have rapid illumination/ chromatic changes. It includes focusing/defocusing effects accompanied by translation motions and scale changes.



(a) Degraded frames (b) Processed frames

Fig. 19. Video sequence 3 acquired at 240 fps under multiple light sources, it includes very fast rotations.

Figure 21 presents other restoration examples on two sequences with artificial flicker allowing us to compare the proposed causal and non causal methods visually. These examples feature large motion between frames, occlusions in the background and compound motion: for example, in the Bird sequence, the camera is moving in the opposite of the bird's direction. In order to perform a quantitative comparison between causal and non-causal approaches, the similarity measures were computed on a non-periodic flicker affected sequence, see Figure 20. It is clear in the figure above, that the non-causal approach provides

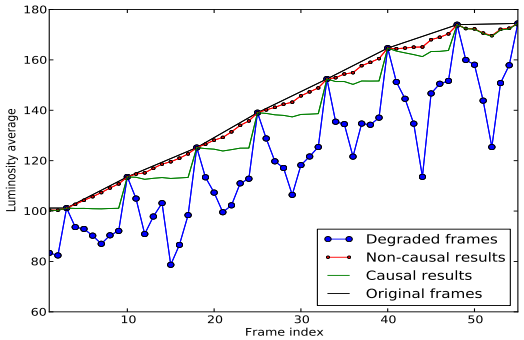


Fig. 20. Luminosity average vs. images sequence for the causal and non-causal approaches, on a non-periodic flicker affected video (Images sequence 4, see Figure 21).

reasonable illumination correction results, as the sudden brightness changes are removed between different flicker periods. The non-causal approach could be used for several applications in addition to high speed imaging, for instance, to stabilize brightness variations in time lapse videos.

Figures 22, 23 and 24 show the benefits of deblocking used after flicker removal to alleviate blocking artifacts caused by occlusions and complex displacements. In order to better assess the performance of the proposed method, we utilize PSNR and and the structural similarity index (SSIM) to measure the similarity between the original and degraded images on the one hand, and then between the original and restored sequences on the other hand. As shown by Figure 25(a), the PSNR average increases from 21 to 48 dB after color correction of image sequence 5. A further 1 dB is gained on average after the deblocking step. We compared our method to another deflickering procedure adapted from [10], which was initially tailored for old grayscale movies, and that we extended to color sequences. This procedure compensates flicker globally and provides a single matrix for the whole image at each timestep. It only improves the PSNR to 32 dB on average, so our gain is significant. Similarly, our SSIM index is very close to 1 after restoration (see Figure 25(b)), and its improvement after removing blocking artifacts and comparing to the global deflickering method is also significant.

Similarly, Figure 26 shows another test of our algorithms performed on the results of image sequence 6 (see Figure 24), and also presents the improvement of PSNR and SSIM indexes after the ficker removal and deblocking procedures.

(a) Reference frame (b) Flicker-affected frame (c) Color correction result (d) Deblocking result (e) Zoom before deblocking (f) Zoom after deblocking

Fig. 22. Sequence with periodic flicker from studio lighting.

(a) Original image (b) Degraded image (c) Color correction result (d) Deblocking result (e) Zoom before deblocking (f) Zoom after deblocking

Fig. 24. Sequence 6 with global motion and periodic artificial flicker.

(a) Degraded frames (b) Causal approach

(c) Non-causal approach

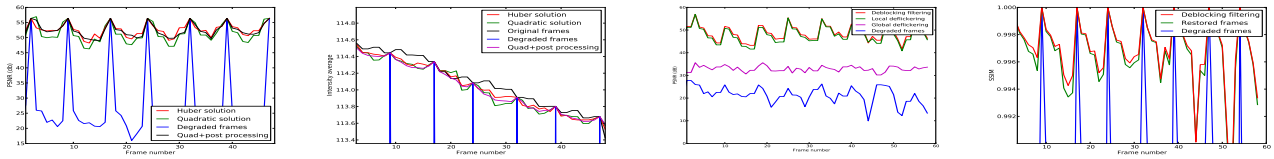
(d) Degraded frames (e) Causal approach

(f) Non-causal approach

Fig. 21. Video sequences 4-5: Comparison between causal and non-causal approaches for color correction on synthetic flicker sequences produced from a flicker-free, naturally lit video.

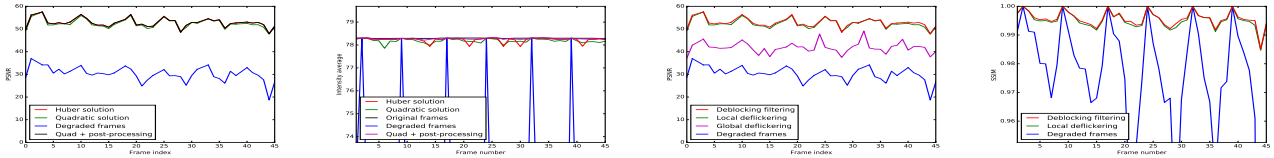
(a) Original image (b) Degraded image (c) Color correction result (d) Deblocking result (e) Zoom before deblocking (f) Zoom after deblocking

Fig. 23. Sequence with large local motion and periodic artificial flicker.



(a) SNR comparison (b) Luminosity variation (c) SNR comparison (d) SSIM comparison

Fig. 25. Similarity measures on video sequence 5.



(a) SNR comparison (b) Luminosity variation (c) SNR comparison (d) SSIM comparison

Fig. 26. Similarity measures on video sequence 6.

VIII. Conclusion

This work introduced a novel flexible method for periodic flicker reduction, suitable for correcting high-speed color video sequences taken under artificial lighting. Our approach mainly relies upon a joint optimized color matrix correction / motion estimation step. We have designed causal and non-causal tracking methods involving blockwise color correction, followed by a per-pixel regularization-based post-processing for removing block artifacts. In order to reduce the processing time, we proposed a pyramidal strategy to estimate motion and color correction parameters at reduced scales, while maintaining close to the same result quality. We have demonstrated the effectiveness of our method on real sequences affected with flicker from studio lights, and on sequences perturbed with an artificial flicker.

In future work, it would be interesting to incorporate variable block sizes in our estimation procedure. and to develop GPGPU implementations of the proposed algorithms.

REFERENCES

[1] A. S. M. Shihavuddin, N. Gracias, and R. García, "Online sunflicker removal using dynamic texture prediction." in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Rome, Italy, 2012, pp. 161–167.

[2] B. Ozer and M. Wolf, "A train station surveillance system: Challenges and solutions," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, Ohio, USA, 2014, pp. 652–657.

[3] S. Baudry, B. Chupeau, M. D. Vito, and G. Doerr, "Modeling the flicker effect in camcorded videos to improve watermark robustness," in *IEEE National Conference on Parallel Computing Technologies*, Bengaluru, India, 2015, pp. 42–47.

[4] É. Decencière, "Restauration automatique de films anciens," Ph.D. dissertation, École Nationale Supérieure des Mines de Paris, France, 1997.

[5] P. M. B. Van Roosmalen, R. L. Lagendijk, and J. Biemond, "Correction of intensity flicker in old film sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1013–1019, 1999.

[6] X. Yang and M. N. Chong, "Enhanced approach to film flicker removal," in *International Symposium on Optical Science and Technology*. San Diego, USA: International Society for Optics and Photonics, 2000, pp. 39–47.

[7] T. Ohuchi, T. Seto, T. Komatsu, and T. Saito, "A robust method of image flicker correction for heavily-corrupted old film sequences," in *IEEE International Conference on Image Processing*, vol. 2, Vancouver, Canada, 2000, pp. 672–675.

[8] A. C. Kokaram, R. Dahyot, F. Pitié, and H. Denman, "Simultaneous luminance and position stabilization for film and video," in *Electronic Imaging*. San Jose, California, USA: International Society for Optics and Photonics, 2003, pp. 688–699.

[9] X. Zhang, A. B. E. Masahide, and M. Kawamata, "Flicker parameters estimation in old film sequences containing moving objects," *Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 94, no. 12, pp. 2836–2844, 2011.

[10] V. Naranjo and A. Albiol, "Flicker reduction in old films," in *IEEE International Conference on Image Processing*, vol. 2, Vancouver, Canada, 2000, pp. 657–659.

[11] J. Delon, "Movie and video scale-time equalization application to flicker reduction," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 241–248, 2006.

[12] F. Pitié, R. Dahyot, F. Kelly, and A. C. Kokaram, "A new robust technique for stabilizing brightness fluctuations in image sequences," in *Statistical Methods in Video Processing*. Springer Berlin Heidelberg, 2004, pp. 153–164.

[13] F. Pitié, B. Kent, B. Collis, and A. C. Kokaram, "Localised deflicker of moving images," in *European Conference on Visual Media Production*, London, United Kingdom, 2006, pp. 134–143.

[14] T. Vlachos, "Flicker correction for archived film sequences using a nonlinear model," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 508–516, 2004.

[15] G. Forbin, T. Vlachos, and S. Tredwell, "Flicker compensation for archived film using a spatially-adaptive nonlinear model," in *IEE European Conference on Visual Media Production*, London, UK, 2005, pp. 241–250.

[16] G. Forbin and T. Vlachos, "Nonlinear flicker compensation for archived film sequences using motion-compensated graylevel tracing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 803–816, 2008.

[17] J. Ren, J. Liu, M. Li, and Z. Guo, "Postprocessing of block-coded videos for deflicker and deblocking," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, 2013, pp. 1631–1635.

[18] A. Jimenez Moreno, E. Martínez-Enríquez, V. Kumar, and F. Diaz-de Maria, "Standard-compliant low-pass temporal filter to reduce the perceived flicker artifact," *IEEE Transactions on Multimedia*, vol. 16, no. 7, pp. 1863–1873, 2014.

[19] C.-S. Fuh and P. Maragos, "Motion displacement estimation using an affine model for image matching," *Optical Engineering*, vol. 30, no. 7, pp. 881–887, 1991.

[20] F.-J. Hampson and J.-C. Pesquet, "Motion estimation in the presence of illumination variations," *Signal Processing: Image Communication*, vol. 16, no. 4, pp. 373–381, 2000.

[21] A. Kanj, H. Talbot, and R. Rodriguez Luparello, "Global image registration and video color correction in presence of illumination variations," in *Fifth IEEE International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, Beirut, Lebanon, 2015, pp. 92–95.

[22] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[23] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." in *International Joint Conference on Artificial Intelligence*, vol. 81, Vancouver, BC, Canada, 1981, pp. 674–679.

[24] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. Society for Industrial and Applied Mathematics, Classics in Applied Mathematics, 1970, vol. 30.

[25] C. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot, "A Majorize-Minimize subspace approach for ℓ_2-ℓ_0 image regularization," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 563–591, 2013.

[26] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. 2nd ed. Springer, New York, 2017.

[27] P. J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[28] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Communications in Statistics-Theory and Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[29] S. Ganan and D. McClure, "Bayesian image analysis: An application to single photon emission tomography," in *Proceeding of the Statistical Computing Section - American Statistical Association*, 1985, pp. 12–18.