Blind uniform motion blur deconvolution for image bursts and video sequences based on sensor characteristics

Yohann Tendero, Stanley Osher*

Department of Mathematics - University of California Los Angeles
520 Portola Plaza, Los Angeles, CA 90095, United States.

Abstract

Video sequences can be enhanced by a spatial deconvolution of any motion blur whose support does not exceed two pixels per frame. However, this deconvolution requires an accurate blur estimation and local deconvolution which is difficult for multiple local motions. We provide a discrete temporal filter whose coefficients are designed 1) to deconvolve blindly any uniform motion blur whose support does not exceed one pixel per frame, 2) to take into account for the sensor dead time between two consecutive frame (duty ratio/inter frame delay). The proposed filter enjoys optimality properties in terms of mean square error. In addition, it is demonstrated on real movies obtained from a smartphone and the *Middlebury* dataset.

Keywords: Motion blur, Poisson noise, flutter shutter, blind deconvolution, coded exposure.

1 Introduction

Most images and video sequences are affected by motion blurs due to camera or scene motions. The difficulty of motion blur is unfolded by its simplest example: the one dimensional uniform motion blur. The result is a convolution of the image with a one dimensional box kernel. The support of the kernel increases linearly with the aperture time and the relative camera scene motion. As soon as the support of this box kernel exceeds two pixels the blur is no more invertible, and the restoration becomes an ill posed problem. Even below two pixels this deconvolution depends on an accurate blur kernel estimate as pointed out in [1] that even state that "the shift-invariant blur assumption made by most algorithms is often violated" [1, "Introduction", p. 1964]. In addition, a multi-frame approach needs an accurate local motion estimation "which is a very challenging task even with user interactions" [2, "abstract"]. In a nutshell, blind motion blur deconvolution remains challenging and not completely solved yet.

A new photography method was proposed in [3]. The authors propose to attach a "flutter shutter" (coded exposure) to a camera. The photon flux is interrupted on sub-intervals of the exposure time and permits to get an invertible motion blur kernel. Numerically a flutter shutter is described by a binary shutter sequence -or flutter shutter code- that gives the intervals where the photon flux is interrupted. If the flutter shutter code is well chosen, a flutter shutter can guarantee the invertibility of any uniform motion blur. These considerations seemed *a priori*

*Yohann Tendero is supported on the Grants No: 12-DARPA-1042, UC Lab 12-LR-236660, ONR N000141210838 and NSF DMS-1118971. Stanley Osher is supported by the Keck foundation, NSF DMS 118971 and DMS 0914561.

limited to the design of flutter shutter cameras. However, an unforeseen consequence of the flutter shutter theory consists in the existence of temporal filters that perform movie blind uniform motion deconvolution [4]. The filter enjoys optimality properties in terms of mean-square-error (MSE) [5]. The utilization of these filters seemed restricted to specific cameras that can acquire photons without interruption between image frame.

In this paper, we extend [4] to cope with on-the-shelf camera that cannot integrate light continuously. Note that this is the case of most camera. Indeed, it is not possible to tune 1) the frame rate so that the video can be displayed using some given video format, 2) the exposure time so that the image frames are correctly exposed (no under/over exposure) and 3) have a 100% duty ratio (the camera integrates photons continuously). Therefore, we propose a convenient formalism that permits to deduce the adequate filter that 1) deblurs blindly video frames, 2) takes the sensor duty ratio into account and 3) is optimal in terms of MSE. The novelty compared to the literature is that we incorporate the sensor duty ratio[1] in the filter model.

Section 2 gives the filter formalism taking the noise model and the sensor duty ratio into account. Section 3 gives numerical details as well as the *explicit* discrete filter coefficients. Experiments on the *Middlebury* data base, camera phone movies and simulations are given in section 4.

2 The deblurring filter formalism

This section provides the formalism of the deblurring filter.

Video temporal filters consist in multiplying the k-th image frame, for $k \in \{0, ..., L-1\}$, by a weight $\alpha_k \in \mathbb{R}$. All images are then added together to get the filtered frame. The filter can be formalized without loss of generality (w.l.o.g.) for an arbitrary motion blur direction. Indeed, the use of a temporal filter does not require its knowledge. If the motion direction is known, any temporal video filter α boils down to the 1D convolution of the 1D restrictions of the observed scene in that direction, with the 1D filter $\frac{1}{|v|}\alpha(\frac{t}{v})$ where v is the velocity measured in (signed) pixels/frame. Since the observation has noise, the expected value at position x of the scene will be denoted by $u(x)$. Thanks to the optical frequency cutoff, the ideal scene u is assumed to be $[-\pi, \pi]$ band limited, and to have finite energy: $u \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. Consider the discrete temporal filter

$$\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+R)\Delta t[}(t), \tag{1}$$

where $\alpha_k \in \mathbb{R}$, $R, \Delta t > 0$, and a scene $u(x - vt)$ moving at velocity v in (signed) pixel/Δt. Here, $\frac{1}{\Delta t}$ represents the camera frame rate and R represents the duty ratio, i.e., the percentage of time where the camera integrates photons. As an example, when $R = 1$ the camera can integrate photons without break between two frames and the model boils down to the one covered in [4]. A more common case, that applies to many more sensors and practical situations, is the case where $R \in (0, 1)$ which implies that the camera has a “dead time” between consecutive frames namely on the time intervals of the form $[(k + R)\Delta t, (k + 1)\Delta t]$. Note that the case $R \in (0, 1)$ arises naturally in many applications. Indeed, it is not possible to tune 1) the frame rate $\frac{1}{\Delta t}$ so that the video can be displayed on some given video format, 2) the exposure time so that the image frames are correctly exposed (no under/over exposure) while having a 100% duty cycle, i.e., $R = 1$ in equation (1).

[1] The ratio of time where the camera integrates photons over the total time needed to produce one image frame.

We wish to design α so that it inverts the motion blur. The filter design is easier in a continuous setting. Therefore, we shall assume first that $\alpha(t) \in L^2(\mathbb{R})$ is arbitrary (not necessarily of the form given by equation (1)). In general such temporal filter is not usable in practice because the frame rate of a video camera can be high but is finite. Fortunately, the following theorem 2.1 permits to get back a filter of the form of (1) from any continuous filter $\alpha \in L^2(\mathbb{R})$.

The temporally filtered frame at pixel n is a linear combination of weighted acquired frames with weights α_k. Therefore, under the classic Poisson image acquisition model the filtered samples are

$$obs(n) \sim \sum_{k=0}^{L-1} \alpha_k \mathcal{P}\left(\int_{k\Delta t}^{(k+R)\Delta t} u(n-vt)dt\right).$$

(The notation $X \sim Y$ means that the random variables X and Y have the same law, $\mathcal{P}(\lambda)$ denotes a Poisson random variable with intensity λ.) Indeed, each observed frame is the realization of a Poisson noise with expectation equal to the motion blurred scene $\int_{k\Delta t}^{(k+R)\Delta t} u(n-vt)dt$.

Consequently, the expected value of the filtered frame is $\mathbb{E}(obs(n)) = (\frac{1}{v}\alpha(\frac{\cdot}{v})) * u)(n)$. (Here and elsewhere, $*$ denotes the classic continuous convolution on $\mathbb{R}$.) If the filtered samples are interpolated by the Shannon-Whittaker method as a $[-\pi,\pi]$ band limited function, the expectation of the Fourier transform of the filtered image is

$$\mathbb{E}(\hat{obs}(\xi)) = \hat{\alpha}(\xi v)\hat{u}(\xi). \tag{2}$$

(Here and elsewhere $\hat{u}$ denotes the classic continuous Fourier transform on $\mathbb{R}$.)

From (2) we deduce that the desired temporal filter is $\hat{\alpha}(\xi v) = \mathbb{1}_{[-\pi,\pi]}(\xi)$ or equivalently $\hat{\alpha}(\xi) = \mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$. Indeed, since u is $[-\pi,\pi]$ band limited this filter α restores perfectly the ideal scene u (in expectation). This means that the temporally filtered frame will be sharp. Indeed, from (2) we have $\mathbb{E}(f(\xi)) = \hat{u}(\xi)$ for $\xi \in [-\pi,\pi]$. In addition this filter provides the lowest variance (lowest MSE). Indeed, with this choice for α, the bias is null. Furthermore, it is shown in [5] that the MSE of the filtered frame is equal to $\int_{-\pi}^{\pi} \frac{\|\alpha\|_{L^2}^2 \|u\|_{L^1}}{|\hat{\alpha}(\xi v)|^2} d\xi$. From this formula and using Jensen's inequality we obtain that for a known velocity v the temporal filter $\alpha(t) = \mathrm{sinc}(tv)$ is not only able to deblur the motion blur but it is also optimal in terms of MSE.

Assuming the existence of a maximal velocity v_{max} we deduce that the use of the filter $\alpha(t) = \mathrm{sinc}(tv_{max})$ will grant a sharp image for all velocities $v \in [-|v_{max}|, |v_{max}|]$. Indeed, for any $v \in [-|v_{max}|, |v_{max}|]$ from (2) we have $\mathbb{E}(f(\xi)) = \mathbb{1}_{[-|v_{max}|\pi,|v_{max}|\pi]}(\xi v)\hat{u}(\xi) = \hat{u}(\xi)$ for any $\xi \in [-\pi,\pi]$. Having seen that the (time continuous) filter we wish to implement comes from a zoomed sinc funtion we can now provide its discretization following the form given by equation (1).

Discrete filter design Consider $\beta \in L^2(\mathbb{R})$ a temporal filter invertible for all velocities below $|v|$, i.e., $\hat{\beta}(\xi v) \neq 0$ for all $\xi \in [-\pi,\pi]$. Since u is $[-\pi,\pi]$ band limited the values of $\hat{\beta}$ outside $[-\pi,\pi]$ do not matter and we can w.l.o.g. assume that $\hat{\beta}$ is zero on $\mathbb{R}\backslash[-\pi,\pi]$. Therefore, we wish to deduce from β a discrete filter α of the form $\alpha(t) = \sum_k \alpha_k \mathbb{1}_{[k\Delta t,(k+R)\Delta t[}(t)$ such that $\hat{\alpha}(\xi v) = \hat{\beta}(\xi v)$ for $\xi \in [-\pi,\pi]$, which is equivalent to $R\Delta t\mathrm{sinc}(\frac{\xi v R\Delta t}{2\pi})e^{\frac{-iR\Delta t\xi v}{2}} \sum_k \alpha_k e^{-ik\Delta t\xi v} = \hat{\beta}(\xi v)$ and therefore to

$$\frac{\hat{\beta}(\xi v)e^{i\frac{R\Delta t\xi v}{2}}}{R\Delta t\mathrm{sinc}(\frac{\xi v R\Delta t}{2\pi})}\mathbb{1}_{[-\pi,\pi]}(\xi) = \sum_k \alpha_k e^{-ik\Delta t\xi v}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{3}$$

As soon as $|v|R\Delta t < 2$ the left-hand side of (3) belongs to $L^2([-\pi,\pi])$. Therefore, formula (3) can be seen as the Fourier series decomposition of the left hand side on the Fourier basis on the interval $[\frac{-T}{2},\frac{T}{2}]$ where $T=\frac{2\pi}{|v|\Delta t}$. Thus, the temporal sampling of the left hand side of (3) requires that $|v|\Delta t < 1$ in order to get $\frac{T}{2} > \pi$ so that $[-\pi,\pi]\subset[-\frac{T}{2},\frac{T}{2}]$. Hence, provided $|v|\Delta t < 1$ equation (3) is valid and $(\alpha_k)_k$ are explicitly given by the Fourier series coefficients of the function $\xi \mapsto \frac{\hat{\beta}(\xi v)e^{i\frac{R\Delta t\xi v}{2}}}{R\Delta t\,\mathrm{sinc}(\frac{\xi v R\Delta t}{2\pi})}\mathbb{1}_{[-\pi,\pi]}(\xi)$, i.e., $\alpha_k=\frac{1}{2\pi}\int_{-\pi|v|\Delta t}^{\pi|v|\Delta t}\frac{\hat{\beta}(\frac{\xi}{\Delta t})e^{i\frac{\xi R}{2}}}{R\Delta t\,\mathrm{sinc}(\frac{\xi R}{2\pi})}e^{ik\xi}d\xi$. Thus we have

Theorem 2.1. *Let $\beta \in L^2(\mathbb{R})$ be a band-limited temporal filter that satisfies $\hat{\beta}(\xi v)\neq 0$ for $\xi\in[-\pi,\pi]$,i.e., invertible for all $[-\pi,\pi]$ band limited functions. If $|v|\Delta t < 1$ there exists a discrete filter of the form $\alpha(t)=\sum_k \alpha_k \mathbb{1}_{[k\Delta t,(k+R)\Delta t[}(t)$, where $R\in(0,1]$ represents the sensor duty ratio, and $(\alpha)_{k\in\mathbb{Z}}\in \ell^2(\mathbb{Z})$ such that $\hat{\alpha}(\xi v)=\hat{\beta}(\xi v)$ on $[-\pi,\pi]$. Moreover, the coefficients are explicitly given by $\alpha_k=\frac{1}{2\pi}\int_{-\pi|v|\Delta t}^{\pi|v|\Delta t}\frac{\hat{\beta}(\frac{\xi}{\Delta t})e^{i\frac{\xi R}{2}}}{R\Delta t\, sinc(\frac{\xi R}{2\pi})}e^{ik\xi}d\xi$.*

We now give examples of discrete temporal filters designed to deblur movies.

3 Examples of discrete filter

As we have seen in section 2 the temporal filter $\alpha(t)=\mathrm{sinc}(v_{max}t)$ permits to guarantee a sharp image for all velocities $v\in[-|v_{max}|,|v_{max}|]$. Indeed, this ideal filter convolves spatially (in motion direction) the observed scene by a zoomed sinc function whose cutoff is beyond the cutoff of observed scene u. Consequently, the frequency content of the observed scene remains unchanged, despite the motion. In practice, this means that no velocity estimation and/or local deconvolution is needed: the observed image does not change if $v=0$ or if $|v|=|v_{max}|$. Thanks to the sampling theorem 2.1 the discrete filters inherit this property. If the frequency cutoff of the sinc function is larger than the frequency cutoff of the scene, the above theory entails that the resulting image is sharp. This remains valid for all velocities not exceeding $|v_{max}|$, where $|v_{max}|=1$ if we measure the velocity in pixel(s)/frame by normalizing the time unit between two frames so that $\Delta t=1$. In addition, the filter works for any motion direction. Therefore, the motion direction needs not to be estimated. To sum up, the deconvolution is blind but mathematically well posed. We now give the numerical method that implements this blind deconvolution and give explicitly the filter coefficients (α_k) varying the sensor duty ratio R.

Given L movie frames $(I_k)_k$ $k\in\{0,...,L-1\}$ the filtered image is $\sum_{k=0}^{L-1}\alpha_k I_k$. The filter coefficients obtained from theorem 2.1 and normalized so that $\int\alpha(t)dt=1$ are (0.00149798, −0.00264995, 0.00588158, −0.0220471, 1.03463, −0.0220471, 0.00588158, −0.00264995, 0.00149798) for a discrete filter of length $L=8$, $v_{max}=1$, $R=\frac{1}{2}$, (0.00404822, −0.00710249, 0.0154352, −0.0534351, 1.08211, −0.0534351, 0.0154352, −0.00710249, 0.00404822) when $L=8$, $v_{max}=1$, $R=\frac{3}{4}$ and (0.00946047, −0.0163343, 0.0342235, −0.106215, 1.15773, −0.106215, 0.0342235, −0.0163343, 0.00946047) when $L=8$, $v_{max}=1$, $R=1$ respectively. The corresponding functions $\alpha(t)$ and their Fourier transforms are depicted in figure 1.

4 Experiments

This section compares the effect of the temporal filter both quantitatively and qualitatively. The simulations are based on [6]. The table 1 contains PSNR values. Real experiments using the *Middlebury* database are shown in figures 2-3. Note that many more experiments and a gallery that allows to alternate between filtered and unfiltered frames is available at `http://www.math.ucla.edu/~tendero/blind_deconvolution/blind_deconvolution.html`.

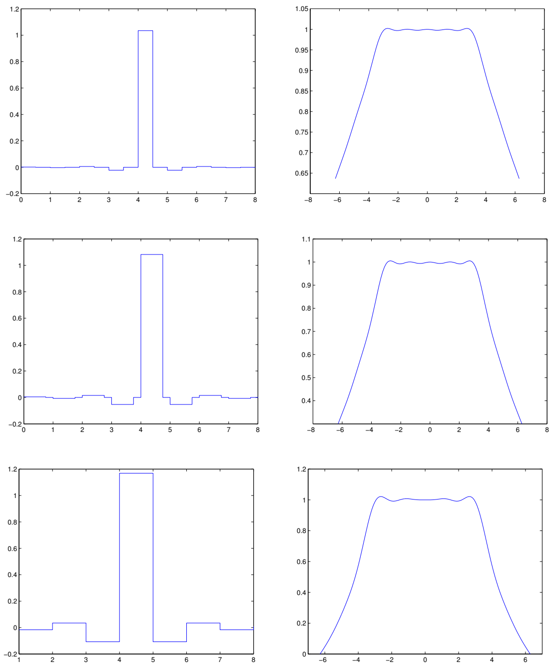

Figure 1: Let panels: the temporal filter value $\alpha(t)$. Right panels: the corresponding Fourier transforms (modulus) of $\alpha(t)$ that approximate the constant Fourier transform of the ideal filter ($\Delta t = 1$) and $v_{max} = 1$. The filter length is $L = 8$, $\Delta t = 1$, $v_{max} = 1$. From top to bottom: $R = \frac{1}{2}$, $\frac{3}{4}$ and $R = 1$.

Table 1: This table provides the PSNR evolution in dBs when the filter length is $L = 8$ and $R = 1$. The method permits to gain up to 3.2dB.

Velocity	$v = 0$	$v = 0.5$	$v = 1$	$v = 1.5$	$v = 2$
House	-0.7	-0.5	$+1.1$	$+2$	$+1.4$
Boat	-0.7	-0.4	$+1.5$	$+3.2$	$+1.1$

Figure 2: This figure shows the result of the proposed sinc filter on two *Middlebury* sequences, *army* (top-left) and *basketball* (bottom-left) and two self made camera phone sequences (top right, bottom right). Each pair shows one frame of the sequence followed by the filtered version. Details are shown in figure 3. The duty ratio factor R was tuned empirically and is set to $R = 0.8$ in these experiments. Note that a gallery that allows the user alternated between filtered and unfiltered image frames is available at `http://www.math.ucla.edu/~tendero/blind_deconvolution/blind_deconvolution.html`.

Figure 3: This figure shows the result of the proposed sinc filter on two self made camera phone sequences. On the left side: one frame of the sequence, on the right side: crops of the filtered image.

5 Conclusion

We have proposed a new filter to post-process video sequences acquired by classic cameras. The filter design takes into account for the inter frame delay or duty ratio (dead times between consecutive frames). We show that video sequences can be deblurred by a fast fixed temporal filter that requires only the knowledge of the maximal observed velocity and the camera frame rate/exposure time set up. In other words, 1) it is possible to turn any camera into a camera that ensures a sharp image for a broader range of velocities than a standard camera 2) it is possible to perform an automatic blind movie deconvolution.

Note Other examples and C++ implementation are available at `http://www.math.ucla.edu/~tendero/blind_deconvolution/blind_deconvolution.html`

References

[1] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1964–1971.

[2] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring using multiple images," *Journal of Computational Physics*, vol. 228, no. 14, pp. 5057–5071, 2009.

[3] A. Agrawal and R. Raskar, "Resolving objects at higher resolution from a single motion-blurred image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[4] Y. Tendero and J.-M. Morel, "An optimal blind temporal motion blur deconvolution filter," *IEEE Signal Processing Letters IEEE*, vol. 20, no. 5, pp. 523–526, 2013.

[5] Y. Tendero, J.-M. Morel, and B. Rougé, "The Flutter Shutter Paradox," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 813–847, 2013.

[6] Y. Tendero, "The Flutter Shutter Camera Simulator," *Image Processing On Line*, vol. 2012, 2012.