SIAM J. Imaging Sciences
Vol. 6, No. 2, pp. 813–847

The Flutter Shutter Paradox*

Yohann Tendero†, Jean-Michel Morel‡, and Bernard Rougé§

Abstract. Photography is the art of acquiring as many photons as possible of a given scene. In classic cameras, the aperture time is irremediably limited by the risk of a motion blur when the camera and the scene are in relative motion. Nevertheless, two recent camera concepts, the Agrawal et al. *flutter shutter* and the Levin et al. *motion-invariant photography* permit one to extend indefinitely the exposure time while guaranteeing an invertible motion blur. In this paper, a complete mathematical theory of these new technologies is proposed. Modeling the capture noise, the theory furnishes explicit formulas for the signal to noise ratio (SNR) of the final image after deconvolution when the motion is uniform. It puts in evidence the existence of two variants, the *analog flutter shutter* and the *numerical flutter shutter*. The results of the resulting quantitative comparison are slightly paradoxical. First, it is shown that the best camera aperture strategies are always flutter shutters, even when the aperture time is a priori fixed. Second, it is shown that the SNR increase obtained by using a *flutter shutter* in the presence of a known motion remains bounded, even with an infinite exposure time. Incidentally, the theory gives the formula of the optimal classic snapshot in the presence of motion and compares its performance to the optimal flutter shutter.

Key words. motion blur, Poisson noise, snapshot, *flutter shutter*, optimization, *motion-invariant photography*, SNR

AMS subject classifications. 42A38, 60G99

DOI. 10.1137/120880665

1. Introduction.

Classic digital cameras are devices counting at each pixel sensor the number of photons emitted by the observed scene during an interval of time Δt called exposure time. Due to the nature of photon emission the counted number of photons is a Poisson random variable. Its mean would be the ideal pixel value. The difference between this ideal mean value and the actual value counted by the sensor is called (shot) noise. The ratio of the mean of the photon count over its standard deviation is called the signal to noise ratio (SNR). At (very) low SNR the noise is so strong compared to the underlying signal that it is almost impossible to distinguish the scene being observed from the noise. Therefore, photography has been striving to achieve the highest possible SNR. In passive imaging systems, the only

*Received by the editors June 12, 2012; accepted for publication (in revised form) January 24, 2013; published electronically May 9, 2013. This work was partially supported by the Direction Générale de l'Armement, the Office of Naval Research under grant N00014-97-1-0839, the European Research Council under advanced grant Twelve labours, and the UCLA Department of Mathematics. The algorithm and simulation facility linked to this paper are the object of a joint submission to Image Processing On Line, which will appear at www.ipol.im under the title "The Flutter-Shutter Camera Simulator."

http://www.siam.org/journals/siims/6-2/88066.html

†Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095 (tendero@math.ucla.edu).

‡Centre de Mathématiques et Leurs Applications (CMLA), École Normale Supérieure de Cachan, 94235 Cachan, France (morel@cmla.ens-cachan.fr).

§Centre d'études spatiales de la biosphère (CESBIO), 31401 Toulouse cedex 9, France (rougebe@gmail.com).

way to increase the SNR is to accumulate more photons by increasing the exposure time Δt.

If the scene being photographed moves during the exposition process, or if the scene is still and the camera moves, the resulting images are degraded by motion blur (see Figure 1). Obtaining longer exposure time without blur can therefore be seen as one of the core problems of photography. The first photographs taken by Nicephore Niepce required several hours, a time incompatible with live subjects or even with outdoor static scenes exposed to the sun. Ever since, photography has been subject to the problem of finding the right compromise between a short exposure time, which avoids the effects of motion blur, and a longer exposure time, which permits many more photons to reach the sensor and therefore increases the SNR.

Motion deblurring is the combination of two dependent problems: (a) the kind of kernel applied to the images which depends here on the motion, and (b) the actual deblurring method, where the kernel may have to be estimated a posteriori or not. Motion blur arises from multiple causes and is very common even for consumer level photography, where it is partly compensated by optical, mechanical, or digital stabilizers. Stabilizers cannot compensate for motion blur of arbitrary length (support) since they are limited by mechanical and technical issues. In most cases the size of the blur support will increase proportionally to the exposure time. Thus they require a "small" exposure time despite the stabilization device. The difficulty of motion blur is illustrated by its simplest example, the one-dimensional (1D) uniform motion blur. The result of a too long exposure during the motion on the image is nothing but a convolution of the image with a 1D window shaped kernel. The support of the kernel increases linearly with the exposure time and the velocity of the motion. As soon as the exposure time is too long, this blur is no longer invertible, and the restoration problem is ill posed.

A revolutionary alternative to classic photography was proposed in [3, 4, 6, 76, 75], where the authors suggest modifications in the acquisition process to get invertible motion blur kernels by using a *flutter shutter*. These authors propose using a binary shutter sequence interrupting the flux of incoming photons on well-chosen time subintervals of the exposure time interval. If the shutter sequence is well chosen, invertibility is guaranteed for blurs with arbitrary size support. Hence, replacing the classic camera shutter by a *flutter shutter*, it becomes possible to use any integration time. This also means that the exposure time on a given scene can be much longer: many more photons are therefore sensed by the camera. Thus, the *flutter shutter* looks like a magic solution that should equip all cameras. Yet, does that mean that one can increase the SNR indefinitely by an increased exposure at no cost from the motion blur side?

This paper starts by modeling the stochastic photon capture by a light sensor realistically, taking into account both the classic shot noise and the obscurity noise. To cope with the fact that the image noise may be colored after deconvolution, the "spectral" SNR function defined in [12] by Boracchi and Foi is used and extended to a "spectral SNR on average" to reflect the final $RMSE$.

The modeling will treat in the same formalism all possible types of *flutter shutter*, including an analog model, a digital model, the classic Agrawal et al. *flutter shutter*, and the Levin et al. *motion-invariant photography* as well. For all, a closed formula will be given for the spectral SNR, permitting us to compare them theoretically.

Among the kinds of possible setups, the most flexible, adaptive to all kinds of motions, is the digital *numerical flutter shutter*, which allows for negative gains. It is proven that it can

Figure 1. *Simulated observed (blurry and noisy) image (left). The blur interval length is* 52 *pixels. Notice the stroboscopic effect of the flutter shutter apparatus. Reconstructed image (right). Such a reconstruction is not possible without a flutter shutter camera.*

also realize the best SNR. One of the striking results of this mathematical analysis is the proof that, when the object velocity is a priori known, the best *numerical flutter shutter* code is given by the Fourier series coefficients of a (zoomed) sinc function. The proposed formalism also permits us to compute by a closed formula the optimal aperture time for a classic snapshot when the velocity of the photographed object is known. This snapshot theory allows us to match on an equal footing the new *flutter shutter* apparatus against a plain old camera. This comparison leads to what we call the two *flutter shutter paradoxes*. The first surprising result is that the *flutter shutter* always beats a standard camera slightly, even when using exactly the same exposure time. On the other hand, an infinite exposure time, accumulating many more photons than a classic snapshot, does not grant an infinite SNR. This rather disappointing fact makes motion photography significantly different from the classic steady photography, where, by increasing the aperture time, any SNR can be achieved.

1.1. Related work.

Blind deconvolution techniques [15, 50, 88, 31, 55, 14, 45] aim at estimating the blur and recovering the sharp image directly from the blurred one. Deconvolution algorithms have been developed intensively [40, 72, 95, 21, 106, 35, 82]. For example, in [111, 109] the authors suggest a modification of the Richardson–Lucy method [81, 58] to control the artifacts of the restored image. Other priors have been investigated in [112, 61, 43]. In [51] Levin et al. use natural image statistics to estimate the blur. In [91, 87, 7, 80, 103, 42, 90, 36, 18, 37, 9, 24, 44, 23, 39, 104, 41, 26, 86, 85] good results are shown for the blur estimation and/or deblurring problem. Using the compressive sensing framework, the question of the order of the pair image estimation/motion estimation for deconvolution is addressed in [38]. Nevertheless, the power spectrum of images acquired with a blur of more than two pixels contains several zero crossings. Thus, useful information for image quality is irreversibly lost. Hence, no matter how sophisticated the image reconstruction is, it is virtually impossible to recover a deblurred image without strong hypotheses on the underlying landscape. Such strong hypotheses are unrealistic for most images. The results are therefore in practice poor [84]. In an attempt to transform the blur problem into a well-posed problem the authors of [16, 17, 77, 19] proposed using two photographs with different blurs instead of one. In [110] the authors use a long exposure image and another one, sharp but noisy, to deblur the first. In [6] the authors suggest taking several images with several exposure times so that the blur in each image is different. If the zeros of each Fourier transform (FT) do not coincide, then it is possible to deblur by picking nonzero coefficients in each image. In [92] a similar hybrid scheme is used where an image at high resolution and long exposure is taken simultaneously with a burst of low resolution and short exposure. In [11] a Mumford–Shah-like variational model is proposed to simultaneously estimate the blur and deblur in the presence of multiple object motions from videos.

In [22] the authors address the question of an automatic tuning of the exposure time to avoid overexposure in the case of still imaging. Finally, in [12] the authors treat the question of the optimal exposure time depending on the SNR of the restored image using a conventional camera. They consider the case of noninvertible blurs with supports larger than two pixels using a regularized deconvolution [27]. In [96] the authors use a full multi-image framework, acquiring a bunch of sharp but noisy images and recovering a sharp image with increased SNR. For a review on multi-image denoising the reader can refer to [13]. Conversely in [34] the authors reconstruct a movie from a single image using a temporally and spatially varying mask placed on the aperture. The mask helps to encode the spatio-temporal information. In [71, 93, 105, 28, 102, 107] the authors use hybrid or complex camera systems. Unfortunately this kind of scheme may lead to other problems, such as an expensive computational cost or hardware issues.

The simplest hardware set up seems to be proposed in [3, 4, 6] by Agrawal and coworkers. The new acquisition process modulates the photon flux into the camera by opening and closing the camera shutter according to certain well-chosen pseudorandom binary codes. In the case of a uniform motion in front of the camera, the resulting blur kernel becomes invertible (there are no zeros in its FT), however big the velocity is. The visual result of an image acquired by a *flutter shutter* is close to a stroboscopic image, which can nonetheless give back a neat image by deconvolution. A compressive sensing *flutter shutter* camera was designed in [89] using random sequences where a blurry and low resolution image is acquired and processed to a neat and high resolution image. Roughly speaking, the *flutter shutter* ensures that no information is lost by the motion blur; the compressed sensing technique deals with the increase of resolution. The compressed sensing technique is also used in [78] for spatio-temporal upsampling. Alternatively the case of periodic events was investigated in [79]. In [46, 99, 63, 70] the authors use an active dynamic lighting pattern in place of the shutter to recreate a *flutter shutter* effect. The theory presented herewith works for this setup. In [64] the *flutter shutter* apparatus is applied to iris images and in [108] to bar-codes. In [62] the authors propose optimizing the binary *flutter shutter* code in function of the velocity of the scene.

In [94] the authors use a local deblurring user-driven scheme on a *flutter shutter* embedded camera to deal with spatially varying blurs caused by the presence of several velocities in the observed scene. In [83] the authors treat the question of denoising an image taken by a *flutter shutter* camera and also suggest a user-assisted estimation of the blur. Their conclusion is that the denoising should be applied both before and after deconvolution. In [25] the authors treat the question of a posteriori motion estimation using a *flutter shutter*. In [32] a per pixel *flutter shutter* is used to build a camera that allows a postcapture balance between spatial and temporal resolutions of movies. A multicamera equipped with *flutter shutters* is investigated in [2] and used to increase the frame rate of a single camera while having an increased amount of light captured compared to the equivalent high-speed camera. A single camera equipped with a mask on the aperture and an array of light sources is used in [48] to construct the visual hull of an object (shape from silhouette). In [100] four projectors projecting a handcrafted pattern on the scene are used to detect the depth edges. Another solution to get an invertible motion blur using only one image was found in [53], where Levin et al. suggested moving the camera in the direction of the motion during the exposure time. The authors use a constant acceleration motion in order to make the resulting kernel invertible and spatially invariant to the velocity. Hence an a priori knowledge of the motion direction is required. This approach has been generalized in [20] to the case of unknown directions, but it uses two images instead of one. In [65] the *motion-invariant photography* apparatus is implemented using the lens of the camera. In all cases, these approaches cause blur in static parts of the scene. Yet, thanks to the invertibility (well-posedness of the recovery problem), in both cases, the sharp image can be recovered by a deconvolution. Notice that only one image is acquired and recovered at the end of the process. Alternatively in [52, 101, 56, 10, 29, 66, 68, 60] the authors use a temporally fixed and spatially varying mask in order to estimate the depth and/or refocus the out of focus part to get an always in focus (neat) image. In [33] the authors deal with the question of the optimal tradeoff between depth of field and exposure time. In [30] the authors take advantage of CMOS imaging sensors to implement a *coded rolling shutter* to trade vertical resolution for an increased dynamic range. The authors of [98] also suggest using a camera equipped with a mask on the aperture camera and taking purposely out of focus images with a mask to increase the dynamic range. Their conclusion is rather negative: "None of the possible combinations of aperture filter and deconvolution algorithm were able to consistently reduce the dynamic range of the captured image without excessively degrading image quality." Another computational camera is designed in [67], where the aperture is equipped with a mask and the sensor is moved at a constant velocity during the exposure. It is used to control the depth of field, creating *bokeh* or a depth invariant blur size. Another camera prototype was designed in [57], where the authors suggest a programmable aperture (mask). It is also used for depth and digital refocusing. An interesting implementation of many computational photography apparatus, the *Frankencamera*, was proposed in [1]. An even more complex scheme involving a fixed mask close to the sensor and a dynamic mask on the aperture is investigated in [5], where the authors explore the feasibility of postprocessing tradeoffs between spatial, angular, and temporal resolutions. Finally, reviews of computational photography can be found in [113, 59, 73, 74].

1.2. Overview. Section 2 proposes a general mathematical framework for image acquisition using a physical Poisson model for the photon capture process, including the obscurity noise. This model suits our context well since all noise terms inherent to image sensing are taken into account without any approximation.

In section 3 the mathematical model of section 2 is used to analyze the *numerical flutter shutter*, a digital implementation of the classic *flutter shutter* method. This setup is the most flexible, adaptive to all motion, and allows for negative gains. The *numerical flutter shutter* does not reduce the number of photons caught by the sensor and it is proven later on that it yields the best possible SNR. It is proven that it actually works, and for *any flutter shutter* gain function a formula providing the SNR of the neat deconvolved image is given. The *numerical flutter shutter* gain function is, in principle, piecewise constant. Nevertheless, it is useful for the theory to extend it to continuous gain functions. In section 3.2 a reverse formula permits us to get back an equivalent piecewise constant *numerical flutter shutter*.

Section 4 investigates classic analog implementation of the *flutter shutter*. This *analog flutter shutter* is a generalization of the original Agrawal et al. *flutter shutter* which allows for smoother, nonbinary, gain functions. For any *analog flutter shutter* apparatus, an explicit formula to measure the SNR of the deconvolved sharp image directly is given.

Section 5 proves that the *numerical flutter shutter* SNR is always larger than the *analog flutter shutter* SNR with the same gain function. A snapshot theory is also developed in section 5. The standard camera apparatus is explored as a particular *flutter shutter* strategy. The SNR of the deconvolved image is calculated for any standard acquisition strategy. The standard camera is optimized to get the best SNR possible, taking the deconvolution into account. This yields a precise definition of the best possible snapshot in the presence of known motion. This best snapshot is used later on as a reference in terms of SNR.

In section 6 the Levin et al. *motion-invariant photography* is proven to be a particular case of the general *analog flutter shutter* theory. The SNR of the *motion-invariant photography* apparatus is computed and compared with the other *flutter shutter* strategies. This section also proposes implementing the *motion-invariant photography* kernel using a *numerical flutter shutter*. This permits us to generalize the *motion-invariant photography* method to the case where the direction of the relative velocity v is not a priori known.

Section 7 proves that the use of *any flutter shutter* does not increase the SNR of the sharp recovered image indefinitely. It is proven that the best *flutter shutter* entails a 17% increase of the SNR compared to the best snapshot. It is also proven that, even though the exposure time remains unchanged, the *flutter shutter* beats the standard camera with classic aperture. These two results are the *flutter shutter paradoxes*.

All of the results are developed for 1D sampling in the direction of motion. For a two-dimensional (2D) sampling coupled with motion blur, some adaptation of the results may be required. Indeed, for the sake of simplicity, we are assuming that the motion blur is parallel to one of the sampling grid axes. Obviously, this is generally not true for uncontrolled camera or unknown object motions. By assuming this parallelism we are simply avoiding unnecessary complications. Nevertheless, the study requires an extension to 2D sampling. Indeed, the results herewith apply exactly to 2D sampling only in the case where the motion is parallel to one axis. Common sense suggests that the conclusions will be essentially the same in a general 2D sampling geometry. But we shall sketch the adaptation to general 2D sampling in a few sentences.

Assuming as we do that the image acquisition is Shannonian, namely, that the frequency cutoff is compatible with the image grid sampling, an easy extension to any 2D grid can be made by considering the rotation resampling operator on $L^2(\mathbb{R}^2)$ that computes from the image samples on the current grid its samples on a grid parallel to the motion. By this image rotation, which is isometric in L^2, a Gaussian noise in the captors remains a white Gaussian noise after resampling. Thus, the extension of the study from one to two dimensions becomes easy if we can replace the white Poisson noise on the initial samples, which is signal dependent, by a white Gaussian noise. By a classical trick, this can be done by the Anscombe transform [8] commonly used in denoising papers [49]. By assuming an application of the Anscombe transform to all samples, we could have presented the whole theory in the Gaussian white noise framework, which is immediately adaptable to any 2D grid. But we preferred in the current exposition to avoid this simplification and to treat the actual Poisson noise.

2. Image acquisition model.

Formalizing the *flutter shutter* requires an accurate continuous stochastic model of the photon capture by a sensor array. Without loss of generality (w.l.o.g.) the formalization will be done in the case where the sensor array is 1D and where the photographed object is conceived as a "landscape" moving in a direction parallel to the sensor array. Let $\mathbf{P}_l : \mathbb{R}^+ \times \mathbb{R}$ be a bidimensional Poisson process of intensity $l(t,x)$ $\forall (t,x) \in \mathbb{R}^+ \times \mathbb{R}$ (here l is called the landscape, and t and x are the time and spatial positions, respectively). This means that the observation of a pixel sensor (photon counter) of unit length centered at x using an exposure time of Δt is a Poisson random variable $\mathbf{P}_l\left([0,\Delta t] \times [x-\frac{1}{2}, x+\frac{1}{2}]\right) \sim \mathcal{P}(\int_0^{\Delta t}\int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(t,y)dydt)$, where Δt is the exposure time, $[x-\frac{1}{2}, x+\frac{1}{2}]$ represents the normalized sensor unit, $X \sim P$ means that a random variable X has law P, and $*$ denotes the convolution (viii). (Here and in the rest of the text, Latin numerals refer to the formulas in the appendix.) In other words the probability of observing k photons coming from the landscape l seen at the position x on the time interval $[0,\Delta t]$ and using a normalized sensor is

$$\frac{\left(\int_0^{\Delta t}\int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(t,y)dydt\right)^k e^{-\int_0^{\Delta t}\int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(t,y)dydt}}{k!}.$$

From now on we assume $l(t,x) = l(x - tv(t))$, and mainly $v(t) \equiv v$. For sampling purposes we assume that the theoretical landscape l is seen through an optical system with a point spread function g.

Definition 2.1. *We call the ideal landscape the deterministic function* $u = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * l$, *where* g *is the point spread function of the optical system providing a cutoff frequency.*

In other words, thanks to the convolution with g the acquisition system is able to sample u. We shall denote by $u(x)$ the ideal (noiseless) pixel landscape value at a pixel centered at x, as it could be obtained only after infinite exposure. Notice that the landscape u contains in itself all spatial integrations required from the PSF g and from the normalized pixel sensor.

Definition 2.2 (ideal acquisition system). *The image acquired by the* ideal *acquisition system, before sampling, corresponds to samples of the Poisson process* $\mathbf{P}_l$. *The intensity* u

(ideal landscape value) is related to the landscape l by (2.1) and is band limited:

(2.1)
$$\mathbf{P}_l\left([t_1,t_2]\times\left[x-\frac{1}{2},x+\frac{1}{2}\right]\right)\sim\mathcal{P}\left(\int_{t_1}^{t_2}\int_{x-\frac{1}{2}}^{x+\frac{1}{2}}(g*l)(y-vt)dydt\right)\sim\mathcal{P}\left(\int_{t_1}^{t_2}u(x-vt)dt\right).$$

Definition 2.3 (real acquisition system with noise included in the landscape). *A realistic acquisition system adds a landscape independent noise also known as dark noise (or obscurity noise or thermal noise). Assuming that this noise has variance η, (2.1) entails*

$$\mathbf{P}_{l+\eta}\left([t_1,t_2]\times\left[x-\frac{1}{2},x+\frac{1}{2}\right]\right)\sim\mathcal{P}\left(\int_{t_1}^{t_2}\left(u(x-vt)+\eta\right)dt\right). \tag{2.2}$$

Since all computations using the "noisy" landscape $u+\eta$ remain formally the same as with a noiseless ideal system, we will assume that u already contains the obscurity noise in itself. Notice that η being a constant, $u+\eta$ and u have the same cutoff frequency. We assume in what follows that $u\in L^1\cap L^2(\mathbb{R})$. This assumption will be necessary to apply some of the mathematical formulas, but it represents no artificial restriction on the acquisition physical model. Indeed, first, the average photon emission is always bounded. Second, taking a large enough support, we can always suppose w.l.o.g. that the landscape has bounded support and that the acquisition time is large but not infinite. Thus we can assume that the noise is zero at infinity. Under these conditions $u\in L^1\cap L^2$.

2.1. Sampling and interpolation. Since the optical kernel g provides a cutoff frequency, u is band limited; namely, $\hat{u}(\xi)$ (see the definition (xxiv) of Fourier transform (FT) in the appendix) is supported in $[-\pi,\pi]$. It could therefore be sampled at unit rate. The discrete sensor observations, or samples, will be denoted by $e(n)$ for $n\in\mathbb{Z}$. Given a discrete array observation $e(n)$, $n\in\mathbb{Z}$, its band-limited interpolate $e(x)$ $x\in\mathbb{R}$ is defined by the Shannon–Whittaker interpolation as

$$e(x)=\sum_{n\in\mathbb{Z}}e(n)\mathrm{sinc}(x-n)\qquad(\mathrm{sinc}(x)=\tfrac{sin(\pi x)}{\pi x}). \tag{2.3}$$

2.2. Noise measurement. We call the signal to noise ratio (SNR) of a random variable X the ratio $SNR(X):=\frac{|\mathbb{E}X|}{\sqrt{var(X)}}$.

For example, if $\hat{\mathbb{u}}_{est}(\xi)$ is an estimation of the landscape $\hat{u}(\xi)$ based on a noisy observation of u, likewise, we call the "spectral SNR" of $\mathbb{u}_{est}$ the frequency dependent ratio defined by

$$SNR^{spectral}(\hat{\mathbb{u}}_{est}(\xi)):=\frac{|\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)|}{\sqrt{(var(\hat{\mathbb{u}}_{est}(\xi))}}\qquad\text{for }\xi\in[-\pi,\pi] \tag{2.4}$$

and introduced by Boracchi and Foi in [12]. We call the "spectral-averaged" SNR the ratio defined by

$$SNR^{averaged}(\hat{\mathbb{u}}_{est}):=\frac{\frac{1}{2\pi}\int|\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)|\mathbb{1}_{[-\pi,\pi]}(\xi)d\xi}{\sqrt{\frac{1}{2\pi}\int var(\hat{\mathbb{u}}_{est}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi))d\xi}}. \tag{2.5}$$

Proposition 2.4. *Given* $\hat{\mathrm{u}}_{est}(\xi)$ *an unbiased estimator of* $\hat{u}(\xi)$*, then*

$$SNR^{spectral-averaged}(\hat{\mathrm{u}}_{est}) = \frac{C}{RMSE(u, u_{est})}.$$

The proof is a direct consequence of Fubini's theorem applied to the bias and variance decomposition of the MSE. In what follows, all estimators are unbiased and the *flutter shutter* is optimized in terms of $RMSE$. In the case of still photography, namely, when $v = 0$, then (2.2) and the above definitions permit us to prove that the SNR satisfies $SNR(x) = \sqrt{u(x)L\Delta t}$, where $L\Delta t$ is the total exposure time. It is therefore proportional to the square root of both the exposure time and the light intensity.

Remark. In a passive optical system we have no control over the landscape light emission $u(x)$. No lighting is possible to boost the photon emission. Thus the only secure way to increase the SNR is to increase the exposure time $L\Delta t$.

From now on we assume $l(t, x) = l(x - tv(t))$, and mainly $v(t) \equiv v$. Hence all the former discussion made on the acquisition system, sampling, and interpolation holds.

Theorem 2.5. *The standard motion blur is equivalent to an image obtained by a convolution of the ideal landscape* u *by a fixed (window shaped) kernel* $\mathbb{1}_{[0,b]}$*, where* b *is the blur length, equal to* $Lv\Delta t$.

Proof. The ideal landscape u is moving in the camera frame at a speed v (counted in pixels per second), and using (2.2) we get that the acquired image at position x can be *any* realization of $\mathbf{P}_l([0, L\Delta t]\times[x-\frac{1}{2}, x+\frac{1}{2}]) \sim \mathcal{P}(\int_0^{L\Delta t} u(x-vt)dt) \sim \mathcal{P}\left((\frac{1}{v}\mathbb{1}_{[0,Lv\Delta t]} * u)(x)\right)$. ■

In this case the expected value and variance of a pixel sensor at position x are equal to $\frac{1}{v}(\mathbb{1}_{[0,Lv\Delta t]} * u)(x)$. The quantity $Lv\Delta t$ is nothing but the length of the blur b (in pixels).

Remark. The convolution with an $h = \mathbb{1}_{[0,Lv\Delta t]}$ (standard blur) function is a noninvertible transformation as soon as the first zero of the FT of h is in the support of $\hat{u}$. This makes ill posed *any* restoration process of u. The purpose of the *flutter shutter* method will be to replace $\mathbb{1}_{[0,Lv\Delta t]}$ with a function whose convolution remains invertible for arbitrary $Lv\Delta t = b$. If the first zero of $\hat{h}$ is outside the support of $\hat{u}$, then the motion blur is said to be negligible and is actually invertible.

3. The numerical flutter shutter.

After having treated the classic image acquisition strategies, we are now in a position to treat the various *flutter shutter* strategies and to compare them to the classic ones. Two things are at stake: first, to prove that the various *flutter shutters* actually work, and, second, to evaluate the SNR of the resulting image and to compare it to the SNR of classic strategies. The hope would be that the *flutter shutter* retains the very interesting feature of the multi-image denoising, namely an increase of the SNR by a factor proportional to $\sqrt{L\Delta t}$, the total exposure time. We shall see that this is not so. The *numerical flutter shutter* method consists of a numerical sensor gain modification taking place *after* the acquisition by the sensor. Roughly speaking the camera takes a burst of L images using an exposure time Δt. The kth image is multiplied, for $k \in 0, \dots, L-1$, by an $\alpha_k \in \mathbb{R}$ gain. Then all images are added to obtain *one* observed image, the *flutter shutter*. The exposure time Δt must be small enough so the blur of each image is negligible (definition in section 5). This technique is similar to the multi-image acquisition strategy but does not use any registration technique. According to, for example, [69, 47] an image sensor can have a duty ratio of nearly 100% (the duty ratio is the ratio of light integration time over readout, storage, and reset times, that is, the percentage of useful time). It means that a sensor can integrate light without interruption. Thus, the *numerical flutter shutter*, as described below without "dead time" between two consecutive gains a_k, is perfectly reasonable from a technological point of view. Nevertheless, it seems that its interest is limited: why not keep all images instead of adding them all? One of the obvious reasons is compression, particularly for Earth observation satellites. In that case the motion blur due to a drift in satellite trajectory estimate could be eliminated by a *flutter shutter*, without any additional transmission (or computational) burden if only the *flutter shutter* image (the sum) is transmitted. The kth acquired elementary image at a pixel at position n is a realization of $\mathcal{P}(\int_{k\Delta t}^{(k+1)\Delta t} u(n-vt)dt)$. The *flutter shutter* observation is obtained by combining the kth output with weight α_k. Thus the *flutter shutter* output at a pixel centered at n is

$$obs(n) \sim \sum_{k=0}^{L-1} \alpha_k \mathcal{P}\left(\int_{k\Delta t}^{(k+1)\Delta t} u(n-vt)dt\right), \tag{3.1}$$

where by construction $obs(n)$ are obtained for $n \in \mathbb{Z}$ and are independent. Indeed, the sensors are disjoint and do not receive the same photons. In the following it will be useful to associate with the *flutter shutter* its *code* defined as the vector $(\alpha_k)_{k=0,\dots,L-1}$, and its *flutter shutter function* defined by $\alpha(t) = \alpha_k$ for $t \in [k\Delta t, (k+1)\Delta t[$.

Definition 3.1. *Let $(\alpha_0, \dots, \alpha_{L-1}) \in \mathbb{R}^L$ be a flutter shutter code. We call numerical flutter shutter samples at position n of the landscape u at velocity v the random variable*

$$obs(n) \sim \sum_{k=0}^{L-1} \alpha_k \mathcal{P}\left(\int_{k\Delta t}^{(k+1)\Delta t} u(n-vt)dt\right). \tag{3.2}$$

We call a numerical flutter shutter its band-limited interpolate $obs(x) \sim \sum_{n\in\mathbb{Z}} obs(n)\mathrm{sinc}(x-n)$. We call a flutter shutter function the function

$$\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t). \tag{3.3}$$

Remark. It is good to keep in mind the following trivial and less trivial examples:

1. $\alpha_k = 1\ \forall k \in \{0,\dots,L-1\}$ (pure accumulation prone to motion blur).
2. $\alpha_k = 0$ or $1\ \forall k \in \{1,\dots,L-1\}$ with $\sum \alpha_k = \frac{L}{2}$ (the Agrawal et al. *flutter shutter* has this generic form).
3. $\alpha_0 = 1$ and $\alpha_k = 0\ \forall k \in \{1,\dots,L-1\}$ (standard snapshot).

Theorem 3.2. *The observed samples of the numerical flutter shutter are such that, for $n \in \mathbb{Z}$,*

$$\mathbb{E}\left(obs(n)\right) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n) \quad \textit{and} \quad var(obs(n)) = \left(\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right) * u\right)(n). \tag{3.4}$$

Proof. From the *numerical flutter shutter* samples definition (3.2),

$$\mathbb{E}\left(obs(x)\right) = \sum_{k=0}^{L-1} \alpha_k \int_{k\Delta t}^{(k+1)\Delta t} u(x - vt)dt = \int_0^{L\Delta t} \alpha(s)u(x - vs)ds. \tag{3.5}$$

Thus,

$$\mathbb{E}\left(obs(x)\right) = \int_0^{Lv\Delta t} \frac{1}{v}\alpha\left(\frac{y}{v}\right) u(x - y)dy = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(x). \tag{3.6}$$

Similarly, from (3.1), $var(obs(x)) = \sum_{k=0}^{L-1} \alpha_k^2 \int_{k\Delta t}^{(k+1)\Delta t} u(x - vt)dt = \left(\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right) * u\right)(x)$. ∎

Notice that $obs(x)$ is not necessarily a Poisson random variable. However, if all α_k are equal to 0 or 1, then by (3.5) $obs(x) \sim \mathcal{P}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(x)$ because we are adding independent Poisson random variables. On the other hand, if for some k, $\alpha_k \notin \{0, 1\}$, then if $X \sim \mathcal{P}(\lambda)$, then $\alpha_k X$ is *not* a Poisson random variable.

Ideal numerical flutter shutter. The formulas of Theorem 3.2 giving the samples obtained by *flutter shutter* appear not to depend on the particular form of α as a piecewise constant function on a finite set of intervals with length Δt. As a matter of fact, we can envisage *any* function $\alpha \in L^2(\mathbb{R})$ to be a *numerical flutter shutter* for an ideally controlled camera where at each instant the gain $\alpha(t)$ is changed. This leads to the following definition and corollary.

Corollary 3.3. *We call a* continuous numerical flutter shutter *any band-limited and bounded gain function $\alpha \in L^2(\mathbb{R})$. Then the formulas* (3.4) *of Theorem* 3.2 *are still valid.*

Proof. By assumption the observed ideal landscape u belongs to $L^1 \cap L^2(\mathbb{R})$. We recall that $L^1 * L^2 \subset L^2$ and $L^2 \cap L^2 \subset C_0(\mathbb{R})$ (the set of continuous functions on $\mathbb{R}$ tending to 0 at infinity). Furthermore, α being band limited is continuous. Also being bounded, the expectation and variance functions of (3.4) are continuous and therefore well defined at any point. It remains to show that these formulas are valid for a general gain function. Consider for this an approximation of $\alpha(t)$ by a finite *numerical flutter shutter* code $(\alpha_k)_k$ such that $(\alpha_k)_k$ tends to α in L^1, L^2, and L^∞ (that is, uniformly) when $k \to \infty$. The formulas (3.4) are valid for $(\alpha_k)_k$, and the corresponding formulas for α are deduced by passing to the limit. ∎

From now on, unless specified otherwise, by *numerical flutter shutter* and by α we shall mean a *continuous numerical flutter shutter.*

3.1. The inverse filter of a numerical flutter shutter.

Step 1: The noiseless case. Let us examine first the discrete noiseless case when $obs(n) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$ and $obs(n)$ is obtained for $n \in \mathbb{Z}$ but being band limited can be interpolated to $obs(x)$ for any $x \in \mathbb{R}$. Then $\mathcal{F}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(\xi) = \hat{u}(\xi)\hat{\alpha}(\xi v)$. By hypothesis (see section 2.1) we assumed that $\hat{u}(\xi) = 0$ for $|\xi| > \pi$. Hence for the invertibility we must only require that $|\hat{\alpha}(\xi v)| > 0$ for $\xi \in [-\pi, \pi]$.

Definition 3.4. *We say that a flutter shutter α is invertible (for velocities $|v|$ smaller than $|v_0|$) if $|\hat{\alpha}(\xi)| > 0$ for $\xi \in [-\pi|v_0|, \pi|v_0|]$.*

If the *flutter shutter* is invertible, we can consider the inverse filter γ defined by

$$\hat{\gamma}(\xi) = \frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}. \tag{3.7}$$

Since $\alpha \in L^1(\mathbb{R})$, $\xi \mapsto \hat{\alpha}(\xi)$ is bounded and continuous. If $\hat{\alpha}(\xi v)$ is nonzero on $[-\pi,\pi]$, $\hat{\gamma}$ will therefore be bounded and supported on $[-\pi,\pi]$. As a consequence, under this assumption, γ is C^∞, bounded, and band limited.

We shall logically define the recovered landscape from noisy data by the formula that would be valid for noiseless data. Assume that we observe $e(n) = \mathbb{E}(obs(n))$ for $n \in \mathbb{Z}$ and wish to compute $\hat{e}(\xi)$ from the discrete observed $(e(n))_{n\in\mathbb{Z}}$. Since $e(x)$ is band limited, we can interpolate it using the band-limited interpolation (2.3). The band-limited interpolate of the ideal observation is

$$e(x) = \sum_{n\in\mathbb{Z}} e(n)\mathrm{sinc}(x-n). \tag{3.8}$$

Then from (3.8) we have

$$\hat{e}(\xi) = \sum_{n\in\mathbb{Z}} e(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{3.9}$$

So the ideal deconvolved landscape $d(x)$ obtained by combining (3.7) and (3.9) is

$$\hat{d}(\xi) = \frac{\sum_{n\in\mathbb{Z}} e(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}. \tag{3.10}$$

Flutter shutter landscape recovery in the real noisy case. We shall now adopt the same formulae for the noisy case.

Definition 3.5. *Assume that a flutter shutter with code α is invertible. We call the estimated landscape $\mathbb{u}_{est,num}$ of the numerical flutter shutter the function defined by*

$$\mathcal{F}(\mathbb{u}_{est,num})(\xi) = \frac{\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}, \tag{3.11}$$

where the observed $obs(n)$ samples (3.1) are used instead of the ideal $e(n)$ in (3.10).

Theorem 3.6. *The numerical flutter shutter has a spectral SNR (2.4) equal to*

$$SNR(\xi) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{|\hat{u}(\xi)||\hat{\alpha}(\xi v)|}{\sqrt{||u||_{L^1}}||\alpha||_{L^2}};$$

the expected value of the estimated landscape $\mathcal{F}(\mathbb{u}_{est,num})(\xi)$ from the observed samples is

$$\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,num})(\xi)) = \hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi), \tag{3.12}$$

and its variance is

$$var(\mathcal{F}(\mathbb{u}_{est,num}(\xi))) = \frac{||\alpha||^2_{L^2}||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}. \tag{3.13}$$

Proof. By (3.4), (3.11),

$$
\begin{aligned}
var(\mathcal{F}(\mathbb{u}_{est,num})(\xi)) &= \frac{var\left(\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{|\hat{\alpha}(\xi v)|^2}\\
&= \frac{\sum_{n\in\mathbb{Z}} var(obs(n))|e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)|^2}{|\hat{\alpha}(\xi v)|^2} = \frac{\sum_{n\in\mathbb{Z}}\left(\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right)*u\right)(n)\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}\\
&= \frac{||\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right)*u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}\\
&= \frac{\frac{1}{v}||\alpha^2\left(\frac{\cdot}{v}\right)||_{L^1}||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} = \frac{\frac{1}{v}||\alpha\left(\frac{\cdot}{v}\right)||^2_{L^2}||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}\\
&= \frac{\frac{v}{v}||\alpha||^2_{L^2}||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} = \frac{||\alpha||^2_{L^2}||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}.
\end{aligned}
\tag{3.14}
$$

In this proof the crucial point is the use of the Poisson summation formula (xxx) in (3.14). Following the same scheme and starting from (3.4), $\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,num}(\xi))(\xi)$ can be computed by using (for the derivation of the third line) the second Poisson formula (xxx) and the fact that u is band limited with $\hat{u}$ supported on $[-\pi,\pi]$:

$$
\begin{aligned}
\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,num}(\xi))) &= \frac{\mathbb{E}\left(\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{\hat{\alpha}(\xi v)} = \frac{\left(\sum_{n\in\mathbb{Z}}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right)*u\right)(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{\hat{\alpha}(\xi v)}\\
&= \frac{\sum_{m\in\mathbb{Z}}\mathcal{F}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right)*u\right)(\xi+2\pi m)\mathbb{1}_{[-\pi,\pi](\xi)}}{\hat{\alpha}(\xi v)} = \frac{\mathcal{F}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right)*u\right)(\xi)\mathbb{1}_{[-\pi,\pi](\xi)}}{\hat{\alpha}(\xi v)}\\
&= \frac{\frac{v}{v}\hat{\alpha}(\xi v)\hat{u}(\xi)\mathbb{1}_{[-\pi,\pi](\xi)}}{\hat{\alpha}(\xi v)} = \hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi).
\end{aligned}
$$

From (3.13) and using the definition of the spectral SNR (2.4), we obtain

$$
SNR^{spectral}(\mathbb{u}_{est,num}(\xi)) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{|\hat{u}(\xi)||\hat{\alpha}(\xi v)|}{\sqrt{||u||_{L^1}}||\alpha||_{L^2}}. \qquad \blacksquare
$$

Remark. From (3.13) we also deduce that $var(\mathcal{F}(\mathbb{u}_{est,num})(\xi))$ is invariant by changing α into $\lambda\alpha$ for $\lambda \neq 0$ (rescaling): as could be expected, the *flutter shutter* code is defined up to a multiplicative constant.

Remark. Going back to the case where α is a discrete *numerical flutter shutter*, we can see a necessary condition on Δt for its invertibility. Indeed, from (3.3) it follows that

$$
\hat{\alpha}(\xi) = \sum_{k=0}^{L-1}\alpha_k\mathcal{F}(\mathbb{1}_{[k\Delta t,(k+1)\Delta t[})(\xi) = \Delta t \,\mathrm{sinc}\left(\frac{\xi\Delta t}{2\pi}\right)e^{\frac{-i\xi\Delta t}{2}}\sum_{k=0}^{L-1}\alpha_k e^{-ik\xi\Delta t}. \tag{3.15}
$$

Notice that this is *not* the DFT (discrete Fourier transform) of the vector α. This means that, in the literature on the *flutter shutter*, the simulations are neglecting the motion blur on the intervals with Δt length and that Δt must satisfy $|v|\Delta t < 2$ to have $\hat{\alpha}(v\xi)$ invertible on the whole support $[-\pi,\pi]$ of $\hat{u}$.

3.2. Flutter shutter design: From continuous to discrete. Even if the above theory deals with continuous and discrete codes as well, in practice any continuous *flutter shutter* code found by some abstract optimization must eventually be realized as a feasible device. Thus it must be replaced by a piecewise constant one on intervals of length Δt. Assume that we have designed a continuous *flutter shutter* function $\beta \in L^2(\mathbb{R})$, invertible for all velocities below $|v|$, which means $\hat{\beta}(\xi v) \neq 0$ for $\xi \in [-\pi, \pi]$. The values of $\hat{\beta}(v\xi)$ outside $[-\pi, \pi]$ do not matter for our scopes, the filter and inverse filter always being applied to band-limited functions. Thus, we can always assume that $\hat{\beta}(v\xi)$ is zero outside $[-\pi, \pi]$. Our goal is to deduce from β a numerical *flutter shutter* code α which coincides with β at velocity v on the spectrum of u. In other words, we want $\hat{\alpha}(v\xi) = \hat{\beta}(v\xi)$ for $\xi \in [-\pi, \pi]$. Under that condition, the observed signal $obs(n) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$ by α or β will be identical. Furthermore, from (3.4) we will have $\mathbb{E}(\widehat{obs}(\xi)) = \hat{\alpha}(v\xi)\hat{u}(\xi) = \hat{\alpha}(v\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)\hat{u}(\xi) = \hat{\beta}(v\xi)\hat{u}(\xi)$, meaning that the expectation of spectrum of the observed signal is unchanged (but not necessarily its variance).

The task is to find an equivalent code function $\alpha(t) = \sum_{k\in\mathbb{Z}} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$, as defined by (3.3), but that is not necessarily compactly supported. Our requirement is that $\hat{\alpha}(\xi v) = \hat{\beta}(\xi v)$ on $[-\pi, \pi]$. By (3.15), a *numerical flutter shutter* has the general form (where we allow for an infinite code $(\alpha_k)_{k\in\mathbb{Z}}$) $\hat{\alpha}(v\xi) = \Delta t \text{sinc}(\frac{v\Delta t\xi}{2\pi}) e^{\frac{-iv\Delta t\xi}{2}} \sum_k \alpha_k e^{-ikv\Delta t\xi}$. We want $\hat{\alpha}(v\xi) = \hat{\beta}(v\xi)$ for $\xi \in [-\pi, \pi]$, which is equivalent to having for $\xi \in [\pi, \pi]$,

$$\Delta t \left(\sum_{k\in\mathbb{Z}} \alpha_k e^{-ik\Delta t v\xi} \right) e^{-i\frac{v\Delta t\xi}{2}} \text{sinc}\left(\frac{v\Delta t\xi}{2\pi}\right) = \hat{\beta}(v\xi),$$

and therefore

$$\frac{\hat{\beta}(v\xi)e^{i\frac{v\Delta t\xi}{2}}}{\Delta t \text{sinc}(\frac{v\Delta t\xi}{2\pi})} \mathbb{1}_{[-\pi,\pi]} = \sum_{k\in\mathbb{Z}} \alpha_k e^{-ikv\Delta t\xi} \mathbb{1}_{[\pi,\pi]}. \tag{3.16}$$

The left-hand member of this equation belongs to $L^2([-\pi, \pi])$, provided the sinc in the denominator does not vanish, which is true if $\Delta t|v| < 2$. The above formula appears to be the Fourier series decomposition of the left-hand member on the Fourier basis on the interval $[\frac{-T}{2}, \frac{T}{2}]$ satisfying $\frac{2\pi}{T} = \Delta t|v|$, which gives $T = \frac{2\pi}{\Delta t|v|}$. Moreover, we assume that $|v|\Delta t \leq 1$. Indeed this supplementary condition is mandatory for the temporal sampling of the left-hand member of (3.16) and to get $\frac{T}{2} > \pi$. Thus, if $|v|\Delta t \leq 1$ $[\frac{-T}{2}, \frac{T}{2}]$ contains $[-\pi, \pi]$, implying that (3.16) is correct and that $(\alpha_k)_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$ are the Fourier series coefficients (provided $|v|\Delta t \leq 1$),

$$\alpha_k = \frac{\Delta t|v|}{2\pi} \int_{-\frac{\pi}{\Delta tv}}^{\frac{\pi}{\Delta tv}} \frac{\hat{\beta}(v\xi)e^{i\frac{v\Delta t\xi}{2}}}{\text{sinc}(\frac{v\Delta t\xi}{2\pi})} \mathbb{1}_{[-\pi,\pi]} e^{ikv\Delta t\xi} d\xi. \tag{3.17}$$

Thus,

$$\begin{aligned}\alpha_k &= \frac{\Delta t|v|}{2\pi}\int_{-\pi}^{\pi}\frac{\hat\beta(v\xi)e^{i\frac{v\Delta t\xi}{2}}}{\mathrm{sinc}(\frac{v\Delta t\xi}{2\pi})}e^{ikv\Delta t\xi}d\xi\\ &= \frac{sign(v)}{2\pi}\int_{-\pi v\Delta t}^{\pi v\Delta t}\frac{\hat\beta(\frac{\xi}{\Delta t})e^{i\frac{\xi}{2}}}{\mathrm{sinc}(\frac{\xi}{2\pi})}e^{ik\xi}d\xi \text{ (where } sign(x)=1 \text{ if } x\geq 0\text{, and 0 otherwise)}\\ &= \frac{1}{2\pi}\int_{-\pi|v|\Delta t}^{\pi|v|\Delta t}\frac{\hat\beta(\frac{\xi}{\Delta t})e^{i\frac{\xi}{2}}}{\mathrm{sinc}(\frac{\xi}{2\pi})}e^{ik\xi}d\xi.\end{aligned}$$

This proves the following theorem.

Theorem 3.7. *Let $\beta \in L^2(\mathbb{R})$ be a band-limited time convolution kernel satisfying $\hat\beta(v\xi)\neq 0$ for $\xi\in[-\pi,\pi]$, i.e., invertible on all band-limited functions and for all velocities below $|v|$. If $|v|\Delta t\leq 1$, there exists an invertible flutter shutter code function*

$$\alpha(t)=\sum_{k\in\mathbb{Z}}\alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t) \tag{3.18}$$

with $(\alpha_k)_{k\in\mathbb{Z}}\in l^2(\mathbb{Z})$ such that $\hat\alpha(v\xi)=\hat\beta(v\xi)$ on $[-\pi,\pi]$. The coefficients α_k of the discrete numerical flutter shutter are explicitly given by

$$\alpha_k=\frac{1}{2\pi}\int_{-\pi|v|\Delta t}^{\pi|v|\Delta t}\frac{\hat\beta(\frac{\xi}{\Delta t})e^{i\frac{\xi}{2}}}{\mathrm{sinc}(\frac{\xi}{2\pi})}e^{ik\xi}d\xi. \tag{3.19}$$

The question arises of whether the discrete *numerical flutter shutter* α function yields a $PSNR$ (peak signal to noise ratio) as good as the original β. According to the formula giving the SNR in Theorem 3.6, we simply have to compare $||\alpha||_{L^2}$ and $||\beta||_{L^2}$. More precisely, the ratio $\frac{||\beta||_{L^2}}{||\alpha||_{L^2}}$ gives the multiplication factor of the SNR obtained with β to get the SNR of the restored image using the discrete filter α. But by assumption, we have $\hat\beta$ supported on $[-\pi|v|,\pi|v|]$ $||\beta||^2_{L^2(\mathbb{R})}=\frac{1}{2\pi}||\hat\beta||^2_{L^2(\mathbb{R})}=\int_{-\pi v}^{\pi v}|\hat\beta(\xi)|^2d\xi$. On the other hand, by construction, $\hat\alpha=\hat\beta$ on $[-\pi v,\pi v]$. It follows that $||\alpha^2||_{L^2(\mathbb{R})}\geq||\beta^2||_{L^2(\mathbb{R})}$.

Thus, we have also proved the following corollary.

Corollary 3.8. *Let β be a continuous numerical flutter shutter. Then its discrete equivalent numerical flutter shutter has a smaller spectral SNR.*

4. The analog flutter shutter. There are two different acquisition tools implementing a *flutter shutter* with a moving sensor (or landscape). The first one has been discussed previously and consists of a mere computational device using the maximal sensor capability. In that case $obs(x)$ is given by (3.1) and is not a Poisson random variable in general. The other technical possibility is to implement the *flutter shutter function* on the sensor as an optical (temporally changing) filter. This setup, which corresponds to the technology proposed by the inventors of the *flutter shutter*, will be called the *analog flutter shutter*. The Agrawal et al. *flutter shutter* method consists of a (binary) temporal mask in front of the sensor. From a practical point of view the shutter of the camera opens and closes during the acquisition process. The proposed generalization uses temporal sunglasses allowing smoother (nonbinary, non–piecewise constant) gain modifications. The gain at time t $\alpha(t)$ is here defined as the proportion of photons coming from the noisy landscape u that are allowed to travel to the pixel sensor, meaning that only *positive* (actually in $[0,1]$) kernels are feasible. The device (roughly speaking a generalized shutter) controlling the percentage of photons from the landscape allowed to travel to the sensor obviously takes place before the sensor. From a practical point of view it is realizable by implementing the filters directly on the stages of a *time delay integration* (TDI) device. Hence the observation *is always* a Poisson random variable. *The analog flutter shutter method consists of the design of an invertible flutter shutter function* $\alpha(t)$.

Definition 4.1 (*analog flutter shutter* function). *Let* $\alpha(t) \in [0,1]$ *be the gain used at time* t. *We call an analog flutter shutter function any* positive *function* $\alpha \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$.

Let α be an *(analog) flutter shutter function*; then the acquired image at position n is (a realization of) $obs(n) \sim \mathcal{P}(\int_{-\infty}^{\infty} \alpha(t)u(n-vt)dt) \sim \mathcal{P}\left(\frac{1}{v}(\alpha(\frac{\cdot}{v}) * u)(n)\right)$, where $obs(n)$ is known only for $n \in \mathbb{Z}$.

Definition 4.2. *Let* α *be an analog flutter shutter function. We call analog flutter shutter samples at position* n *of the landscape* u *at velocity* v *the random variable*

$$obs(n) \sim \mathcal{P}\left(\frac{1}{v}\left(\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)\right). \tag{4.1}$$

We call an analog flutter shutter its band limited interpolate $obs(x) \sim \sum_{n\in\mathbb{Z}} obs(n)\mathrm{sinc}(x-n)$.

Theorem 4.3. *The observed samples of the analog flutter shutter are such that, for* $n \in \mathbb{Z}$,

$$\mathbb{E}(obs(x)) = \frac{1}{v}\left(\alpha\left(\frac{\cdot}{v}\right) * u\right)(x) \quad and \quad var(obs(x)) = \frac{1}{v}\left(\alpha\left(\frac{\cdot}{v}\right) * u\right)(x). \tag{4.2}$$

Proof. This comes directly from the *analog flutter shutter* samples definition (4.1). ∎

The main difference with the *numerical flutter shutter* is that the observed image is always a Poisson random variable. The calculations on the *analog flutter shutter* are almost identical to those of the *numerical flutter shutter*.

Theorem 4.4. *The analog flutter shutter method has a spectral* SNR *equal to*

$$SNR^{spectral}(\mathcal{F}(\mathbb{u}_{est,ana})(\xi)) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{|\hat{u}(\xi)||\hat{\alpha}(\xi v)|}{\sqrt{||u||_{L^1}||\alpha||_{L^1}}};$$

the expected value of the estimated landscape $\mathcal{F}(\mathbb{u}_{est,ana})(\xi)$ *from the observed samples is*

$$\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,ana})(\xi))) = \hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi),$$

and the variance is

$$var(\mathcal{F}(\mathbb{u}_{est,ana}(\xi))) = \frac{||\alpha||_{L^1}||u||_{L^1}}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{4.3}$$

Proof. Similarly to the *numerical flutter shutter* the inverse filter is the inverse filter defined by $\frac{1}{\hat{\alpha}(v\xi)}$; then

$$
\begin{aligned}
var\left(\mathcal{F}(\mathbb{u}_{est,ana})(\xi)\right) &= var\left(\frac{\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}}{\hat{\alpha}(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi)\right) = \frac{\sum_{n\in\mathbb{Z}} \frac{1}{v}(\alpha\left(\frac{\cdot}{v}\right) * u)(n)}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi) \\
&= \frac{\frac{1}{v}||\alpha\left(\frac{\cdot}{v}\right) * u)||_{L^1}}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi) \qquad \text{(by (xxx))} \\
&= \frac{\frac{1}{v}||\alpha\left(\frac{\cdot}{v}\right)||_{L^1}||u||_{L^1}}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi) = \frac{||\alpha||_{L^1}||u||_{L^1}}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi).
\end{aligned}
$$

Moreover, by the same calculations as for the *numerical flutter shutter*,

$$
\mathbb{E}\left(\mathcal{F}(\mathbb{u}_{est,ana})(\xi)\right) = \left(\frac{\mathbb{E}\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}}{\hat{\alpha}(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi)\right) = \hat{u}(\xi).
$$

Therefore, $SNR^{spectral}(\mathcal{F}(\mathbb{u}_{est,ana}))(\xi) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{|\hat{u}(\xi)||\hat{\alpha}(\xi v)|}{\sqrt{||u||_{L^1}||\alpha||_{L^1}}}$. ■

A brief summary pointing out the differences between the *analog flutter shutter* and the *flutter shutter* is given in Table 1. The *analog flutter shutter* controls the percentage of photons allowed to travel to the sensor, and therefore only positive functions are implementable. It decreases the number of sensed photons and therefore tends to decrease the resulting SNR. On the other hand, the *numerical flutter shutter* requires piecewise constant *flutter shutter* functions. Consequently, if a *flutter shutter* function is positive and piecewise constant (implementable on both cameras), the numerical *flutter shutter* should always be chosen, as it leads to a better SNR of the reconstructed image. The question of choice of the *flutter shutter* type in the general case is answered in section 5.

5. Comparison of a piecewise constant analog flutter shutter with the numerical flutter shutter and snapshot optimization. The question arises of whether it is better to apply an *analog flutter shutter* or the equivalent *numerical flutter shutter* with exactly the same code $0 \leq \alpha \leq 1$. (From the technological viewpoint, an *analog flutter shutter* could be easily implemented with a classic CCD and a numerical one with a CMOS).

First, we observe that the variance of the *analog flutter shutter* observation (4.2) is larger than or equal to the variance of the *numerical flutter shutter* observation (3.4), with equality when $\forall k\ \alpha_k = 0$ or 1.

Indeed, since $0 \leq \alpha(t) \leq 1$ (because it is a proportion of incoming photons allowed to travel through the sensor), we have $\alpha(\frac{t}{v}) \geq \alpha^2(\frac{t}{v})$. Hence $(\alpha\left(\frac{\cdot}{v}\right) * u)(x) \geq (\alpha^2\left(\frac{\cdot}{v}\right) * u)(x)$ (because $u \geq 0$). Using (4.2) and (3.4) concludes the proof.

Moreover, the expected value of (4.2) is equal to the expected value of the *numerical flutter shutter* (see (3.4)) and the inverse filter is equal to the inverse filter of the *numerical flutter shutter* (3.7). The next result is a decider for the *numerical flutter shutter* (when it is possible to implement it with the same code as an *analog flutter shutter*, meaning that α is piecewise constant).

Let $0 \leq \alpha \leq 1$ be a piecewise constant code function for the *analog flutter shutter*. Then the spectral SNR of the *analog flutter shutter* method is smaller than or equal to the spectral SNR of the *numerical flutter shutter* with the same code.

Table 1

This table summarizes the results on numerical flutter and analog flutter shutters. The first column describes the structure of the numerical flutter shutter, and the second describes the analog flutter shutter. The first line indicates which kind of flutter shutter functions $\alpha(t)$ are implementable with respect to the flutter shutter type. The second (resp., the third) gives the expected value (resp., variance) of the (observed) flutter shutter. The fourth shows the inverse filter to be applied to the flutter shutter in order to deconvolve. The fifth (resp., the sixth) gives the expected value (resp., variance) of the deconvolved. Given any flutter shutter function $\alpha(t)$ the last one gives the spectral SNR of both methods. Provided that a flutter shutter function $\alpha(t)$ is implementable on both kinds of flutter shutters the spectral SNR of the analog flutter shutter is lower than the spectral SNR of the numerical flutter shutter (see the previous page).

Type of *flutter shutter*	*Numerical flutter shutter*	*Analog flutter shutter*
Flutter shutter function $\alpha(t)$	$\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$ (with $\alpha_k \in \mathbb{R}$ and $\Delta t > 0$)	$\alpha(t) \in [0,1]$
$\mathbb{E}\,(obs(n))$ (observed)	$\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$	$\frac{1}{v}(\alpha\left(\frac{\cdot}{v}\right) * u)(n)$
$var(obs(n))$ (observed)	$\left(\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right) * u\right)(n)$	$\frac{1}{v}(\alpha\left(\frac{\cdot}{v}\right) * u)(n)$
Inverse filter $\hat{\gamma}(\xi)$	$\frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}$	$\frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}$
$\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,num})(\xi)))$ (deconvolved)	$\hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)$	$\hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)$
$var(\mathcal{F}(\mathbb{u}_{est,num}(\xi)))$ (deconvolved)	$\frac{\vert\vert\alpha\vert\vert_{L^2}^2 \vert\vert u\vert\vert_{L^1}}{\vert\hat{\alpha}(\xi v)\vert^2}\mathbb{1}_{[-\pi,\pi]}(\xi)$	$\frac{\vert\vert\alpha\vert\vert_{L^1}\vert\vert u\vert\vert_{L^1}}{\vert\hat{\alpha}\vert^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi)$
(spectral) $SNR(\xi)$	$\frac{\vert\hat{u}(\xi)\vert\vert\hat{\alpha}(\xi v)\vert}{\sqrt{\vert\vert u\vert\vert_{L^1}}\vert\vert\alpha\vert\vert_{L^2}}\mathbb{1}_{[-\pi,\pi]}(\xi)$	$\frac{\vert\hat{u}(\xi)\vert\vert\hat{\alpha}(\xi v)\vert}{\sqrt{\vert\vert u\vert\vert_{L^1}\vert\vert\alpha\vert\vert_{L^1}}}\mathbb{1}_{[-\pi,\pi]}(\xi)$

The *analog flutter shutter* method has a spectral SNR equal to

$$SNR(\mathcal{F}(\mathbb{u}_{est,ana})(\xi)) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{|\hat{u}(\xi)||\hat{\alpha}(\xi v)|}{\sqrt{||u||_{L^1}||\alpha||_{L^1}}},$$

and the spectral SNR of the *numerical flutter shutter* is

$$SNR(\mathcal{F}(\mathbb{u}_{est,num})(\xi)) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{|\hat{u}(\xi)||\hat{\alpha}(\xi v)|}{\sqrt{||u||_{L^1}}||\alpha||_{L^2}}.$$

A comparison of both formulas shows that the announced inequality amounts to proving that $\sqrt{\int\alpha} \geq \sqrt{\int\alpha^2}$, which boils down to $\int\alpha \geq \int\alpha^2$. This last inequality follows immediately from $0 \leq \alpha \leq 1$.

This result also implies that the variance of the estimated landscape (4.3) using an *analog flutter shutter* method $var(\mathcal{F}(\mathbb{u}_{est,ana}))$ is larger than or equal to $var(\mathcal{F}(\mathbb{u}_{est,num}))$ (3.13) using a *numerical flutter shutter* method when α is positive and piecewise constant.

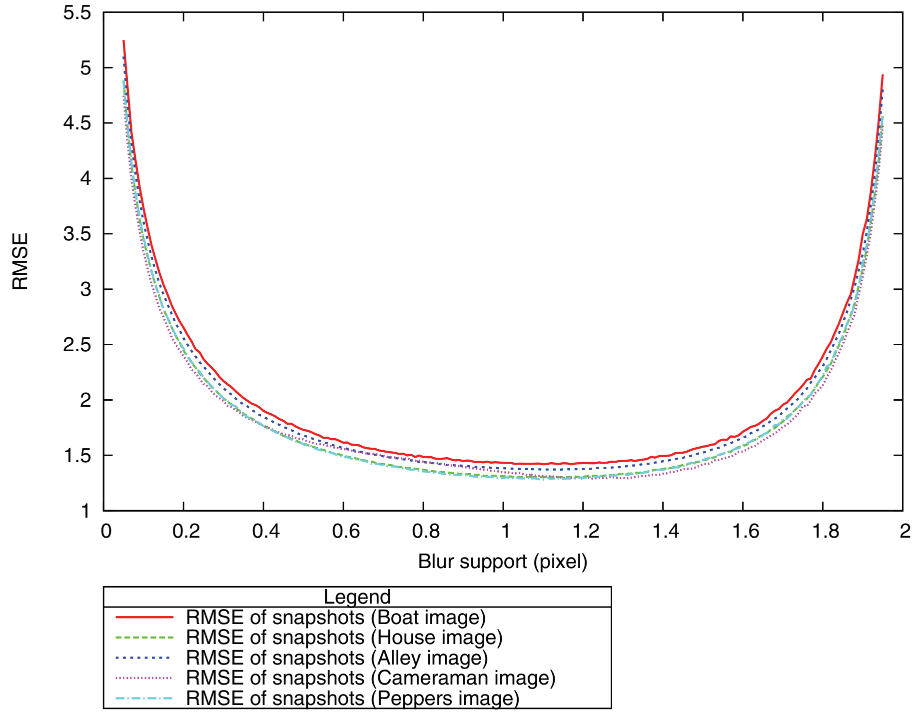


Figure 2. *This figure shows the $RMSE$ curves for different snapshot kinds on five test images (House, Alley, Boat, Cameraman, Peppers). On the x-axis, the blur support ($|v|\Delta t$) is in pixels; on the y-axis is the corresponding $RMSE$. Notice that some of these curves are so close to each other that they superimpose. W.l.o.g., by the arguments developed in the proof of Theorem 5.1, the comparison is made with a fixed $v = 1$. From the proof of Theorem 5.1 we get that the best snapshot depends only on the blur support $|v|\Delta t$. However, from (5.1) it can be deduced that for a fixed landscape u the value of this SNR is a function of the two variables Δt and $v\Delta t$. Furthermore, since our estimator is unbiased, by Proposition 2.4 we get that minimizing the variance is equivalent to minimizing the $RMSE$ of the deconvolved image. Therefore, the curves confirm that, on average on multiple images, the blur support for a standard camera should be of approximatively $\Delta t^* = 1.0909$ pixels. Moreover, from Theorem 5.1 we get that for a fixed v and Δt the value of the $RMSE$ depends on the landscape u, explaining the differences between curves. Theses curve also show that our SNR definition is indeed proportional to the $RMSE$ of the deconvolved image. A larger support would lead to a better SNR on the observed image samples, but the deconvolution would entail a lower SNR (and a bigger $RMSE$) on the deconvolved image. The best snapshot is a compromise between the number of photons caught during a time span Δt and the deconvolution kernel. It gives a reference to compare all flutter shutter strategies in terms of SNR.*

The above results shall be used in what follows to compare the *flutter shutter* with classic cameras and provide a thorough definition and analysis of the classic flutterless photography. Uniform motion blurs using a standard camera have been studied nicely, for example, in [12]. A figure providing the $RMSE$ of snapshot varying the exposure time Δt is given in Figure 2.

The acquired image at position x for a short snapshot is (a realization of)

$$\mathbf{P}_l\left([0,\Delta t]\times\left[x-\frac{1}{2},x+\frac{1}{2}\right]\right)\sim\mathcal{P}\left(\frac{1}{v}(\mathbb{1}_{[0,v\Delta t]}*u)(x)\right)\sim obs(x), \tag{5.1}$$

where (5.1) is known only for $x\in\mathbb{Z}$. In short, a snapshot is nothing but a *flutter shutter* (analog or numerical) with code $\alpha(t) = \mathbb{1}_{[0,\Delta t](t)}$. Thus

$$\mathcal{F}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right)\right) = 2\frac{\sin(\frac{\xi v \Delta t}{2})}{v\xi}e^{-i\xi\frac{v\Delta t}{2}}. \tag{5.2}$$

From (5.2) we see that we *must* have $|v|\Delta t < 2$ to guarantee the invertibility of the blur kernel on $[-\pi, \pi]$. Similarly to the *flutter shutter* formalism developed in section 3, we call a "standard snapshot" the use of an integration time (Δt) such that $|v|\Delta t < 2$. We call *snapshot samples* at position n of the landscape u at velocity v the random variables $obs(n) \sim \mathcal{P}\left(\frac{1}{v}(\mathbb{1}_{[0,v\Delta t]} * u)(n)dt\right)$. We call a *band-limited interpolated snapshot* its band-limited interpolate (2.3) $obs(x) \sim \sum_{n\in\mathbb{Z}} obs(n)\text{sinc}(x-n)$. By definition of the standard snapshot, Δt is small enough so (5.2) has no zero on $[-\pi, \pi]$ (the support of $\hat{u}$). Thus (5.1), (5.2) lead to the definition of the inverse filter γ satisfying $\hat{u}(\xi) = \hat{\gamma}(\xi)\mathcal{F}(\mathbb{E}(obs))(\xi)$, implying

$$\hat{\gamma}(\xi) = \frac{v\mathbb{1}_{[-\pi,\pi]}(\xi)}{2\frac{\sin(\frac{\xi v\Delta t}{2})}{\xi}e^{-i\xi\frac{v\Delta t}{2}}}. \tag{5.3}$$

We call the estimated landscape $\mathbb{u}_{est,sna}$ of the standard snapshot the function defined by

$$\mathcal{F}(\mathbb{u}_{est,sna})(\xi) = \hat{\gamma}(\xi)\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{5.4}$$

The following estimation of variance and SNR are direct applications of the same quantities for the numerical (or analog) *flutter shutter*. Moreover, the spectral SNR (2.4) of the standard snapshot using an exposure time of Δt is

$$SNR(\xi) = \mathbb{1}_{[-\pi,\pi]}(\xi)|\hat{u}(\xi)|\sqrt{\frac{\Delta t}{||u||_{L^1}}}\left|2\frac{\sin(\frac{\xi v\Delta t}{2})}{\xi v\Delta t}\right|.$$

Furthermore, the expected value of the estimated landscape $\mathcal{F}(\mathbb{u}_{est,sna})(\xi)$ from the observed samples is

$$\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,sna})(\xi)) = \hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi), \tag{5.5}$$

and the variance is

$$var(\mathcal{F}(\mathbb{u}_{est,sna}(\xi))) = \frac{||u||_{L^1}\mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t\left|2\frac{\sin(\frac{\xi v\Delta t}{2})}{\xi v\Delta t}\right|^2}. \tag{5.6}$$

Indeed, since $\alpha(t) = \mathbb{1}_{[0,\Delta t]}(t)$, $\|u\|_{L^2}^2 = \Delta t$, and $\hat{\alpha}(\xi) = \frac{2\sin(\Delta t\xi)}{\xi}e^{-i\frac{\Delta t\xi}{2}}$, these formulas are direct applications of Theorem 3.6.

The only remaining, but crucial, task is the computation of the best exposure time Δt for a known v providing the best SNR without the use of a *flutter shutter*.

Theorem 5.1 (best exposure time for landscape recovery). *Consider a landscape $u(x - vt)$ moving at velocity v. Then for a snapshot the $SNR^{spectral-averaged}$ (2.5) is maximized when $|v|\Delta t^* \approx 1.0909$" and is equal to*

$$SNR^{spectral-averaged} = \frac{\sqrt{\frac{\Delta t^*}{2\pi}} \int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{||u||_{L^1} \int_{-\pi}^{\pi} \frac{d\xi}{\left| \frac{\sin(\frac{\xi v \Delta t^*}{2})}{\frac{\xi v \Delta t^*}{2}} \right|^2}}} \approx \frac{0.1359}{\sqrt{|v|}} \frac{\int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{||u||_{L^1}}}.$$

Proof. From (5.6) the energy (variance of the noise) to be minimized in order to guarantee the best $SNR^{spectral-averaged}$ *after* deconvolution is

$$E(\Delta t) = \frac{1}{\Delta t} \int_{-\pi}^{\pi} \frac{d\xi}{\left| 2 \frac{\sin(\xi \frac{v\Delta t}{2})}{\xi v \Delta t} \right|^2} d\xi = \frac{v^2 \Delta t}{4} \int_{-\pi}^{\pi} \frac{\xi^2}{\sin^2(\xi \frac{v\Delta t}{2})} d\xi.$$

Then its derivative vanishes when $b^* = v\Delta t^* \approx 1.0909$ (see Figure 3). Then (2.5), (5.5), (5.6) entail

$$SNR^{spectral-averaged} = \frac{\sqrt{\frac{1}{2\pi}} \int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\frac{||u||_{L^1}}{\Delta t^*} \int_{-\pi}^{\pi} \frac{d\xi}{\left| \frac{\sin(\frac{\xi v \Delta t^*}{2})}{\frac{\xi v \Delta t^*}{2}} \right|^2}}} \approx \frac{0.1359}{\sqrt{v}} \frac{\int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{||u||_{L^1}}}. \qquad \blacksquare$$

This means that, using a standard camera, the best $SNR^{spectral-averaged}$ of the recovered image is achieved by a finite blur whose support is of approximatively ≈ 1.0909 pixels. The use of a bigger exposure time can give a better SNR *before* deconvolution, but this advantage is lost by the deconvolution. The previous also applies to "time delay and integration" devices (commonly used in satellites as a SNR booster) where the number of stages defines the time exposure.

6. Motion-invariant photography.

We shall prove that the *motion-invariant photography* method proposed in [53, 54] is equivalent to an *analog flutter shutter* method using a specific *flutter shutter function*. Thus, we are able to compute its SNR and compare it with the other *flutter shutter* methods. This fact comes as a surprise, as the gain in a *flutter shutter* is controlled, while with the *motion-invariant photography* method the shutter remains fully open during the whole aperture time. Thus, its gain remains constant and equal to one on the normal time scale. Nevertheless, a time renormalization gives a variable gain. The *motion-invariant photography* apparatus consists in *moving the camera at a constant acceleration in the direction of v while the landscape is moving at a constant velocity v*. Thus, to use the *motion-invariant photography* method the direction of v must be a priori known. Furthermore, this means that the apparent relative velocity (between the camera and the landscape) is $v(t) = -at - v$. The *motion-invariant photography* was discovered by searching among all camera motions for one providing the same kernel for all velocities v of the landscape. With the formalism proposed in the former sections the observed value of the *motion-invariant*

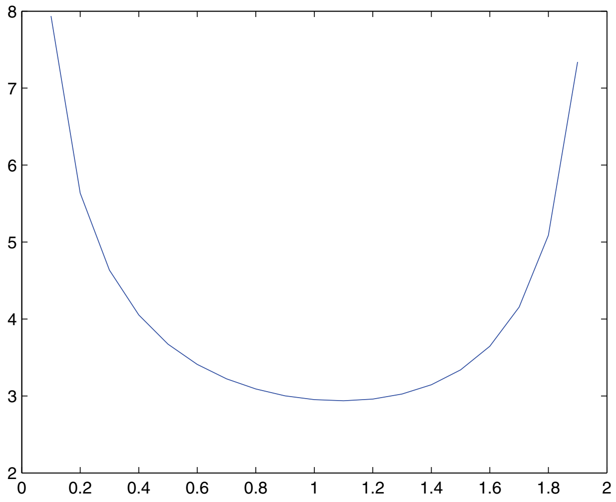


Figure 3. *This figure shows the square root of the energy $E(\Delta t)$ of Theorem 5.1. The energy $E(\Delta t)$ measures the variance of the Poisson noise* after *deconvolution, for* any *snapshot, in function of the exposure time Δt. Since our estimator is unbiased and by using Proposition 2.4 we get that minimizing the variance is equivalent to minimizing the RMSE of the deconvolved image. Therefore, we ought to minimize $E(\Delta t)$ in order to guarantee the best $SNR^{spectral-averaged}$ and the smallest RMSE taking the deconvolution into account: x-axis blur ($|v|\Delta t$) in pixels; y-axis the standard deviation of the noise taking the deconvolution into consideration. The minimum is reached for a blur of approximately* 1.0909 *pixels. W.l.o.g., by the arguments developed in the proof of Theorem 5.1, the curve has been drawn for $v = 1$.*

photography using a finite aperture time T on the centered time interval $[\frac{-T}{2}, \frac{T}{2}]$ is a realization of

$$obs(x) \sim \mathcal{P}\left(\int_{\frac{-T}{2}}^{\frac{T}{2}} u\left(x - \frac{a}{2}t^2 - v.t\right) dt\right). \tag{6.1}$$

Then by a nonlinear time scale we have

$$obs(x) \sim \mathcal{P}\left(\int_{-\infty}^{\infty} \left(\underbrace{\underbrace{\frac{\mathbb{1}_{]a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a},a(\frac{T}{2}+|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{2\sqrt{a(t+\frac{v^2}{2a})}}}_{A} + \underbrace{\frac{\mathbb{1}_{]-\frac{v^2}{2a},a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}}_{B}}_{\alpha_{MIP}(t)} \right) u(x-t)dt \right). \tag{6.2}$$

The denominator of B is arbitrarily close to 0. Thus, $\alpha_{MIP}(t)$ can become larger than one. This means that, in general, it could not be realized *stricto sensu* by an *analog flutter shutter*, where the relative motion v of landscape and camera would be uniform. Fortunately, the above formula shows that the *motion-invariant photography* apparatus is mathematically equivalent to an *analog flutter shutter* and can be analyzed in terms of SNR like any other *flutter shutter*. It is not a *numerical flutter shutter* since the *flutter shutter function* modifies the intensity of the Poisson random variables directly. This means that the observed samples of the *motion-invariant photography* are always Poisson random variables. The claim raised in [53, 54] that the method is motion invariant comes from the fact that the kernel depends only on $|\frac{v}{a}|$. Thus, if $|a|$ is large enough, the relative variations of $|\frac{v}{a}|$ toward v are small. Under that assumption the kernel $\alpha_{MIP}(t)$ is indeed nearly invariant with respect to the velocity v. Notice that when $T \to +\infty$, the "A" part of $\alpha_{MIP}(t)$ tends to 0 since $a(\frac{T}{2} - |\frac{v}{a}|)^2 \to sign(a)\infty$.

Theorem 6.1. *The motion-invariant photography using a finite aperture time T is equivalent to an analog flutter shutter with a flutter function equal to*

$$\alpha_{MIP}(t) = \frac{\mathbb{1}_{]a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a},a(\frac{T}{2}+|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{2\sqrt{a(t+\frac{v^2}{2a})}} + \frac{\mathbb{1}_{]-\frac{v^2}{2a},a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}.$$

Proof. This is a direct consequence of (6.1)–(6.2) and section 4. ∎

The question arises of whether or not the kernel

$$\alpha_{MIP}(t) = \frac{\mathbb{1}_{]a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a},a(\frac{T}{2}+|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{2\sqrt{a(t+\frac{v^2}{2a})}} + \frac{\mathbb{1}_{]-\frac{v^2}{2a},a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}$$

is indeed invertible for all band-limited functions whose FT lies on $[-\pi, \pi]$. This finite aperture scheme is a technically feasible approximation of the ideal *motion-invariant photography* using an infinite aperture time with an accelerating camera. Let the aperture time $T \to +\infty$; then, provided $a > 0$,

$$\alpha_{MIP}(t) \to \frac{\mathbb{1}_{]-\frac{v^2}{2a},+\infty[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}$$

in L^1_{loc} and in the tempered distribution sense. Thus, skipping the time translation, we get the ideal *motion-invariant photography* "flutter" function $\alpha_{MIP-ideal}(t) := \frac{\mathbb{1}_{]0,+\infty[}(t)}{\sqrt{at}}$. Notice that when $a < 0$,

$$\alpha_{MIP}(t) \to \frac{\mathbb{1}_{]-\frac{v^2}{2a},-\infty[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}},$$

whose FT is $\hat{\alpha}_{MIP-ideal}(-\xi)$. Thus *asymptotically* the choice of the direction of the acceleration has no influence on the invertibility of the *motion-invariant photography*.

Lemma 6.2 (invertibility of the *motion-invariant photography* method). *Using a large enough aperture time the motion-invariant photography kernel is invertible, whatever the sign of a.*

Proof. Indeed, when $T \to +\infty$, $\alpha_{MIP} \to \alpha_{MIP-ideal} = \frac{\mathbb{1}_{]0,+\infty[}(t)}{\sqrt{at}}$, where $\hat{\alpha}_{MIP-ideal} = \frac{1}{\sqrt{|a\xi|}}e^{-i\frac{\pi}{4}sign(\xi)}$, which does not depend on the sign of a. These calculations are valid up to an irrelevant multiplicative constant factor for the *numerical flutter shutter*, dropping also the time translation. Thus the convergence of α_{MIP} to $\alpha_{MIP-ideal}$ is true in the tempered distribution sense. It follows that also $\hat{\alpha}_{MIP}$ tends to $\hat{\alpha}_{MIP-ideal}$ in the tempered distribution sense, and the limit indeed does not vanish. ■

Lemma 6.3 (efficiency of the ideal *motion-invariant photography* method). *When $T \to +\infty$ the ideal motion-invariant photography method has a spectral SNR*

$$SNR^{spectral}(\xi) = \begin{cases} \frac{|\hat{u}(0)|}{\sqrt{||u||_{L1}}}\infty & \text{at } \xi = 0, \\ 0 & \text{elsewhere.} \end{cases}$$

In consequence, the average SNR is zero: $SNR^{spectral-averaged} = 0$.

Proof. We have, when $T \to \infty$, $\alpha_{MIP} \to \alpha_{MIP-ideal}$, and thus at $\xi = 0$ $\hat{\alpha}_{MIP-ideal}(0) = ||\alpha_{MIP-ideal}||_{L^1}$ since $\alpha_{MIP-ideal}(t) \geq 0$, which proves, by Theorem 4.4, that $SNR^{spectral}(0) = \lim_{x\to\infty} \frac{x}{\sqrt{x}} \frac{|\hat{u}(0)|}{\sqrt{||u||_{L1}}} = \frac{|\hat{u}(0)|}{\sqrt{||u||_{L1}}}\infty$. Now let $\xi \neq 0$. Then

$$\begin{aligned} var(\mathcal{F}(\mathbb{u}_{est,ana}(\xi))) &= \frac{||\alpha_{MIP-ideal}||_{L^1}||u||_{L^1}}{\frac{1}{|a\xi|}} \mathbb{1}_{[-\pi,\pi]}(\xi) \\ &= |a\xi|||\alpha_{MIP-ideal}||_{L^1}||u||_{L^1}\mathbb{1}_{[-\pi,\pi]}(\xi) = \infty \end{aligned}$$

(since $||\alpha_{MIP-ideal}||_{L^1} = \infty$ and $|\hat{\alpha}_{MIP-ideal}(\xi)| < \infty$). This entails, again by Theorem 4.4, that $SNR^{spectral}(\xi) = 0$. The last result comes from the fact that the variance is infinite on a set $[-\pi, \pi] \setminus \{0\}$, and thus $SNR^{spectral-averaged} = 0$. ■

Lemma 6.3 means that the *motion-invariant photography* behaves like a standard camera using an infinite time exposure: only the null frequency is preserved. Indeed an invertible kernel does not guarantee a good SNR *after* deconvolution (except on the unreal case where the acquired samples are noiseless). A convolution against a kernel $\alpha(t)$ having a small but nonzero $|\hat{\alpha}(\xi)|$ on the support of $\hat{u}(\xi)$ (for example, a Gaussian with a large standard deviation) would lead to the same result. The *motion-invariant photography* is therefore a perfect example of the *flutter shutter* paradox (section 7). To sense many more photons does not necessarily imply a better SNR after deconvolution. The authors of [53] wrote as a drawback of the *flutter shutter* that it was losing half the photons while the *motion-invariant photography* kept them all: "(about the Agrawal et al. *flutter shutter*) ... the amount of recorded light is halved. Because of the loss of light, the vertical budget is reduced from $2T$ to T for each ω_x." The number of acquired photons can be arbitrarily large using a *flutter shutter* or a *motion-invariant photography* apparatus. Nevertheless, the SNR of the image obtained *after* deconvolution is lower than the SNR (see Table 2) of the best snapshot acquiring few photons (comparatively), despite the fact that the snapshot "spends energy outside the slope wedge and thus does not make a full usage of the vertical $\hat{k}_{\omega_x}$ budget" [53]. We now turn to practical aspects of the *motion-invariant photography*. For obvious practical reasons it is not possible to accelerate the camera infinitely. Thus $\hat{\alpha}_{MIP}$, using a finite aperture time, is nonetheless an approximation of $\hat{\alpha}_{MIP-ideal}$. It may seem surprising, at first sight, that the finite aperture approximation has a better SNR than the ideal one. This comes from the fact that, for a finite time aperture, α_{MIP} belongs to $L^1(\mathbb{R})$. Its FT may have zeros, but, for finite and large enough T's, they are *outside* $[-\pi, \pi]$, the support of $\hat{u}$. This fact is illustrated in Table 3, where the value $SNR^{spectral-averaged}$ is given for a variety of choices for a and T. To

Table 2

This table provides the relative $SNR^{spectral-averaged}$ of all standard flutter shutter strategies compared to the best snapshot. A number greater than one means an increase of the SNR and less than one a loss.

Flutter shutter strategy	$SNR^{spectral-averaged}$
Best snapshot	1
Agrawal et al. *flutter shutter* (code) ($v = 1$ $\Delta t = 1$)	0.5636
Ideal motion-invariant photography (infinite time exposure)	0
Motion-invariant photography (at $\lvert\frac{v}{a}\rvert = 1$ and $T = 1$)	0.6233
Ideal flutter shutter (sinc) (infinite time exposure)	1.17

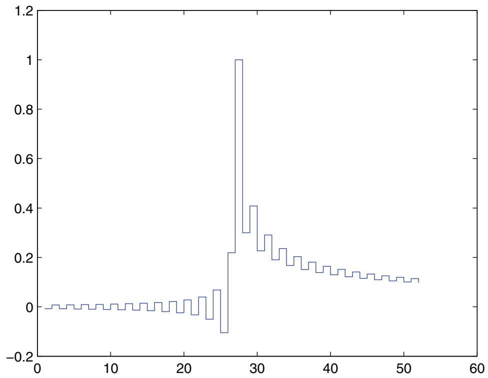
Table 3

This table provides the relative $SNR^{spectral-averaged}$ compared to the best snapshot. A number greater than one means an increase of the SNR and less than one a loss. This fact illustrates the asymptotic result on the motion-invariant photography (Lemma 6.3): the bigger T is, the worse the results become (the noisier the deconvolved is).

	$\lvert\frac{v}{a}\rvert = 1$	$\lvert\frac{v}{a}\rvert = 10^{-1}$	$\lvert\frac{v}{a}\rvert = 10^{-2}$	$\lvert\frac{v}{a}\rvert = 10^{-3}$	$\lvert\frac{v}{a}\rvert = 10^{-4}$
$T = 1$	0.6233	0.4538	0.1743	0.1451	0.0550
$T = 10$	0.0812	0.1080	0.0338	0.0157	0.0017
$T = 100$	$6.8270.10^{-2}$	$8.9420.10^{-3}$	$3.9406.10^{-4}$	$2.9002.10^{-4}$	$4.7470.10^{-6}$
$T = 1000$	$4.4610.10^{-3}$	$6.0796.10^{-4}$	$4.6485.10^{-5}$	$6.9466.10^{-6}$	$1.3826.10^{-6}$
$T = 10000$	$1.7162.10^{-4}$	$3.9338.10^{-6}$	$7.3618.10^{-8}$	$1.3089.10^{-9}$	$2.4434.10^{-11}$

compare, on an equal footing, all *flutter shutters* and the *motion-invariant photography* we propose finding a piecewise constant *flutter shutter code* approximating $\alpha_{MIP-ideal}$ using the framework of Theorem 3.7. This permits us to override a drawback of the method. Indeed, a *flutter shutter* implementation will work *without* the a priori knowledge of the direction of the velocity v of the landscape. It is a bit clumsy to directly approximate α_{MIP} since it already is an approximation of the ideal $\alpha_{MIP-ideal}$ *motion-invariant photography* function and would result inevitably in a lower SNR (and therefore an unfair comparison). To do so, we remark that $\hat{\alpha}_{MIP-ideal}$ does not belong to $L^1(\mathbb{R})$ or to $L^2(\mathbb{R})$, and to avoid this pitfall we change it for $\hat{\alpha}_{MIP-ideal}(\xi)\mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$. Indeed, frequencies outside the interval $[-\pi|v|, \pi|v|]$ are of no interest for our scope since u is band limited on $[-\pi, \pi]$, the expected value of the observation being $\mathbb{E}\left(obs(n)\right) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$ (see Theorem 3.2). This change does not ensure that $\hat{\alpha}_{MIP-ideal}(\xi)\mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$ belongs to $L^2(\mathbb{R})$. Nevertheless, we can at least compute its Fourier expansion (as the Fourier expansion of an $L^1(\mathbb{R})$ function).

Now, we are able to apply Theorem 3.7 (with $\hat{\alpha}_{MIP-ideal} = \frac{1}{\sqrt{|a\xi|}}e^{-i\frac{\pi}{4}sign(\xi)}\mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$) and to compute the code. Being Hermitian this function provides a real code (i.e., coming from a real function in the space domain). The only loss incurred in applying Theorem 3.7 with a function that does not belong to $L^2(\mathbb{R})$ is the goodness of the convergence, which is reduced to a tempered distribution convergence. Nonetheless, by the localization principle and the Riemann–Lebesgue theorem, we also have at the very least a pointwise convergence everywhere, except in 0, of the Fourier expansion. Thus, it is no surprise that the proposed



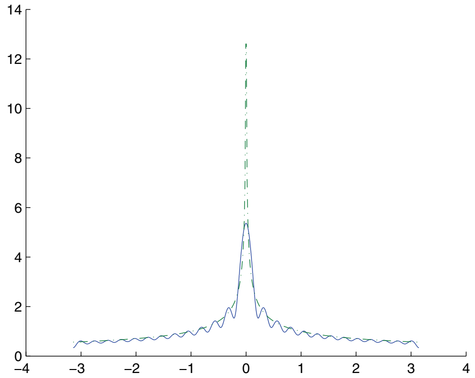


Figure 4. *Left: the flutter shutter gain function for the motion-invariant photography code (w.l.o.g. for $v = 1$), x-axis: k; y-axis: the gain α_k. On the right: the FT (modulus) of the motion-invariant photography code (solid line) and of the ideal motion-invariant photography function $\hat{\alpha}_{MIP-ideal}$ (dashed-dotted line). As predicted the proposed approximation does not vanish on $[-\pi, \pi]$. Thus the convolution of a band-limited function by the motion-invariant photography code is invertible, x-axis: frequency ξ; y-axis: $|\hat{\alpha}(\xi)|$.*

numerical approximation (which is a trigonometric polynomial and thus $C^\infty(\mathbb{R})$) works well and is indeed invertible for all band-limited functions such that $\hat{u}$ is supported on $[-\pi, \pi]$. The obtained code (w.l.o.g. for $v = 1$ and $\Delta t = 1$) and its FT are shown Figure 4 and will be compared advantageously to the Agrawal et al. code in [97]. The proposed implementation of *motion-invariant photography* using a *numerical flutter shutter* is simpler from a technical point of view since it does not require controlling the camera motion itself. This permits us to compute SNRs for *any* finite code and compare the *motion-invariant photography* like any other *flutter shutter* setup. The above formalism, paradoxes, and comparison also apply to the "motion blur removal with orthogonal parabolic exposures" [20], a recent extension of the *motion-invariant photography* using two images, namely two orthogonal motion invariant apparatuses. Indeed it is equivalent to the use of two *flutter shutters*, and, in that case, a fair comparison shall also involve the acquisition of two *flutter shutter* images. Surprisingly, the *numerical flutter shutter* permits us to approximate this ideal function with a finitely supported *flutter shutter* function (that is, a finite code) while avoiding an unrealistic infinite acceleration. It also permits us to get rid of the exigence of an a priori knowledge of the direction of v. As a consequence, like any other *flutter shutter* the coded-*motion-invariant photography* will work for any direction of v. Finally, it increases the efficiency of the method compared to the classic one involving an accelerating camera. Indeed it permits us to control the FT and to concentrate it easily on the support of $\hat{u}(\xi)$, i.e., where the information is (contrarily to $\hat{\alpha}_{mip-ideal}$, which is supported on the whole $\mathbb{R}$). Predictive results are shown in Table 2 and simulations in [97].

7. The flutter shutter paradoxes. In this section we compute the *best flutter shutter* function and the best snapshot and compare them.

Theorem 7.1 (ideal *flutter shutter* function). *Consider a landscape $u(x - vt)$ moving at velocity v. Then an optimal continuous numerical flutter shutter gain function maximizing the* *average spectral* SNR (2.5) *is equal to* $\alpha^*(t) = \operatorname{sinc}(tv)$.

Proof. Among all gain control functions $\alpha(t)$ the one giving the best $SNR^{spectral-averaged}$ (2.5) is given by minimizing the averaged variance of $\hat{\mathrm{u}}_{est}$ (3.13),

$$F(\alpha) = ||\alpha||_2^2 \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{d\xi}{|\hat{\alpha}(v\xi)|^2} \text{ (dropping the irrelevant constants, } u \text{ being fixed)}$$
$$\geq ||\alpha||_2^2 \frac{1}{\frac{1}{2\pi}\int_{-\pi}^{\pi}|\hat{\alpha}(v\xi)|^2 d\xi},$$

where the inequality is Jensen's inequality applied to the strictly convex function $x > 0 \mapsto \frac{1}{x}$. Because of this strict convexity, the equality occurs when $|\hat{\alpha}(\xi)|^2 \equiv 1$ on $[-\pi v, \pi v]$, up to an irrelevant multiplicative constant for a *numerical flutter shutter* (see the second Remark in section 3.1). Thus, an optimal *numerical flutter shutter* function is $\alpha^*(t) = \operatorname{sinc}(tv)$ (up to a normalization constant). ■

Notice that the proposed optimal *flutter shutter function* has a constant FT on the support of $\hat{u}$ for any velocity $|\tilde{v}| \leq |v|$. This means that this *flutter shutter* code is "self-deconvolving." Being nonpositive this ideal gain control function is *not* implementable using an *analog flutter shutter*, and being non–piecewise constant it is *not directly* implementable using a *numerical flutter shutter* strategy. However, a piecewise constant approximation can be used in a *numerical flutter shutter* strategy with Theorem 3.7 as soon as $|v|\Delta t \leq 1$, and it is enough to let $\Delta t \to 0$ to approximate the optimal *flutter shutter*.

Corollary 7.2 (upper bound on the SNR). *Consider a landscape* $u(x-vt)$ *moving at velocity* v. *The ideal numerical flutter shutter strategy using* $\alpha^*(t) = \operatorname{sinc}(tv\Delta t)$ *has a spectral* SNR (2.4) *equal to* $SNR^{spectral}(\xi) = \frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\sqrt{v}} \frac{|\hat{u}(\xi)|}{\sqrt{||u||_{L^1}}}$. *Moreover, the averaged spectral* SNR (2.5) *is equal to* $SNR^{spectral-averaged} = \frac{1}{2\pi\sqrt{v}} \frac{\int_{-\pi}^{\pi}|\hat{u}(\xi)|d\xi}{\sqrt{||u||_{L^1}}}$.

Proof. By Theorem 7.1, an optimal *flutter shutter* strategy satisfies $|\hat{\alpha}^*(\xi)| = \mathbb{1}_{[-\pi v,\pi v]}$. Using Parseval's formula we deduce that $||\alpha^*||_{L^2}^2 = v$. Then from Theorem 3.6 we deduce that $SNR^{spectral}(\xi) = \frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\sqrt{v}} \frac{|\hat{u}(\xi)|}{\sqrt{||u||_{L^1}}}$. It follows that $SNR^{spectral-averaged} = \frac{1}{2\pi\sqrt{v}} \frac{\int_{-\pi}^{\pi}|\hat{u}(\xi)|d\xi}{\sqrt{||u||_{L^1}}}$. ■

Corollary 7.3 (the *flutter shutter* paradox). *The use of a flutter shutter strategy increasing the total time exposure does not permit us to achieve an arbitrary* SNR. *Consider a landscape* $u(x - vt)$ *moving at velocity* v. *Then the* $SNR^{spectral-average}$ *of any flutter shutter strategy is bounded independently of the total exposure time. In other words, increasing the exposure time has a limited effect on the* SNR.

Proof. This is a direct consequence of Corollary 7.2. Indeed, the exposure time is the (infinite) length of the support of α^*, but the SNR of the restored image is nevertheless finite.

Moreover, $SNR(analog\ flutter) \leq SNR(numerical\ flutter)$ for *any analog flutter shutter* function and $SNR(numerical\ flutter) \leq SNR(best\ numerical\ flutter) < \infty$ by Corollary 7.2. Thus,

$$SNR(analog\ flutter) \leq SNR(numerical\ flutter) \leq SNR(best\ numerical\ flutter) < \infty,$$

implying that the SNR of any *analog flutter shutter* is bounded as well (and smaller than or equal to the *numerical flutter shutter*). ■

Corollary 7.4 (efficiency of the numerical *flutter shutter*). *Consider a landscape $u(x-vt)$ moving at velocity v. Then the ratio R of $SNR^{spectral-average}$ between the ideal flutter shutter and the best snapshot with exposure time equal to Δt^* is equal to $R = \frac{SNR^{spectral-average}(flutter,ideal)}{SNR^{spectral-average}(snapshot)} \approx \frac{\frac{1}{2\pi}}{0.1359} \approx 1.171$.*

Proof. This is a direct consequence of Theorem 5.1 and Corollary 7.2. ■

This result is surprising and disappointing. The gain of the most flexible *flutter shutter* that could be envisaged, a *numerical flutter shutter* with a continuous gain function, is insignificant with respect to the best classic snapshot. Nevertheless, even if the aperture time is the same, a *numerical flutter shutter* beats the standard snapshot slightly.

Corollary 7.5 (deconvolution gain). *For a landscape $u(x-vt)$ moving at velocity v, consider its best classic snapshot with exposure time equal to Δt^* and the flutter shutter strategy using $\alpha = \alpha^* \mathbb{1}_{[-\frac{\Delta t^*}{2}, \frac{\Delta t}{2}]}$. Then the spectral SNR, $SNR^{spectral-average}$ is larger for this restricted flutter shutter than for the best snapshot. The ratio of the SNRs is approximately* 1.04.

Proof. This is a mere numerical estimation using Theorems 5.1 and 7.1. ■

In short, the number of collected photons is not larger, but they are better combined. The resulting *flutter shutter* kernel is slightly better than the snapshot kernel. These positive and negative results constitute what we shall call the *flutter shutter paradoxes*. If the velocity of the observed object is known, none of the *flutter shutter* strategies beats the optimal standard snapshot adapted to this velocity significantly. Nevertheless, the *flutter shutter* strategy is always (slightly) better.

Table 4 summarizes the results on the various *flutter shutter* strategies explored, focusing on the resulting SNR.

Table 4

This table summarizes the results on the different flutter shutter strategies and their SNRs. In the first column the types of flutter are indicated. The second and last rows give the optimal flutter shutter function in two categories: the best simple snapshot, and the best numerical flutter shutter. The best numerical flutter shutter function is a (zoomed) sinc *function and has a finite SNR in spite of using an infinite exposure time. This is what we call the flutter shutter paradox. M.I.P. stands for motion-invariant photography. The second column shows the form of the α function. The Agrawal et al. code is analog discrete, with $\alpha_k \in \{0,1\}$. The last column gives the SNR of each method in its more adequate presentation: the first line shows that the accumulation is the best strategy, the only one able to increase the SNR indefinitely. The second and last lines compare the average spectral SNRs for the best snapshot and the best numerical flutter shutter (av. stands for average). The SNR gain with the numerical flutter shutter with respect to the best snapshot is only approximately* 1.171. *The spectral SNR formulas for the analog flutter shutter and the numerical flutter shutter are similar but distinct. The analogue involves the L^1 norm of α and the numerical L^2 norm.*

Flutter type	Flutter function $\alpha(t)$	(av. spectral) SNR
Accumulation	$\mathbb{1}_{[0,T[}(t)$, $v=0$	(spatial) $\sqrt{u(x)T}$
Best snapshot	$\mathbb{1}_{[0,\frac{1.0909}{v}[}(t)$	(av.) $\approx \frac{0.1359}{\sqrt{\lvert v\rvert}} \frac{\int_{-\pi}^{\pi}\lvert\hat{u}(\xi)\rvert d\xi}{\sqrt{\lvert\lvert u\rvert\rvert_{L1}}}$
Analog discrete	$\sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$, $\alpha_k \in [0,1]$	$\mathbb{1}_{[-\pi,\pi]}(\xi)\frac{\lvert\hat{\alpha}(\xi v)\rvert}{\lvert\lvert\alpha\rvert\rvert_{L1}} \frac{\lvert\hat{u}(\xi)\rvert}{\sqrt{\lvert\lvert u\rvert\rvert_{L1}}}$
Analog continuous	$\alpha(t) \geq 0$	*Idem*
Numerical	$\sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$, $\alpha_k \in \mathbb{R}$	$\mathbb{1}_{[-\pi,\pi]}(\xi)\frac{\lvert\hat{\alpha}(\xi v)\rvert}{\lvert\lvert\alpha\rvert\rvert_{L2}} \frac{\lvert\hat{u}(\xi)\rvert}{\sqrt{\lvert\lvert u\rvert\rvert_{L1}}}$
M.I.P. (numerical)	(approximating code)	*Idem*
Best numerical	$\mathrm{sinc}(tv)$	(av.) $\frac{1}{2\pi\sqrt{\lvert v\rvert}} \frac{\int_{-\pi}^{\pi}\lvert\hat{u}(\xi)d\xi\rvert}{\sqrt{\lvert\lvert u\rvert\rvert_{L1}}}$

8. Conclusion. This paper started by modeling the stochastic photon acquisition of a moving landscape by a light sensor. The model intrinsically contains noise terms due to the Poisson photon emission. This model permits us to formalize and analyze a general *flutter shutter* theory which includes the standard photography, the original Agrawal et al. *flutter shutter*, two suggested generalizations of the *flutter shutter* and the Levin et al. *motion-invariant photography*. A formula providing the SNR of the sharp recovered images directly has been given for all these methods. It also permits us to prove what we called the *flutter shutter paradoxes*. A well-optimized *flutter shutter* always beats the traditional camera, even using the same aperture time. And, for an infinite exposure time accumulating many more photons than a snapshot, the SNR remains finite (contrarily to the classic still photography). Two kinds of *flutter shutter* setups have been considered: an *analog flutter shutter*, and a *numerical flutter shutter* permitting smoother, negative gain-control-functions and leading to the best SNR of the restored images. It also appeared that the *motion-invariant photography* is a particular case of an *analog flutter shutter*. The *motion-invariant photography* has been generalized to the case of unknown velocity direction by using a *numerical flutter shutter*. Optimized snapshots have been considered leading to the definition of best blur. It gives the best aperture time to use in a standard camera and can be used, for example, to compute the ideal number of stages of the *time delay and integration* device commonly used in push broom satellites. It is proven that knowing the velocity the best *flutter shutter* code comes from the Fourier series coefficients of a (zoomed) sinc function. The SNR raise is of 17% compared to the best snapshot, leading to a poor efficiency of such an acquisition system, even if the exposure time is infinite.

Appendix. Main notations and formulae.

(i) $t \in \mathbb{R}^+$ time variable
(ii) Δt length of a time interval
(iii) $x \in \mathbb{R}$ spatial variable
(iv) $X \sim Y$ means that the random variables X and Y have the same law
(v) $\mathbb{P}(A)$ probability of an event A
(vi) $\mathbb{E}X$ expected value of a random variable X
(vii) $var(X)$ variance of a random variable X
(viii) $f * g$ convolution of two $L^2(\mathbb{R})$ functions $(f*g)(x) = \int_{-\infty}^{+\infty} f(y)g(x-y)dy$
(ix) $l(t,x) > 0\ \forall x \in \mathbb{R}^+ \times \mathbb{R}$ continuous landscape before passing through the optical system
(x) $\mathcal{P}(\lambda)$ Poisson random variable with intensity $\lambda > 0$
(xi) $\mathbf{P}_l$ bidimensional Poisson process on $\mathbb{R}^+ \times \mathbb{R}$ associated with the intensity field $l(t,x)$, $\mathbf{P}_l([t_1,t_2]\times[a,b]) \sim \mathcal{P}(\int_{t_1}^{t_2}\int_a^b l(t,x)dtdx)$
(xii) g *point-spread-function* of the optical system
(xiii) $u = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * l$ ideal observable landscape just before sampling. Assumption: $u \in L^1 \cap L^2$, band limited
(xiv) $obs(n)$, $n \in \mathbb{Z}$, observation of the landscape at pixel n
(xv) $e(n) = \mathbb{E}(obs(n))$, $n \in \mathbb{Z}$, and $e(x)$ its band-limited interpolation
(xvi) $\mathbf{P}_u$ Poisson process associated with the intensity field $u > 0$: $= \mathbf{P}_u \sim \mathbf{P}_{\mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}*g*l}$
(xvii) $\prod_a = \sum_{n\in\mathbb{Z}} \delta_{na}$ Dirac comb
(xviii) v relative velocity between the landscape and the camera (unit: pixels per second)
(xix) b apparent length of the support of the blur (unit: pixels), $Lv\Delta t = b$
(xx) $\alpha(t)$ piecewise constant or continuous gain control function for the *analog flutter shutter* and *numerical flutter shutter* methods
(xxi) $w(x) \geq 0$ weight function associated with the probability distribution of the velocity

v

(xxii) $\alpha^*(t)$ optimal gain control function

(xxiii) $||f||_{L^1} = \int |f(x)|dx$, $||f||_{L^2} = \sqrt{\int |f(x)|^2 dx}$

(xxiv) Let $f, g \in L^1(\mathbb{R})$ or $L^2(\mathbb{R})$; then $\mathcal{F}(f)(\xi) := \hat{f}(\xi) := \int_{-\infty}^{\infty} f(x)e^{-ix\xi}dx$ and $\mathcal{F}^{-1}(\mathcal{F}(f))(x) = f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathcal{F}(f)(\xi)e^{ix\xi}d\xi$. Moreover, $\mathcal{F}(f{*}g)(\xi) = \mathcal{F}(f)(\xi)\mathcal{F}(g)(\xi)$ and (Plancherel)

$$\int_{-\infty}^{\infty} |f(x)|^2 dx = ||f||_{L^2}^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |\mathcal{F}(f)|^2(\xi)d\xi = \frac{1}{2\pi}||\mathcal{F}(f)||_{L^2}^2$$

(xxv) $SNR(X) := \frac{|\mathbb{E}X|}{\sqrt{var(X)}}$, SNR of a random variable X

(xxvi) Let $\mathbb{u}_{est}(x)$ be an estimation of the landscape u based on a stochastic observation of u; then $SNR(\mathbb{u}_{est}(x)) := \frac{|\mathbb{E}\mathbb{u}_{est}(x)|}{\sqrt{var(\mathbb{u}_{est}(x))}}$

(xxvii) Let $\hat{\mathbb{u}}_{est}$ be an estimation of $\hat{u}$. Then $SNR^{spectral}(\mathbb{u}_{est}(\xi)) := \frac{|\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)|}{\sqrt{(var(\hat{\mathbb{u}}_{est}(\xi))}}$ for $\xi \in [-\pi, \pi]$

(xxviii) Let $\hat{\mathbb{u}}_{est}$ be an estimation of $\hat{u}$. Then

$$SNR^{spectral-averaged}(\hat{\mathbb{u}}_{est}) := \frac{\frac{1}{2\pi}\int |\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)|\mathbb{1}_{[-\pi,\pi]}(\xi)d\xi}{\sqrt{\frac{1}{2\pi}\int var(\hat{\mathbb{u}}_{est}(\xi))\mathbb{1}_{[-\pi,\pi]}(\xi)d\xi}}$$

(xxix) $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} = \frac{1}{2\pi}\mathcal{F}(\mathbb{1}_{[-\pi,\pi]})(x) = \mathcal{F}^{-1}(\mathbb{1}_{[-\pi,\pi]})(x)$

(xxx) (Poisson summation formula) Let $f \in L^1(\mathbb{R})$ be band limited. Then $\sum_n f(n) = \sum_m \hat{f}(2\pi m)$, and hence if $\hat{f}(\xi) = 0\ \forall|\xi| > \pi$, then $\sum_n f(n) = \hat{f}(0)$. Moreover, if f is positive, then $\sum_{n\in\mathbb{Z}} f(n) = \hat{f}(0) = ||f||_{L^1}$. An easy variant of the first Poisson formula gives the second Poisson formula we use, $\sum_n f(n)e^{-in\xi} = \sum_m \hat{f}(2\pi m + \xi)$. It is easily obtained by applying the first Poisson formula to $g(x) := f(x)e^{-ix\xi}$.

REFERENCES

[1] A. Adams, D.E. Jacobs, J. Dolson, M. Tico, K. Pulli, E.V. Talvala, B. Ajdin, D. Vaquero, H. Lensch, M. Horowitz, S.H. Park, N. Gelfand, J. Baek, W. Matusik, and M. Levoy, *The Frankencamera: an experimental platform for computational photography*, ACM Trans. Graph., 29 (2010), 29.

[2] A. Agrawal, M. Gupta, A. Veeraraghavan, and S.G. Narasimhan, *Optimal coded sampling for temporal super-resolution*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2010, pp. 599–606.

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

[4] A. Agrawal and R. Raskar, *Optimal single image capture for motion deblurring*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 2560–2567.

[5] A. Agrawal, A. Veeraraghavan, and R. Raskar, *Reinterpretable imager: Towards variable post-capture space, angle and time resolution in photography*, in Computer Graphics Forum, Vol. 29, Wiley-Blackwell, New York, 2010, pp. 763–772.

[6] A. Agrawal and Y. Xu, *Coded exposure deblurring: Optimized codes for PSF estimation and invertibility*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 2066–2073.

[7] C. Ancuti, C.O. Ancuti, and P. Bekaert, *Deblurring by matching*, in Computer Graphics Forum, Vol. 28, Wiley Online Library, 2009, pp. 619–628.

[8] F.J. Anscombe, *The transformation of Poisson, binomial and negative-binomial data*, Biometrika, 35 (1948), pp. 246–254.

[9] M. Backes, T. Chen, M. Duermuth, H. Lensch, and M. Welk, *Tempest in a teapot: Compromising reflections revisited*, in Proceedings of the 30th IEEE Symposium on Security and Privacy-Volume 00, 2009, pp. 315–327.

[10] Y. Bando and T. Nishita, *Towards digital refocusing from a single photograph*, in Proceedings of the Pacific Conference on Computer Graphics and Applications, IEEE Computer Society, Washington, DC, 2007, pp. 363–372.

[11] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro, *A variational framework for simultaneous motion estimation and restoration of motion-blurred video*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2007, pp. 1–8.

[12] G. Boracchi and A. Foi, *Uniform motion blur in Poissonian noise: Blur/noise tradeoff*, IEEE Trans. Image Process., 20 (2011), pp. 592–598.

[13] T. Buades, Y. Lou, J.M. Morel, and Z. Tang, *A note on multi-image denoising*, in Local and Non-Local Approximation in Image Processing (LNLA), IEEE International Workshop on, 2009, pp. 1–15.

[14] J.F. Cai, H. Ji, C. Liu, and Z. Shen, *Blind motion deblurring from a single image using sparse approximation*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 104–111.

[15] J.F. Cai, H. Ji, C. Liu, and Z. Shen, *Blind motion deblurring using multiple images*, J. Comput. Phys., 228 (2009), pp. 5057–5071.

[16] J.F. Cai, H. Ji, C. Liu, and Z. Shen, *High-quality curvelet-based motion deblurring from an image pair*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 1566–1573.

[17] W.G. Chen, N. Nandhakumar, and W.N. Martin, *Image motion estimation from motion smear-a new computational model*, IEEE Trans. Pattern Anal. Mach. Intell., 18 (1996), pp. 412–425.

[18] S. Cho and S. Lee, *Fast motion deblurring*, ACM Trans. Graph., 28 (2009), 145.

[19] S. Cho, Y. Matsushita, and S. Lee, *Removing non-uniform motion blur from images*, in Computer Vision, (ICCV), IEEE International Conference on, 2007, pp. 1–8.

[20] T.S. Cho, A. Levin, F. Durand, and W.T. Freeman, *Motion blur removal with orthogonal parabolic exposures*, in Computational Photography (ICCP), IEEE International Conference on, 2010, pp. 1–8.

[21] T. Cho, C. Zitnick, N. Joshi, S. Kang, R. Szeliski, and W. Freeman, *Image restoration by matching gradient distributions*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2012), pp. 683–694.

[22] S. Cvetkovic, H. Jellema, and P.H.N. de With, *Automatic level control for video cameras towards HDR techniques*, EURASIP J. Image Video Process., 2010 (2010), 13.

[23] S. Dai and Y. Wu, *Motion from blur*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2008, pp. 1–8.

[24] S. Dai and Y. Wu, *Removing partial blur in a single image*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 2544–2551.

[25] Y. Ding, S. McCloskey, and J. Yu, *Analysis of motion blur with a flutter shutter camera for non-linear motion*, in Proceedings of the 11th European Conference on Computer Vision: Part I, Springer-Verlag, Berlin, 2010, pp. 15–30.

[26] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman, *Removing camera shake from a single photograph*, ACM Trans. Graph., 25 (2006), pp. 787–794.

[27] A. Foi, S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, *A spatially adaptive poissonian image deblurring*, in Image Processing (ICIP), IEEE International Conference on, Vol. 1, 2005, pp. I–925.

[28] M.E. Gehm, R. John, D.J. Brady, R.M. Willett, and T.J. Schulz, *Single-shot compressive spectral imaging with a dual-disperser architecture*, Optics Express, 15 (2007), pp. 14013–14027.

[29] J.P. Godbaz, M.J. Cree, and A.A. Dorrington, *Extending AMCW lidar depth-of-field using a coded aperture*, in Proceedings of the 10th Asian Conference on Computer Vision: Part IV, Springer-Verlag, Berlin, 2010, pp. 397–409.

[30] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar, *Coded rolling shutter photography: Flexible space-*

time sampling, in Computational Photography (ICCP), IEEE International Conference on, 2010, pp. 1–8.
[31] A. Gupta, N. Joshi, C.L. Zitnick, M. Cohen, and B. Curless, *Single image deblurring using motion density functions*, in Proceedings of the 11th European Conference on Computer Vision: Part I, Springer-Verlag, Berlin, 2010, pp. 171–184.
[32] M. Gupta, A. Agrawal, A. Veeraraghavan, and S.G. Narasimhan, *Flexible voxels for motion-aware videography*, in Proceedings of the 11th European Conference on Computer Vision: Part I, Springer-Verlag, Berlin, 2010, pp. 100–114.
[33] S.W. Hasinoff and K.N. Kutulakos, *Light-efficient photography*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 2203–2214.
[34] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S.K. Nayar, *Video from a single coded exposure photograph using a learned over-complete dictionary*, in Computer Vision (ICCV), IEEE International Conference on, 2011.
[35] T. Hou, S. Wang, and H. Qin, *Image deconvolution with multi-stage convex relaxation and its perceptual evaluation*, IEEE Trans. Image Process., 20 (2011), pp. 3383–3392.
[36] P. Hsu and B.Y. Chen, *Blurred image detection and classification*, in Proceedings of the 14th International Conference on Advances in Multimedia Modeling, Springer-Verlag, Berlin, 2008, pp. 277–286.
[37] N. Jacobs, S. Schuh, and R. Pless, *On Unusual Pixel Shapes and Image Motion*, Tech. report, Computer Science and Engineering, Washington University, St. Louis, MO, 2009.
[38] N. Jacobs, S. Schuh, and R. Pless, *Compressive sensing and differential image-motion estimation*, in Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference on, 2010, pp. 718–721.
[39] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia, *Estimation of blur and noise parameters in remote sensing*, in Acoustics, Speech, and Signal Processing (ICASSP), IEEE International Conference on, Vol. 4, 2002, pp. IV–3580.
[40] P.A. Jansson and M. Richardson, *Deconvolution of images and spectra*, Opt. Eng., 36 (1997), pp. 3224–3225.
[41] H. Ji and C. Liu, *Motion blur identification from image gradients*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2008, pp. 1–8.
[42] J. Jia, *Single image motion deblurring using transparency*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2007, pp. 1–8.
[43] N. Joshi, S.B. Kang, C.L. Zitnick, and R. Szeliski, *Image deblurring using inertial measurement sensors*, ACM Trans. Graph., 29 (2010), 30.
[44] N. Joshi, R. Szeliski, and D.J. Kriegman, *PSF estimation using sharp edge prediction*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2008, pp. 1–8.
[45] T. Kenig, Z. Kam, and A. Feuer, *Blind image deconvolution using machine learning for three-dimensional microscopy*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 2191–2204.
[46] S.J. Koppal, S. Yamazaki, and S.G. Narasimhan, *Exploiting DLP illumination dithering for reconstruction and photography of high-speed scenes*, Int. J. Comput. Vis., 96 (2012), pp. 125–144.
[47] H. Kozuka, *Image sensor*, US Patent 6,473,538, 2002.
[48] D. Lanman, R. Raskar, A. Agrawal, and G. Taubin, *Shield fields: Modeling and capturing 3D occluders*, ACM Trans. Graph., 27 (2008), 131.
[49] M. Lebrun, M. Colom, A. Buades, and J. M. Morel, *Secrets of image denoising cuisine*, Acta Numer., 21 (2012), pp. 475–576.
[50] A. Levin, *Blind motion deblurring using image statistics*, Adv. Neural Inform. Process. Syst., 19 (2007), pp. 841–848.
[51] A. Levin, R. Fergus, F. Durand, and W.T. Freeman, *Deconvolution Using Natural Image Priors*, Tech. report, Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA, 2007.
[52] A. Levin, R. Fergus, F. Durand, and W.T. Freeman, *Image and depth from a conventional camera with a coded aperture*, ACM Trans. Graph., 26 (2007), 70.
[53] A. Levin, P. Sand, T.S. Cho, F. Durand, and W.T. Freeman, *Motion-invariant photography*, ACM Trans. Graph., 27 (2008), 71.
[54] A. Levin, P. Sand, T.S. Cho, F. Durand, and W.T. Freeman, *Method and apparatus for motion invariant imaging*, US Patent 20,090,244,300, 2009.

[55] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, *Efficient marginal likelihood optimization in blind deconvolution*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2011, pp. 2657–2664.

[56] F. Li, J. Sun, J. Wang, and J. Yu, *Dual-focus stereo imaging*, J. Electron. Imaging, 19 (2010), 043009.

[57] C.K. Liang, T.H. Lin, B.Y. Wong, C. Liu, and H.H. Chen, *Programmable aperture photography: Multiplexed light field acquisition*, ACM Trans. Graph., 27 (2008), 55.

[58] L.B. Lucy, *An iterative technique for the rectification of observed distributions*, Astron. J., 79 (1974), pp. 745–754.

[59] R. Lukac, *Computational Photography: Methods and Applications*, Vol. 2, CRC Press, Boca Raton, FL, 2010.

[60] M. Martinello, T.E. Bishop, and P. Favaro, *A Bayesian approach to shape from coded aperture*, in Image Processing (ICIP), IEEE International Conference on, 2010, pp. 3521–3524.

[61] M. Martinello and P. Favaro, *Single image blind deconvolution with higher-order texture statistics*, in Video Processing and Computational Video, Springer-Verlag, Berlin, 2011, pp. 124–151.

[62] S. McCloskey, *Velocity-dependent shutter sequences for motion deblurring*, in Proceedings of the 11th European Conference on Computer Vision: Part VI, Springer-Verlag, Berlin, 2010, pp. 309–322.

[63] S. McCloskey, *Temporally coded flash illumination for motion deblurring*, in Computer Vision (ICCV), IEEE International Conference on, 2011, pp. 683–690.

[64] S. McCloskey, W. Au, and J. Jelinek, *Iris capture from moving subjects using a fluttering shutter*, in Biometrics: Theory Applications and Systems (BTAS), IEEE International Conference on, 2010, pp. 1–6.

[65] S. McCloskey, K. Muldoon, and S. Venkatesha, *Motion invariance and custom blur from lens motion*, in Computational Photography (ICCP), IEEE International Conference on, 2011, pp. 1–8.

[66] L. Mei, X. Cai, and W. Liu, *Defocus deblurring with a coded aperture*, in Progress in Informatics and Computing (PIC), IEEE International Conference on, Vol. 2, 2010, pp. 916–919.

[67] A. Mohan, D. Lanman, S. Hiura, and R. Raskar, *Image destabilization: Programmable defocus using lens and sensor motion*, in Computational Photography (ICCP), IEEE International Conference on, 2009, pp. 1–8.

[68] H. Nagahara, C. Zhou, T. Watanabe, H. Ishiguro, and S.K. Nayar, *Programmable aperture camera using LCoS*, in Proceedings of the 11th European Conference on Computer Vision: Part VI, Springer-Verlag, Berlin, 2010, pp. 337–350.

[69] K. Nakamura, H. Ohzu, and I. Ueno, *Image sensor in which reading and resetting are simultaneously performed*, US Patent 5,262,870, 1993.

[70] S.G. Narasimhan, S.J. Koppal, and S. Yamazaki, *Temporal dithering of illumination for fast active vision*, in Proceedings of the 10th European Conference on Computer Vision: Part IV, Springer-Verlag, Berlin, 2008, pp. 830–844.

[71] S.K. Nayar and M. Ben-Ezra, *Motion-based motion deblurring*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 689–698.

[72] A. Raj and R. Zabih, *A graph cut algorithm for generalized image deconvolution*, in Computer Vision (ICCV), IEEE International Conference on, Vol. 2, 2005, pp. 1048–1054.

[73] R. Raskar, *Less is more: Coded computational photography*, in Proceedings of the 8th Asian Conference on Computer Vision: Part I, Springer-Verlag, Berlin, 2007, pp. 1–12.

[74] R. Raskar, *Computational photography: Epsilon to coded photography*, in Selected and Invited Papers, Lecture Notes in Comput. Sci. 5416, Springer-Verlag, New York, 2009, pp. 238–253.

[75] R. Raskar, *Method and apparatus for deblurring images*, US Patent 7,756,407, 2010.

[76] R. Raskar, J. Tumblin, and A. Agrawal, *Method for deblurring images using optimized temporal coding patterns*, US Patent 7,580,620, 2009.

[77] A. Rav-Acha and S. Peleg, *Two motion-blurred images are better than one*, Pattern Recognition Lett., 26 (2005), pp. 311–317.

[78] D. Reddy, A. Veeraraghavan, and R. Chellappa, *$P2C2$: Programmable pixel compressive camera for high speed imaging*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2011, pp. 329–336.

[79] D. Reddy, A. Veeraraghavan, and R. Raskar, *Coded strobing photography for high-speed periodic events*, in Imaging Systems, Optical Society of America, Washington, DC, 2010.

[80] C. Ren, W. Chen, and I. Shen, *Three-stage motion deblurring from a video*, in Proceedings of the 8th Asian Conference on Computer Vision: Part II, Springer-Verlag, Berlin, 2007, pp. 53–62.

[81] W.H. Richardson, *Bayesian-based iterative method of image restoration*, J. Opt. Soc. Amer., 62 (1972), pp. 55–59.

[82] P.D. Samarasinghe, R.A. Kennedy, and H. Li, *On non-blind image restoration*, in Signal Processing and Communication Systems (ICSPCS), IEEE International Conference on, 2009, pp. 1–7.

[83] A. Sarker and L.G.C. Hamey, *Improved reconstruction of flutter shutter images for motion blur reduction*, in Digital Image Computing: Techniques and Applications (DICTA), IEEE International Conference on, 2010, pp. 417–422.

[84] S. Schuon and K. Diepold, *Comparison of motion de-blur algorithms and real world deployment*, Acta Astronautica, 64 (2009), pp. 1050–1065.

[85] A. Sellent, M. Eisemann, and M. Magnor, *Calculating Motion Fields from Images with Two Different Exposure Times*, Tech. report, Computer Graphics Lab, Technical University of Braunschweig, Braunschweig, Germany, 2008.

[86] A. Sellent, M. Eisemann, and M. Magnor, *Motion field and occlusion time estimation via alternate exposure flow*, in Computational Photography (ICCP), IEEE International Conference on, 2009, pp. 1–8.

[87] K. Sengupta and F. Porikli, *Geometric Sequence (GS) imaging with Bayesian smoothing for optical and capacitive imaging sensors*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 90–97.

[88] Q. Shan, J. Jia, and A. Agarwala, *High-quality motion deblurring from a single image*, ACM Trans. Graph., 27 (2008), 73.

[89] G. Shi, D. Gao, D. Liu, and L. Wang, *High resolution image reconstruction: A new imager via movable random exposure*, in Image Processing (ICIP), IEEE International Conference on, 2009, pp. 1177–1180.

[90] M. Shi and J.Y. Zheng, *A slit scanning depth of route panorama from stationary blur*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, Vol. 1, 2005, pp. 1047–1054.

[91] C. Song, H. Zhao, W. Jing, and H. Zhu, *PSO based motion deblurring for single image*, in Proceedings of the 13th Annual ACM Conference on Genetic and Evolutionary Computation, 2011, pp. 85–92.

[92] Y.W. Tai, H. Du, M.S. Brown, and S. Lin, *Image/video deblurring using a hybrid camera*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2008, pp. 1–8.

[93] Y.W. Tai, H. Du, M.S. Brown, and S. Lin, *Correction of spatially varying image and video motion blur using a hybrid camera*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 1012–1028.

[94] Y.W. Tai, N. Kong, S. Lin, and S.Y. Shin, *Coded exposure imaging for projective motion deblurring*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2010, pp. 2408–2415.

[95] Y. Tai, P. Tan, and M. Brown, *Richardson-Lucy deblurring for scenes under projective motion path*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 1603–1618.

[96] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis, *Synthetic shutter speed imaging*, in Computer Graphics Forum, Vol. 26, Wiley Online Library, 2007, pp. 591–598.

[97] Y. Tendero, *The Flutter Shutter Camera Simulator*, Image Processing On Line, 2012 (2012).

[98] M. Trentacoste, C. Lau, M. Rouf, R. Mantiuk, and W. Heidrich, *Defocus Techniques for Camera Dynamic Range Expansion*, Proc. SPIE 7537, SPIE, Bellingham, WA, 2010.

[99] R. Tsai, *Pulsed control of camera flash*, US Patent 7,962,031, 2011.

[100] D.A. Vaquero, R. Raskary, R.S. Feris, and M. Turk, *A projector-camera setup for geometry-invariant frequency demultiplexing*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2009, pp. 2082–2089.

[101] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin, *Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing*, ACM Trans. Graph., 26 (2007), 69.

[102] A. Veeraraghavan, D. Reddy, and R. Raskar, *Coded strobing photography: Compressive sensing of high speed periodic events*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 671–686.

[103] V.S. Veeravasarapu and J. Sivaswamy, *Motion deblurring as optimisation*, in Proceedings of the Sev-

enth ACM Indian Conference on Computer Vision, Graphics and Image Processing, 2010, pp. 267–273.

[104] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, *Non-uniform deblurring for shaken images*, in Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2010, pp. 491–498.

[105] B. Wilburn, N. Joshi, V. Vaish, E.V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, *High performance imaging using large camera arrays*, ACM Trans. Graph., 24 (2005), pp. 765–776.

[106] S. Xiang, G. Meng, Y. Wang, C. Pan, and C. Zhang, *Image deblurring with matrix regression and gradient evolution*, Pattern Recognition, 45 (2012), pp. 2164–2179.

[107] S. Xu, H. Liang, D. Tu, and G. Li, *A deburring technique for large scale motion blur images using a hybrid camera*, in Image and Signal Processing (CISP), IEEE International Congress on, Vol. 2, 2010, pp. 806–810.

[108] W. Xu and S. McCloskey, 2*D Barcode localization and motion deblurring using a flutter shutter camera*, in Applications of Computer Vision (WACV), IEEE Workshop on, 2011, pp. 159–165.

[109] W. Yongpan, F. Huajun, X. Zhihai, L. Qi, and D. Chaoyue, *An improved Richardson-Lucy algorithm based on local prior*, Optic Laser Tech., 42 (2010), pp. 845–849.

[110] L. Yuan, J. Sun, L. Quan, and H.Y. Shum, *Image deblurring with blurred/noisy image pairs*, ACM Trans. Graph., 26 (2007), 1.

[111] L. Yuan, J. Sun, L. Quan, and H.Y. Shum, *Progressive inter-scale and intra-scale non-blind image deconvolution*, ACM Trans. Graph., 27 (2008), 74.

[112] J. Zhao, H. Feng, Z. Xu, and Q. Li, *An improved image restoration approach using adaptive local constraint*, Optik-International Journal for Light and Electron Optics, 123 (2012), pp. 982–985.

[113] C. Zhou and S. Nayar, *Computational cameras: Convergence of optics and processing*, IEEE Trans. Image Process., 20 (2011), pp. 3322–3340.