

The Flutter Shutter Paradox

Yohann Tundero Jean-Michel Morel Bernard Rougé

November 4, 2011

1 Abstract

Acquiring good quality images of moving objects by a digital camera remains a valid question, particularly if the velocity of the photographed object is only partially known, it is virtually impossible to tune an optimal exposure time. The same question arises when a solid object is being photographed by a moving camera. This is not only frequent, but even necessary when the camera is embarked on terrestrial vehicle, a plane, or an Earth observation satellite. In the latter cases, the camera velocity is constrained by physical laws. For this reason the recent Raskar and al. *flutter shutter* and the Durand and al. *motion invariant photography* methods have generated much interest. This paper proposes a mathematical analysis of moving photography which permits to treat in the same formalism the *flutter shutter* and the *motion invariant photography*. A general *flutter shutter* method is defined and explored, which allows for nonbinary shutter sequences and permits to formalize the question of an optimal *flutter shutter* code. The first result of this analysis is actually negative, and raises what we call the *flutter shutter paradox*. We prove that both the *flutter shutter* and the *motion-invariant photography* methods cannot increase indefinitely the signal to noise ratio compared to an exposure short enough to avoid motion blur despite the use of a possibly infinite exposure time. Bounds are shown. We also resolve the *flutter shutter* paradox under the assumption that the relative velocity of object and camera is not *a priori* known. Then a significant increase of the average SNR can be expected provided there is an *a priori* knowledge of the probability density for the observed velocities. In this set up we show how to compute *analytically* an optimal *flutter shutter* strategy.

Key words: Motion blur, flutter shutter, optimization, motion invariant photography, SNR.

2 Introduction

If a scene being photographed moves during the exposition process, or if the scene is steady and the camera moves, the resulting images are degraded by motion blur. Motion deblurring is the combination of two dependent problems a) the kind of kernel applied to the images which depends here on the motion b) the actual deblurring method, where the kernel may have to be estimated *a posteriori*, or not. Motion blur arises from multiples causes and is very common even for consumer level photography where it is partly compensated by optical, mechanical, or digital stabilizers. Lens based optical stabilizers use a floating sensor moving orthogonally to the optical axis. Vibrations are detected using gyroscopic sensors estimating the camera motion in the x and y axis of the focal plane. Hence it cannot compensate for camera rotations. Sensor shift stabilizers work similarly but do not use any specific lens. Hence, they tend to be cheaper. These techniques 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 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 one dimensional 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 more invertible and makes the restoration process to an ill posed problem.

An exciting alternative to classic photography was proposed in [1, 21] where the authors suggest modifications in the acquisition process to get invertible blur kernels by using a *flutter shutter*. 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.

This problem is the core problem of photography. The first photographs taken by Nicephore Niepce required several hours a time incompatible with live subjects or even with outdoors 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.

At first sight the *flutter shutter* looks like a magic solution that should equip all cameras. It could indefinitely increase the SNR by increased exposure, at no cost from the motion blur side. We shall see that this hasty conclusion is actually wrong, and that the SNR increase is limited, no mat-

ter how long the exposure time is. It is this surprising fact that we shall call the “*flutter shutter paradox*”. However, we shall also see that the method can make more flexible the use of certain cameras, particularly when their motion or when the motion of the scene can be modeled.

2.1 Related work

Blind deconvolution techniques [7, 19, 31, 13, 22] aimed at estimating the blur and recovering the sharp image directly from the blurred one. Deconvolution algorithms have been developed intensively [15, 25]. For example in [38] the authors suggest a modification of the Richardson Lucy method [28, 23] to control the artifacts of the restored image. In [20] Fergus and al. use natural image statistics to estimate the blur. In [18, 24, 10, 14, 35, 17] good results are shown on the blur estimation and/or deblurring. However the power spectrum of the acquired images with blur of more than two pixels contains several zeros. Thus useful information for image quality is lost. Hence, no matter how sophisticated the image reconstruction is, it is virtually impossible to recover a de-blurred image without very strong hypotheses on the underlying landscape. Such strong hypotheses are unrealistic for most images. The results are therefore in practice poor [29].

In an attempt to transform the blur problem into a well posed problem the authors of [6, 8, 30, 27, 9] proposed to use two photographs with different blurs instead of one. In [37] the authors use a long exposure taken image and another one, sharp but noisy. In [33] the authors use a full multi-image framework acquiring a bunch of sharp but noisy images and recovering a sharp image with increased SNR.

In [4, 32, 36, 12, 34] 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.

A solution was proposed in [1, 3, 2] by Raskar and al. The new acquisition process modulates the photon flux into the camera by opening and closing the camera shutter according to certain well chosen pseudo random binary codes. In the case of a uniform motion in front of the camera, the resulting blur kernel becomes invertible (there is no zero in its Fourier transform), however big the velocity is. The visual result of an image acquired by *flutter shutter* is close to a stroboscopic image, which can nonetheless give back a neat image by deconvolution.

Similarly in [21] Levin and al. suggested to move 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. Nevertheless, this approach causes blur in static parts of the scene.

Yet, thanks to the invertibility (well posedness of the recovering 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.

2.2 Overview

Section 3 proposes a general mathematical framework for image acquisition using a physical Poisson model for the photons capture process, including the obscurity noise. This model suits well our context since all noise terms inherent to image sensing are taken into account without any approximation. The model is detailed in a static context in section 3.1 and applied in 3.1.3. The non stationarity motion induced case is introduced in section 3.2 and applied to standard image acquisition : snapshots 3.3 and multi-images 3.2.2. In section 4 the mathematical model is used to analyze the *flutter shutter* methods and verify that they actually work. Generalizations for digital 4.1 and analogical 4.2 implementations of the flutter shutter are proposed and analyzed. Section 5 shows that the numerical *flutter shutter* beats the analog *flutter shutter* for the image quality and is more flexible to use. Nevertheless, Section 6 shows that the use of the *flutter shutter* does not improve on indefinitely the SNR of the sharp recovered image compared to a traditional snapshot with a (“*quick-enough-to-be-sharp*”) aperture time. This is the *flutter shutter paradox*. In section 4.3 the *motion-invariant photography* is shown to be a particular case of our generalized *Flutter-Shutter*. Results are illustrated in section 7. Section 8 proposes a solution to the *flutter shutter paradox* theorem provided the probability density of the observed velocities is known. Section 9.1 constructs optimized codes for centered-Gaussian and uniform (with different std-dev σ and ranges) distribution over the velocity. Simulations, predictive results and evaluation on simulations are shown and discussed in section 9.

3 Mathematical modeling

This section presents a continuous stochastic model of photons captured by a sensor array. The model applies to a standard image acquisition on still or moving landscapes, provided the motion is uniform and stationary. Without loss of generality the formalization will be done in the case where the sensor array is 1D and the landscape moves in the direction given by the sensor. Let $\mathbf{P}_l : \mathbb{R}^+ \times \mathbb{R}$ be a bi-dimensional Poisson process of intensity $l(t, x)$, $\forall (t, x) \in \mathbb{R}^+ \times \mathbb{R}$ (here l is called landscape, t resp. x are the time resp. spatial positions). This means that for every a, b, t_1, t_2 (with $a < b$ and $0 \leq t_1 < t_2$) $\mathbf{P}_l([t_1, t_2] \times [a, b])$ is a Poisson random variable with intensity $\int_{t_1}^{t_2} \int_a^b l(t, x) dx dt$. The theoretical observation of a pixel sensor (photon counter) of unit length centered at x during the time span $[0, \Delta t]$ is a Poisson

random variable

$$\mathbf{P}_l \left([0, \Delta t] \times \left[x - \frac{1}{2}, x + \frac{1}{2} \right] \right) \sim \mathcal{P} \left(\int_0^{\Delta t} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(t, y) dy dt \right)$$

where $[x - \frac{1}{2}, x + \frac{1}{2}]$ represents the normalized sensor unit, and by $X \sim P$ we mean that a random variable X has law P . In other terms the probability to observe 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 equal to

$$\mathbb{P} \left(\mathbf{P}_l \left([0, \Delta t] \times \left[x - \frac{1}{2}, x + \frac{1}{2} \right] \right) = k \right) = \frac{\left(\int_0^{\Delta t} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(t, y) dy dt \right)^k e^{-\left(\int_0^{\Delta t} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(t, y) dy dt \right)}}{k!}.$$

3.1 The still photography case

In the classic still photography set up, $l(t, x) = l(x)$, which makes of $\mathbf{P}_l : \mathbb{R}^+ \times \mathbb{R}$ a bi-dimensional time stationary and spatially inhomogeneous Poisson process of intensity $l(x)$, $\forall (t, x) \in \mathbb{R}^+ \times \mathbb{R}$. Thus for every a, b, t_1, t_2 (s.t $a < b$ and $0 \leq t_1 < t_2$) $\mathbf{P}_l([t_1, t_2] \times [a, b])$ is a Poisson random variable with intensity $\int_{t_1}^{t_2} \int_a^b l(x) dx dt = |t_2 - t_1| \int_a^b l(x) dx$ (by stationarity of the process). Then the theoretical observation of a pixel sensor (photon counter) of unit length centered at x using a exposure time of Δt is a Poisson random variable

$$\begin{aligned} \mathbf{P}_l \left([0, \Delta t] \times \left[x - \frac{1}{2}, x + \frac{1}{2} \right] \right) &\sim \mathcal{P} \left(\int_0^{\Delta t} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(y) dy dt \right) \\ &\sim \mathcal{P} \left(\Delta t \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} l(y) dy \right) \sim \mathcal{P} \left(\Delta t (\mathbb{1}_{[-\frac{1}{2}, \frac{1}{2}]} * l)(x) \right) \end{aligned}$$

where Δt is the exposure time, using a normalized sensor of unit length, and $*$ denotes the convolution (**viii**). (Here and in the rest of the text, Latin numerals refer to the formulas in the final glossary page 68.) For sampling purposes we assume that the theoretical landscape l is seen through an optical system with a *point spread function* g .

Definition We call ideal landscape the deterministic function

$$u = \mathbb{1}_{[-\frac{1}{2}, \frac{1}{2}]} * g * l \tag{1}$$

where g is the *point spread function* of the optical system providing a cut off frequency.

In other words, thanks to the convolution with g the acquisition system is able to sample u . From the imaging point of view if the center of the sensor is x then $u(x)$ represents the ideal (noiseless) pixel landscape value, which unfortunately can only be obtained after infinite exposure.

Definition (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 (1) and is band limited.

$$\mathbf{P}_l([t_1, t_2] \times [x - \frac{1}{2}, x + \frac{1}{2}]) \sim \mathcal{P} \left(\int_{t_1}^{t_2} \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} u(y) dy dt \right) \sim \mathcal{P}(|t_2 - t_1|u(x)). \quad (2)$$

Definition (Real acquisition system with noise included in the landscape.) A more 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) entails

$$\mathbf{P}_{l+\eta}([t_1, t_2] \times [x - \frac{1}{2}, x + \frac{1}{2}]) \sim \mathcal{P}(|t_2 - t_1|(u(x) + \eta)). \quad (3)$$

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 cut off frequency. We will also always assume in the sequel that $u \in L^1 \cap L^2(\mathbb{R})$. This assumption will be necessary to apply some of the mathematical formulas, but 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 $u \in L^1 \cap L^2$.

3.1.1 Sampling, interpolation

In the following u is assumed band limited, namely \hat{u} (see the definition **(xxiii)** of Fourier transform in the glossary) is supported in $[-\pi, \pi]$ and therefore can be sampled at a unit rate. This hypothesis is again no restriction, being justified by the frequency cutoff provided by the optical kernel g . The discrete sensor observations, or samples, will be denoted by $e(n)$ for $n \in \mathbb{Z}$. (Under a very long exposure T of a static scene, by the law of large numbers we have $\frac{e(n)}{T} \rightarrow u(n)$, and $u(n) = \mathbb{E}e(n)$.)

Definition Given a discrete array observation $e(n)$, $n \in \mathbb{Z}$, its band limited interpolate $e(x)$ $x \in \mathbb{R}$ can be defined by Shannon-Whittaker interpolation as

$$e(x) = \sum_{n \in \mathbb{Z}} e(n) \text{sinc}(x - n) \quad (\text{see in } \mathbf{(xxviii)} \text{ the definition of } \text{sinc}.) \quad (4)$$

Notice that in the deterministic ideal case, $e(n) = u(n)$ and, u being band-limited, we would deduce $e(x) = u(x)$ from $e(n) = u(n)$. By (4) we also have

$$\hat{e}(\xi) = \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi). \quad (5)$$

3.1.2 Noise measurement

Definition In the following we call signal to noise ratio (SNR) of a random variable X the ratio

$$SNR(X) := \frac{|\mathbb{E}X|}{\sqrt{\text{var}(X)}}. \quad (6)$$

For example if $\mathfrak{u}_{est}(x)$ (resp. $\hat{\mathfrak{u}}_{est}(\xi)$) is an estimation of the landscape $u(x)$ (resp. $\hat{u}(\xi)$) based on a noisy observation of u ,

$$SNR(\mathfrak{u}_{est}(x)) := \frac{|\mathbb{E}\mathfrak{u}_{est}(x)|}{\sqrt{\text{var}(\mathfrak{u}_{est}(x))}}. \quad (7)$$

Likewise, we call “spectral SNR ” of \mathfrak{u}_{est} the frequency dependent ratio defined by

$$SNR^{spectral}(\hat{\mathfrak{u}}_{est}(\xi)) := \frac{|\mathbb{E}\hat{\mathfrak{u}}_{est}(\xi)|}{\sqrt{\text{var}(\hat{\mathfrak{u}}_{est}(\xi))}} \quad \text{for } \xi \in [-\pi, \pi]; \quad (8)$$

$$SNR^{spectral-averaged}(\hat{\mathfrak{u}}_{est}) := \frac{\frac{1}{2\pi} \int |\mathbb{E}\hat{\mathfrak{u}}_{est}(\xi)| \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi}{\sqrt{\frac{1}{2\pi} \int \text{var}(\hat{\mathfrak{u}}_{est}(\xi)) \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi}}. \quad (9)$$

The reason of these definitions is to compare in terms of SNR the *flutter shutter* method to a standard “quick enough to be sharp” snapshot image acquisition strategy in the sense of (6), (7), (8), (9).

3.1.3 Standard acquisition, the SNR with no blur

From the acquisition system definition (3) the observed image at position x using the standard image acquisition strategy with an exposure time $N\Delta t$ can be any realization of

$$\mathbf{P}_l([0, N\Delta t] \times [x - \frac{1}{2}, x + \frac{1}{2}]) \sim \mathcal{P} \left(\int_{[0, N\Delta t]} u(x) dt \right) \sim \mathcal{P}(N\Delta t u(x)). \quad (10)$$

Notice that in this case the expected value and variance of a pixel sensor at position x are equal to $N\Delta t u(x)$.

Theorem 3.1. (Fundamental theorem of photography)

In the case of a still image, the SNR satisfies $SNR(x) = \sqrt{u(x)N\Delta t}$, where $N\Delta t$ 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 $u(x)$. Thus the only secure way to increase the SNR is to increase the exposure time $N\Delta t$.

Proof. Let $\mathbf{u}_{est} \sim \frac{\mathbf{P}_l([0, N\Delta t] \times [x - \frac{1}{2}, x + \frac{1}{2}])}{N\Delta t}$ be the estimate of the ideal landscape u . The relation (10) entails

$$\mathbf{u}_{est}(x) \sim \frac{\mathcal{P}(N\Delta t u(x))}{N\Delta t}.$$

Then $\mathbb{E}(\mathbf{u}_{est}(x)) = u(x)$ and the variance of $\mathbf{u}_{est}(x)$ is

$$var(\mathbf{u}_{est}(x)) = \frac{N\Delta t u(x)}{(N\Delta t)^2} = \frac{u(x)}{N\Delta t}.$$

It follows from (7) that $SNR(\mathbf{u}_{est})(x) = \sqrt{u(x)N\Delta t}$. □

3.2 Image acquisition with a moving landscape

In the following the camera focal plane is always taken as reference frame. The camera is always assumed to move parallel to its focal plane in a straight trajectory in the x direction (Fig. 1). Hence the acquisition process is made in the x direction only. The camera moves straight at a speed $v(t)$ (counted in pixels per second) and the landscape is static. These assumptions may seem simplistic, but are actually technologically relevant, as they include the scanning of planar scenes like documents, or of scenes seen at a long distance, like aerial video and push broom satellites.

Thus from now on we assume $l(t, x) = l(x - tv(t))$, and mainly $v(t) \equiv v$ where l is the underlying stationary landscape model (section 3.1). Hence all the former discussion made on the acquisition system, sampling and interpolation holds. This section focuses on the mathematical model for standard photography, where there is either a motion blur in a single continuous exposure, or the acquisition of an image burst (several consecutive snapshots with shorter exposure). We shall give the conditions under which the exposure is short enough to get a quality photograph, and evaluate the SNR in all three cases: snapshots, longer exposure with motion blur, and image burst made of short enough snapshots.

3.2.1 Standard motion blur

In the following we fix a exposure time unit Δt , and measure exposure times as its multiples $N\Delta t$.

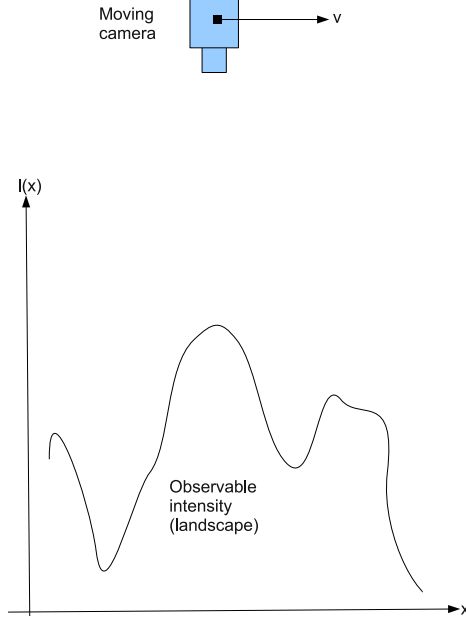


Figure 1: Acquisition system: the camera plane moves parallel to the focal plane in fixed direction, with scalar speed $v(t)$. The exposure time $N\Delta t$ is assumed short enough, or the camera elevation large enough to ensure that there is no apparent deformation of the landscape due to its varying height.

Theorem 3.2. *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 $Nv\Delta t$.*

Proof. The ideal landscape u is moving in the camera frame at a speed v (counted in pixels per second) and using (3) we get that the acquired image at position x is (a realization of)

$$\begin{aligned} \mathbf{P}_l([0, N\Delta t] \times [x - \frac{1}{2}, x + \frac{1}{2}]) &\sim \mathcal{P} \left(\int_0^{N\Delta t} u(x - vt) dt \right) \\ &\sim \mathcal{P} \left(\int_0^{Nv\Delta t} \frac{1}{v} u(x - t) dt \right) \sim \mathcal{P} \left(\left(\frac{1}{v} \mathbb{1}_{[0, Nv\Delta t]} * u \right)(x) \right). \end{aligned}$$

□

In this case the expected value and variance of a pixel sensor at position

x are equal to $\frac{1}{v}(\mathbb{1}_{[0, Nv\Delta t]} * u)(x)$. The quantity $Nv\Delta t$ is nothing but the length of the blur b (in pixels).

Remark The convolution with $h = \mathbb{1}_{[0, Nv\Delta t]}$ (standard blur) function is a non-invertible transformation as soon as the first zero of the Fourier transform (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 $\mathbb{1}_{[0, Nv\Delta t]}$ with a function whose convolution remains invertible for arbitrary $Nv\Delta t = b$. If instead, as considered in the next paragraph, the first zero of \hat{h} is outside the support of \hat{u} , then the motion blur is said negligible and it is actually invertible.

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

$$\begin{aligned} \mathbf{P}_l([0, \Delta t] \times [x - \frac{1}{2}, x + \frac{1}{2}]) &\sim \mathcal{P} \left(\int_0^{\Delta t} u(x - vt) dt \right) \sim \mathcal{P} \left(\int_0^{v\Delta t} \frac{1}{v} u(x - t) dt \right) \\ &\sim \mathcal{P} \left(\frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x) \right) \sim obs(x) \end{aligned} \quad (11)$$

where (11) is known only for $x \in \mathbb{Z}$. Taking the Fourier transform (**xxiii**) of the expected value in (11) yields

$$\frac{1}{v} \mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi) \hat{u}(\xi). \quad (12)$$

Moreover we have (**xxiii**)

$$\mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi) = \int_0^{v\Delta t} e^{-ix\xi} dx = 2 \frac{\sin(\frac{\xi v\Delta t}{2})}{\xi} e^{-i\xi \frac{v\Delta t}{2}}. \quad (13)$$

From (13) we see that we *must* have $|v|\Delta t < 2$ to guarantee the invertibility of the blur kernel on $[-\pi, \pi]$.

Definition Given v , we call “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

$$obs(n) \sim \mathcal{P} \left(\frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(n) \right).$$

We call *band limited interpolated snapshot* its band limited interpolate (5)

$$obs(x) \sim \sum_{n \in \mathbb{Z}} obs(n) sinc(x - n).$$

By definition of the standard snapshot, Δt is small enough so (13) has no zero on $[-\pi, \pi]$ (the support of \hat{u}). Thus (12, 13) lead to the definition of the inverse filter γ satisfying

$$\hat{u}(\xi) = \hat{\gamma}(\xi) \mathcal{F}(\mathbb{E}(obs))(\xi)$$

implying

$$\hat{\gamma}(\xi) = \frac{v \mathbb{1}_{[-\pi, \pi]}(\xi)}{2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi} e^{-i\xi \frac{v \Delta t}{2}}}. \quad (14)$$

Remark The inverse filter γ would give back the ideal observable landscape u if we knew $\mathbb{E}(obs)(x)$. But, in fact we can only apply it to the band limited interpolate of $\mathbb{E}(obs)(n)$, $n \in \mathbb{Z}$, which contains noise. It is nevertheless interesting to explore the noiseless ideal case.

The noiseless discrete case This is the ideal case where (from (11))

$$e(n) = \mathbb{E} \left(\mathcal{P} \left(\frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(n) dt \right) \right) = \frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(n).$$

We need now to compute $\hat{e}(\xi)$ from the samples $(e(n))_{n \in \mathbb{Z}}$ in order to perform the deconvolution. Since u is band limited (see section 3.1), $e(x)$ is also band limited and we can interpolate it using the sinc interpolation (4). Thus the band limited interpolated ideal observation is

$$e(x) = \sum_{n \in \mathbb{Z}} e(n) \text{sinc}(x - n). \quad (15)$$

Then from (15) we have

$$\hat{e}(\xi) = \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi). \quad (16)$$

So the ideal deconvolved image is obtained by combining (14) and (16) to give

$$\hat{d}(\xi) = \hat{\gamma}(\xi) \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi). \quad (17)$$

Step2: the real, noisy case

Definition We call estimated landscape $\mathfrak{u}_{est, sna}$ of the standard snapshot the function defined by (using the observed $obs(n)$ samples (11) instead of the ideal $e(n)$ in (17))

$$\mathcal{F}(\mathfrak{u}_{est, sna})(\xi) = \hat{\gamma}(\xi) \sum_{n \in \mathbb{Z}} obs(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi). \quad (18)$$

Theorem 3.3. For a standard snapshot using a exposure time of Δt the spectral SNR (8) is

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

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

$$\mathbb{E}(\mathcal{F}(\mathbf{u}_{est, sna})(\xi)) = \hat{u}(\xi) \mathbb{1}_{[-\pi, \pi]}(\xi); \quad (19)$$

and the variance is

$$var(\mathcal{F}(\mathbf{u}_{est, sna}(\xi))) = \frac{\|u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi)}{\Delta t \left| 2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi v \Delta t} \right|^2}. \quad (20)$$

Proof. Now we need to compute $var(\mathcal{F}(\mathbf{u}_{est, sna}(\xi)))$ in order to estimate the effect of the deconvolution of the final restored image. From (18) we have

$$var(\mathcal{F}(\mathbf{u}_{est, sna}(\xi))) = |\hat{\gamma}(\xi)|^2 \sum_{n \in \mathbb{Z}} var(obs(n)) \mathbb{1}_{[-\pi, \pi]}(\xi). \quad (21)$$

The above formula works under the realistic assumption that $obs(n)$ are independent random variables. Indeed, they correspond to disjoint sensors in the CCD and receive separate sets of photons. They are therefore two independent Poisson variables. By (11), $obs(n)$ being Poisson random variables we have $var(obs(n)) = \frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(n)$. Combining this with the above (21) equation and (14), we obtain

$$\begin{aligned} var(\mathcal{F}(\mathbf{u}_{est, sna}(\xi))) &= \frac{v \sum_{n \in \mathbb{Z}} (\mathbb{1}_{[0, v\Delta t]} * u)(n) \mathbb{1}_{[-\pi, \pi]}(\xi)}{|e^{-i\xi \frac{v\Delta t}{2}} 2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi}|^2} \\ &= \frac{v \|\mathbb{1}_{[0, v\Delta t]} * u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi)}{|e^{-i\xi \frac{v\Delta t}{2}} 2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi}|^2} \quad (\text{using the Poisson summation formula (xxix)}) \\ &= \frac{v \|\mathbb{1}_{[0, v\Delta t]}\|_{L^1} \|u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi)}{|2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi}|^2} = \frac{\|u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi)}{\Delta t \left| 2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi v \Delta t} \right|^2}. \end{aligned} \quad (22)$$

Here the crucial point is the use of the Poisson summation formula in equation (22). From (11) since $obs(n)$ are Poisson random variables $\mathbb{E}(obs(n)) = \frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(n)$ and following the same scheme we can compute $\mathbb{E}(\mathcal{F}(\mathbf{u}_{est}))(\xi)$.

$$\begin{aligned}
\mathbb{E}(\mathcal{F}(\mathbf{u}_{est, sna})(\xi)) &= \frac{\sum_{n \in \mathbb{Z}} \mathbb{E}(\text{obs}(n)) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi)}{\mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi)} = \frac{\sum_{n \in \mathbb{Z}} (\mathbb{1}_{[0, v\Delta t]} * u)(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi)}{\mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi)} \\
&= \frac{\sum_{m \in \mathbb{Z}} \mathcal{F}(\mathbb{1}_{[0, v\Delta t]} * u)(\xi + 2\pi m) \mathbb{1}_{[-\pi, \pi]}(\xi)}{\mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi)} \quad \text{using the first form of (xxix)} \\
&= \frac{\mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi) \hat{u}(\xi) \mathbb{1}_{[-\pi, \pi]}(\xi)}{\mathcal{F}(\mathbb{1}_{[0, v\Delta t]})(\xi)} = \hat{u}(\xi) \mathbb{1}_{[-\pi, \pi]}(\xi).
\end{aligned}$$

Combining these two last results and using the definition of the spectral SNR (8) we obtain $SNR^{\text{spectral}}(\xi) = \mathbb{1}_{[-\pi, \pi]}(\xi) |\hat{u}(\xi)| \sqrt{\frac{\Delta t}{\|u\|_{L^1}}} |2 \frac{\sin(\frac{\xi v \Delta t}{2})}{\xi v \Delta t}|$. \square

The only remaining 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 3.4. (Best exposure time for landscape recovery)

Consider a landscape $u(x - vt)$ moving at velocity v . Then for a snapshot the $SNR^{\text{spectral-averaged}}$ (9) is maximized when $|v| \Delta t^* \approx 1.0909$ and is equal to

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

Proof. From (20) the energy to be minimized in order to guarantee the best $SNR^{\text{spectral-averaged}}$ after deconvolution is

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

Then

$$\begin{aligned}
E'(\Delta t) &= \frac{v^2}{4} \left(\int_{-\pi}^{\pi} \frac{\xi^2}{\sin^2(\xi \frac{v \Delta t}{2})} d\xi + \Delta t \int_{-\pi}^{\pi} \frac{-\xi^3 v \cos(\xi \frac{v \Delta t}{2})}{\sin^3(\xi \frac{v \Delta t}{2})} d\xi \right) \\
&= \frac{v^2}{4} \int_{-\pi}^{\pi} \frac{\xi^2 (\sin(\xi \frac{v \Delta t}{2}) - \xi v \Delta t \cos(\xi \frac{v \Delta t}{2}))}{\sin^3(\xi \frac{v \Delta t}{2})} d\xi.
\end{aligned}$$

Which vanishes when $b^* = v\Delta t^* \approx 1.0909$. Then (9, 19, 20) entail

$$\begin{aligned} SNR^{spectral-averaged} &= \frac{\sqrt{\frac{1}{2\pi}} \int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\frac{\|u\|_{L^1}}{\Delta t^*} \int_{-\pi}^{\pi} \frac{d\xi}{\left|\frac{\sin(\frac{\xi v \Delta t^*}{2})}{\frac{\xi v \Delta t^*}{2}}\right|^2}}} = \frac{\sqrt{\frac{\Delta t^*}{2\pi}} \int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\|u\|_{L^1} \int_{-\pi}^{\pi} \frac{d\xi}{\left|\frac{\sin(\frac{\xi v \Delta t^*}{2})}{\frac{\xi v \Delta t^*}{2}}\right|^2}}} \\ &\approx \frac{\sqrt{\frac{1.0909}{2\pi v}} \int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\|u\|_{L^1} \int_{-\pi}^{\pi} \frac{d\xi}{\left|\frac{\sin(\frac{1.0909\xi}{2})}{\frac{1.0909\xi}{2}}\right|^2}}} \approx \frac{0.1359 \int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{v} \sqrt{\|u\|_{L^1}}}. \end{aligned}$$

□

Definition (Best snapshot.)

Given a moving landscape $u(x - vt)$ moving at velocity v , we call best snapshot strategy the use of the time-exposure $\Delta t^* \approx \frac{1.0909}{|v|}$.

3.2.2 The multi-image fusion to improve the SNR

The multi-image acquisition strategy consists in taking a burst of N images using a Δt exposure time. Δt is taken small enough, so that the elementary blur of each one of the N images is negligible (see section 3.2.1). Each image is registered and added to the former ones to obtain a single accumulated image. See [5] for an analysis of the feasibility of this technique. We shall assume here that the registration operation is error-free, so that the operation amounts to an increased exposure time.

Theorem 3.5. *The multi-image acquisition strategy using N images increases the SNR by a factor of \sqrt{N} compared to a standard snapshot.*

Proof. Before the registration operation the k -th image observed (for $k \in \{0, N - 1\}$) at position x is (a realization of)

$$\begin{aligned} \mathbf{P}_l([k\Delta t, (k+1)\Delta t] \times [x - \frac{1}{2}, x + \frac{1}{2}]) &\sim \mathcal{P} \left(\int_{k\Delta t}^{(k+1)\Delta t} u(x - vt) dt \right) \sim \mathcal{P} \left(\int_0^{\Delta t} u(x - vk\Delta t - vt) dt \right) \\ &\sim \mathcal{P} \left(\int_0^{v\Delta t} \frac{1}{v} u(x - vk\Delta t - t) dt \right) \sim \mathcal{P} \left(\frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x - vk\Delta t) \right) \end{aligned}$$

The registration operation consists in a perfect translation $\tau(x) = x + vk\Delta t$ of the previous observed, hence simulating a stationary landscape. After the registration operation the k -th image observed for $k \in \{0, N - 1\}$ at position x is (a realization of)

$$\mathcal{P} \left(\frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x) dt \right).$$

Finally the image is constructed by adding all observations, so it is (a realization of)

$$\sum_{k=0}^{N-1} \mathcal{P} \left(\frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x) dt \right) \sim \mathcal{P} \left(\sum_{k=0}^{N-1} \frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x) dt \right) \sim \mathcal{P} \left(N \frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x) dt \right). \quad (23)$$

(This last equivalence uses the independence of the Poisson random variables, justified by the disjoint time intervals). Following the same scheme used in (3.3) from (23) we get the estimated

$$u_{est, mul} \sim \frac{v}{N} \mathcal{P} \left(N \frac{1}{v} (\mathbb{1}_{[0, v\Delta t]} * u)(x) dt \right)$$

whose variance is N times smaller. Thus the N factor remains in all equations. Finally from (11, 20) we deduce that the SNR is increased by a \sqrt{N} factor. In other words the use of a multi-image fusion with a sufficient N permits to achieve *any* SNR. \square

4 The *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 $\sqrt{N\Delta t}$, the total exposure time. We shall see that this is not so.

4.1 The *numerical flutter shutter*

The *numerical flutter shutter* method consists in a numerical sensor gain modification taking place *after* the acquisition by the sensor. Roughly speaking the camera takes a burst of N images using a exposure time Δt . The k -th image is multiplied, for $k \in 0, \dots, N-1$, by an $\alpha_k \in \mathbb{R}$ gain. Then all images are added to obtain *one* observed image, the flutter shutter. Δt must be small enough so the blur of each image is negligible (definition in section 3.2.1). This technique is similar to the multi-image acquisition strategy but doesn't use any registration technique. If it is realizable, it seems that its interest is limited: why not keeping 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 burden if only the flutter shutter image (the sum) is transmitted.

The k -th acquired elementary image at a pixel at position n is a realization of

$$\mathcal{P} \left(\int_{k\Delta t}^{(k+1)\Delta t} u(n - vt) dt \right).$$

The observation flutter shutter observation is obtained by combining the k -th output with weight α_k . Thus the flutter shutter output at a pixel centered at n is

$$obs(n) \sim \sum_{k=0}^{N-1} \alpha_k \mathcal{P} \left(\int_{k\Delta t}^{(k+1)\Delta t} u(n - vt) dt \right) \quad (24)$$

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 $(\alpha_k)_{k=0, \dots, N-1}$, and its *flutter shutter function* defined by $\alpha(t) = \alpha_k$ for $k \in [k, k + 1[$.

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

$$obs(n) \sim \sum_{k=0}^{N-1} \alpha_k \mathcal{P} \left(\int_{k\Delta t}^{(k+1)\Delta t} u(n - vt) dt \right). \quad (25)$$

We call *numerical flutter shutter* its band limited interpolate

$$obs(x) \sim \sum_{n \in \mathbb{Z}} obs(n) sinc(x - n).$$

We call *flutter shutter function* the function

$$\alpha(t) = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1[}(t). \quad (26)$$

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

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

The motionless $v = 0$ case

In order to check what this means, we shall first consider the ideal motionless case where $v = 0$. In this situation the acquired image has simply lost some photons with respect to an accumulation image, because some of the α_k are null. Thus in the motionless case, the flutter shutter cannot intuitively but decrease the SNR. This is easily confirmed by the following calculation. We have

$$obs(x) \sim \sum_{k=0}^{N-1} \alpha_k \mathcal{P}(\Delta t u(x)).$$

Thus

$$\mathbb{E}(obs(x)) = \sum_{k=0}^{N-1} \alpha_k \Delta t u(x) \quad (27)$$

and

$$var(obs(x)) = \sum_{k=0}^{N-1} \alpha_k^2 \Delta t u(x). \quad (28)$$

From (27) we deduce that the estimated landscape \mathfrak{u}_{est} is obtained using

$$\mathfrak{u}_{est}(x) \sim \frac{obs(x)}{\sum_{k=0}^{N-1} \alpha_k \Delta t}. \quad (29)$$

Let $r \sim \mathfrak{u}_{est} - u$ be the residual error then from (27-28) we get

$$\mathbb{E}(r(x)) = \frac{\mathbb{E}(obs(x))}{\sum_{k=0}^{N-1} \alpha_k \Delta t} - u(x) = 0.$$

Hence $\mathcal{F}(\mathfrak{u}_{est})$ is an unbiased estimator for \hat{u} . Moreover, from (28, 29) we deduce

$$var(r(x)) = \frac{var(obs(x))}{(\sum_{k=0}^{N-1} \alpha_k)^2 \Delta t^2} = \frac{u(x) \sum_{k=0}^{N-1} \alpha_k^2}{(\sum_{k=0}^{N-1} \alpha_k)^2 \Delta t} = \frac{u(x) \sum_{k=0}^{N-1} \alpha_k^2}{(\sum_{k=0}^{N-1} \alpha_k)^2 \Delta t}.$$

Thus, by (7), $SNR(\mathfrak{u}_{est, sna}) = \sqrt{u(x) \Delta t \frac{(\sum_{k=0}^{N-1} \alpha_k)^2}{\sum_{k=0}^{N-1} \alpha_k^2}}$. Since always $\sum_k \alpha_k^2 \leq (\sum_k \alpha_k)^2$, the flutter shutter is less favorable than a mere accumulation. To give an example, in the case of a Raskar code where $\alpha_k = 0$ or 1 and the number of non null α_k is $N/2$. Then the SNR is $\sqrt{u(x) \Delta t \frac{(N/2)^2}{N/2}} = \sqrt{\frac{u(x) \Delta t N}{2}}$, to be compared to the pure accumulation case where $\alpha_k = 1$ and the SNR is $\sqrt{u(x) \Delta t \frac{N^2}{N}} = \sqrt{u(x) \Delta t N}$. Thus the Raskar flutter shutter entails in the motionless case a $\sqrt{2}$ loss factor for the SNR. We now pass to the general more interesting case, namely when there is a motion.

Flutter shutter in presence of motion

Theorem 4.1. *The observed samples of the numerical flutter shutter are such that, for $n \in \mathbb{Z}$*

$$\mathbb{E}(\text{obs}(n)) = \left(\frac{1}{v} \alpha \left(\frac{\cdot}{v\Delta t} \right) * u \right) (n) \quad (30)$$

and

$$\text{var}(\text{obs}(n)) = \left(\frac{1}{v} \alpha^2 \left(\frac{\cdot}{v\Delta t} \right) * u \right) (n). \quad (31)$$

Proof. From the numerical flutter shutter samples definition (25),

$$\mathbb{E}(\text{obs}(x)) = \sum_{k=0}^{N-1} \alpha_k \int_{k\Delta t}^{(k+1)\Delta t} u(x-vt) dt = \sum_{k=0}^{N-1} \alpha_k \int_k^{(k+1)} \Delta t u(x-v\Delta t s) ds = \int_0^N \Delta t \alpha(s) u(x-v\Delta t s) ds \quad (32)$$

where we recall that

$$\alpha = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1[}.$$

Thus,

$$\mathbb{E}(\text{obs}(x)) = \int_0^{Nv\Delta t} \frac{1}{v} \alpha \left(\frac{y}{v\Delta t} \right) u(x-y) dy = \int_{-\infty}^{+\infty} \frac{1}{v} \alpha \left(\frac{y}{v\Delta t} \right) u(x-y) dy = \left(\frac{1}{v} \alpha \left(\frac{\cdot}{v\Delta t} \right) * u \right) (x). \quad (33)$$

Similarly from (24)

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

□

Notice that $\text{obs}(x)$ is not necessarily a Poisson random variable. However, if all α_k are equal to 0 or 1 then by (32)

$$\text{obs}(x) \sim \mathcal{P} \left(\frac{1}{v} \alpha \left(\frac{\cdot}{v\Delta t} \right) * u \right) (x),$$

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.

4.1.1 Computing the inverse filter

Step 1: the noiseless case Let us examine first the discrete noiseless case, when $obs(n) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v\Delta t}\right) * u\right)(n)$ 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

$$\mathcal{F}\left(\frac{1}{v}\alpha\left(\frac{\cdot}{v\Delta t}\right) * u\right)(\xi) = \Delta t \hat{u}(\xi) \hat{\alpha}(\xi v \Delta t).$$

By hypothesis (3.1) we assumed that $\hat{u}(\xi) = 0$ for $|\xi| > \pi$. Hence for the invertibility we must only require that $|\hat{\alpha}(\xi v \Delta t)| > 0$ for $\xi \in [-\pi, \pi]$.

Definition We say that the flutter shutter with code $(\alpha_k)_{k=0, \dots, N-1}$ is invertible (for velocities v smaller than v_0) if $|\hat{\alpha}(\xi)| > 0$ for $\xi \in [-\pi v_0 \Delta t, \pi v_0 \Delta t]$.

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

$$\hat{\gamma}(\xi) = \frac{\mathbb{1}_{[-\pi, \pi]}(\xi)}{\Delta t \hat{\alpha}(\xi v \Delta t)}. \quad (34)$$

Since $\alpha \in L^1(\mathbb{R})$, $\xi \mapsto \hat{\alpha}(\xi)$ is bounded, continuous, nonzero on $[-\pi, \pi]$ and therefore $\hat{\gamma}$ is bounded and supported on $[-\pi, \pi]$. Therefore, γ is C^∞ , bounded, and band limited.

We shall as usual define the recovered landscape from noisy data by the formulae that would be valid for noiseless data. Assume 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 (4).

The band limited interpolate of the ideal observation is

$$e(x) = \sum_{n \in \mathbb{Z}} e(n) \text{sinc}(x - n). \quad (35)$$

Then from (35) we have

$$\hat{e}(\xi) = \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi). \quad (36)$$

So the ideal deconvolved landscape $d(x)$ obtained by combining (34) and (36) is

$$\hat{d}(\xi) = \frac{\sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbb{1}_{[-\pi, \pi]}(\xi)}{\Delta t \hat{\alpha}(\xi v \Delta t)}. \quad (37)$$

We shall now adopt the same formulae for the noisy case.

Flutter shutter landscape recovery in the real noisy case

Definition Assume that the flutter shutter with code $(\alpha_k)_{k=0,\dots,N-1}$ is invertible. We call estimated landscape $\mathbf{u}_{est,num}$ of the *numerical flutter shutter* the function defined by (using the obtained $obs(n)$ samples (24) instead of the ideal $e(n)$ in (37))

$$\mathcal{F}(\mathbf{u}_{est,num})(\xi) = \frac{\sum_{n \in \mathbb{Z}} obs(n) e^{-in\xi} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t \hat{\alpha}(\xi v \Delta t)}. \quad (38)$$

Theorem 4.2. *The numerical flutter shutter has a spectral SNR (8) equal to*

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

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

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

and the variance is

$$var(\mathcal{F}(\mathbf{u}_{est,num})(\xi)) = \frac{\|\alpha\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t |\hat{\alpha}(\xi v \Delta t)|^2}. \quad (39)$$

Proof. By (31, 38),

$$\begin{aligned} var(\mathcal{F}(\mathbf{u}_{est,num})(\xi)) &= \frac{var\left(\sum_{n \in \mathbb{Z}} obs(n) e^{-in\xi} \mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} = \frac{\sum_{n \in \mathbb{Z}} var(obs(n)) |e^{-in\xi} \mathbb{1}_{[-\pi,\pi]}(\xi)|^2}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} \\ &= \frac{\sum_{n \in \mathbb{Z}} \left(\frac{1}{v} \alpha^2 \left(\frac{\cdot}{v \Delta t}\right) * u\right)(n) \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} \\ &= \frac{\|\frac{1}{v} \alpha^2 \left(\frac{\cdot}{v \Delta t}\right) * u\|_{L^1} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} \quad (\text{by Poisson summation formula}) \\ & \quad (40) \\ &= \frac{\frac{1}{v} \|\alpha^2 \left(\frac{\cdot}{v \Delta t}\right)\|_{L^1} \|u\|_{L^1} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} = \frac{\frac{1}{v} \|\alpha \left(\frac{\cdot}{v \Delta t}\right)\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} \\ &= \frac{\frac{v \Delta t}{v} \|\alpha\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t^2 |\hat{\alpha}(\xi v \Delta t)|^2} = \frac{\|\alpha\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t |\hat{\alpha}(\xi v \Delta t)|^2}. \end{aligned}$$

Here again the crucial point is the use of the Poisson summation formula **(xxix)** in equation (40). Following the same scheme and starting from (33) $\mathbb{E}(\mathcal{F}(\mathbf{u}_{est,num})(\xi))(\xi)$ can be computed by the Poisson formula, using the fact that u is band limited in $[-\pi, \pi]$.

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

From (39) and using the definition of the spectral SNR (8), we obtain

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

□

Remark From (39) we deduce that $var(\mathcal{F}(\mathbf{u}_{est,num})(\xi))$ 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 From (26) follows that

$$\hat{\alpha}(\xi) = \sum_{k=0}^{N-1} \alpha_k \mathcal{F}(\mathbb{1}_{[k,k+1]})(\xi) = \frac{2 \sin(\frac{\xi}{2})}{\xi} \sum_{k=0}^{N-1} a_k e^{-i\xi \frac{2k+1}{2}} = \text{sinc}\left(\frac{\xi}{2\pi}\right) e^{-\frac{i\xi}{2}} \sum_{k=0}^{N-1} a_k e^{-ik\xi}. \quad (41)$$

Notice that this is *not* the DFT of the vector α , which means that in the literature on the flutter shutter, the simulations are neglecting the motion blur on the intervals with Δt length.

Remark 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 in a mere computational device, using the maximal sensor capability. In that case $obs(x)$ is given by (24) 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 case is discussed below as *Analog Flutter-Shutter*.

4.2 The *analog flutter shutter*

Raskar's flutter shutter method consists in 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 than a simple shutter could perform. 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. Hence the observation *is always* a Poisson random variable. *The analog flutter shutter method consists in the design of an invertible flutter shutter function $\alpha(t)$.*

Definition (Analog flutter shutter function.)

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

Let α be an (*analog*) *flutter shutter function* then the acquired image at position n is (a realization of)

$$obs(x) \sim \mathcal{P} \left(\int_{-\infty}^{\infty} \alpha(t) u(n - vt) dt \right) \sim \mathcal{P} \left(\int_{-\infty}^{\infty} \frac{1}{v} \alpha\left(\frac{s}{v}\right) u(n - s) ds \right) \sim \mathcal{P} \left(\frac{1}{v} (\alpha(\frac{\cdot}{v}) * u)(n) \right)$$

where $obs(n)$ is known only for $n \in \mathbb{Z}$.

Definition 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

$$obs(n) \sim \mathcal{P} \left(\frac{1}{v} (\alpha(\frac{\cdot}{v}) * u)(n) \right). \quad (42)$$

We call *analog flutter shutter* its band limited interpolate

$$obs(x) \sim \sum_{n \in \mathbb{Z}} obs(n) sinc(x - n).$$

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

$$\mathbb{E}(obs(x)) = \frac{1}{v} (\alpha(\frac{\cdot}{v}) * u)(x) \quad (43)$$

and

$$var(obs(x)) = \frac{1}{v} (\alpha(\frac{\cdot}{v}) * u)(x). \quad (44)$$

Proof. Directly from the *analog flutter shutter samples* definition (42). \square

Remark 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*

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

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

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

and the variance is

$$var(\mathcal{F}(\mathbf{u}_{est,ana})(\xi)) = \frac{\|\alpha\|_{L^1}\|u\|_{L^1}}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi). \quad (45)$$

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

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

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

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

Therefore,

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

□

4.3 Motion Invariant Photography

We shall see that the *motion-invariant photography* method proposed in [21] is equivalent to a *flutter shutter* method using a specific (non piecewise-constant) gain-function $\alpha_{MIP} = \mathbb{1}_{[0,a(N\Delta t)^2]} \frac{1}{2\sqrt{at}}$, where a is the acceleration and $N\Delta t$ is the exposure time. In the case of the *motion-invariant photography* method the gain remains constant equal to one. *The landscape is moving at a constant acceleration*, meaning that the apparent induced velocity is $v(t) = a.t + v$. More precisely, the landscape is moving at a constant speed v and the camera is moving in the opposite direction with a constant acceleration. Thus, with the formalism proposed in the former sections the

observed value is a realization of

$$obs(x) \sim \mathcal{P} \left(\int_0^{N \cdot \Delta t} u(x - v \cdot t + t \cdot v(t)) dt \right) \sim \mathcal{P} \left(\int_0^{N \cdot \Delta t} u(x - at^2) dt \right) \quad (46)$$

$$\sim \mathcal{P} \left(\int_0^{a(N \cdot \Delta t)^2} \frac{1}{2\sqrt{at}} u(x - t) dt \right) \sim \mathcal{P} \left((\mathbb{1}_{[0, a(N \cdot \Delta t)^2]}(t) \frac{1}{2\sqrt{at}} * u)(x) \right). \quad (47)$$

Thus the *motion-invariant photography* method is equivalent to the use of an *analog flutter shutter* method.

Corollary 4.5. *The Motion Invariant Photography (MIP) inherits of the same limitation as the analog flutter shutter concerning the SNR.*

Proof. A consequence of (46-47 and section 4.2) is that the *motion-invariant photography* is an analog flutter shutter with a flutter function of the form of $\alpha_{MIP} = \mathbb{1}_{[0, a(N \cdot \Delta t)^2]} \frac{1}{2\sqrt{at}}$. \square

4.4 Piecewise constant analog flutter shutter

The goal of this subsection is to give an easy comparison between analog and numerical flutter shutter in term of SNR. The comparison is to be done for positive piecewise constant flutter-shutter functions to satisfy the requirements of both numerical (section 4.1) and analog (section 4.2) flutter shutters.

Theorem 4.6. (Piecewise constant analog flutter shutter.) *Let $\alpha(t)$ be a piecewise constant analog flutter shutter function. Then the analog flutter shutter samples are*

$$obs(n) \sim \mathcal{P} \left(\frac{1}{v} \left(\alpha \left(\frac{\cdot}{v \Delta t} \right) * u \right) (n) \right). \quad (48)$$

Proof. Similarly to (26) let $\alpha(t) = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1[}(t)$ be a piecewise constant analog flutter shutter function where $\alpha_k \in [0, 1]$ is the gain used on the time interval $[k, k + 1[$. Then the acquired image at position n is (a realization of)

$$\begin{aligned} obs(n) &\sim \mathcal{P} \left(\sum_{k=0}^{N-1} \alpha_k \int_{k\Delta t}^{(k+1)\Delta t} u(n - vt) dt \right) \sim \mathcal{P} \left(\sum_{k=0}^{N-1} \alpha_k \int_k^{(k+1)} \Delta t u(n - v\Delta t s) ds \right) \\ &\sim \mathcal{P} \left(\int_0^N \Delta t \alpha(s) u(n - v\Delta t s) ds \right) \sim \mathcal{P} \left(\int_0^{Nv\Delta t} \frac{1}{v} \alpha \left(\frac{s}{v\Delta t} \right) u(n - s) ds \right) \sim \mathcal{P} \left(\frac{1}{v} \left(\alpha \left(\frac{\cdot}{v\Delta t} \right) * u \right) (n) \right) \end{aligned}$$

\square

Corollary 4.7. *The observed samples of the analog flutter shutter are such that, for $n \in \mathbb{Z}$*

$$\mathbb{E}(\text{obs}(x)) = \frac{1}{v}(\alpha(\frac{\cdot}{v\Delta t}) * u)(x)$$

and

$$\text{var}(\text{obs}(x)) = \frac{1}{v}(\alpha(\frac{\cdot}{v\Delta t}) * u)(x).$$

Proof. Directly from (48). \square

The only significant difference is illustrated in the next lemma.

Lemma 4.8. *The variance of the analog flutter shutter observation (44) is larger or equal to the variance of the numerical flutter shutter observation, (31), with equality when $\forall k \alpha_k = 0$ or 1.*

Proof. 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 $(\alpha(\frac{\cdot}{v}) * u)(x) \geq (\alpha^2(\frac{\cdot}{v}) * u)(x)$ (because $u \geq 0$). Using (44) and (31) concludes the proof. \square

Since the expected value of (43) is equal to the expected value of the numerical flutter shutter (30) the inverse filter is equal to the inverse filter of the numerical flutter shutter (34).

Theorem 4.9. *Let α a positive piecewise constant flutter shutter function then the analog flutter shutter method has a spectral SNR equal to*

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

which is smaller or equal to the spectral SNR of the numerical flutter shutter ,

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

Proof. Since the inverse filter is the inverse filter defined by (34) we have

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

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

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

Therefore,

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

From (45) and using Lemma 4.8 we obtain

$$var(\mathcal{F}(\mathbf{u}_{est,ana})(\xi)) \geq \frac{\sum_{n \in \mathbb{Z}} \frac{1}{v} (\alpha^2(\frac{\cdot}{v\Delta t}) * u)(n)}{\Delta t^2 |\hat{\alpha}|^2(\xi v \Delta t)} \mathbb{1}_{[-\pi, \pi]}(\xi) = var(\mathcal{F}(\mathbf{u}_{est,num})(\xi)).$$

Taking into account that the expected values of *numerical* and *analog flutter shutter* are equal (30,43) yields

$$SNR^{spectral}(\mathcal{F}(\mathbf{u}_{est,ana}))(\xi) \leq SNR^{spectral}(\mathcal{F}(\mathbf{u}_{est,num}))(\xi).$$

□

This also means that the variance of the estimated landscape (45) using an *analog flutter shutter* method $var(\mathcal{F}(\mathbf{u}_{est,ana}))$ is larger or equal to the variance of $var(\mathcal{F}(\mathbf{u}_{est,num}))$ (39) using a *numerical flutter shutter* method when α is positive and piecewise constant. Notice that (45) is not invariant by changing α into $t\alpha$ for $t \neq 0$ (rescaling) anymore. This means that the optimization of α will be easier with the *numerical Flutter-Shutter*.

5 Equivalence of piece-wise constant and continuous flutter shutter functions

As stated in section (4) the *numerical flutter shutter* can deal with piecewise constant positive or negative gains. An *analog flutter shutter* can be used with continuous (at least theoretically) but positive gains. It may seem at first sight that neither is more flexible than the other. Indeed piecewise constant functions are easier to implement practically on the camera. On the other hand there is no reason for the optimal gain function to be neither piecewise constant nor positive. However we shall see a way to deal with continuous function with a *numerical flutter shutter* and compare the resulting SNR.

Definition (Equivalent flutter shutter function.)

Consider a *numerical flutter shutter* code $(\alpha_k)_{k=0, \dots, N-1}$ and an *analog flutter shutter function* $\beta(t)$. We say that the *numerical flutter shutter* function using $\alpha(t) = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1[}(t)$ is equivalent to the *analog flutter shutter* function using $\beta(t)$ if their observed samples have the same expected value, namely if $\mathbb{E}(obs_\alpha(n)) = \mathbb{E}(obs_\beta(n))$ for $n \in \mathbb{Z}$.

In other words two flutter-shutter functions are said equivalent if the observed values are equal up to the noise. This is a rather necessary definition. Indeed the landscape u being band limited, from (30, 43) we deduce that there are many different flutter shutter functions leading to the “same” observation. This does not mean that the acquisition strategies are equivalent as their variance may differ.

Theorem 5.1. (*Bridge between coded and continuous flutter shutter functions*)

Let $\alpha(t)$ be a piecewise constant gain function, then there exists $\beta(t)$ a continuous gain function such that the two flutter shutter strategies using $\alpha(t)$ and $\beta(t)$ are equivalent.

Similarly from a continuous gain function $\beta(t)$ we are able to construct $\alpha(t)$ a piecewise constant gain function such that the two flutter shutter strategies are equivalent.

Proof. Step1. Consider a landscape $u(x - vt)$ moving at velocity v and α a numerical flutter shutter function $\alpha = \sum_{k \in \mathbb{Z}} a_k \mathbb{1}_{[k-\frac{1}{2}, k+\frac{1}{2}[}$ be a piecewise constant function (with $\hat{\alpha}(\xi) \neq 0$ for $\xi \in [-\pi v \Delta t, \pi v \Delta t]$ for obvious reason) then from (30) we have

$$\mathbb{E}(\hat{obs}(\xi)) = \Delta t \hat{\alpha}(\xi v \Delta t) \hat{u}(\xi) = \Delta t \hat{\alpha}(\xi v \Delta t) \mathbb{1}_{[-\pi, \pi]}(\xi) \hat{u}(\xi) = \hat{\beta}(\xi v) \hat{u}(\xi).$$

The previous relations emphasizes the fact that the observable landscape u is band-limited with \hat{u} supported by $[-\pi, \pi]$. Consequently, no matter what $\hat{\alpha}(\xi v \Delta t)$ is equal to outside $[-\pi, \pi]$. The observed remains unchanged. In other words the *apparent* gain control function is equal to $\beta = \Delta t \mathcal{F}^{-1} \left(\hat{\alpha}(\xi \Delta t) \mathbb{1}_{[-\pi, \pi]} \left(\frac{\xi}{v} \right) \right)$, which is C^∞ .

Step2. Reciprocally consider a landscape $u(x - vt)$ moving at velocity v and $\beta(t)$ be a C^0 (or C^∞) function. The goal is here to construct $\alpha(t)$ a piecewise constant function such that (from 30, 43) : $\hat{\alpha}(\xi v \Delta t) = \frac{\hat{\beta}(v \xi)}{\Delta t}$ for $\forall \xi \in \mathbb{1}_{[-\pi, \pi]}$. Let

$$\Delta t = \frac{1}{|v|} \text{ and } a_k = \frac{1}{\Delta t} \int_{-\pi}^{\pi} \frac{\hat{\beta}(vs)}{\text{sinc}(\frac{sv \Delta t}{2\pi}) e^{-\frac{is}{2}}} e^{iks} ds \quad (49)$$

Moreover let $\alpha = \sum_{k \in \mathbb{Z}} a_k \mathbb{1}_{[k-\frac{1}{2}, k+\frac{1}{2}[}$. Let $\xi \in [-\pi, \pi]$ then from (41) we

have

$$\begin{aligned}
\hat{\alpha}(\xi v \Delta t) &= \text{sinc}\left(\frac{\xi}{2\pi}\right) e^{\frac{-i\xi}{2}} \sum_{k \in \mathbb{Z}} a_k e^{-ik\xi} \quad (\text{from (49) } v\Delta t = 1) \quad (50) \\
&= \text{sinc}\left(\frac{\xi}{2\pi}\right) e^{\frac{-i\xi}{2}} \sum_{k \in \mathbb{Z}} \frac{1}{\Delta t} \left(\int_{-\pi}^{\pi} \frac{\hat{\beta}(vs)}{\text{sinc}\left(\frac{s}{2\pi}\right) e^{\frac{-is}{2}}} e^{iks} ds \right) e^{-ik\xi} \\
&= \frac{1}{\Delta t} \text{sinc}\left(\frac{\xi}{2\pi}\right) e^{\frac{-i\xi}{2}} \int_{-\pi}^{\pi} \left(\frac{\hat{\beta}(vs)}{\text{sinc}\left(\frac{s}{2\pi}\right) e^{\frac{-is}{2}}} \sum_{k \in \mathbb{Z}} e^{ik(t-\xi)} \right) ds \\
&= \frac{1}{\Delta t} \text{sinc}\left(\frac{\xi}{2\pi}\right) e^{\frac{-i\xi}{2}} \int_{-\pi}^{\pi} \frac{\hat{\beta}(vs)}{\text{sinc}\left(\frac{s}{2\pi}\right) e^{\frac{-is}{2}}} \delta_0(t-\xi) ds = \frac{\hat{\beta}(v\xi)}{\Delta t}. \quad (51)
\end{aligned}$$

Such $\alpha(t)$ being piecewise constant by construction can be implemented with a numerical flutter shutter. Then (50- 51) implies from (30, 43) that the analog flutter shutter using β is equivalent to the numerical flutter shutter using the constructed α . In other words for any continuous flutter shutter function there exists an implementable equivalent numerical flutter function. \square

Notice that since we choose $\Delta t = \frac{1}{|v|}$ (49) the requirement (41) namely $|v|\Delta t \leq 2$ is reached. No relation between the spatial sampling of u and the temporal sampling of α can be stated except $|v|\Delta t \leq 2$. Thus in the case of an implementation using the lines of a *time delay integration* device where Δt is imposed, the drift between any line and the next one must be smaller than two pixels.

In Thm. 4.9 we compared the analog and numerical flutter shutter for positive piecewise constant functions. The left to be done to conclude is to compare when $\alpha(t)$ is an analog flutter shutter function. Being an analog flutter shutter function $\alpha(t)$ is not piecewise constant in general. Hence we shall compare both strategies using the framework of Thm. 5.1.

Theorem 5.2. (Comparative efficiency of numerical and analog flutter shutter)

Let β be an analog flutter shutter function and α its equivalent numerical flutter shutter function. Then the spectral SNR of the analog flutter shutter using β

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

is smaller or equal to the spectral SNR of the numerical flutter shutter using α ,

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

Proof. From Thm. 5.1 let $a_k = \frac{1}{\Delta t} \int_{-\pi}^{\pi} \frac{\hat{\beta}(vs)}{\text{sinc}(\frac{sv\Delta t}{2\pi})e^{-is}} e^{iks} ds$, and $\alpha = \sum_{k \in \mathbb{Z}} a_k \mathbb{1}_{[k-\frac{1}{2}, k+\frac{1}{2}]}$.

Thus by construction the expected value of the analog flutter shutter $\mathbb{E}(\hat{obs}_{ana}(\xi))$ using β is equal to $\mathbb{E}(\hat{obs}_{num}(\xi))$ the expected value of its equivalent numerical flutter shutter, namely the numerical flutter shutter using α . This imply that the filters to deconvolve and get \mathbf{u}_{est} in both case are equal so are $\mathbb{E}(\mathbf{u}_{est, \hat{ana}}(\xi))$ and $\mathbb{E}(\mathbf{u}_{est, \hat{num}}(\xi))$. Thus to estimate the SNR the only left is to compare the variances, namely (45)

$$\text{var}(\mathbf{u}_{est, \hat{ana}}(\xi)) = \frac{\|\beta\|_{L^1} \|u\|_{L^1}}{|\hat{\beta}|^2(\xi v)} \mathbb{1}_{[-\pi, \pi]}(\xi)$$

and (39)

$$\text{var}(\mathbf{u}_{est, \hat{num}}(\xi)) = \frac{\|\alpha\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi)}{\Delta t |\hat{\alpha}(\xi v \Delta t)|^2}. \quad (52)$$

By construction of α we have (50, 51)

$$\Delta t |\hat{\alpha}(\xi v \Delta t)|^2 = \frac{|\hat{\beta}(\xi v)|^2}{\Delta t}.$$

Which yields to (52)

$$\text{var}(\mathbf{u}_{est, \hat{num}}(\xi)) = \frac{\Delta t \|\alpha\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi)}{|\hat{\beta}(\xi v)|^2}.$$

Here we shall compare $\|\beta\|_{L^1}$ and $\Delta t \|\alpha\|_{L^2}^2$ but thanks to the construction of α we have $\Delta t = \frac{1}{v}$. Then (49) entails

$$\begin{aligned} \Delta t \|\alpha\|_{L^2}^2 &= \frac{1}{2\pi v} \|\hat{\alpha}\|_{L^2}^2 \\ &= \frac{1}{2\pi v} \int |\hat{\alpha}(\xi)|^2 d\xi = \frac{1}{2\pi v} \int |\hat{\alpha}(\xi v \Delta t)|^2 d\xi \text{ by (49)} \\ &= \frac{1}{2\pi v} \int \left| \frac{\hat{\beta}(\xi v)}{\Delta t} \right|^2 d\xi \text{ by construction of } \alpha \text{ (50- 51)} \\ &= \frac{v^2}{2\pi v} \int |\hat{\beta}(\xi v)|^2 d\xi \text{ by (49) again} \\ &= \frac{v}{2\pi} \int |\hat{\beta}(\xi v)|^2 d\xi = \frac{1}{2\pi} \int |\hat{\beta}(\xi)|^2 d\xi = \|\beta\|_{L^2}^2 \end{aligned}$$

Then the only remaining is to compare $\|\beta\|_{L^1}$ and $\|\beta\|_{L^2}^2$. β being an analog flutter shutter function we have $\beta(t) \in [0, 1]$ implying $\beta(t) = |\beta(t)| \geq |\beta(t)|^2$. Thus we have $\|\beta\|_{L^1} \geq \|\beta\|_{L^2}^2$. Thus

$$\text{var}(\mathbf{u}_{est, \hat{num}}(\xi)) \leq \text{var}(\mathbf{u}_{est, \hat{ana}}(\xi)).$$

which conclude the proof (8). □

6 The *flutter shutter* paradox theorem

Theorem 6.1. (The ideal *flutter shutter* function)

Consider a landscape $u(x - vt)$ moving at velocity v . Then the ideal continuous gain control function is equal to $\alpha^*(t) = \text{sinc}(tv\Delta t)$.

Proof. Among all gain control functions $\alpha(t)$ one giving the best $\text{SNR}^{\text{spectral-averaged}}$ (9) is given by minimizing the averaged variance of \hat{u}_{est} (39)

$$\begin{aligned} F(\alpha) &= \frac{\|\alpha\|_2^2}{2\pi\Delta t} \int_{-\pi}^{\pi} \frac{d\xi}{|\hat{\alpha}(\xi v\Delta t)|^2} \quad (\text{dropping the irrelevant constant in } u) \\ &= \frac{\|\alpha\|_2^2}{\Delta t} \int_{-\pi v\Delta t}^{\pi v\Delta t} \frac{1}{|\hat{\alpha}(\xi)|^2} \frac{d\xi}{2\pi v\Delta t} \geq \frac{\|\alpha\|_2^2}{\Delta t} \frac{1}{\int_{-\pi v\Delta t}^{\pi v\Delta t} \frac{d\xi}{2\pi v\Delta t}}. \end{aligned}$$

The crucial point here is the use of the Jensen formula. Notice that since $x > 0 \mapsto \frac{1}{x}$ is strictly convex the equality occurs when $|\hat{\alpha}(\xi)|^2 = \mathbb{1}_{[-\pi v\Delta t, \pi v\Delta t]}(\xi)$ up to a meaningless constant renormalization for a numerical flutter shutter implementation (see 4.1.1). This leads to $\alpha^*(t) = \text{sinc}(tv\Delta t)$ (up to a normalization constant). \square

Being non positive this ideal gain control function is *not* implementable using an *analog Flutter-Shutter*, and being non piecewise-constant 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 Thm. 5.1.

Corollary 6.2. (Upper bound on the SNR)

Consider a landscape $u(x - vt)$ moving at velocity v . The ideal numerical flutter shutter strategy using $\alpha^*(t) = \text{sinc}(tv\Delta t)$ has a spectral SNR (8) equal to

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

Moreover the averaged spectral SNR (9) is equal to

$$\text{SNR}^{\text{spectral-averaged}} = \frac{1}{2\pi\sqrt{v}} \frac{\int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\|u\|_{L^1}}}.$$

Proof. From Thm. 6.1 and using the Parseval formula we deduce that $\|\alpha^*\|_{L^2}^2 = v\Delta t$. Then from Thm. 4.2 we deduce that $\text{SNR}^{\text{spectral}}(\xi) = \frac{\mathbb{1}_{[-\pi, \pi]}(\xi)}{\sqrt{v}} \frac{|\hat{u}(\xi)|}{\sqrt{\|u\|_{L^1}}}$. Moreover from (39) we have

$$\begin{aligned} \int_{-\pi}^{\pi} \text{var}(\mathcal{F}(\mathbf{u}_{est, num})(\xi)) d\xi &= \int_{-\pi}^{\pi} \frac{\|\alpha^*\|_{L^2}^2 \|u\|_{L^1}}{\Delta t |\hat{\alpha}^*(\xi v\Delta t)|^2} d\xi = \int_{-\pi v\Delta t}^{\pi v\Delta t} \frac{v\Delta t \|u\|_{L^1}}{v\Delta t^2 |\alpha^*(\xi)|^2} d\xi \\ &= \int_{-\pi v\Delta t}^{\pi v\Delta t} \frac{v\Delta t \|u\|_{L^1}}{v\Delta t^2} d\xi = 2\pi v \|u\|_{L^1}. \end{aligned}$$

Then $\frac{1}{2\pi} \int_{-\pi}^{\pi} \text{var}(\mathcal{F}(\mathbf{u}_{est,num})(\xi)) d\xi = v \|u\|_{L^1}$ and $SNR^{\text{spectral-averaged}} = \frac{1}{2\pi\sqrt{v}} \frac{\int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\|u\|_{L^1}}}$. \square

Corollary 6.3. (The flutter shutter paradox)

The use of a flutter shutter strategy increasing the total time-exposure does not permits to achieve an arbitrary SNR.

Consider a landscape $u(x-vt)$ moving at velocity v . Then the $SNR^{\text{spectral-averaged}}$ of the flutter shutter strategy is bounded independently of the total exposure time $N\Delta t$. In other words increasing the time-exposure has a limited effect on the SNR.

Proof. This is a direct consequence of Cor. 6.2 since the measure of the support of α^* which is the exposure time is infinite, but the SNR of the restored is nevertheless finite. Notice that both analog and numerical flutter shutter inherit this behavior. \square

Corollary 6.4. (Efficiency of the numerical flutter shutter)

Consider a landscape $u(x-vt)$ moving at velocity v . Then the ratio R of $SNR^{\text{spectral-averaged}}$ between the ideal flutter shutter and the best snapshot with exposure time equal to Δt^* is equal to $R = \frac{SNR^{\text{spectral-averaged}}(\text{flutter,ideal})}{SNR^{\text{spectral-averaged}}(\text{snapshot})} \approx$

$$\sqrt{\frac{1.0909 \times 2\pi}{\int_{-\pi}^{\pi} \frac{d\xi}{\left| \frac{\sin\left(\frac{\xi v \Delta t^*}{2}\right)}{\frac{\xi v \Delta t^*}{2}} \right|^2}} \approx 1.1715.$$

Proof. This is a direct consequence of Thm. 3.4 and Cor. 6.2. \square

Those negative results are the *flutter shutter paradox* namely none of those flutter shutter strategies are efficient.

7 Experiments

The purpose of this section is to compare experimentally different acquisition strategies : snapshot, *flutter shutter* using Raskar’s code [1], a random uniform over $[-1,1]$ code and the sinc code. All strategies are compared using the RMSE, the contrast invariant RMSE ($RMSE^{CI}$) and the visual image quality. A benchmark of acquisition strategies is given (Fig. 12) to illustrate the *flutter shutter* paradox theorem 6.3.

7.1 Contrast invariant image quality measure

In many cases reconstruction errors inherent to a method can be quantified using the Root-Mean-Squared-Error (RMSE) (**xxx**). However, when a

	flutter function $\alpha(t)$	SNR
Accumulation	$\mathbb{1}_{[0,T]}(t), v = 0$	(spatial) $\sqrt{u(x)T}$
Best snapshot	$\mathbb{1}_{[0, \frac{1.0909}{v}]}(t)$	(spectral) $\mathbb{1}_{[-\pi, \pi]}(\xi) \hat{u}(\xi) \sqrt{\frac{1.0909}{v \ u\ _{L^1}}} 2 \frac{\sin(\frac{1.0909}{2v})}{1.0909}$
Raskar	$\sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1]}(t), a_k \in \{0, 1\}$	(spectral) $\mathbb{1}_{[-\pi, \pi]}(\xi) \frac{\sqrt{\Delta t} \hat{\alpha}(\xi v \Delta t) }{\ \alpha\ _{L^1}} \frac{ \hat{u}(\xi) }{\sqrt{\ u\ _{L^1}}}$
Analog flutter	$\alpha(t) \geq 0$	(spectral) $\mathbb{1}_{[-\pi, \pi]}(\xi) \frac{ \hat{\alpha}(\xi v) }{\ \alpha\ _{L^1}} \frac{ \hat{u}(\xi) }{\sqrt{\ u\ _{L^1}}}$
Motion invariant photography	$\alpha(t) = \frac{1}{\sqrt{t}} \mathbb{1}_{[0, T]}(t)$	(spectral) $\mathbb{1}_{[-\pi, \pi]}(\xi) \frac{ \hat{\alpha}(\xi v) }{\ \alpha\ _{L^1}} \frac{ \hat{u}(\xi) }{\sqrt{\ u\ _{L^1}}}$
Numerical flutter	$\sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1]}(t), a_k \in \mathbb{R}$	(spectral) $\mathbb{1}_{[-\pi, \pi]}(\xi) \frac{\sqrt{\Delta t} \hat{\alpha}(\xi v \Delta t) }{\ \alpha\ _{L^2}} \frac{ \hat{u}(\xi) }{\sqrt{\ u\ _{L^1}}}$
Ideal flutter shutter	$\text{sinc}(tv\Delta t)$	(spectral) $\frac{\mathbb{1}_{[-\pi, \pi]}(\xi)}{\sqrt{v}} \frac{ \hat{u}(\xi) }{\sqrt{\ u\ _{L^1}}}$

Figure 2: This summarize the results on the different flutter shutter strategies and their SNR. The best flutter shutter is a *sinc* and has a finite SNR but use an infinite exposure time. This is the flutter shutter paradox.

small contrast change occurs between the original and the processed image, the RMSE can become substantial, while the images remain perceptually indistinguishable. For example take an image $I(m, n)$ defined over a sub-domain of \mathbb{Z}^2 , and another one $J = I + 10$. Then $RMSE(I, J) = 10$ is large but does not reflect the quality of the reconstruction J . Comparatively a convolution with for example a Gaussian can give a smaller RMSE while making considerable damage. This bias is avoided by normalizing the images before computing the RMSE. The principle of the normalization is that two images related to each other by a contrast change are perceptually equivalent. Their distance should reflect this fact and be zero. The midway equalization [11] is best suited for that purpose, because it equalizes the image histogram to a “midway” histogram depending on both images. By the midway operation both images undergo a minimal distortion and adopt exactly the same histogram. Thus we shall define the contrast invariant RMSE ($RMSE^{CI}$) by

$$RMSE^{CI} = RMSE(I_{mid(I, J)}, J_{mid(I, J)})$$

where $mid(I, J)(= mid(J, I))$ is the midway histogram between I and J . $I_{mid(I, J)}$ is the image I specified on the $mid(I, J)$ histogram (having an histogram equal to $mid(J, I)$) and $J_{mid(I, J)}$ is J specified on $mid(I, J)$. In all experiments results will be compared by comparing their visual quality and quantitatively using the $RMSE$ (**xxx**).

7.2 Numerical flutter shutter simulations

7.2.1 Numerical implementation

From (33, 31) the acquired image can be simulated using the following scheme (without loss of generality, for simplicity we choose $\Delta t = 1$ and $v = 1$ in the experiments):

1. Simulation of the observed
 - (a) Take a sharp image
 - (b) Convolve with the gain control function α , (26). This $I1$ image is a simulation of the expected value of the acquired one. Notice this can be done in Fourier domain using (41)
 - (c) Convolve with the squared gain control function α^2 (31). As previously this can be done using (41). This $I2$ image (map) is a simulation of the noise variance
 - (d) The deterministic part of the observed is $(\frac{1}{v}u * \alpha(\frac{\cdot}{v})) - (\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))$. The stochastic part is a Poisson random variable with intensity $\lambda(x) = (\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))(x)$.
 - (e) The simulated observed value at pixel x is obtained by simulating the random variable $obs(x) \sim (\frac{1}{v}u * \alpha(\frac{\cdot}{v}))(x) - (\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))(x) + Poisson((\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))(x))$ with the desired SNR
 - (f) Crop the result omitting the last N columns to avoid unrealistic periodization effects
2. Restoration
 - (a) Use classic mirror-extension in the direction of the blur
 - (b) Divide the Fourier transform of I by the Fourier transform of the gain control function α , (26)
 - (c) Crop the result to avoid border effects

Notice that the sharp snapshot can be simulated using a “degenerated” gain control function defined from (26) and using $\alpha_0 = 1$ and $\alpha_k = 0 \forall k \geq 1$.

7.3 Analog flutter shutter simulations

7.3.1 Numerical implementation

From (43,44) the acquired image can be simulated using the following scheme (without loss of generality and for simplicity we choose again $\Delta t = 1$ and $v = 1$):

1. Simulation of the observed

- (a) Take a sharp image
- (b) Simulate the observed pixel value at pixel x by simulating a Poisson random variable with intensity $\lambda(x) = \left(\frac{1}{v}u * \alpha(\frac{\cdot}{v})\right)(x)$ with the desired SNR
- (c) Crop the result omitting the last N columns to avoid unrealistic periodization effects

2. Restoration

- (a) Use classic mirror-extension in the direction of the blur
- (b) Divide the Fourier transform of I by the Fourier transform of the gain controlled function (26)
- (c) Crop the result to avoid border effects

The snapshot is simulated using the scheme described in (7.2.1).

7.4 Examples

The test image is the "boat" (Fig. 3) image. Four acquisition strategies are simulated and evaluated : snapshot, *flutter shutter* using Raskar's code [1], a random uniform over $[-1, 1]$ code and the sinc code. For the four strategies the observed image is obtained by simulating the underlying Poisson random variable (Figs. 4,6,8,10), for a normalized velocity $v = 1$ and a SNR of 100 for the level 100. In other words each elementary image of the *numerical flutter shutter* acquisition process (and therefore each elementary image of an *analog flutter shutter* with gain 1) has a SNR equal to 100 for the level 100. Details of *flutter shutter* codes including the modulus of their Fourier transform are given (Fig. 13). The RMSE is calculated on truncated images (omitting the last N columns) to avoid border effects (Figs. 5, 7, 9, 11) which are negligible for big enough images (depending on the ratio of the blur support b over the image support). A benchmark of acquisition strategies is given (12) to illustrate the *flutter shutter* paradox theorem 6.3.

7.4.1 Snapshot

7.4.2 Using Raskar's code

Raskar's code given in [1] is a binary sequence of length 52 defined by (1 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 0 0 0 0 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 1 1 1 0 0 1 0 0 1 1 0 0 1 1 1) found by extensive binary research. Since it is binary the *numerical* and *analog flutter shutter* methods are here equivalent. The acquired image presents the (Fig. 6) stroboscope like pattern resulting from both the motion and the code. Notice that the restored image (Fig. 7) has a bigger RMSE (2.30) than the restored image (Fig. 5) using a snapshot (1.38). This illustrates the *flutter shutter* paradox (Thm. 6.3).



Figure 3: Test-image.



Figure 4: Acquired image (snapshot), the (log) modulus its Fourier transform .

7.4.3 Using a random code

The following code of length 52 was generated randomly (we picked the first one thus generated) from a uniform distribution over $[-1, 1]$. The obtained code was (0.5491, -0.4903, -0.6919, -0.5125, -0.9079, 0.3560, -



Figure 5: Restored image (snapshot) ($RMSE = 1.38$), and the residual noise (contrast adjusted to $[0, 255]$). Notice that there is almost no boundary effect, the support of the code being small.

0.6357, 0.0275, 0.9568, 0.8670, 0.4087, 0.4097, 0.8659, 0.3344, 0.9198, -0.8389, 0.7476, 0.9808, 0.1366, -0.7247, 0.9320, 0.8069, -0.5848, 0.9493, -0.1682, 0.9533, -0.2173, -0.5834, 0.7483, 0.4785, 0.2266, 0.9764, -0.4708, 0.6723, 0.8312, 0.0084, 0.6892, -0.5245, -0.8651, 0.3417, 0.5183, 0.1317, -0.1301, 0.3432, -0.3262, 0.4367, -0.9771, -0.2120, 0.1160, 0.6648, 0.9446, 0.0590).

Notice that this sequence is not *positive*, thus cannot be implemented with an *analog flutter shutter* method. The acquired image (Fig. 8) seems a bit more degraded than with Raskar's code. Nevertheless the RMSE (2.09) of the restored image (Fig. 9) is slightly better than the RMSE (2.30) of the restored using Raskar's code (Fig. 7).

7.4.4 Using The sinc code

The sinc code is (0.0002, -0.0002, 0.0002, -0.0003, 0.0003, -0.0003, 0.0003, -0.0004, 0.0004, -0.0005, 0.0005, -0.0006, 0.0007, -0.0008, 0.0009, -0.0011, 0.0014, -0.0017, 0.0021, -0.0027, 0.0037, -0.0053, 0.0082, -0.0141, 0.0296, -0.0917, 1.0000, -0.0917, 0.0296, -0.0141, 0.0082, -0.0053, 0.0037, -0.0027, 0.0021, -0.0017, 0.0014, -0.0011, 0.0009, -0.0008, 0.0007, -0.0006, 0.0005, -0.0005, 0.0004, -0.0004, 0.0003, -0.0003, 0.0003, -0.0003, 0.0002, -0.0002).

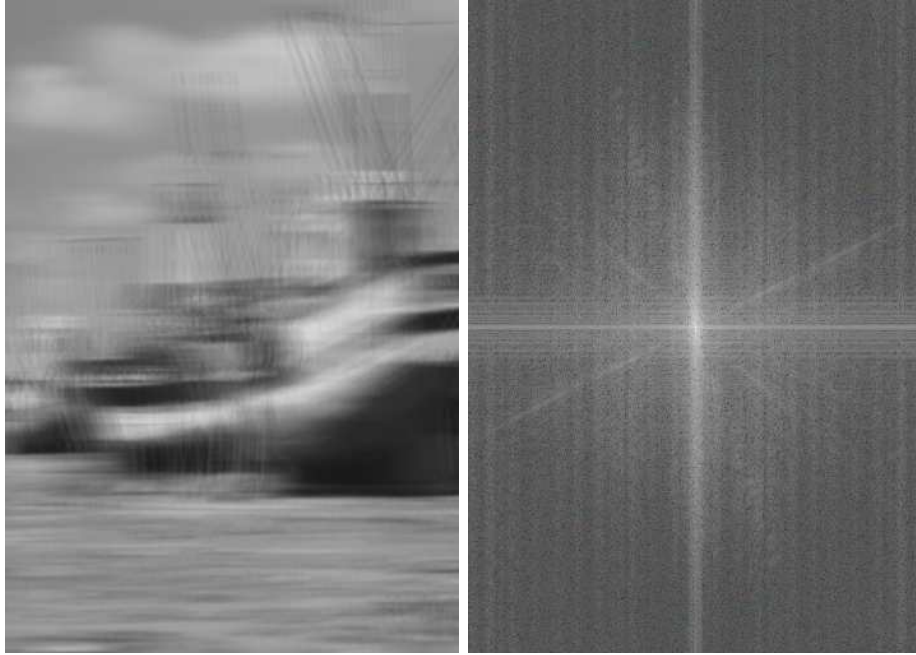


Figure 6: Acquired image with Raskar’s code and the (log) modulus of its Fourier transform. Notice the vertical pattern trace of the *flutter shutter* code.

This is the best L2 approximation of the ideal gain function namely, given v , the sinc with cutoff frequency equal to πv (given here for $|v| = 1$). The acquired image presents the (Fig. 6) stroboscope like pattern resulting from both the motion and the code. Notice that the restored image (Fig. 11) has a RMSE (1.42) slightly better than the restored image (Fig. 5) using a snapshot (1.44).

8 Why is the *flutter shutter* still interesting ?

When the motion is perfectly known then the best flutter shutter function is a sinc as proved in Thm. 6.1. However as stated in Cor. 6.4 the use of this sinc function is not a breakthrough compared to the best snapshot since the SNR is increased by small factor. Nevertheless, if the motion is not well-known then without some *flutter shutter* method the only way to ensure a sharp image for a broad range of velocities v is to use a very small exposure time leading to a poor SNR. It is then possible to achieve better results *on average*, and to ensure anyway always a sharp image. To optimize the SNR gain, however, we must know the probability distribution of the velocities. Fixing the norm of the code and maximizing its DFT as done in



Figure 7: Restored image using the Raskar’s code ($RMSE = 2.30$, $RMSE^{CI} = 2.32$), and the residual noise (contrast adjusted to $[0, 255]$). Notice the boundary effects which depend on the blur support. In practice if the image is big enough this should be negligible.

[16, 3, 2] does not lead to optimal codes in a strong sense.

8.1 Computation of the optimal $\alpha(t)$ function

Let $\rho(v)$ be a probability density for $v \in \mathbb{R}$ (which is equivalent to a probability density over the blur length), such that $\rho(v) = 0, \forall |v| > v_{max}$. Notice that without the knowledge of the maximum speed v_{max} it is virtually impossible to guarantee $|v|\Delta t < 2$ (see section 3.2.1) for v feasible (in the support of ρ). Eventually ρ can be estimated during a phase of calibration taking a bunch of images without *flutter shutter* and using them to compute the actual blur density function by tracking zeros in the DFT of the observed for instance.

We want to discover α the best flutter function (the best code), designed to obtain the minimal noise for the restored image. The noise level is measured by the variance of the restored, leading to the best $SNR^{spectral-average}$ thus from (39) as

$$\int \frac{\|\alpha\|_{L^2}^2 \|u\|_{L^1} \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi}{\Delta t |\hat{\alpha}(\xi v \Delta t)|^2} d\xi \frac{\|u\|_{L^1} \|\alpha\|_2^2}{\Delta t} \int \frac{\mathbb{1}_{[-\pi, \pi]}(\xi) d\xi}{|\hat{\alpha}|^2(\xi v \Delta t)} = \frac{\|u\|_{L^1} \|\alpha\|_2^2}{\Delta t} \int \frac{\mathbb{1}_{[-\pi v \Delta t, \pi v \Delta t]}(\xi) d\xi}{v \Delta t |\hat{\alpha}|^2(\xi)}.$$

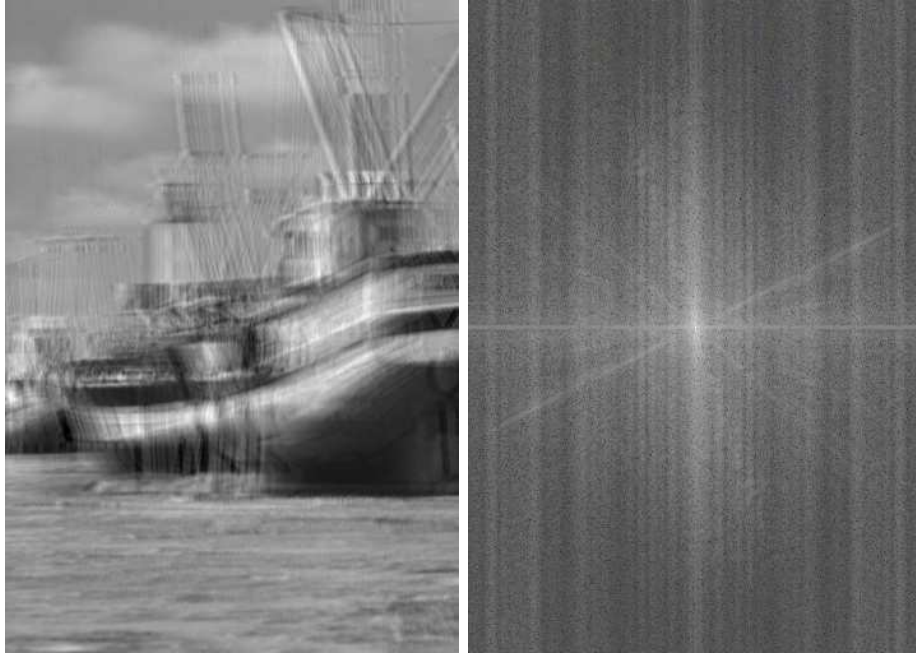


Figure 8: Acquired image with a random code and the (log) modulus of its Fourier transform. Notice the vertical pattern trace of the *numerical flutter shutter* code.

Thus, dropping the multiplicative constants we posit the following energy to be minimized for a fixed v ,

$$E_v(\hat{\alpha}) = \|\hat{\alpha}\|_2^2 \int \frac{\mathbb{1}_{[-\pi v \Delta t, \pi v \Delta t]}(\xi) d\xi}{|v| |\hat{\alpha}|^2(\xi)}.$$

As stated previously if v is known the optimization of E_v is easy. Notice that for $\rho(v) = \delta_{v_0}$ the best code is *not* a snapshot. The *flutter shutter* compensate to avoid de-whitening which yields to a *sinc*-like code which concentrate the support (over $[-\pi v \Delta t, \pi v \Delta t]$). The real challenge is to find the bests a_k coefficients when the motion v is known up to a density of probability. Remark that this is the case in many situations, as for example

1. When the scene is not planar, the speed of objects will change according to their distance;
2. For camera shakes (with non-constant and or unknown amplitudes);
3. In Raskar's examples : the background is static, the observed vehicle moves.



Figure 9: Restored image from the random code ($RMSE = 2.09$, $RMSE_{CI} = 2.09$), and the residual noise (contrast adjusted to $[0, 255]$). Notice the boundary effects, which depend on the blur support. In practice if the image is large enough this should be negligible.

Given $\rho(v)$ the α function leading to the best possible SNR on average is given by minimizing

$$\begin{aligned}
E(\hat{\alpha}) &= \int_{\mathbb{R}} E_v(\alpha) \rho(v) dv = \int_{\mathbb{R}} \|\hat{\alpha}\|_2^2 \int_{-\infty}^{\infty} \frac{\mathbb{1}_{[-\pi v \Delta t, \pi v \Delta t]}(\xi) d\xi}{|v| |\hat{\alpha}|^2(\xi)} \rho(v) dv \\
&= \|\hat{\alpha}\|_2^2 \int_{\mathbb{R}} \int_{-\infty}^{\infty} \frac{\mathbb{1}_{[-\pi v \Delta t, \pi v \Delta t]}(\xi)}{|v| |\hat{\alpha}|^2(\xi)} \rho(v) dv d\xi \quad (\text{Fubini}) \\
&= \|\hat{\alpha}\|_2^2 \int_{-\infty}^{\infty} \frac{1}{|\hat{\alpha}|^2(\xi)} \int_{\mathbb{R}} \frac{\rho(v) \mathbb{1}_{[-v \Delta t \pi, v \Delta t \pi]}(\xi)}{|v|} dv d\xi \\
&= \|\hat{\alpha}\|_2^2 \int_{-\infty}^{\infty} \frac{1}{|\hat{\alpha}|^2(\xi)} \left(\int_{\mathbb{R}} \frac{\rho(v) \mathbb{1}_{[-v \Delta t \pi, v \Delta t \pi]}(\xi)}{|v|} dv \right) d\xi. \tag{53}
\end{aligned}$$

Then from (53) defining

$$w(\xi) = \int_{\mathbb{R}} \frac{\rho(v) \mathbb{1}_{[-v \Delta t \pi, v \Delta t \pi]}(\xi)}{|v|} dv \tag{54}$$

the minimization reduces to

$$E(\hat{\alpha}) = \|\hat{\alpha}\|_2^2 \int_{-\infty}^{\infty} \frac{w(\xi)}{|\hat{\alpha}|^2(\xi)} d\xi.$$



Figure 10: Acquired image with the sinc code and the (log) modulus of its Fourier transform. For a band limited image, the convolution a small enough sinc function is equivalent to a convolution against a dirac. This explains that almost no blur can be seen.

Notice that $w(\xi) = 0$ for $|\xi| \geq v_{max}\Delta t\pi$, and that

$$\int_{\mathbb{R}} |w(\xi)| d\xi = \int_{\mathbb{R}} 2v\Delta t\pi \frac{\rho(v)}{v} dv = 2\pi\Delta t.$$

Thus w is in $L^1(\mathbb{R})$ is compactly supported, so that $\sqrt[4]{w} \in L^1 \cap L^2(\mathbb{R})$.

Theorem 8.1. *The optimal functions α^* minimizing $E(\hat{\alpha})$ satisfy $|\hat{\alpha}^*| = \sqrt[4]{w}$ on the support of w . If we extend $\hat{\alpha}^*$ by zero outside this support, we obtain bounded continuous codes belonging to $L^2(\mathbb{R})$.*

Proof. The easy to compute unknown is $f(\xi) := |\alpha|^2(\xi)$. All integrals are on \mathbb{R} and we shall omit the ξ variable in the integrals. We have $E(f) = (\int f) \int \frac{w(\xi)}{f(\xi)} d\xi$. Thus its weak differential in the direction of every bounded and compactly supported perturbation g satisfies

$$E'(f)(g) = \left(\int g\right) \int \frac{w}{f} - \left(\int f\right) \int \frac{w}{f^2} g.$$



Figure 11: Restored image using the sinc code ($RMSE = 1.42$, $RMSE^{CI} = 2.32$), and the residual noise (contrast adjusted to $[0, 255]$). Despite the deconvolution the noise remains almost a white noise since the Fourier transform of the sinc code is nearly constant.

Acquisition strategy	Snapshot	Raskar's code	Random code	Sinc code
$RMSE$	1.44	2.550	2.16	1.42
$RMSE^{CI}$	1.44	2.76	2.49	1.44

Figure 12: Quantitative ($RMSE$) comparison of different strategies for fixed velocity $v = 1$ (so the blur support is of 52 pixels, except for the snapshot). The best $RMSE$ is obtained using a the sinc code beating only slightly the snapshot as predicted *flutter shutter* paradox (Thm. 6.3).

A minimal f therefore satisfies for every bounded compactly supported g $E'(f)(g) = 0$ which yields

$$\int g \left[\left(\int \frac{w}{f} \right) - \left(\int f \right) \frac{w}{f^2} \right] = 0$$

and therefore $f = C\sqrt{w}$. Being nonnegative this solution is admissible and we deduce $|\hat{\alpha}(\xi)| = \sqrt[4]{w}$. as announced. It remains to show that this is our solution. To this aim, we notice that the minimization of $E(f)$ is equivalent to minimizing $F(f) := \int \frac{w(\xi)}{f(\xi)}$ under the constraints $\int f = 1$, $f \geq 0$, $f = 0$

outside the support of w . But we can replace these constraints by $\int f \leq 1$, $f \geq 0$, $f = 0$ outside the support of w , since if $\int f < 1$ by picking $\tilde{f} := \frac{f}{\int f}$, we get back $\int \tilde{f} = 1$ and obviously $F(\tilde{f}) < F(f)$. Thus our problem is equivalent to minimizing the strictly convex energy F over the convex set $\int f \leq 1$, $f \geq 0$, $f = 0$ outside the support of w . The solution of this problem, if it exists, is therefore unique. On the other hand the function $f_0 := \frac{\sqrt{w}}{\int \sqrt{w}}$ is a critical point for this problem, and therefore is this unique solution.

Nevertheless, this calculation only fixes the modulus of $\hat{\alpha}$. Since w is symmetric with respect to the origin, so is $|\hat{\alpha}|$ and there are therefore many optimal real codes α . □

Here the only thing left is, given N , to compute $\alpha_k \forall k \in \{0, \dots, N-1\}$ coefficients approximating the above solution.

Lemma 8.2. *The function $\alpha = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k, k+1[}$ is, using (26, 41) the best L^2 approximation of $\hat{\alpha}$ using a piecewise constant function, when the a_k are given (up to a constant normalization constant) by*

$$a_k = \int \frac{\xi}{2\sin(\frac{\xi}{2})} \mathcal{F}(\alpha^*)(\xi) e^{i\xi \frac{2k+1}{2}} d\xi = \int \frac{\xi}{2\sin(\frac{\xi}{2})} \sqrt[4]{w}(\xi) e^{i\xi \frac{2k+1}{2}} d\xi.$$

Nevertheless, there is no guarantee that $\hat{\alpha}(\xi) \neq 0$ for $\xi \in [-\pi, \pi]$.

8.2 A “reverse engineering” of Raskar’s code

Given a code $(\alpha_0, \dots, \alpha_{N-1}) \in \mathbb{R}^N$ we are able to estimate its underlying velocity probability distribution. The algorithm is the following:

1. compute $\hat{\alpha}$ using (26)
2. estimate w the weight (54) function by $\tilde{w}(\xi) = |\hat{\alpha}(\xi)|^4$
3. Now $\rho(x) \approx -x\tilde{w}(\pi\Delta tx)$

Indeed from (54) we have by differentiating,

$$\begin{aligned} w'(\xi) &= \int_{\mathbb{R}} \frac{\rho(v)\delta_{-\pi v\Delta t}(\xi) - \delta_{\pi v\Delta t}(\xi)}{v} dv \\ &= \frac{-\pi\Delta t}{\xi} \rho\left(\frac{-\pi\Delta t}{\xi}\right) - \frac{\pi\Delta t}{\xi} \rho\left(\frac{\pi\Delta t}{\xi}\right) \end{aligned}$$

Hence

$$w'(\pi\Delta tx) = \frac{-2}{x} \rho(x), \quad x \geq 0 \text{ assuming } \rho \text{ symmetric.}$$

Applying this scheme to Raskar’s code gives (using a normalized $\Delta t = 1$) the probability density of Fig 14. This distribution can be understood in two different ways. It either means that there is a high probability that the scene is steady and then otherwise uniformly distributed motions occur on a certain interval of velocities. But this is an unlikely model for a camera motion. Instead, it is a quite natural model given the kind of motions considered in the Raskar et al. codes. Indeed the velocity histogram of Fig. 14 considers that there is a fixed background with a large area and then some moving objects whose velocity probability distribution is indeed close to uniform in a broad interval. All proposed codes [2, 26, 1] are actually optimal - contrarily to several statements - but for non understood -and probably unlikely- camera or objects motion.

9 Experiments on optimized codes

The goal of this section is to explore numerically different natural models of motion, give the corresponding optimized codes and to compare the efficiency to the snapshot. The considered models of motion are centered Gaussians and uniforms. We begin (Figs. 16,17 & 19) by showing how the codes are changing according to the probability densities they have been optimized for (Gaussian and uniform for various standard deviations and ranges). Then we illustrate, for a fixed code, how the quality of the restored image is evolving with various blur supports. A comparison is made with the Raskar code given in [2] and “brute-force” optimization by a random search among binary sequences.

We shall predict the impact on the noise of the *flutter shutter* analyzing (39) for the *numerical* and (45) for the *analog* case. Thus the pessimistic case (guaranteeing an upper bound on the SNR) is given by the noise boost factor defined from (Thm. 4.2) by

$$\min_{\xi \in [-\pi, \pi]} \left(\frac{\|\alpha\|_{L^2}}{|\hat{\alpha}(\xi v)|} \right)$$

for the *numerical* and from (4.4) by

$$\max_{\xi \in [-\pi, \pi]} \left(\frac{\|\alpha\|_{L^1}}{|\hat{\alpha}(\xi v)|} \right)$$

for the *analog* case.

The normalized noise boost factor reflect the fact that even for a small exposure time the resulting window shaped (standard blur) kernel doesn’t have a constant Fourier transform. Thus to ease the comparison between the different methods we shall normalize the noise boost factor using the corresponding snapshot ($\alpha_0 = (1, 0, \dots, 0)$ with the same N) leading to the

following definitions

$$\min_{\xi \in [-\pi, \pi]} \left(\frac{\|\alpha\|_{L^2} |\hat{\alpha}_0(\xi v)|}{|\hat{\alpha}(\xi v)|} \right)$$

for the *numerical* and by

$$\max_{\xi \in [-\pi, \pi]} \left(\frac{\|\alpha\|_{L^1} |\hat{\alpha}_0(\xi v)|}{|\hat{\alpha}(\xi v)|} \right)$$

for the *analog* case.

In the following worst case scenarios will be predicted using the normalized version of the noise boost factor. Notice that a normalized noise boost factor smaller than one means that the noise is actually reduced (compared to the snapshot).

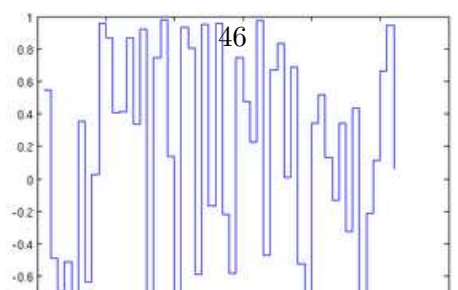
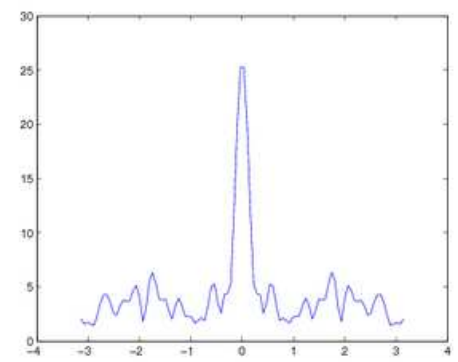
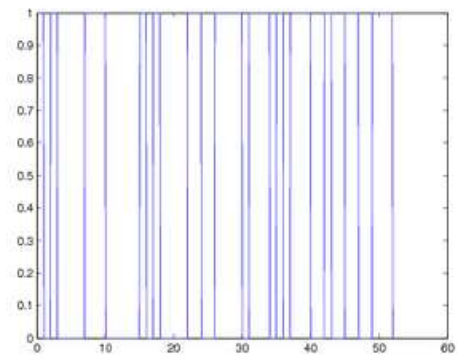
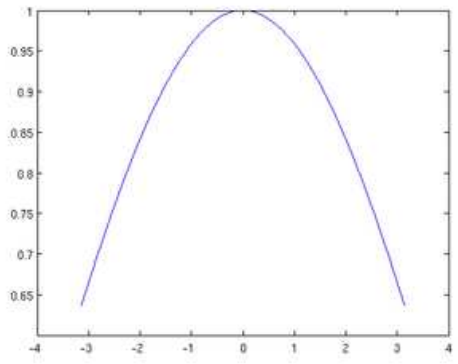
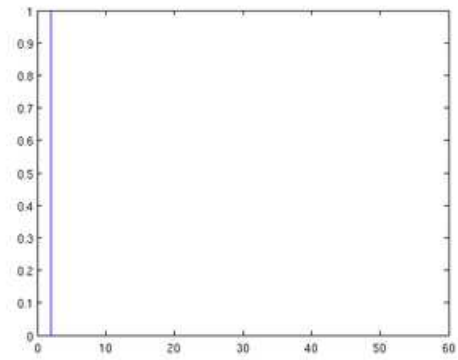
A summary of the impact on the noise is given both on a worst case noise boost factor (Figs. 21, 22, 23, 24)- and on simulations using RMSE (Figs. 35 & 36) for Gaussianly distributed motion in case of both *numerical* and *analog Flutter-Shutter*. The same is made in made in (Figs. 25, 26, 27, 28) and in (Figs. 37 & 38) for the RMSE in the case of uniformly distributed motion. Simulation of a *flutter shutter* acquisition device are shown (Figs. 29, 30, 31, 32, 33, 34).

9.1 Codes associated with various probability densities ρ

To ease the comparison with Raskar's code all experiments are made with $N = 52$ which is the length of the Raskar's code. All following codes are the best possible choices for α_k approximating the ideal α^* (defined in 8.1).

9.1.1 Gaussian distribution over the velocity

Optimal codes are shown assuming a centered Gaussian density -for several standard deviation (σ)- *hypotheses* on the velocity. Notice that the weight-function w (defined in 54) being concave, one can deduce that the modulus of the Fourier transform of the *ideal* gain-control function (8.1) is also concave. Thus all optimized codes (Figs. 16 & 17) inherit this low-pass behavior - more or less peaked at the null frequency- depending on σ , the standard deviation of the (centered) Gaussian assumed for the optimization. In the following σ is measured in pixels. Optimized codes are given explicitly in (Fig. 18). To summarize the results, if increasing the exposure time by a factor of N leads to a Gaussian motion blur with $\approx \frac{N}{12}$ standard deviation, (only) then the SNR can be increased by an ≈ 3 factor using a *numerical Flutter-Shutter*.



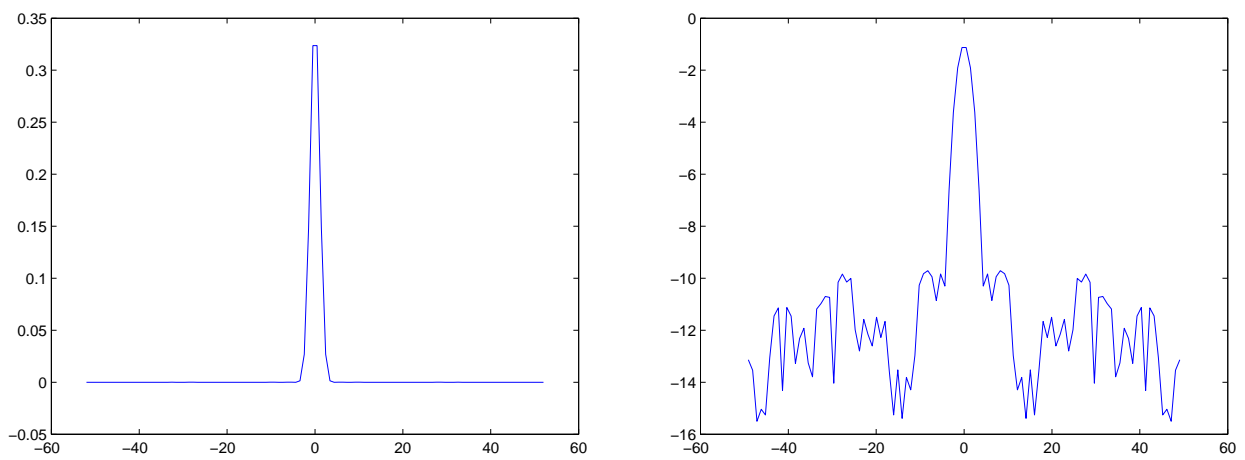


Figure 14: On the left : The density of probability associated with the Raskar et al code : x-axis (signed) motion (in pixels), y-axis probability. On the right : with a logarithmic scale, notice that the probability is nonzero even for big blurs (velocity). It corresponds to a steady background and some fast moving objects in a broad velocity interval. This is precisely the caption model used in the experiments: a fixed camera (a steady background) and some moving vehicles at an unknown speed.

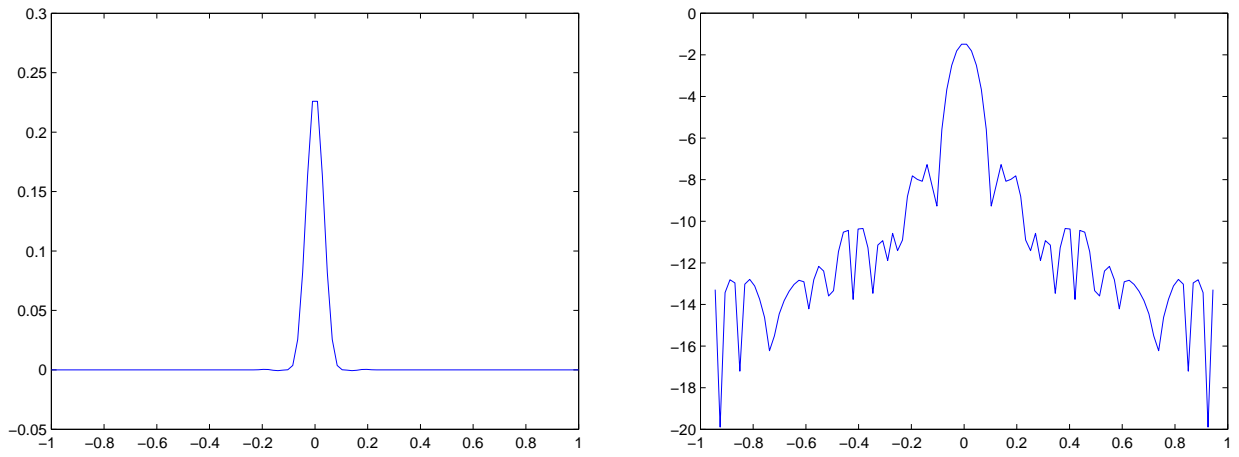
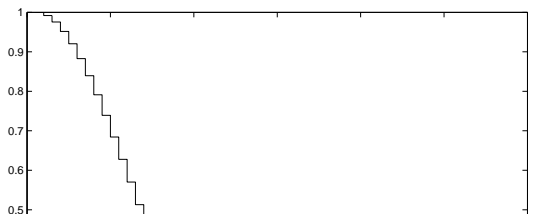
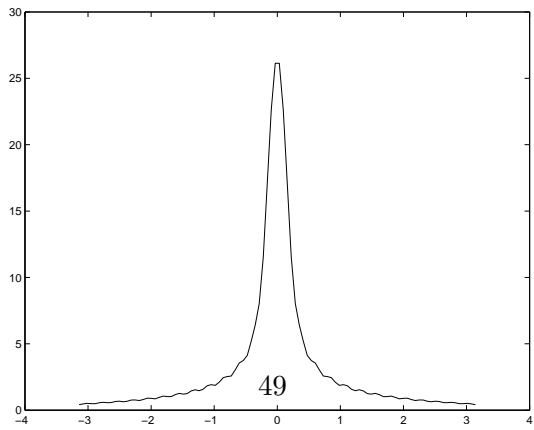
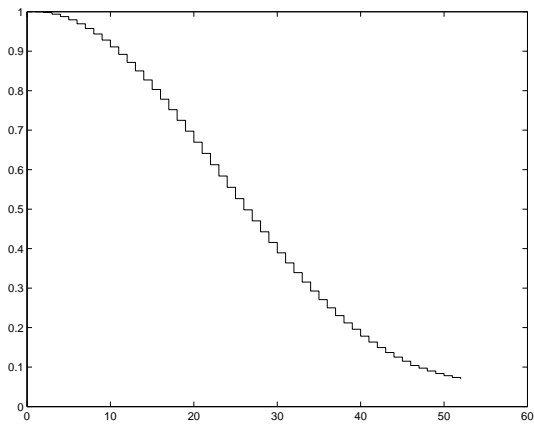
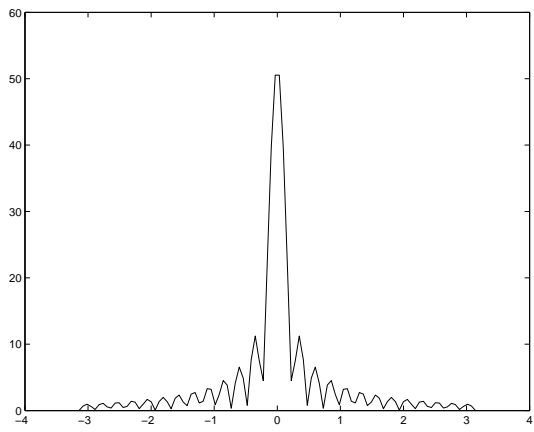
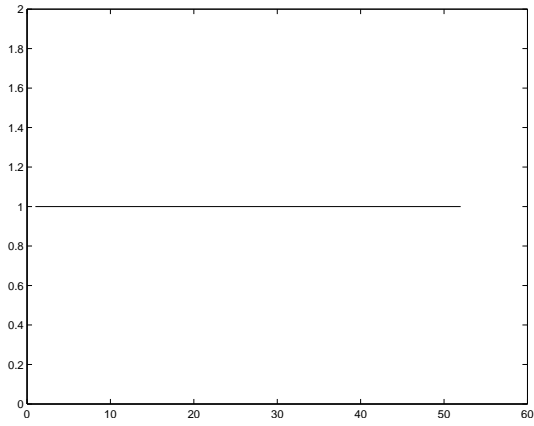


Figure 15: On the left : The density of probability associated with the Raskar et al code : x-axis (signed) motion (in pixels), y-axis probability. On the right : with a logarithmic scale, notice that the probability is nonzero even for big blurs (velocity). It corresponds to an attempts to optimize both the SNR and the PSF estimation.



Notice that the optimal codes coming from a Gaussian probability distribution on the motions are positive. Thus we can explicitly compare the *analog* and *numerical flutter-shutter*.

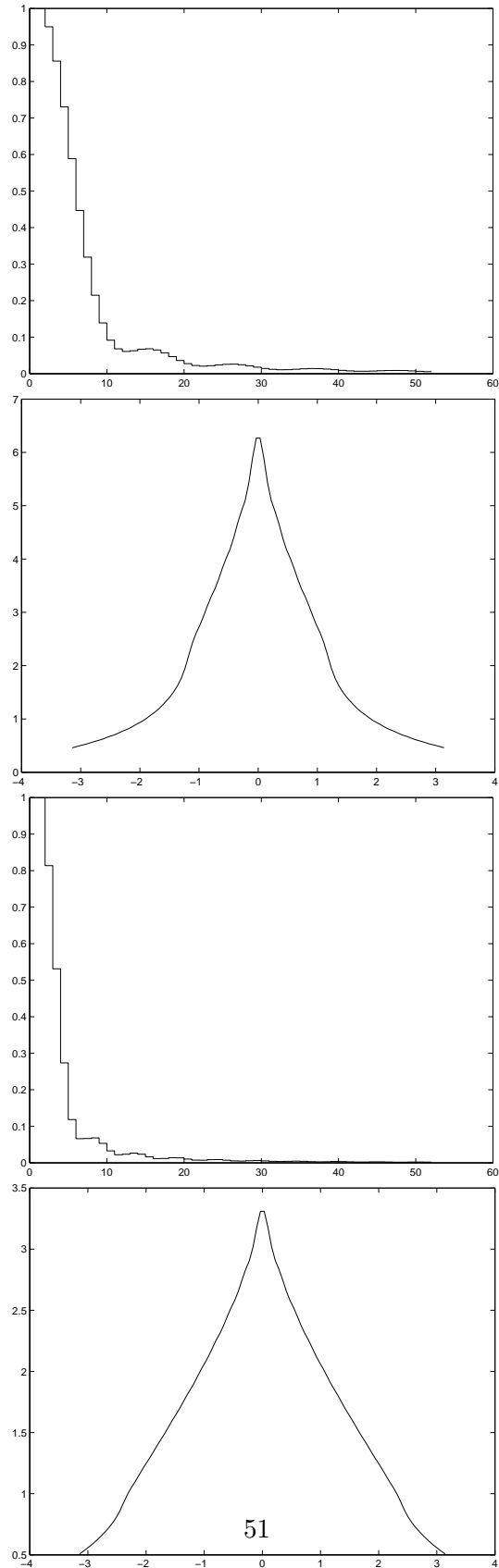


Figure 17: Codes obtained assuming a Gaussian density over the velocities for several standard deviations (σ), the blur being measured in pixels. On the left from top to bottom: the code for $\sigma = 5, \sigma = 10$ (in pixels). For Gaussian velocities the blur length (support) was cropped to $b_{max} := 4\sigma$ thus $v_{max} = \frac{b_{max}}{N}$ at the optimization step. On the right the corresponding

9.1.2 Uniform distribution

For several ranges optimal codes have been computed using a uniform *hypothesis* on the motion velocities. The weight function w (54) being concave, we shall deduce that the modulus of the Fourier transform of the *ideal* gain control function (8.1) is also concave. Thus all optimized codes (Fig. 19) inherit this low pass behavior (with a peak at the zero frequency) depending on the range assumed on the motion. Codes are given explicitly in (Fig. 20). To summarize, if increasing the exposure time by a N factor leads to a uniform motion over $[0, \frac{N}{10}]$, then the SNR can be increased by an ≈ 3 factor using a *numerical Flutter-Shutter*.

9.2 Comparison of acquisition strategies

This comparison is made with the only alternative choice : a standard snapshot. Actually without *flutter shutter* the only way to guarantee a sharp image is to take a standard, “quick enough to be sharp” snapshot. This method has a nearly constant SNR for all blur supports (all v such that $v\Delta t < 2$). Now, because of the *flutter shutter* paradox for each $v \in \text{support}(\rho)$ the code is never better than the corresponding snapshot using the adapted exposure time. Nevertheless, compared to the very restrictive case when the snapshot is fixed to v_{max} (which is the only way to ensure a sharp image in all cases without a *flutter shutter*) we shall observe better results, *on average*. Of course the SNR will depend on the blur support. Actually the better SNRs we shall observe depend on the shape of ρ , the assumed probability density of velocities. Roughly speaking the idea is to accept more risks, to expect a better SNR on average. The optimization step guarantees the smallest risk for average SNR. Necessarily, the optimization concentrates the gains on the more probable motions while things are made worse for unlikely motions. We can't expect to increase all results, the *flutter shutter* paradox indicating that the worst case optimal code is realized by classic photography.

9.3 Simulation of acquired images

Let $(\alpha_0, \dots, \alpha_{N-1})$ be a code found using (8.1). We need to simulate the obtained images for several blur supports (v) but using the *same* code to illustrate the increase in SNR and the effect of the optimization on the restored image.

9.3.1 Numerical implementation

From (33, 31) the acquired image can be simulated using the following scheme (without loss of generality, for simplicity we choose $\Delta t = 1$ and $v = 1$ in the experiments):

1. Simulation of the observed

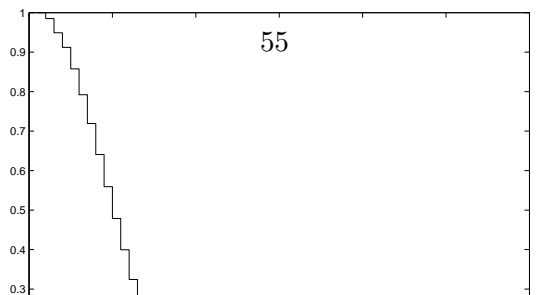
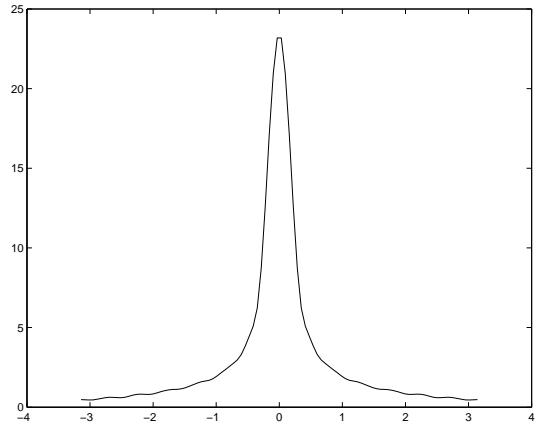
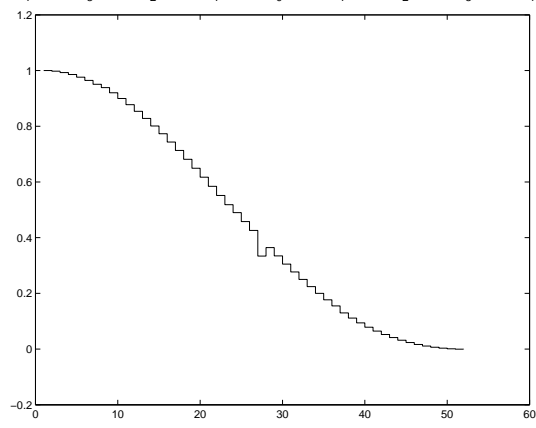
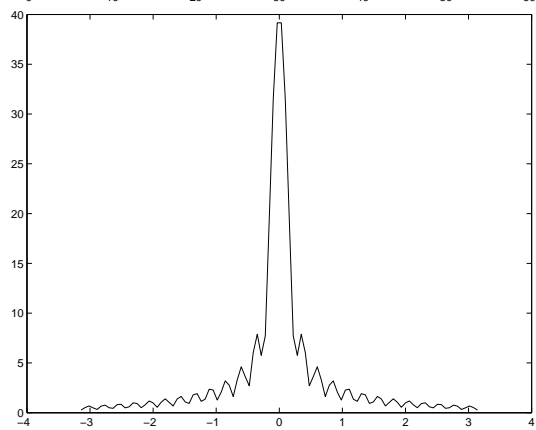
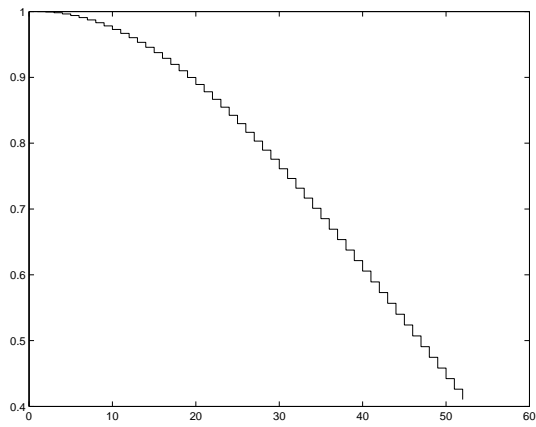
- (a) Take a sharp image
- (b) Convolve with the gain control function α , (26). This $I1$ image is a simulation of the expected value of the acquired one. Notice this can be done in Fourier domain using (41)
- (c) Convolve with the squared gain control function α^2 (31). As previously this can be done using (41). This $I2$ image (map) is a simulation of the noise variance
- (d) The deterministic part of the observed is $(\frac{1}{v}u * \alpha(\frac{\cdot}{v})) - (\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))$. The stochastic part is a Poisson random variable with intensity $\lambda(x) = (\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))(x)$.
- (e) The simulated observed value at pixel x is obtained by simulating the random variable $obs(x) \sim (\frac{1}{v}u * \alpha(\frac{\cdot}{v}))(x) - (\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))(x) + Poisson((\frac{1}{v}u * \alpha^2(\frac{\cdot}{v}))(x))$ with the desired SNR
- (f) Crop the result omitting the last N columns to avoid unrealistic periodization effects

2. Restoration

- (a) Use classic mirror-extension in the direction of the blur
- (b) Divide the Fourier transform of I by the Fourier transform of the gain control function α , (26)
- (c) Crop the result to avoid border effects

Notice that the sharp snapshot can be simulated using a “degenerated” gain control function defined from (26) and using $\alpha_0 = 1$ and $\alpha_k = 0 \forall k \geq 1$.

9.3.2 Examples



$\mathcal{U}_{[0,1]}$	(1.0000 0.9994 0.9982 0.9964 0.9939 0.9909 0.9873 0.9831 0.9783 0.9729 0.9670 0.9605 0.9534 0.9459 0.9377 0.9291 0.9200 0.9103 0.9002 0.8896 0.8786 0.8671 0.8553 0.8430 0.8303 0.8173 0.8039 0.7902 0.7762 0.7619 0.7473 0.7324 0.7174 0.7021 0.6866 0.6709 0.6551 0.6392 0.6231 0.6070 0.5907 0.5745 0.5582 0.5419 0.5255 0.5093 0.4930 0.4768 0.4608 0.4448 0.4289 0.4131)
$\mathcal{U}_{[0,1]}$	(1.0000 0.9976 0.9927 0.9855 0.9759 0.9640 0.9499 0.9337 0.9154 0.8952 0.8731 0.8494 0.8241 0.7973 0.7692 0.7401 0.7099 0.6790 0.6473 0.6152 0.5828 0.5502 0.5176 0.4851 0.4529 0.4211 0.3900 0.3595 0.3298 0.3011 0.2734 0.2469 0.2215 0.1975 0.1748 0.1534 0.1335 0.1151 0.0981 0.0826 0.0685 0.0560 0.0448 0.0350 0.0266 0.0194 0.0135 0.0088 0.0052 0.0026 0.0009 0.0001)
$\mathcal{U}_{[0,1]}$	(1.0000 0.9849 0.9552 0.9121 0.8569 0.7918 0.7189 0.6406 0.5595 0.4781 0.3987 0.3235 0.2542 0.1924 0.1390 0.0946 0.0594 0.0331 0.0152 0.0047 0.0004 0.0012 0.0057 0.0126 0.0206 0.0287 0.0359 0.0417 0.0455 0.0471 0.0466 0.0441 0.0400 0.0346 0.0285 0.0221 0.0160 0.0106 0.0061 0.0028 0.0008 0.0001 0.0005 0.0019 0.0040 0.0066 0.0092 0.0118 0.0139 0.0155 0.0163 0.0164)
$\mathcal{U}_{[0,1]}$	(1.0000 0.9407 0.8306 0.6849 0.5228 0.3638 0.2250 0.1177 0.0466 0.0100 0.0008 0.0092 0.0248 0.0391 .0467 0.0457 0.0376 0.0255 0.0134 0.0045 0.0004 0.0012 0.0053 0.0106 0.0148 0.0165 0.0153 0.0118 0.0072 0.0031 0.0006 0.0002 0.0016 0.0040 0.0065 0.0081 0.0082 0.0070 0.0048 0.0025 0.0008 0.0001 0.0005 0.0017 0.0033 0.0045 0.0050 0.0047 0.0036 0.0021 0.0009 0.0001)

Figure 20: Optimized codes for an uniformly distributed blur model. Notice that the supports of the codes are decreasing with the support of the probability. Again this a consequence of the *flutter shutter* paradox (Thm. 6.3).

v	blur support (pixels)	<i>Snapshot</i>	Raskar	Rand	$\alpha_{\mathcal{N}(0,1)}$	$\alpha_{\mathcal{N}(0,2)}$	$\alpha_{\mathcal{N}(0,5)}$	$\alpha_{\mathcal{N}(0,10)}$
0	0	1	0.1961	0.6359	0.1646	0.2120	0.3136	0.4306
0.25	13	1	2.6414	1.5641	1.7006	1.1422	0.5827	0.6172
0.5	26	1	3.0608	2.5447	3.3463	2.3018	1.3571	0.8159
0.75	39	1	3.0608	3.3463	4.8680	3.4322	2.1059	1.1739
1	52	1	3.0608	3.3463	6.6639	4.4776	2.7418	1.7973
1.5	78	1	3.0608	41.1231	8.5474	5.7872	3.5916	2.4850

Figure 21: (Normalized) noise boost factor guaranteed for the *numerical flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account (see Thm.4.2)) for different codes. For noise boost factors lesser than one the noise is reduced. Notice that for $v = 1$ for all codes the boost factor is larger than or equal to one, this is a consequence of the *flutter shutter* paradox (Thm. 6.3).

	<i>Snapshot</i>	$\alpha_{\mathcal{N}(0,1)}$	$\alpha_{\mathcal{N}(0,2)}$	$\alpha_{\mathcal{N}(0,5)}$	$\alpha_{\mathcal{N}(0,10)}$
Averaged noise boost factor	1	0.1764	0.2463	0.3895	0.5483
Std-dev on the average	0	0.0183	0.0390	0.0630	0.0877

Figure 22: Averaged (according to the assumed density of probability) noise boost factor guaranteed for the *numerical flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account (see Thm.4.2)) for different codes. On the second row : the standard deviation of this averaged value.

v	blur support (pixels)	<i>Snapshot</i>	Raskar	$\alpha_{\mathcal{N}(0,1)}$	$\alpha_{\mathcal{N}(0,2)}$	$\alpha_{\mathcal{N}(0,5)}$	$\alpha_{\mathcal{N}(0,10)}$
0	0	1	0.6359	0.2077	0.2610	0.3977	0.5481
0.25	13	1	2.6414	2.0028	1.4061	0.7391	0.7857
0.5	26	1	3.0608	3.9319	2.8335	1.7212	1.0386
0.75	39	1	3.0608	5.7330	4.2251	2.6710	1.4943
1	52	1	3.0608	7.8480	5.5119	3.4775	2.2877
1.5	78	1	3.0608	10.0661	7.1214	4.5553	3.1631

Figure 23: Noise boost factor guaranteed for the *analog flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account (see lem.4.8)) for several codes. For noise boost factors lesser than one the noise is reduced. The random code being not positive is not implementable with an *analog flutter shutter* device.

	<i>Snapshot</i>	$\alpha_{\mathcal{N}(0,1)}$	$\alpha_{\mathcal{N}(0,2)}$	$\alpha_{\mathcal{N}(0,5)}$	$\alpha_{\mathcal{N}(0,10)}$
Averaged noise boost factor	1	0.2077	0.3032	0.4921	0.6979
Std-dev on the average	0	0.0216	0.0480	0.0799	0.1160

Figure 24: Averaged (according to the assumed density of probability) noise boost factor guaranteed for the *analog flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account (see Thm.4.2)) for different codes. On the second row : the standard deviation of this averaged value.

v	blur support (pixels)	<i>Snapshot</i>	Raskar	$\alpha_{\mathcal{U}[0,1]}$	$\alpha_{\mathcal{U}[0,2]}$	$\alpha_{\mathcal{U}[0,5]}$	$\alpha_{\mathcal{U}[0,10]}$
0	0	1	0.6359	0.1426	0.1771	0.2638	0.3649
0.25	13	1	2.6414	3.4811	1.5938	0.9116	0.5409
0.5	26	1	3.0608	6.8435	3.1732	1.9424	1.2650
0.75	39	1	3.0608	9.8925	4.6119	2.8933	1.9789
1	52	1	3.0608	13.6382	5.8726	3.7645	2.5729
1.5	78	1	3.0608	17.5944	7.6794	4.8722	3.4086

Figure 25: Noise boost factor guaranteed for the *numerical flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account) for several codes. For noise boost factors lesser than one the noise is reduced. The random code being not positive is not implementable with an *analog flutter shutter* device.

	<i>Snapshot</i>	$\alpha_{\mathcal{U}[0,1]}$	$\alpha_{\mathcal{U}[0,2]}$	$\alpha_{\mathcal{U}[0,5]}$	$\alpha_{\mathcal{U}[0,10]}$
Averaged noise boost factor	1	0.1473	0.1889	0.3021	1.4262
Std-dev on the average	0	0.0043	0.0108	0.0276	0.0383

Figure 26: Averaged (according to the density of probability assumed) noise boost factor guaranteed for the *numerical flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account (see Thm.4.2)) for different codes. On the second row : the standard deviation of this averaged value.

v	blur support (pixels)	<i>Snapshot</i>	Raskar	$\alpha_{\mathcal{U}[0,1]}$	$\alpha_{\mathcal{U}[0,2]}$	$\alpha_{\mathcal{U}[0,5]}$	$\alpha_{\mathcal{U}[0,10]}$
0	0	1	0.6359	0.1576	0.2064	0.3162	0.4408
0.25	13	1	2.6414	3.8477	1.8572	1.0926	0.6534
0.5	26	1	3.0608	7.5643	3.6976	2.3161	1.5281
0.75	39	1	3.0608	11.0340	5.3741	3.4678	2.3906
1	52	1	3.0608	15.0746	6.8432	4.5120	3.1081
1.5	78	1	3.0608	19.4476	8.9486	5.8397	4.1177

Figure 27: Noise boost factor guaranteed for the *analog flutter shutter* (in the worst case when the power spectrum of the image or the PSF g is not taken into account) for several codes. For noise boost factors smaller than one the noise is reduced. The random code being not positive is not implementable with an *analog flutter shutter* device.

	<i>Snapshot</i>	$\alpha_{\mathcal{U}[0,1]}$	$\alpha_{\mathcal{U}[0,2]}$	$\alpha_{\mathcal{U}[0,5]}$	$\alpha_{\mathcal{U}[0,10]}$
Averaged noise boost factor	1	0.1628	0.2201	0.3621	0.5149
Std-dev on the average	0	0.0047	0.0126	0.0331	0.0463

Figure 28: Averaged (according to the density of probability assumed) noise boost factor guaranteed for the *analog flutter shutter* (in the worst case, when the power spectrum of the image or the PSF g is not taken into account (see Thm.4.2)) for different codes. On the second row : the standard deviation of this averaged value.



Figure 29: On the left : the acquired image for $v = 1$, using the optimized code for Gaussian distributed motion with $\sigma = 1$. On the right its Fourier transform (on a logarithmic scale).



Figure 30: On the left : the restored image for $v = 1$, using the optimized code for Gaussian distributed motion with $\sigma = 1$, ($RMSE = 5.6628, RMSE_{CI} = 5.6266$). On the right : the noise (rescaled on $[0, 255]$).



Figure 31: Acquired image for $v = 0.25$, using the optimized code for Gaussian distributed motions with $\sigma = 1$. On the right its Fourier transform (on a logarithmic scale).



Figure 32: On the left : the restored image for $v = 0.25$, using the optimized code for Gaussian distributed motions with $\sigma = 1$, ($RMSE = 1.6600, RMSE_{CI} = 1.2572$). On the right : the noise (rescaled on $[0, 255]$).



Figure 33: On the left : the acquired image for $v = 0$, using the optimized code for Gaussian distributed motions with $\sigma = 1$. On the right its Fourier transform (on a logarithmic scale).



Figure 34: On the left : the restored image for $v = 0$, using the optimized code for Gaussian distributed motion with $\sigma = 1$, ($RMSE = 0.1897, RMSE_{CI} = 0.4319$). On the right : the noise (rescaled on $[0, 255]$)

9.4 Comparison with the Raskar’s code

The following results illustrate how the noise can be reduced using a *flutter shutter* method (depending on the motion probabilistic model).

v	blur support (pixels)	<i>Snapshot</i>	<i>Raskar’s</i>	Rand	$\alpha_{\mathcal{N}(0,1)}$
0	0	1.1525—1.2454	0.2259—0.4844	0.7336—0.8874	0.1897 —0.4319
0.25	13	1.1591—1.2540	1.6022—1.6635	1.1161—1.2125	1.6600 —1.2572
0.5	26	1.1972—1.2842	2.0533—2.0966	1.5275—1.5908	2.3945 —2.4258
0.75	39	1.2677—1.3493	1.9574—1.9988	1.6761—1.7221	3.8271 —3.8244
1	52	1.3822—1.4559	2.2669—2.3136	2.0834—2.0978	5.6688 —5.6266
1.5	78	1.8836—1.9341	4.0575—4.0273	4.4067—4.3047	12.2717—11.9193

v	blur support (pixels)	$\alpha_{\mathcal{N}(0,2)}$	$\alpha_{\mathcal{N}(0,5)}$	$\alpha_{\mathcal{N}(0,10)}$
0	0	0.2444—0.4960	0.3618—0.6099	0.4961—0.7065
0.25	13	0.7841—0.9245	0.5687—0.7631	0.6290—0.8070
0.5	26	1.6722—1.7311	0.9508—1.0657	0.8058—0.9476
0.75	39	2.6946—2.7185	1.5974—1.6594	1.0242—1.1298
1	52	3.9826—3.9759	2.4276—2.4580	1.5435—1.6093
1.5	78	8.7119—8.5549	5.4171—5.3805	3.6735—3.6723

Figure 35: *Numerical flutter shutter* RMSE for several Gaussian probability distributions, $N = 52$, $\Delta t = 1$ for the velocity. We denote by $\alpha_{\mathcal{N}(0,\sigma)}$ the codes optimized for a Gaussian distribution of std-dev σ . As previously mentioned the noise boost factor is quite pessimistic (see Fig 21).

10 Estimating the kernel

In a real context to perform the deconvolution we need to estimate, v the (*a priori* unknown) velocity of the motion. If the apparatus allows to build simultaneously two images, the problem can be made easier (by *storing* a classic motion blur, image without *flutter shutter*, which shall be used to estimate the blur (by tracking zeros in its Fourier transform for example) and another one with *flutter shutter* which is thereafter deconvolved).

Otherwise, we can try to estimate v from the observed image itself. This is an apparently ill posed problem, since any observed pattern could come either from the landscape or the code applied. Nevertheless, codes like Raskar’s, because of the oscillating pattern of their Fourier transform leave a distinguishable trace in the Fourier transform of the observed image (Fig. 6).

Omitting the noise term, we are observing the landscape u convolved with $\alpha_{v^*}(t) = \frac{1}{v^*}\alpha(t/v^*)$ for an unknown v^* . The oscillating pattern of the Fourier transform modulus of the observed image most likely belongs

v	blur support (pixels)	<i>Snapshot</i>	Raskar	$\alpha_{\mathcal{N}(0,1)}$
0	0	1.1525—1.2454	0.2259—0.4844	0.2233—0.4769
0.25	13	1.1591—1.2540	1.6022—1.6635	1.3663—1.4414
0.5	26	1.1972—1.2842	2.0533—2.0966	2.8235—2.8445
0.75	39	1.2677—1.3493	1.9574—1.9988	4.4963—4.4815
1	52	1.3822—1.4559	2.2669—2.3136	6.6898—6.6184
1.5	78	1.8836—1.9341	4.0575—4.0273	14.5246—14.0169

v	blur support (pixels)	$\alpha_{\mathcal{N}(0,2)}$	$\alpha_{\mathcal{N}(0,5)}$	$\alpha_{\mathcal{N}(0,10)}$
0	0	0.3007—0.5582	0.4584—0.6872	0.6317—0.8086
0.25	13	0.9361—1.0564	0.7021—0.8649	0.7951—0.9373
0.5	26	2.0512—2.0945	1.1856—1.2753	1.0060—1.1134
0.75	39	3.3129—3.3211	2.0289—2.0718	1.2932—1.3733
1	52	4.9086—4.8822	3.0678—3.0806	1.9534—1.9980
1.5	78	10.6810—10.4235	6.8663—6.7884	4.6624—4.6437

Figure 36: *Analog flutter shutter* RMSE for various probability distributions, $N = 52$, $\Delta t = 1$. Notice that the cost is $\approx +20\%$ of the noise compared to the *numerical flutter shutter* except for binary codes like Raskar’s. The random code used previously is not positive and cannot be implemented in an *analog flutter shutter* context. We denote by $\alpha_{\mathcal{N}(0,\sigma)}$ the code optimized for a centered Gaussian distribution of std-dev σ . As seen in (Fig. 21) the noise boost factor was quite pessimistic.

to the α kernel, and not to the landscape (see Fig 6). We shall use this observation to extract the α kerne. Notice that without *Flutter-Shutter*, even for a *snapshot* to perform a total deconvolution an estimation of the blur is needed and not easier.

The modulus of the observed image is $\log(|\hat{obs}(\xi)|) = \log(|\frac{1}{v^*}\hat{\alpha}(t/v^*)|(\xi)) + \log(|\hat{u}(\xi)|)$. Let us compute $\log(|\hat{obs}(\xi)|) - \log(|\frac{1}{v}\hat{\alpha}(t/v)|(\xi))$ for a set of values for v . For most of the v values, the result will present all sorts of oscillating patterns, because we are subtracting phase-shifted oscillations (see Fig 39). On the other hand for v close to v^* the oscillations cancel (see Fig 39). Thus for wrong v $\log(|\hat{obs}(\xi)|)$ will have a large total variation norm. Enforcing the total variation norm to be finite using a convolution with a small Gaussian (which also perform a denoising) leads to the following definition of v_{est} estimator of v

$$v_{est} = \operatorname{argmin}_{v \in \operatorname{support}(\rho)} \|\nabla \left(\log(|\hat{obs}|) - \log\left(|\frac{1}{v}\hat{\alpha}(t/v)|\right) \right) (\xi)\|_{L^1}.$$

10.1 Numerical implementation

1. For each line on the observed image in the direction of the blur:

v	blur support (pixels)	<i>Snapshot</i>	Raskar	Rand	$\alpha_{\mathcal{U}[0,1]}$
0	0	1.1525—1.2454	0.2259—0.4844	0.7336—0.8874	0.1645—0.4192
0.25	13	1.1591—1.2540	1.6022—1.6635	1.1161—1.2125	1.7480—1.8029
0.5	26	1.1972—1.2842	2.0533—2.0966	1.5275—1.5908	3.4470—3.4507
0.75	39	1.2677—1.3493	1.9574—1.9988	1.6761—1.7221	5.3906—5.3579
1	52	1.3822—1.4559	2.2669—2.3136	2.0834—2.0978	8.1403—8.0122
1.5	78	1.8836—1.9341	4.0575—4.0273	4.4067—4.3047	17.7451—16.9643

v	blur support (pixels)	$\alpha_{\mathcal{U}[0,2]}$	$\alpha_{\mathcal{U}[0,5]}$	$\alpha_{\mathcal{U}[0,10]}$
0	0	0.2043—0.4556	0.3039—0.5613	0.4213—0.589
0.25	13	1.0882—1.1863	0.6425—0.8145	0.5621—0.7583
0.5	26	2.2641—2.3023	1.3902—1.4642	0.8860—1.0141
0.75	39	3.6157—3.6181	2.2559—2.2925	1.5022—1.5688
1	52	5.3427—5.3086	3.3456—3.3537	2.2849—2.3201
1.5	78	11.6713—11.3558	7.3583—7.2592	5.1182—5.0867

Figure 37: *Numerical flutter shutter* RMSE for several probability distributions, $N = 52$, $\Delta t = 1$. $\alpha_{\mathcal{U}[0,x]}$ denotes the codes optimized for an uniform distribution over $[0, x]$. As stated previously the noise boost factor is quite pessimistic.

v	blur support (pixels)	<i>Snapshot</i>	Raskar	$\alpha_{\mathcal{U}[0,1]}$
0	0	1.1525—1.2454	0.2259—0.4844	0.1814—0.4277
0.25	13	1.1591—1.2540	1.6022—1.6635	1.9373—1.9835
0.5	26	1.1972—1.2842	2.0533—2.0966	3.8119—3.8110
0.75	39	1.2677—1.3493	1.9574—1.9988	5.9764—5.9269
1	52	1.3822—1.4559	2.2669—2.3136	8.9632—8.8031
1.5	78	1.8836—1.9341	4.0575—4.0273	19.6137—18.6535

v	blur support (pixels)	$\alpha_{\mathcal{U}[0,2]}$	$\alpha_{\mathcal{U}[0,5]}$	$\alpha_{\mathcal{U}[0,10]}$
0	0	0.2367—0.4951	0.3645—0.6156	0.5091—0.7188
0.25	13	1.2643—1.3481	0.7476—0.8977	0.6646—0.8327
0.5	26	2.6311—2.6537	1.6468—1.7057	1.1573—0.6646
0.75	39	4.2456—4.2355	2.6937—2.7175	1.8069—1.8595
1	52	6.2489—6.1826	4.0332—4.0249	2.7650—2.7869
1.5	78	13.6281—13.1823	8.7709—8.6144	6.1573—6.0979

Figure 38: *Analog flutter shutter* RMSE for different distribution of probability, $N = 52$, $\Delta t = 1$. Notice that the cost is $\approx +20\%$ of noise compared to the *numerical flutter shutter* except for binary codes like Raskar's. The random code used previously being not positive cannot be implemented in an *analog flutter shutter* context.

- (a) Compute $\|\nabla ((\log(|\hat{obs}|) - \log(|\frac{1}{v}\hat{\alpha}(t/v)|)) * G)(\xi)\|_{L^1}$, for different $v \in \text{support}(\rho)$;
 - (b) keep the one that minimizes the previous;
2. average the estimator of each line to get v_{est} .

10.2 Example : Raskar's code

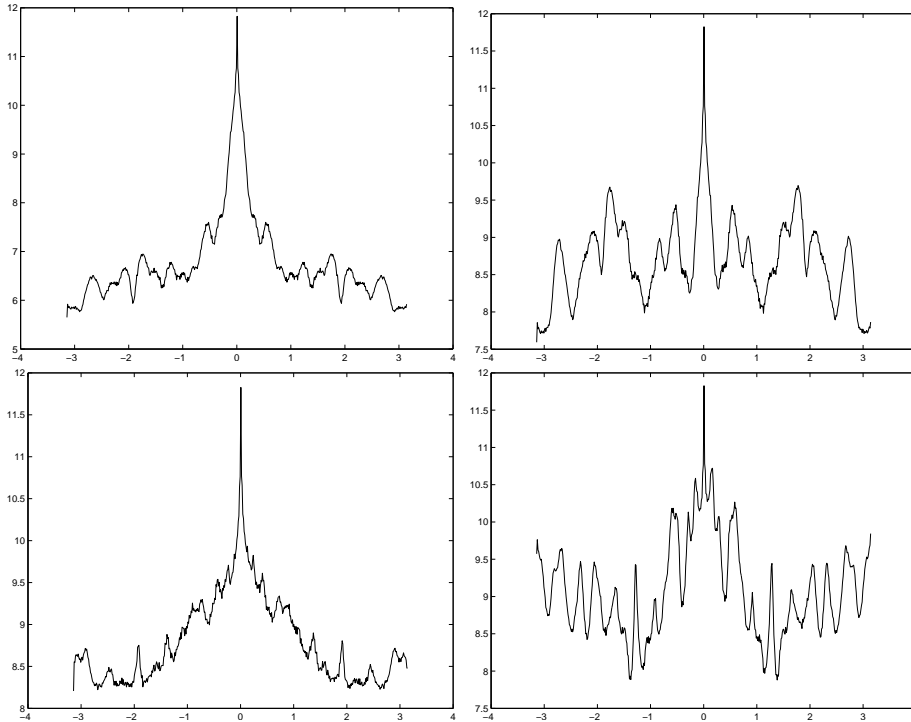


Figure 39: The observed image is simulated for a velocity $v = 1$. Here is shown the log of the sum of the (modulus) Fourier transform in the direction of the blur minus the log of the (modulus) Fourier transform of the code applied for several velocities v ($\log(|\hat{obs}|) - \log(|\frac{1}{v}\hat{\alpha}(t/v)|)$). From top to bottom and left to right for $v = 0$ (TV-norm=31.28), $v = 0.5$ (TV-norm=33), $v = 1$ (TV-norm=25), $v = 1.5$ (TV-norm=44). The estimated velocity is 1 since the estimated velocity is the velocity which minimizes the TV-norm of $\log(|\hat{obs}|) - \log(|\frac{1}{v}\hat{\alpha}(t/v)|)$. In other words deconvolving with wrong kernels leads to an oddly oscillating (modulus) of the Fourier transform which is not credible in a natural image.

Unfortunately optimal codes have to be smooth, hence the previous scheme is not robust in practice. However codes like Raskar's work. Moreover, trivial constant codes equal to one permit to guess the velocity v and

the direction analyzing the Fourier transform of such an acquired image. Hence the reasonable scheme (particularly for aerial or satellite imaging) would be to build simultaneously two images, one with, and one without a *Flutter-Shutter*, use the standard image to estimate accurately v , then use this value of v and to deconvolve the *flutter shutter* image.

11 Conclusion

This paper has started by modeling the stochastic photon acquisition model of a landscape by a light sensor. The model intrinsically contains noise terms due to the Poisson photon emission. This model permits to formalize the *Flutter-Shutter* and to prove what we called the *flutter shutter paradox* namely the fact that even for an infinite exposure time the SNR remains finite, contrarily to the classic steady photography. Two kinds of flutter shutter setups have been considered: an *analog flutter shutter* and a *numerical flutter shutter* permitting smoother, negative gain-control-function and leading to a better SNR of the restored image. It also appeared that the *motion-invariant photography* is a particular case of an *analog Flutter-Shutter*. It is proved that knowing the velocity the best *flutter shutter* function is a sinc, however the SNR raise is small compared to the best snapshot leading to a poor efficiency of such an acquisition system. Eventually a better mouse trap was set up to increase the efficiency of the *Flutter-Shutter*. On average, the SNR can increase significantly provided we *a priori* know the probability density on the motion blur. The conclusion is that the flutter shutter is useful in presence of (unknown) motion blur, and can even become SNR efficient if the density probability of motions is known.

12 Glossary of Notations, Formulas

- (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) $\text{var}(X)$ 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 bi-dimensional Poisson process on $\mathbb{R}^+ \times \mathbb{R}$ associated to the intensity field $l(t, x)$, $\mathbf{P}_l([t_1, t_2] \times [a, b]) \sim \mathcal{P}\left(\int_{t_1}^{t_2} \int_a^b l(t, x) dt dx\right)$
- (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) $\text{obs}(n)$, $n \in \mathbb{Z}$ observation of the landscape at pixel n
- (xv) $e(n) = \mathbb{E}(\text{obs}(n))$, $n \in \mathbb{Z}$ and $e(x)$ its band limited interpolation
- (xvi) $\prod_a = \sum_{n \in \mathbb{Z}} \delta_{na}$ Dirac comb
- (xvii) v relative velocity between the landscape and the camera (unit: pixels per second)
- (xviii) b apparent length of the support of the blur (unit: pixels), $Nv\Delta t = b$
- (xix) $\alpha(t)$ piecewise constant gain control function (*analog flutter shutter & numerical flutter shutter methods*)
- (xx) $w(x) \geq 0$ weight function associated with the probability distribution of the velocity v
- (xxi) $\alpha^*(t)$ optimal gain control function
- (xxii) $\|f\|_{L^1} = \int |f(x)| dx$, $\|f\|_{L^2} = \sqrt{\int |f(x)|^2 dx}$

(xxiii) let $f, g \in L^1(\mathbb{R})$ or $L^2(\mathbb{R})$, then

$$\mathcal{F}(f)(\xi) := \hat{f}(\xi) := \int_{-\infty}^{\infty} f(x)e^{-ix\xi} dx$$

$$\mathcal{F}^{-1}(\mathcal{F}(f))(x) = f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(f)(\xi)e^{ix\xi} d\xi$$

Moreover $\mathcal{F}(f * g)(\xi) = \mathcal{F}(f)(\xi)\mathcal{F}(g)(\xi)$ and (Plancherel)

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

(xxiv) $SNR(X) := \frac{|\mathbb{E}X|}{\sqrt{\text{var}(X)}}$, signal to noise ratio of a random variable X

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

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

(xxvii) Let $\hat{\mathfrak{u}}_{est}$ be an estimation of \hat{u} . Then $SNR^{spectral-averaged}(\hat{\mathfrak{u}}_{est}) := \frac{\frac{1}{2\pi} \int |\mathbb{E}\hat{\mathfrak{u}}_{est}(\xi)| \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi}{\sqrt{\frac{1}{2\pi} \int \text{var}(\hat{\mathfrak{u}}_{est}(\xi)) \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi}}$

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

(xxix) (Poisson summation formula) Let $f \in L^1(\mathbb{R})$ be band-limited. Then $\sum_n f(n) = \sum_m \hat{f}(2\pi m)$, hence if $\hat{f}(\xi) = 0 \forall |\xi| > \pi$ then $\sum_n f(n) = \hat{f}(0)$. Moreover if f is positive then $\sum_{n \in \mathbb{Z}} f(n) = \hat{f}(0) = \|f\|_{L^1}$

(xxx) Root-mean-squared-error (RMSE) between a groundtruth u and an estimation \mathfrak{u}_{est} based on a noisy observation. When u and \mathfrak{u}_{est} are defined on a domain D ($D \subset \mathbb{R}^2$ for images) $RMSE(u, \mathfrak{u}_{est}) := \sqrt{\frac{\int_D |u(x) - \mathfrak{u}_{est}(x)|^2 dx}{\text{measure}(D)}}$

(xxxi) Contrast invariant RMSE (see 7.1) $RMSE^{CI}(u, \mathfrak{u}_{est}) := RMSE(u_{mid(u, \mathfrak{u}_{est})}, \mathfrak{u}_{est_{mid(u, \mathfrak{u}_{est})}})$

References

- [1] A. Agrawal and R. Raskar. Resolving objects at higher resolution from a single motion-blurred image. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [2] A. Agrawal and Y. Xu. Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2066–2073. Citeseer, 2009.
- [3] A. K. Agrawal and R. Raskar. Optimal single image capture for motion deblurring. In *CVPR*, pages 2560–2567. IEEE, 2009.
- [4] M. Ben-Ezra and S.K. Nayar. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, 2004.
- [5] T. Buades, Y. Lou, JM Morel, and Z. Tang. A note on multi-image denoising. In *Local and Non-Local Approximation in Image Processing, 2009. LNLA 2009. International Workshop on*, pages 1–15. IEEE, 2009.
- [6] 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, 2009. CVPR 2009. IEEE Conference on*, pages 1566–1573. IEEE.
- [7] J.F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring using multiple images. *Journal of Computational Physics*, 228(14):5057–5071, 2009.
- [8] W.G. Chen, N. Nandhakumar, and W.N. Martin. Image motion estimation from motion smear—a new computational model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):412–425, 1996.
- [9] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.
- [10] S. Dai and Y. Wu. Motion from blur. In *In Proc. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008.
- [11] J. Delon. Midway image equalization. *Journal of Mathematical Imaging and Vision*, 21(2):119–134, 2004.
- [12] M. E. Gehm, R. John, J.D. Brady, R.M. Willett, and T.J. Schulz. Single-shot compressive spectral imaging with a dual-disperser architecture. *Opt. Express*, 2007.

- [13] A. Gupta, N. Joshi, C. Lawrence Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. *ECCV*, 2010.
- [14] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Estimation of blur and noise parameters in remote sensing. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3580. IEEE, 2002.
- [15] P.A. Jansson and M. Richardson. Deconvolution of images and spectra. *Optical Engineering*, 36:3224, 1997.
- [16] J. Jelinek. Designing the optimal shutter sequences for the flutter shutter imaging method. In *Proceedings of SPIE*, volume 7701, page 77010N, 2010.
- [17] H. Ji and C. Liu. Motion blur identification from image gradients. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.
- [18] J. Jia. Single image motion deblurring using transparency. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
- [19] A. Levin. Blind motion deblurring using image statistics. *Advances in Neural Information Processing Systems*, 19:841, 2007.
- [20] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Deconvolution using natural image priors. *ACM Trans. Graphics*, 26(3).
- [21] A. Levin, P. Sand, T.S. Cho, F. Durand, and W.T. Freeman. Motion-invariant photography. In *ACM SIGGRAPH 2008 papers*, pages 1–9. ACM, 2008.
- [22] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. *CVPR*, 2011.
- [23] L.B. Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974.
- [24] N. Joshi, R. Szeliski, and D. Kriegman. Psf estimation using sharp edge prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.
- [25] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1048–1054. IEEE, 2005.
- [26] Raskar; Ramesh and Tumblin; Jackand Agrawal; Amit. Method for deblurring images using optimized temporal coding patterns, 05 2006.

- [27] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3):311–317, 2005.
- [28] W.H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972.
- [29] S. Schuon and K. Diepold. Comparison of motion de-blur algorithms and real world deployment. *Acta Astronautica*, 64(11-12):1050–1065, 2009.
- [30] A. Sellent, M. Eisemann, and M. Magnor. Calculating motion fields from images with two different exposure times. Technical report, Cite-seer, 2008.
- [31] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM SIGGRAPH 2008 papers*, pages 1–10. ACM, 2008.
- [32] 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 Transactions on Pattern Analysis and Machine Intelligence*, pages 1012–1028, 2010.
- [33] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis. Synthetic shutter speed imaging. In *Computer Graphics Forum*, volume 26, pages 591–598. Wiley Online Library, 2007.
- [34] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high speed periodic videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):671–686, 2011.
- [35] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 491–498. IEEE, 2010.
- [36] 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. volume 24, pages 765–776. ACM, 2005.
- [37] L. Yuan, J. Sun, L. Quan, and H.Y. Shum. Image deblurring with blurred/noisy image pairs. In *ACM SIGGRAPH 2007 papers*, pages 1–es. ACM, 2007.
- [38] L. Yuan, J. Sun, L. Quan, and H.Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. In *ACM SIGGRAPH 2008 papers*, pages 1–10. ACM, 2008.