

A Theory of Optimal Flutter Shutter for Probabilistic Velocity Models*

Yohann Tendo[†] and Jean-Michel Morel[‡]

Abstract. *Flutter shutter* (coded exposure) is a new paradigm for cameras that allows for an arbitrary increase of the exposure time when the relative camera/scene motion is uniform. The photon flux is interrupted according to a *flutter shutter code*. For arbitrarily severe uniform motion blur a well chosen code guarantees an invertible blur kernel. Yet, when the relative camera/scene velocity is a *known constant*, a *flutter shutter* cannot gain more than a 1.17 factor in terms of root mean-squared error compared to the optimal *snapshot*. In this paper, we prove that this optimality bound can be relaxed under the realistic assumption that a random model for the velocities is available. We give analytical formulae for the *optimal flutter shutter code* and the *optimal snapshots* associated with a random velocity distribution. Conversely we also prove formulae that reveal the velocity distribution underlying a given *flutter shutter code*.

Key words. motion blur, Poisson noise, snapshot, *flutter shutter*, optimization, signal-to-noise ratio (SNR), mean-squared error (MSE), coded exposure, stochastic motion model

AMS subject classifications. 68U10, 42A38, 60G99

DOI. 10.1137/15M1035872

1. Introduction. Digital cameras count at each pixel sensor the number of photons emitted by the observed scene during a time span called the exposure time. The photon count follows a Poisson random variable. Its mean is the ideal (noiseless) pixel value. The difference between this ideal pixel value and the observed sensor photon count is called (shot) noise. The ratio of the mean of the photon count over its standard-deviation is called the signal-to-noise ratio (SNR). In passive imaging systems there is no control over the scene lighting. Thus, the only safe way to increase the SNR is to integrate more photons by increasing the exposure time. Yet, when the scene or the camera moves during the exposure process, the resulting images are degraded by a motion blur. If the support of a motion blur kernel exceeds two pixels, the blur is no longer invertible. This limits the exposure time and therefore the image quality of a *snapshot*. A setup solving this photography trade-off between noise and blur was proposed in [3, 4, 6, 37, 38, 39] for uniform camera-scene motions. The authors of these papers attached a *flutter shutter* to a camera to get an invertible motion blur kernel. A *flutter shutter*

*Received by the editors December 9, 2013; accepted for publication (in revised form) December 30, 2015; published electronically DATE. The algorithm and simulation facility linked to this paper has been separately published as [44].

<http://www.siam.org/journals/siims/x-x/M103587.html>

[†]LTCI, CNRS, Télécom ParisTech, Université Paris-Saclay, 75013 Paris, France (yohann.tendero@telecom-paristech.fr). The research of this author was partially supported by the grant ONR N000141410683. Part of this author's work was completed at the UCLA Mathematics Department.

[‡]Centre de Mathématiques et Leurs Applications (CMLA), École Normale Supérieure de Cachan, 94235 Cachan, France (morel@cmla.ens-cachan.fr). The research of this author was partially supported by the European Research Council (advanced grant Twelve Labours 246961), by the Office of Naval Research (ONR grant N00014-14-1-0023), and by ANR-DGA project ANR-12-ASTR-0035.



Figure 1. Illustration of the formulae proved in this paper. Left: Simulated observed (blurry and noisy) image using the patented [39] code of Agrawal and coworkers (see [3], [38, p. 799]) with a 52 pixel code length. Middle: The image reconstructed by direct deconvolution. Right: The formalism that we shall develop in this paper allows us to deduce the probabilistic velocity distribution for which this code is near-optimal. The x-axis is the motion (in signed pixels), and the y-axis represents the logarithm of the probability density ($\log(1 + \rho(v))$). The probability that $v = 0$ is massive.

is described by a binary shutter sequence or *flutter shutter code* that gives the intervals when the photon flux is interrupted. If the *flutter shutter code* is well chosen, invertibility of the motion blur kernel can be guaranteed for arbitrarily high velocities. This fact can be checked numerically in the online *flutter shutter* camera simulator [42] (see also Figure 1).

Since the *flutter shutter* allows for an arbitrarily long exposure time, does that mean that one could indefinitely decrease the mean-squared error (MSE) of the deconvolved image? The answer is negative, as proved in [48] for ideal observation conditions. The answer is still negative, as proved in [46], which takes the observation conditions into account, i.e., lighting, sensor readout, and obscurity noise of finite variances. More precisely, given a scene moving in uniform translation at a known velocity $v \in \mathbb{R}$, the optimal *flutter shutter* strategy can gain no more than a 1.17 factor compared to an optimal *snapshot* in terms of root mean-squared error (RMSE). This 1.17 gain is significant, but clearly not massive. In addition, obtaining this 1.17 bound requires the technologically more demanding setup of a *numerical flutter shutter* that allows for signed codes.

A *numerical flutter shutter* requires the acquisition of an image sequence, for example by a high speed camera, which is numerically treated by a temporal filter. According to [20, 36] an image sensor can have a duty ratio¹ of nearly 100%. Thus, a sensor can integrate light without interruption. Thus, a *numerical flutter shutter* without “dead time” is feasible from a technological point of view.

Yet, in such a situation, there are better solutions than a temporal filter to increase the SNR. The classic one consists of registering the images locally or globally and performing a burst denoising [9]. The fusion of two images divides the MSE by two. This means that a *flutter shutter* is useless unless the considered application has some strong physical or numerical constraints preventing the separate acquisition and fusion of short exposure frames. In such severely constrained situations, an *analog flutter shutter* becomes the only other option. Earth observation satellites, for example, suffer from uniform motion blur that limits the exposure

¹The duty ratio is the ratio of light integration time over readout, storage, or reset times—that is, the percentage of useful time.

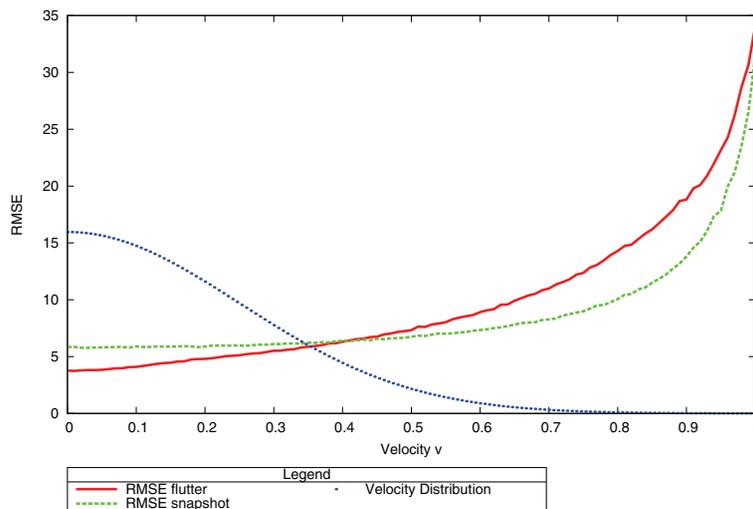


Figure 2. This figure illustrates the setup proposed in this paper. In this experiment, we assume that the relative camera/scene velocity is proportional to the (Gaussian-like) probability distribution depicted by the blue curve. The formalism that we shall develop here allows us to deduce an optimal code and the optimal exposure time for the snapshot taking the velocity distribution into account. The red (resp., green) curve depicts the evolution of the RMSE of the optimal flutter shutter (resp., optimal snapshot) as a function of the relative camera/scene velocity measured without loss of generality in pixels per Δt (Δt is the time step of the flutter shutter). The optimization permits us to concentrate the gain for the most probable observed velocities (see also Figure 3). For instance, with this model, the probability of observing a velocity smaller than 0.4 is larger than 0.89. This simulation is based on [42], using the “boat” image.

time and have limited transmission, storage, and computational resources.

A *flutter shutter* setup is also to be envisaged for the observation of very fast phenomena like explosions, where the readout time is too long to ensure that the support of the motion kernel is under two pixels/frame.

This paper proposes a framework that permits us to optimize *flutter shutter* cameras beyond the aforementioned 1.17 bound [48] on the RMSE gain. This is possible, provided that the velocity law in the scene is a priori known or can be learned. We prove a new closed formula that permits one to compute optimal codes for any probability density of the expected velocities (see Theorems 3.2 and 3.3, Figure 2). With this different setup we shall give closed formulae for the RMSE gain (see section 5) and show that, depending on the velocity distribution, it can exceed the 1.17 bound (see section 6.1).

In addition, we prove an inverse formula that deduces from each *flutter shutter code* its associated velocity distribution (see Theorem 4.1, Figure 1, and section 6.2). This permits us to review and interpret *flutter shutter* cameras and *flutter shutter codes* recently proposed in the literature, including the patented ones. In short, these formulae permit one to associate with any existing *flutter shutter code* the velocity distribution for which it is optimal.

In section 1.1 we start with an analysis of the literature on the *flutter shutter*. In section 1.2 we fix the terminology needed to justify the plan of the paper, which is given in section 1.3.

1.1. Related work. The simplest hardware setup to get invertible uniform motion blurs was proposed by Agrawal, Raskar, and coworkers [3, 4, 6, 37, 38, 39]. The camera shutter

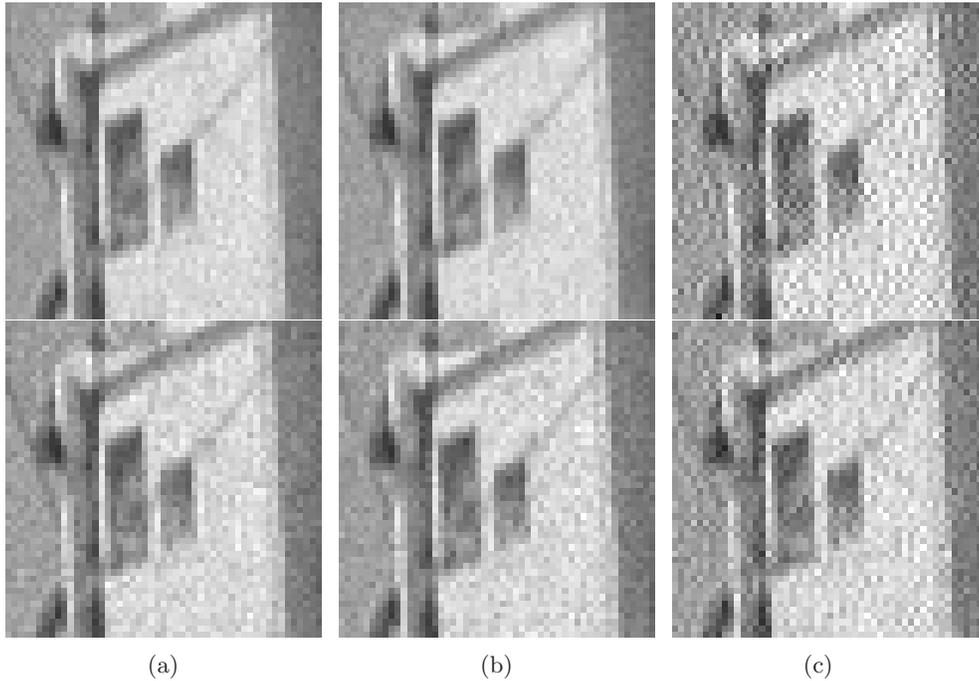


Figure 3. This experiment shows three (crops) of the boat image used to generate the curve of Figure 2. The three images at the top are the restored images using an optimal flutter shutter. The three at the bottom are the restored images using the optimal snapshot. Both are optimized for the velocity distribution of Figure 2. The considered velocities are $v = 0$ (column (a)), $v = 0.1$ (column (b)), and $v = 0.75$ (column (c)), measured in pixels per Δt (Δt is the time step of the flutter shutter). With this velocity model the probability that the velocity v satisfies $|v| \leq 0.1$ is larger than 0.3, while the probability that v satisfies $|v| \geq 0.75$ is lower than 0.0027. The optimal flutter shutter is less noisy than the optimal snapshot for the most probable velocities.

opens/closes according to a pseudorandom binary code called *flutter shutter code*. Following the *flutter shutter* literature [3, 4, 6, 11, 17, 18, 19, 28, 29, 30, 31, 37, 38, 39, 43, 47, 48, 50, 56], only well-posed deconvolution strategies will be compared. Indeed, if the *flutter shutter code* is well chosen, arbitrarily severe uniform motion blurs are made invertible. As illustrated in Figure 1, an image acquired by *flutter shutter* looks like a stroboscopic image. Nonetheless, one can recover a neat image by a simple well-posed deconvolution. The *flutter shutter* camera simulator developed in [42] permits one to test the code proposed in [38, p. 5] and patent application [39] and several other classic *flutter shutter codes* from the literature.

In their recent paper [11], Cossairt, Gupta, and Nayar claimed that (1) an upper bound for the gain of the *flutter shutter* with respect to a *snapshot* in terms of the RMSE is $\sqrt{1 + \sigma_r^2/J}$ [11, eq. (11), p. 5], where J is the mean photon emission and σ_r^2 is the sensor readout variance, and (2) “gain for computational imaging is significant only when the average signal level J is considerably smaller than the read noise variance σ_r^2 ” [11, p. 5]. These two claims actually rely on an oversimplified acquisition model and have been proved incorrect in [46].

In [30] the *flutter shutter* apparatus is applied to iris images, and in [28, 56] to bar-codes. In [27] the authors propose to optimize the binary *flutter shutter code* as a function of the

velocity of the scene. The optimization involves an accelerated random search among *binary* sequences with fixed sum and length. The authors state in their conclusion that there exists no universal *flutter shutter code* in the sense that “a shutter sequence will produce noninvertible blur when the velocity is more than twice a nominal velocity” [27, section 7, p. 13]. This fact is obvious when looking at the form of the Fourier transform of any *flutter shutter code* (see, e.g., (B.4) below). Thus, the authors of [27] look for (binary) sequences that are optimal for a certain known velocity. No sequence is optimal for all velocities at the same time. Indeed, for a known *maximal* velocity $v \in \mathbb{R}$ an optimal *flutter shutter code* in terms of MSE exists [48] and is derived from a cardinal sine function of the form $\text{sinc}(vt)$. It is therefore neither binary nor nonnegative and depends on v . In [41] the authors use a local deblurring user-driven scheme on a *flutter shutter* embedded camera to deal with spatially varying blurs caused by the presence of several velocities in the observed scene. In [40] the authors treat the question of denoising an image taken by a *flutter shutter* camera. The authors also suggest a user-assisted estimation of the blur. Their conclusion is that the denoising should be applied both before and after deconvolution. In [12] the authors treat the question of a posteriori motion estimation using a *flutter shutter*. In [15] a per pixel *flutter shutter* is used to build a camera that permits a postcapture balance between spatial and temporal resolutions of movies. A multicamera equipped with *flutter shutters* is investigated in [2] and used to increase the frame rate of a single camera while having an increased number of photons captured compared to the equivalent high-speed camera. A single camera equipped with a mask on the aperture and an array of light sources is used in [21] to construct the visual hull of an object (shape from silhouette).

Another solution for getting an invertible motion blur using only one image was found in [23]. There Levin et al. suggested moving the camera with constant acceleration in the direction of the motion during the exposure time. The resulting kernel is invertible, and its approximated invariance with respect to the velocity is proved in [48], which also proves that the resulting MSE is infinite. This approach, which requires an a priori knowledge of the motion direction, has been generalized in [10] to the case of unknown directions. It requires acquiring two images instead of one. In contrast, [48] proves that this setup is equivalent to an *analog flutter shutter*, for which a *flutter shutter code* was proposed. This permits us to avoid the burden of accelerating the camera and works in any motion direction, like any other *flutter shutter*. In [32] the *motion-invariant photography* apparatus is implemented using the lens of the camera. Note that the approaches that require moving the camera cause blur even in static parts of the scene. Therefore, they cannot be optimal for every velocity in terms of MSE.

In [7, 13, 22, 24, 26, 33, 35, 52] the authors use a temporally fixed and spatially varying mask in order to estimate the depth, and/or they refocus the out-of-focus part to get an always-in-focus (neat) image. In [16] the authors deal with the question of the optimal trade-off between depth of field and exposure time. In [14] the authors take advantage of complementary metal oxide semiconductor (CMOS) imaging sensors to implement a *coded rolling shutter* to trade vertical resolution for an increased dynamic range. The authors of [51] also suggest using a camera equipped with a mask on the aperture camera and to take purposely out-of-focus images with a mask to increase the dynamic range. Their conclusion is rather negative, since they state that “None of the possible combinations of aperture filter and deconvolution

algorithm were able to consistently reduce the dynamic range of the captured image without excessively degrading image quality” [51, section 8, p. 10]. Another computational camera is designed in [34], where the aperture is equipped with a mask, and the sensor is moved at a constant velocity during the exposure. The mask is used to control the depth of field, creating *bokeh* or a depth-invariant blur size. Another camera prototype was designed in [25], where the authors suggest a programmable aperture (mask), which is also used for depth and digital refocusing. An interesting implementation, the *Frankencamera*, was proposed in [1]. It permits “control and synchronization of the sensor and image processing pipeline at the microsecond time scale, as well as the ability to incorporate and synchronize external hardware like lenses and flashes” [1, abstract]. The authors of [1] investigate six computational photography applications. An even more complex scheme involving a fixed mask close to the sensor and a dynamic one on the aperture is investigated in [5], where the authors explore the feasibility of postprocessing trade-offs between spatial, angular, and temporal resolutions.

Most of these works develop more complex hardware setups than the original *flutter shutter*. However, the common denominator is obtaining a sharp image by an increased exposure time. Thus, the ultimate criterion for this should be the MSE gain with respect to a standard camera aperture strategy with optimal exposure time. The underlying model for all the setups discussed above is an encoded exposure followed by a deconvolution. Therefore, the upper bound of 1.17% [46, 48] of the gain in terms of RMSE is in principle applicable to all. It is therefore important to discuss by which means and assumptions this performance might be increased above the 1.17 bound, which applies as soon as the velocity is known. Our main goal in this paper is to provide a mathematically founded answer to this question, along with simple and new closed formulae linking velocity distributions and optimal *flutter shutter codes* (see Theorems 3.2, 3.3, and 4.1).

1.2. The *flutter shutter* implementations and their optimization. There are three different ways to implement a *flutter shutter* (coded exposure).

The first technical possibility consists of implementing the *flutter shutter exposure function* with a shutter device that opens and closes on very short subintervals of the exposure time. (In the following we denote the function describing the time exposure function of the *flutter shutter* by “*flutter shutter gain function*.”) In [48] the authors proposed considering smoother and/or nonbinary *flutter shutter gain functions*. A nonbinary *flutter shutter gain function* requires the use of an optical filter in place of a simple shutter. As an example, this optical filter can be implemented on the stages of a time delay and integration (TDI) device for remote sensing applications. The optical filter opacity varies temporarily and controls the percentage of photons that are allowed to travel to the sensor (like temporarily controlled sunglasses), called the “gain.” When the gains are piecewise constant, the *flutter shutter gain function* can be encoded by a “code” called *flutter shutter code*. If, in addition, the code is binary, we retrieve the “historical” implementation considered by the inventors [3, 4, 6, 37, 38, 39]. Notice that with this setup, the number of caught photons is reduced compared to those caught by a *snapshot* with the same exposure time. This generalization is called *analog flutter shutter* [48]. For example, *motion-invariant photography* is equivalent to an *analog flutter shutter* [48]. An *analog flutter shutter* camera allows in principle for any nonnegative, bounded from above by 1, and $L^1(\mathbb{R})$ *flutter shutter gain function*.

The second technical possibility for implementing a *flutter shutter* is the *numerical flutter shutter*. In contrast to the *analog flutter shutter*, the *numerical flutter shutter* is a temporal filter. It allows for nonbinary and even negative gains. The camera takes a burst of L images using an exposure time Δt . The k th elementary image is assigned a numerical gain $\alpha_k \in \mathbb{R}$. The final observed image is obtained as the weighted sum of elementary images with weights $(\alpha_k)_{k \in \{0, \dots, L-1\}}$. A *numerical flutter shutter* camera permits the use of any $L^2(\mathbb{R})$ piecewise constant function (code) for the weighted sum [48]. In addition, it does not require one to attach a fast physical shutter device to the camera. Indeed, the “*flutter shutter effect*” that guarantees the invertibility of the motion kernel is obtained by the numerical weights. Such a setup can be easily implemented, for example, with a CMOS sensor.

The third kind of *flutter shutter* is the *postprocessing flutter shutter*, which consists of applying a temporal filter on a sliding window to the frames of a video sequence. Thus, in contrast to the *numerical* or *analog flutter shutters*, it offers no compression. The *postprocessing flutter shutter* can be used for video postprocessing, e.g., for blind motion blur deconvolution [45, 49]. A well chosen filter blindly deconvolves *any* uniform motion blur if the blur of each frame is less than or equal to 1 pixel/frame [45, 49]. Another example of *postprocessing flutter shutter* is given in [55]; its goal is not a deconvolution, but to boost the high temporal frequencies of videos to enhance the motion effects.

As proved in [48], the best performance for all *flutter shutters* of all sorts in terms of MSE is attained with *numerical flutter shutters*. The argument is that, using the same *flutter shutter code*, the *numerical flutter shutter* beats the *analog flutter shutter* in terms of MSE. In consequence, any performance bound for *numerical flutter shutters* applies to *analog flutter shutters* as well.

Table 1 summarizes the three possible *flutter shutter* setups, *analog*, *numerical*, and *postprocessing flutter shutters*.

Table 1

The three possible implementations of a flutter shutter.

<i>Flutter shutter type</i>	Shutter modification	Number of image(s) to store/transmit
<i>Analog</i> [3, 4, 6, 17, 18, 19, 28, 29, 30, 31, 37, 38, 39, 48, 56]	Fast physical shutter (or liquid crystal device or flash light)	1
<i>Numerical</i> [48]	None	1
<i>Postprocessing</i> [45, 49, 55]	None	All, a temporal filter is applied as post-processing

Table 2 provides, to the best of our knowledge, the state of the art concerning the optimization of a *flutter shutter* and points out the considered application fields. The quoted papers limit their study to binary exposure sequences. Yet the current technology allows for *numerical flutter shutters* (see, e.g., [20, 36]).

1.3. Plan of the paper. Section 2 recalls a common formalism proposed in [48] for *analog*, *numerical*, and *postprocessing flutter shutters* and for *motion-invariant photography*. For each of these *flutter shutters*, a closed formula for the MSE as a function of the *flutter shutter code* and of the velocity probability density is given.

Table 2

This table describes the state of the art concerning the optimization of a flutter shutter. Little is known about optimal codes. In particular, most articles consider only binary codes.

Articles	Shutter type	Code type	Optimization type	Application
Agrawal, Raskar, and colleagues [3, 4, 6, 37, 38, 39]	Analog	Binary	Random search	Traffic surveillance
He, Huang, McCloskey, Jelinek, Xu et al. [17, 18, 19, 28, 30, 31, 56]	Analog	Binary	Accelerated search among binary codes	Traffic surveillance, iris image deblurring (biometric), 2D bar-codes, remote sensing, character recognition
Tendero et al. [45, 48, 49]	Numerical	Real-valued codes	Analytic, exact	Video postprocessing

Section 3 proves analytical formulae for the optimal *analytically* continuous or piecewise constant *flutter shutter codes* associated with a given probability density for the velocities (see Theorems 3.2 and 3.3).

Not only does the proposed formalism permit a forward analysis from velocity densities to *flutter shutter codes*, but also it proposes a backward analysis. Indeed, Theorem 4.1 in section 4 states the conditions under which a given *flutter shutter code* is optimal for some velocity distribution. It also gives a formula that associates its underlying velocity density to each given optimal code.

This formalism is also applied in section 5 to compute the optimal exposure time of a standard camera for a given velocity distribution (see Proposition 5.1). This new *snapshot* theory allows us to compare optimal standard cameras and optimal *flutter shutter* cameras for *any* velocity distribution.

All of these theoretical results are illustrated in section 6. Section 6.1 analyzes three natural stochastic velocity models. It provides the optimal codes and a performance comparison between the *numerical flutter shutter* and the optimal *snapshot* for a centered-Gaussian, a uniform, and a trimodal velocity distribution. As a last application, section 6.2 performs the reverse engineering of several classic, often patented, *flutter shutter codes* by providing the underlying velocity distribution for which each code is optimal.

This study will be conducted on *numerical flutter shutters*. It therefore also gives upper bounds for the gain of any *analog flutter shutter* with respect to the optimal *snapshot* in terms of RMSE. These upper bounds are sufficient to discuss the pros and the cons of the *flutter shutter* method (section 7). A glossary of notation is available in Appendix E. (Latin numeral references in the text refer to formulae listed there.)

2. A simple mathematical formalism for the *flutter shutter* [48]. The relative camera-landscape motion can be associated with a one-dimensional (1D) box kernel. The support of this kernel increases linearly with the exposure time $\Delta t > 0$ and the velocity $v \in \mathbb{R}$ of the motion. If the exposure time is too long and the blur support exceeds two pixels, its deconvolution is an ill-posed problem [8]. Instead the *flutter shutter*, or coded exposure, ensures an invertible motion kernel for arbitrarily severe motion blur. There are two different acquisition tools that implement a *flutter shutter* with a moving sensor (or scene). The *flutter*

shutter gain function can be implemented as an optical (temporally changing) filter. This filter controls the percentage of incoming photons allowed to travel to the sensor. The filtering function is generally assumed to be piecewise constant [3, 4, 6, 39, 37] with a *flutter shutter code* $(\alpha_k)_{k \in \{0, \dots, L-1\}}$, where L is the length of the code. This setup corresponding to the initial technology of the inventors will be called the *analog flutter shutter*.

A more flexible setup, the *numerical flutter shutter*, is a mere temporal filter applied to a sequence of L frames. The k th raw frame is assigned a numerical gain $\alpha_k \in \mathbb{R}$. The processed frame is obtained as the weighted sum of the raw frames with weights $(\alpha_k)_{k \in \{0, \dots, L-1\}}$. *Analog* and *digital flutter shutters* are associated to a *flutter shutter code*, but the formulae for the resulting image are not exactly the same, as summarized in Table 3.

Table 3

This table summarizes and compares the main formulae of numerical and analog flutter shutters. Using the same code, the MSE of a numerical flutter shutter is lower than the MSE of an analog flutter shutter. All codes usable with an analog flutter shutter are usable with a numerical flutter shutter, while the converse is not true. See text.

Type of <i>flutter shutter</i>	Numerical <i>flutter shutter</i>	Analog <i>flutter shutter</i>
<i>Flutter shutter gain function</i> $\alpha(t)$	$\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t, (k+1)\Delta t]}(t)$ (with $\alpha_k \in \mathbb{R}$ and $\Delta t > 0$)	$\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t, (k+1)\Delta t]}(t)$ (with $\alpha_k \in [0, 1]$ and $\Delta t > 0$)
<i>Continuous flutter shutter gain function</i> $\alpha(t)$	$\alpha(t) \in L^2(\mathbb{R})$	$\alpha(t) \in L^1(\mathbb{R})$, $\alpha(t) \in [0, 1]$
Observed samples $obs(n)$	$obs(n) \sim \sum_{k=0}^{L-1} \alpha_k \mathcal{P}\left(\int_{k\Delta t}^{(k+1)\Delta t} u(n-vt) dt\right)$	$obs(n) \sim \mathcal{P}\left(\frac{1}{ v }(\alpha(\frac{\cdot}{v}) * u)(n)\right)$
$\mathbb{E}(obs(n))$ (observed)	$\left(\frac{1}{ v }\alpha(\frac{\cdot}{v}) * u\right)(n)$	$\frac{1}{ v }(\alpha(\frac{\cdot}{v}) * u)(n)$
$\text{var}(obs(n))$ (observed)	$\left(\frac{1}{ v }\alpha^2(\frac{\cdot}{v}) * u\right)(n)$	$\frac{1}{ v }(\alpha(\frac{\cdot}{v}) * u)(n)$
Inverse filter $\hat{\gamma}(\xi)$	$\frac{\mathbb{1}_{[-\pi, \pi]}(\xi)}{\hat{\alpha}(\xi v)}$	$\frac{\mathbb{1}_{[-\pi, \pi]}(\xi)}{\hat{\alpha}(\xi v)}$
$\mathbb{E}(\hat{u}_{est}(\xi))$ (deconvolved)	$\hat{u}(\xi) \mathbb{1}_{[-\pi, \pi]}(\xi)$	$\hat{u}(\xi) \mathbb{1}_{[-\pi, \pi]}(\xi)$
MSE (u_{est}) (deconvolved)	(2.1) $\frac{1}{2\pi} \int_{\mathbb{R}} \frac{\ \alpha\ _{L^2(\mathbb{R})}^2 \ u\ _{L^1(\mathbb{R})}}{ \hat{\alpha}(\xi v) ^2} \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi$	(2.2) $\frac{1}{2\pi} \int_{\mathbb{R}} \frac{\ \alpha\ _{L^1(\mathbb{R})} \ u\ _{L^1(\mathbb{R})}}{ \hat{\alpha}(\xi v) ^2} \mathbb{1}_{[-\pi, \pi]}(\xi) d\xi$

This *flutter shutter* study is performed as though the image were a 1D signal, recorded on a line in the direction of the camera-landscape motion. The convolution and deconvolution model is applied on each line of the image. From the mathematical viewpoint, the *flutter shutter* therefore boils down to the 1D convolution of a *flutter shutter gain function* α with a 1D stochastic observed landscape. The expected value at position x of this stochastic landscape will be denoted by $u(x)$. In all statements, this ideal (noiseless) landscape u is assumed to have finite energy, $u \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$, and to be $[-\pi, \pi]$ band limited (thanks to the combined camera and sensor frequency cut-off). Therefore, u is well sampled at a unit rate.

The whole formalism of the *flutter shutter* is summarized in Table 3. This table is self-contained, and all formulae are straightforward except the MSE formulae of the last row; for their proof we refer to [48]. The first row of the table indicates the kind of implementable *flutter shutter gain function* α , depending on the *flutter shutter* type and with a discrete code. In the second row, for the sake of simplicity in calculations, the *flutter shutter* formalism is extended to deal with time-continuous as well as piecewise constant *flutter shutter gain functions*. The *flutter shutter gain function* is $\alpha(t)$, meaning that the gain can change continuously with time. A formula proved in [48] permits us to convert a time-continuous *flutter shutter gain function* into a classic code.

The third row of the table gives the exact formula of the observed samples. The notation $X \sim Y$ means that the random variables X and Y have the same law. The notation $\mathcal{P}(\lambda)$ denotes a Poisson random variable with intensity λ . For the *analog flutter shutter*, the observed digital image at pixel n is a Poisson noise with intensity $\frac{1}{|v|}(\alpha(\frac{\cdot}{v}) * u)(n)$, where $*$ denotes the convolution in $L^1(\mathbb{R})$; see (ix) of Appendix E. The scaling factor $v \in \mathbb{R}$ corresponds to the relative camera scene velocity, which for the formulae in this table is assumed to be known. The formula of the observation $obs(n)$ is identical for both *numerical* and *analog flutter shutters*. In both cases the observed samples $obs(n)$ are obtained for $n \in \mathbb{Z}$. As shown in the fourth row, the expected value of the (observed) image is also identical in both cases. However, there is a significant difference in the variance formulae of the fifth row. The variance of the observed value at a pixel n depends on the square of the *flutter shutter gain function* α for the *numerical flutter shutter*, while the dependency is linear for the *analog flutter shutter*. The fourth and fifth rows are immediately derived from the third by expectation and variance calculations.

The sixth row gives the formula of the deconvolution filter. As usual in the literature on the *flutter shutter* [3, 4, 6, 11, 17, 18, 19, 28, 29, 30, 31, 37, 38, 39, 48, 56], only well-posed deconvolution strategies are considered. Thus, we always assume that the *flutter shutter gain function* α does not vanish on $[-\pi|v|, \pi|v|]$ and that the velocity $v \in \mathbb{R}$ is a known constant. Hereinafter \hat{f} denotes the classic continuous Fourier transform on \mathbb{R} , and \check{f} the inverse Fourier transform on \mathbb{R} ; see (xx). Under this condition the deconvolution of the motion-blurred observed image is a well-posed problem. This inverse filter is nothing but the inverse of the Fourier transform of the convolution kernel, $\frac{1}{|v|}(\alpha(\frac{\cdot}{v}) * \text{sinc})(n)$. The sinc function ensures that it is applied only on the $[-\pi, \pi]$ frequencies. Indeed, u is assumed to be $[-\pi, \pi]$ band limited.

The seventh row gives the expected value of the deconvolved image. The inverse filter is designed to give back the ideal landscape (in expectation) by simply inverting the invertible *flutter shutter* kernel α .

The last row gives the main two formulae proved in [48], namely the MSE (or variance) of the restored signal. These formulae give the MSE of the restored signal value with respect to its expectation. In other words, this MSE is the expectation of the square of their difference, and this difference is the deconvolved acquisition noise. Note that the observed landscape u intervenes in the above formulae as a mere multiplication factor by the constant $\|u\|_{L^1(\mathbb{R})}$. Thus, optimizing a *flutter shutter* amounts to finding *flutter shutter gain functions* α that minimize (2.1) or (2.2), which are different for the *analog* and *numerical flutter shutter*.

It immediately follows from the MSE formulae (2.1) and (2.2) of the last row of Table 3 that if a *flutter shutter gain function* $0 \leq \alpha(t) \leq 1$ is implementable on both kinds of *flutter shutters*, the MSE of the *analog flutter shutter* is bigger than the MSE of the *numerical flutter shutter*. Indeed, these conditions on $\alpha(t)$ imply $\alpha^2(t) \leq \alpha(t)$ and therefore $\|\alpha\|_{L^2(\mathbb{R})}^2 \leq \|\alpha\|_{L^1(\mathbb{R})}$. Notice that the MSE of the *numerical flutter shutter* does not change by changing α for $\lambda\alpha$ if $\lambda \in \mathbb{R} \setminus \{0\}$. This is not true for the *analog flutter shutter*, where for evident physical reasons, $0 \leq \alpha(t) \leq 1$ and (e.g.) $\frac{\alpha}{2}$ has a higher MSE than α .

3. From a velocity distribution to its optimal *flutter shutter code*. We now address the subject of the present paper, which is to extend the above formalism to compute optimal *flutter shutter codes* in the presence of a (known) velocity distribution. Our first step is to find optimal, in the sense of the MSE of (2.1), real-valued, *time-continuous flutter shutter gain functions* in $L^2(\mathbb{R})$. Our second step is to find optimal, real-valued, piecewise constant *flutter shutter gain functions* in $L^2(\mathbb{R})$ that can be implemented with a *numerical flutter shutter*.

Step 1. Optimal time-continuous *flutter shutter gain function*. In what follows we assume that the probability density $\rho(v)$ for the relative camera/scene velocities is known, and that the possible velocities are bounded; namely, $\rho(v) = 0$ for $|v| > |v_{max}|$. The density ρ is, for example, easily obtained from an optical flow algorithm.

The question is how to derive from ρ a real-valued *flutter shutter gain function* $\alpha \in L^2(\mathbb{R})$ that gives a minimal MSE for the reconstructed signal. As we have seen in section 2, for a given velocity $v \in \mathbb{R}$ the MSE of the final deconvolved image of the *numerical flutter shutter* is given by (see (2.1))

$$(3.1) \quad \int_{\mathbb{R}} \frac{\|\alpha\|_{L^2(\mathbb{R})}^2 \|u\|_{L^1(\mathbb{R})} \mathbb{1}_{[-\pi, \pi]}(\xi)}{|\hat{\alpha}(\xi v)|^2} d\xi.$$

Since by definition the optimum, if it exists, satisfies $\alpha \in L^2(\mathbb{R})$, we can look for $\hat{\alpha} \in L^2(\mathbb{R})$ without loss of generality (w.l.o.g.). In addition, from Plancherel's identity (xx) we have $\|\alpha\|_{L^2(\mathbb{R})}^2 = \frac{1}{2\pi} \|\hat{\alpha}\|_{L^2(\mathbb{R})}^2$. Thus, by a change of variables we deduce that, for a fixed v and dropping the positive multiplicative constant $\frac{\|u\|_{L^1(\mathbb{R})}}{2\pi}$, minimizing (3.1) with respect to $\alpha \in L^2(\mathbb{R})$ is equivalent to minimizing

$$(3.2) \quad E_v(\hat{\alpha}) := \|\hat{\alpha}\|_{L^2(\mathbb{R})}^2 \int_{\mathbb{R}} \frac{\mathbb{1}_{[-\pi|v|, \pi|v|]}(\xi) d\xi}{|v| |\hat{\alpha}|^2(\xi)}$$

with respect to $\hat{\alpha} \in L^2(\mathbb{R})$. Taking the velocity distribution $\rho(v)$ into account, from (3.2) we deduce the functional that characterizes optimal *flutter shutter gain functions*

$$(3.3) \quad \begin{aligned} E(\hat{\alpha}) &:= \int_{\mathbb{R}} E_v(\hat{\alpha}) \rho(v) dv = \int_{\mathbb{R}} \|\hat{\alpha}\|_{L^2(\mathbb{R})}^2 \int_{-\infty}^{\infty} \frac{\mathbb{1}_{[-\pi|v|, \pi|v|]}(\xi) d\xi}{|v| |\hat{\alpha}|^2(\xi)} \rho(v) dv \\ &= \|\hat{\alpha}\|_{L^2(\mathbb{R})}^2 \int_{-\infty}^{\infty} \frac{1}{|\hat{\alpha}|^2(\xi)} \left(\int_{\mathbb{R}} \frac{\rho(v) \mathbb{1}_{[-|v|\pi, |v|\pi]}(\xi)}{|v|} dv \right) d\xi, \end{aligned}$$

where we used Fubini's theorem for the last equality. The relation (3.3) leads us to associate

to the velocity distribution the function $\mathbb{R} \ni \xi \mapsto w(\xi)$ defined by

$$(3.4) \quad w(\xi) := \int_{\mathbb{R}} \frac{\rho(v) \mathbb{1}_{[-|v|\pi, |v|\pi]}(\xi)}{|v|} dv = \int_{\mathbb{R} \setminus \left[\frac{-|\xi|}{\pi}, \frac{|\xi|}{\pi} \right]} \frac{\rho(v)}{|v|} dv.$$

Remark 1. From (3.4), we deduce that w is even, that $w \geq 0$. In addition, from (3.4) we deduce that the function $(0, +\infty) \ni \xi \mapsto w(\xi)$ is nonincreasing. Indeed, the integrand is fixed, but the integration interval decreases (in the sense of the inclusion) for $\xi > 0$. Furthermore, it follows from (3.4) that w' is odd and satisfies for any $\xi > 0$

$$w'(\xi) = \frac{-\pi}{\xi} \left(\rho \left(\frac{\xi}{\pi} \right) + \rho \left(\frac{-\xi}{\pi} \right) \right),$$

this derivative being understood in the distribution sense if ρ is just $L^1(\mathbb{R})$. (In which case a primitive of ρ is absolutely continuous, and its classic derivative is almost everywhere equal to ρ .) In addition, from (3.4), we deduce that

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

Thus, $w \in L^1(\mathbb{R})$ and is compactly supported, so that $\sqrt[4]{w} \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$. By assumption, we have that $\rho(v) = 0$ for any $v \in \mathbb{R}$ such that $|v| > |v_{max}|$. Therefore, from (3.4), we deduce that $w(\xi) = 0$ for any $\xi \in \mathbb{R}$ such that $|\xi| \geq |v_{max}|\pi$.

With the help of the function w we can finally formulate the energy that minimizes (3.3) in a closed form.

Definition 3.1. *Given a velocity probability density ρ , we call optimal flutter shutter gain function for ρ any function $\alpha \in L^2(\mathbb{R})$ that minimizes*

$$(3.5) \quad E(\hat{\alpha}) = \|\hat{\alpha}\|_{L^2(\mathbb{R})}^2 \int_{-\infty}^{\infty} \frac{w(\xi)}{|\hat{\alpha}|^2(\xi)} d\xi,$$

where w is linked to ρ by (3.4).

The energy (3.5) is invariant to arbitrary translations and scalings of α , i.e., satisfies $E(C_1 \hat{\alpha}(\cdot) e^{-iC_2 \cdot}) = E(\hat{\alpha})$, for any constants $C_1 \in \mathbb{R} \setminus \{0\}$ and $C_2 \in \mathbb{R}$. Therefore, minimizers of (3.5) among functions $\hat{\alpha} \in L^2(\mathbb{R})$ are not unique, assuming there is one.

Theorem 3.2 (optimal time-continuous flutter shutter gain functions). *Let ρ be a probability density supported on $[-|v_{max}|, |v_{max}|]$, and consider w obtained from ρ by (3.4). A flutter shutter gain function $\alpha \in L^2(\mathbb{R})$ is optimal in terms of MSE (3.5) iff, for some $C > 0$, $\hat{\alpha}$ satisfies $|\hat{\alpha}| = C \sqrt[4]{w}$ on the support of w , and $\hat{\alpha} = 0$ outside the support of w .*

Proof. See Appendix A for the proof. ■

Remark 2. Theorem 3.2 implies that optimal flutter shutter gain functions $\alpha \in L^2(\mathbb{R})$ are bounded, continuous, and band-limited. In addition, (3.4) implies that

$$w(\xi) = \int_{\mathbb{R} \setminus \left[\frac{-|\xi|}{\pi}, \frac{|\xi|}{\pi} \right]} \frac{\rho(v) + \rho(-v)}{2|v|} dv.$$

We deduce that optimal *flutter shutter gain functions* $\alpha \in L^2(\mathbb{R})$ depend only on the even part of ρ . Indeed, the choice of a positive direction for the velocities v is arbitrary and does not change the MSE.

Step 2. Computing optimal flutter shutter codes. Theorem 3.2 gives the formula of optimal *time-continuous flutter shutter gain functions* $\alpha \in L^2(\mathbb{R})$. However, these time-continuous *flutter shutter gain functions* must be turned into a piecewise constant function to be implementable by a *numerical flutter shutter*.

Theorem 3.3 (optimal flutter shutter codes in terms of MSE). *Let ρ, w be as in Theorem 3.2, and let Δt be such that $|v_{max}|\Delta t \leq 1$. Consider a sequence $(\alpha_k)_k \in \ell^2(\mathbb{Z})$ and the $L^2(\mathbb{R})$ piecewise constant flutter shutter gain function uniquely associated with $(\alpha_k)_k$,*

$$(3.6) \quad \alpha(t) = \sum_{k \in \mathbb{Z}} \alpha_k \mathbb{1}_{[k\Delta t, (k+1)\Delta t)}(t),$$

where $\Delta t > 0$. A sequence $(\alpha_k)_k \in \ell^2(\mathbb{Z})$ is optimal with respect to the MSE (3.5) iff $(\alpha_k)_k$ satisfies

$$\left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right| = C \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}}{\sqrt{\text{sinc}\left(\frac{\xi}{2\pi}\right)}}$$

for some fixed $C > 0$ and for any $\xi \in [-\pi, \pi]$.

In addition, the real values α_k , for $k \in \mathbb{Z}$, explicitly given by

$$\alpha_k = \frac{1}{2\pi} \int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)} \cos(ks)}{\sqrt{\text{sinc}\left(\frac{\xi}{2\pi}\right)}} d\xi,$$

define an optimal flutter shutter gain function $\alpha \in L^2(\mathbb{R})$ with respect to the energy (3.5), among all real-valued functions in $L^2(\mathbb{R})$ of the form (3.6).

Proof. See Appendix B for the proof. ■

This theorem directly links optimal *flutter shutter codes* with the distribution of the camera-scene velocities. It produces *flutter shutter codes* that are implementable with a *numerical flutter shutter*. However, in general nothing guarantees that the codes will be nonnegative. Thus, in general one cannot expect the produced code to be implementable with an *analog flutter shutter*.

4. The reverse path: From flutter shutter gain functions to their underlying velocity distributions. By the formulae of the previous section we are now able to check whether a *flutter shutter gain function* $\alpha \in L^2(\mathbb{R})$ is optimal for some velocity distribution, and to compute its underlying velocity distribution. Remark 2 above implies that from a given optimal *flutter shutter gain function* $\alpha \in L^2(\mathbb{R})$, associated with some unknown probability density ρ , one can recover only the even part of ρ , namely $\frac{\rho(\cdot) + \rho(-\cdot)}{2}$. Indeed, the optimal *flutter shutter gain function* $\alpha \in L^2(\mathbb{R})$ depends only on the even part of ρ . Thus, throughout this section we shall assume that ρ is even.

Theorem 4.1 (an optimality test for flutter shutter codes and a formula for their underlying velocity distribution). *Let $\alpha \in L^2(\mathbb{R})$ be a flutter shutter gain function. If α is time-continuous*

(not of the form of (3.6)), then α is optimal in the sense of (3.5) for some velocity distribution $\rho(v)$ only if the function $(0, +\infty) \ni \xi \mapsto |\hat{\alpha}(\xi)|$ is nonincreasing. Moreover, if $(0, +\infty) \ni \xi \mapsto |\hat{\alpha}(\xi)|$ is nonincreasing, then

$$(4.1) \quad \rho(v) = -\frac{v}{2}w'(\pi v) = -\frac{vC}{2\pi} (|\hat{\alpha}|^4)'(\pi v), \quad v \neq 0,$$

where C is a positive constant, w is given by (3.4), and the derivatives in (4.1) are understood in the distribution sense if ρ is just in $L^1(\mathbb{R})$. Assume that $\alpha \in L^2(\mathbb{R})$ has the form of (3.6) and that $|v_{\max}|\Delta t \leq 1$. Then α is optimal in the sense of (3.5) only if

$$\left(0, \frac{\pi}{\Delta t}\right] \ni \xi \mapsto \frac{|\hat{\alpha}(\xi)|^4}{\left|\operatorname{sinc}\left(\frac{\xi\Delta t}{2\pi}\right)\right|^2}$$

is nonincreasing. Moreover, if $(0, \frac{\pi}{\Delta t}] \ni \xi \mapsto \frac{|\hat{\alpha}(\xi)|^4}{\left|\operatorname{sinc}\left(\frac{\xi\Delta t}{2\pi}\right)\right|^2}$ is nonincreasing, then

$$(4.2) \quad \rho(v) = -\frac{v}{2}w'(\pi v) = -\frac{vC}{2} \left(\frac{|\hat{\alpha}(\xi)|^4}{\left|\operatorname{sinc}\left(\frac{\xi\Delta t}{2\pi}\right)\right|^2} \right)'(\pi v) \quad \text{for } v \in \left[\frac{-1}{\Delta t}, \frac{1}{\Delta t}\right] \setminus \{0\},$$

where C is a positive constant, w is given by (3.4), and the derivatives in (4.2) are understood in the distribution sense if ρ is just in $L^1(\mathbb{R})$.

Proof. See Appendix C for the proof. ■

Note that for any discrete α of the form $\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t, (k+1)\Delta t)}(t)$ we have $\alpha \in L^2(\mathbb{R})$, and therefore Theorem 4.1 applies. Moreover, Theorem 4.1 gives a direct algorithm that computes ρ . This numerical method is detailed in Algorithm 1.

Remark 3. As we shall see in section 6.2, most classic codes do not strictly satisfy the conditions of Theorem 4.1. Fortunately, for these codes the set where $|\hat{\alpha}|$ is increasing has small measure. In addition, $|\hat{\alpha}|$ is small on this set. Thus, we can apply Algorithm 1 by modifying α (or w) only slightly by replacing (4.2) by

$$(4.3) \quad \rho(v) = -\frac{v}{2\pi}w'(\pi v)\mathbb{1}_{\{-vw'(\pi v) \geq 0\}}(v) \quad \text{for } v \neq 0$$

and normalizing ρ so that $\int \rho = 1$.

Algorithm 1, for which [44] also provides a peer-reviewed implementation and online demo, will be applied in section 6.2 to several classic (patented or not) codes to uncover their underlying velocity distribution $\rho(v)$.

5. Optimal snapshot and comparison flutter shutter/snapshot definitions. The formulae that give the exposure time and MSE of an optimal *snapshot* are needed to compare optimal *flutter shutters* with optimal *snapshots* for the same velocity distribution. The framework of section 3 is applicable, a *snapshot* being a *flutter shutter gain function* of the form $\alpha = \mathbb{1}_{[0, \Delta t]}$.

ALGORITHM 1. Pseudocode computing the velocity distribution associated with a given code.

input : a *flutter shutter code* $(\alpha_k)_{k \in \{0, \dots, L-1\}}$, Δt the time step of the *flutter shutter*.

output: underlying probability density ρ for which the code is optimal.

1. compute $\hat{\alpha}(\xi)$ ($= \Delta t \operatorname{sinc}(\frac{\xi \Delta t}{2\pi}) e^{-\frac{i\xi \Delta t}{2}} \sum_{k=0}^{L-1} \alpha_k e^{-ik\xi \Delta t}$; see Table 3);
2. compute the function w defined in (3.4) by

$$w(\xi) = \left(\frac{|\hat{\alpha}(\xi)|^4}{|\operatorname{sinc}(\frac{\xi \Delta t}{2\pi})|^2} \right) (\xi)$$

(see Theorem 3.2);

3. estimate $w'(x)$ by $w'(x) \approx \frac{w(x+h) - w(x)}{h}$;
4. compute $\rho(v) = -\frac{v}{2\pi} w'(\pi v) \mathbb{1}_{\{-vw'(\pi v) \geq 0\}}(v)$, $v \in [\frac{-1}{\Delta t}, \frac{1}{\Delta t}] \setminus \{0\}$;
5. normalize so that $\int_{\mathbb{R}} \rho(v) dv = 1$.

Following the formalism of section 3, given a probability density $\rho(v)$ for the camera-scene velocities v , we want to minimize the MSE (3.1) among all *flutter shutter gain functions* of the form of $\alpha = \mathbb{1}_{[0, \Delta t]}$. This leads to minimizing

$$E(\Delta t) = \int \int_{-\pi}^{\pi} \frac{\xi^2}{\sin^2(\frac{\xi v \Delta t}{2})} \frac{v^2 \Delta t}{4} \rho(v) d\xi dv,$$

which therefore yields

$$E'(\Delta t) = \int \int_{-\pi}^{\pi} \frac{v^2 \xi^2 \left(\sin(\frac{\xi v \Delta t}{2}) - \xi v \Delta t \cos(\frac{\xi v \Delta t}{2}) \right)}{4 \sin^3(\frac{\xi v \Delta t}{2})} \rho(v) d\xi dv.$$

Notice that $\Delta t < \frac{2}{|v_{max}|}$ is necessary for the invertibility of a *snapshot* for any velocity v such that $|v| \leq |v_{max}|$. The existence and uniqueness of such a *snapshot* is formalized in the following proposition.

Proposition 5.1 (and definition: optimal snapshot). *Let ρ be as in Theorem 3.2. We consider the use of the unique Δt^* that minimizes the MSE,*

$$(5.1) \quad E(\Delta t) = \int \int_{-\pi}^{\pi} \frac{\xi^2}{\sin^2(\frac{\xi v \Delta t}{2})} \frac{v^2 \Delta t}{4} \rho(v) d\xi dv,$$

to be an optimal snapshot. In addition, when $\rho(v)$ is uniform over $[-|v_{max}|, |v_{max}|]$, we have that $\Delta t^ |v_{max}|$ is constant ($\Delta t^* |v_{max}| \approx 1.42$).*

Similarly, when $\rho(v)$ is a truncated Gaussian, i.e., $\rho(v) \propto \mathbb{1}_{[-4\sigma, 4\sigma]}(v) \exp(\frac{-v^2}{2\sigma^2})$, then $\Delta t^ |v_{max}|$ is also constant ($\Delta t^* |v_{max}| \approx 1.94$).*

Proof. See Appendix D for the proof. ■

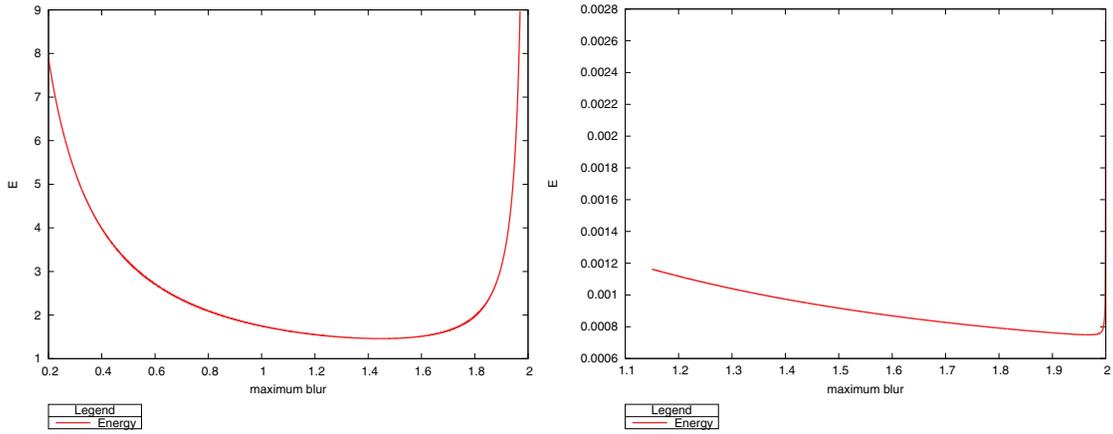


Figure 4. The energies E defined in (5.1). The x-axis is maximum blur ($|v_{max}|\Delta t$) in pixels, and the y-axis the value of E , i.e., the MSE of the deconvolved image. Left: For a uniform velocity distribution $\mathcal{U}([-1, 1])$, the minimum is reached for a maximum blur of approximately 1.44 pixels. Right: For a truncated Gaussian velocity distribution $\rho(v) \propto \mathbb{1}_{[-1, 1]}(v) \exp\left(-\frac{v^2}{2(1/4)^2}\right)$, the minimum is reached for a maximum blur of approximately 1.96 pixels.

Proposition 5.1 means that for a standard camera, assuming a uniform or a truncated Gaussian motion model, the exposure time should be tuned so that the blur support never exceeds the constant $\Delta t^*|v_{max}|$. This constant depends on the motion model, as illustrated in Figure 4. The zero of $E'(\Delta)$ was computed numerically.

We now turn to the second goal of this section, namely, giving the definitions needed to theoretically analyze the RMSE gain of the *numerical flutter shutter* with respect to the *snapshot*.

Gain evaluation. The RMSE gain depends on the velocity v . Thus, it is useful for the analysis to define the gain $G(v)$ at velocity v of the optimal *flutter shutter* with respect to the optimal *snapshot* in terms of RMSE by the ratio

$$(5.2) \quad G(v) = \frac{\text{RMSE}(\text{snapshot})}{\text{RMSE}(\text{flutter})} = \sqrt{\frac{\int_{-\pi}^{\pi} \frac{1}{\Delta t^* \left| 2 \frac{\sin(\frac{\xi v \Delta t^*}{2})}{\xi v \Delta t^*} \right|^2} d\xi}{\int_{-\pi}^{\pi} \frac{\|\alpha\|_{L^2(\mathbb{R})}^2}{|\hat{\alpha}(\xi v)|^2} d\xi}},$$

where v is the support of the velocity distribution ρ and Δt^* is the exposure time of the optimal *snapshot*. This optimal *snapshot* is defined in Proposition/definition 5.1. Recall that a piecewise constant *flutter shutter gain function* α has the generic form (3.6), $\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t, (k+1)\Delta t)}(t)$. Moreover, its time step is Δt , and $\Delta t \neq \Delta t^*$ in general. The coefficients α_k of the *flutter shutter code* are explicitly given by Theorem 3.3 as a function of the velocity distribution ρ . The average gain of the *flutter shutter* in terms of RMSE with respect to the optimal *snapshot* is defined by

$$(5.3) \quad \int_{\mathbb{R}} G(v) \rho(v) dv,$$

and the associated standard deviation is

$$(5.4) \quad \sqrt{\int_{\mathbb{R}} \left| G(v) - \int_{\mathbb{R}} G(u)\rho(u)du \right|^2 \rho(v)dv}.$$

6. Numerical experiments. Section 6.1 gives the optimal *flutter shutter codes* and computes the gain of the optimal *flutter shutter* with respect to the optimal *snapshot* for three natural velocity distributions: a (truncated) Gaussian velocity model, a uniform velocity model, and a trimodal “traffic-like” velocity model. Section 6.2 gives the reverse engineering of classic *flutter shutter codes* in the literature. We refer to [44] for a detailed pseudocode and an implementation.

6.1. Simulations on optimized codes. The goal of this section is to numerically explore three natural velocity distributions. For each velocity distribution, we give the corresponding optimal code and compare its efficiency in terms of RMSE with the optimal *snapshot*.

Recall that the parameters of a *flutter shutter* are as follows: (1) L the length of the code $(\alpha_k)_{k \in \{0, \dots, L-1\}}$, (2) the velocity motion model $\rho(v)$, and (3) the time step Δt of the *flutter shutter gain function* α , so that $L\Delta t$ is the total exposure time of the *flutter shutter*. The parameter for the optimal *snapshot* is only the velocity model $\rho(v)$. The optimal *snapshot* provides Δt^* , the optimal exposure time for a standard camera, i.e., without using a *flutter shutter*. In order to simplify the comparison with the code of Agrawal et al. [6, 39, 37, 31, 3], all experiments are made with the code length $L = 52$ used in those papers.

Three velocity motion models are considered and compared: a truncated Gaussian (in section 6.1.1), a uniform velocity distribution model (in section 6.1.2), and a three modal velocity distribution (in section 6.1.3). To facilitate the comparison of the *flutter shutter* with the optimal *snapshot*, the time step Δt of the *flutter shutter* was chosen so that $L\Delta t = c\Delta t^*$, where $c \in \mathbb{N}^+$.

The codes of Figures 5 and 7 are the optimal choices for the *flutter shutter codes* $(\alpha_k)_{k \in \{0, \dots, L-1\}}$ in terms of MSE (see Theorem 3.3). These experiments compare two strategies using a finite exposure time, which is mandatory for a practical solution. This comparison was already performed in [48] for cases when the velocity v_0 is known, i.e, $\rho(v) = \delta_{v_0}(v)$. In that case the optimal *numerical flutter shutter code* is derived from a sinc function. The optimal *snapshot* has an exposure time Δt^* satisfying $|v_0|\Delta t^* \approx 1.0909$, and the RMSE gain of the optimal *flutter shutter* with respect to this optimal *snapshot* is a 1.17 factor [48]. In the case of a probabilistic velocity distribution, the RMSE gains should therefore be compared to this 1.17 bound which optimizes the worst-case scenario, based on the maximal velocity. Without loss of generality we shall normalize the maximal velocity to $v_{max} = 1$. Indeed, *ceteris paribus* a scale change of the velocity model simply results in a scale change of the function $w(\xi)$ and therefore in a zoom of the code.

6.1.1. Optimal codes, Gaussian velocity model. This section provides the optimal codes for a truncated Gaussian $\mathcal{N}(0, \frac{1}{4})$ velocity motion model explicitly given by

$$(6.1) \quad \rho(v) \propto \mathbb{1}_{[-1,1]}(v) \exp\left(\frac{-v^2}{2\left(\frac{1}{4}\right)^2}\right).$$

Optimal *flutter shutter codes* are explicitly depicted in Figures 5(a) and 5(c). The Fourier transforms of the corresponding *flutter shutter gain functions* are given in Figures 5(b) and 5(d). In Figure 5 between the two plots at the top (Figures 5(a) and 5(b)) and the two at the bottom (Figures 5(c) and 5(d)), the discretization step Δt of the *flutter shutter gain functions* is adapted to meet a given value of the exposure time factor c . For Figures 5(a) and 5(b) we have $c = 5$, while for Figures 5(c) and 5(d) we have $c = 10$. (Recall that in these experiments we have $52\Delta t = c\Delta t^*$.) Thus, the support of the *flutter shutter gain function* doubles between the top and bottom parts of Figure 5. The green curves show $\sqrt[4]{w(\xi)}$ to gauge the quality of the finitely supported approximation of $\sqrt[4]{w(\xi)}$ by the *flutter shutter code*. The function $\sqrt[4]{w(\xi)}$ does not change. Indeed, the function $w(\xi)$ defined in (3.4) depends only on the velocity motion model and is fixed. Notice that the approximation is slightly better for the larger exposure factor $c = 10$, shown in the bottom part of Figure 5. This is no surprise. Indeed, since $\widehat{w(\xi)}$ has compact support, the ideal time-continuous *flutter shutter gain function* $\sqrt[4]{w(\xi)}$ is supported on \mathbb{R} .

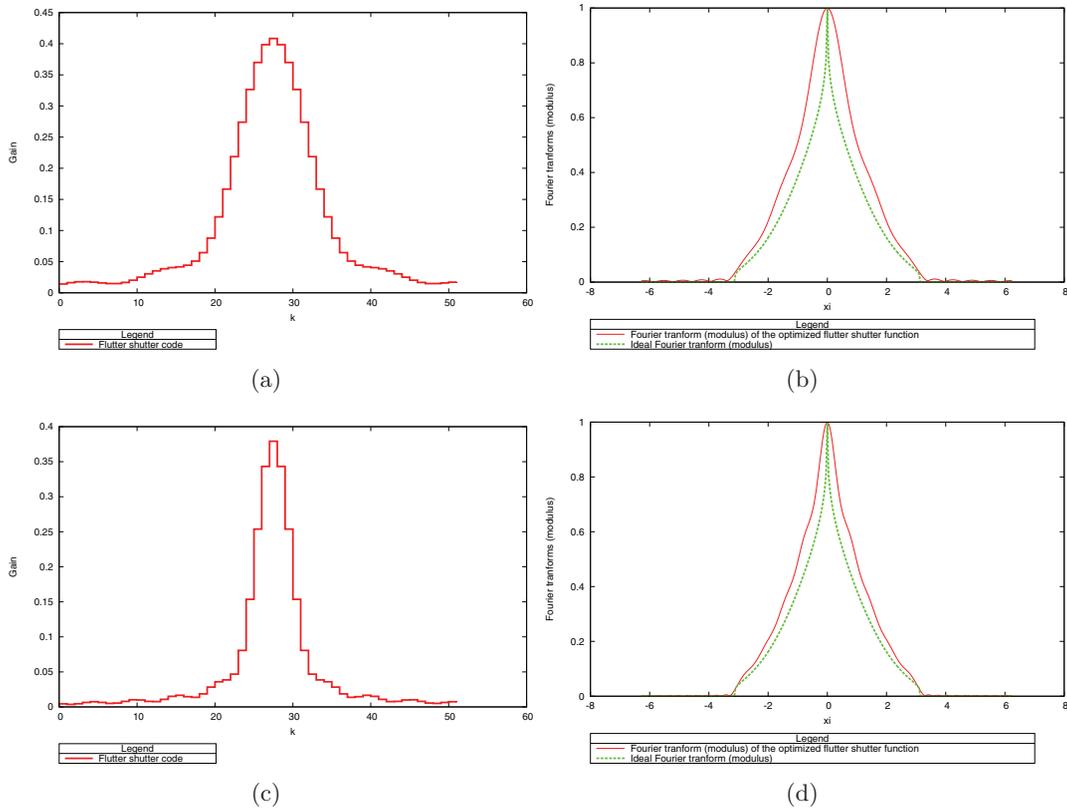


Figure 5. Codes obtained for a truncated Gaussian velocity density explicitly given in (6.1). Left: The flutter shutter code coefficients α_k , using an exposure time of 5 (a) or 10 (c) times larger than for the optimal snapshot. Right: The modulus of the corresponding Fourier transforms (red), and the Fourier transforms of the optimal time-continuous flutter shutter gain function $\sqrt[4]{w(\xi)}$ defined in (3.4) (green). The convergence is quite good, even for small exposure time factors. The exact equality $\hat{\alpha}(\xi) = \sqrt[4]{w(\xi)}$ requires an infinitely supported flutter shutter gain function because $\sqrt[4]{w(\xi)}$ has compact support.

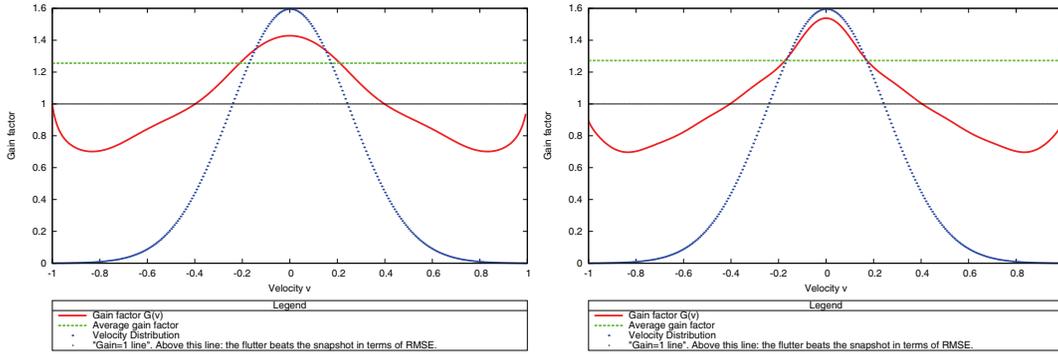


Figure 6. Red: The gain $G(v)$ in terms of RMSE (defined by (5.2)) of the optimal flutter shutter code with respect to the optimal snapshot for the truncated Gaussian velocity distribution. Results are shown for exposure time factors of 5 (left) or 10 (right). The dotted blue curve represents the probability density of the truncated Gaussian velocity distribution. The green curves show the average gain μ as it is defined in (5.3). The optimization permits us to concentrate the gain on most probable velocities, as expected. On average the gain is substantial compared to the bound of [48] that optimizes the maximal velocity (worst case).

Figure 6 provides the comparison with the optimal *snapshot* in terms of RMSE. For the velocities v in the support of the velocity motion model, the red curves of Figure 6 show the RMSE gain $G(v)$ (see (5.2)) of the *flutter shutter* with respect to the optimal *snapshot*. Figure 6(left) corresponds to an exposure factor $c = 5$, meaning that the *flutter shutter* integrates five times longer than the optimal *snapshot*. Figure 6(right) corresponds to $c = 10$.

The dotted blue curve provides the probability of the velocity according to the velocity distribution. It permits us to verify that the optimization concentrates the RMSE gain on the most probable velocities. On the other hand, for higher but less likely velocities v the optimized *flutter shutter* performs worse than the optimal *snapshot*. The green line shows the expectation of $G(v)$ (see (5.3)).

Table 4 provides both the average gain, defined by (5.3), and its associated standard deviation, defined by (5.4). It permits us to measure “how risky” the optimization is, i.e., how the gain will vary when one observes velocities according to the motion model explicitly given in (6.1). Notice that the asymptotic bound of [48] is beaten by approximately 50%.

Table 4

Average gain of the optimized flutter shutter compared to the optimal snapshot, assuming a truncated Gaussian velocity distribution explicitly given in (6.1). As supposed from Figure 6, the gain is substantial, and the increase is approximately 50% compared to the asymptotic of [48].

Exposure time factor c	5	10
Code length L	52	52
Average gain μ from (5.3)	1.2556	1.2721
Standard deviation σ from (5.4)	0.1700	0.1994

6.1.2. Optimal codes, uniform motion model. This section provides the optimal codes for a $\mathcal{U}[-1, 1]$ velocity motion model explicitly given by

$$(6.2) \quad \rho(v) = \frac{1}{2} \mathbb{1}_{[-1,1]}(v).$$

The setup is exactly the same as in section 6.1.1. Optimal *flutter shutter codes* are explicitly depicted in Figures 7(a) and 7(c). The Fourier transforms of the corresponding *flutter shutter gain functions* are given in Figures 7(b) and 7(d). For Figures 7(a) and 7(b) we have $c = 5$, while for the Figures 7(c) and 7(d) we have $c = 10$, as in section 6.1.1.

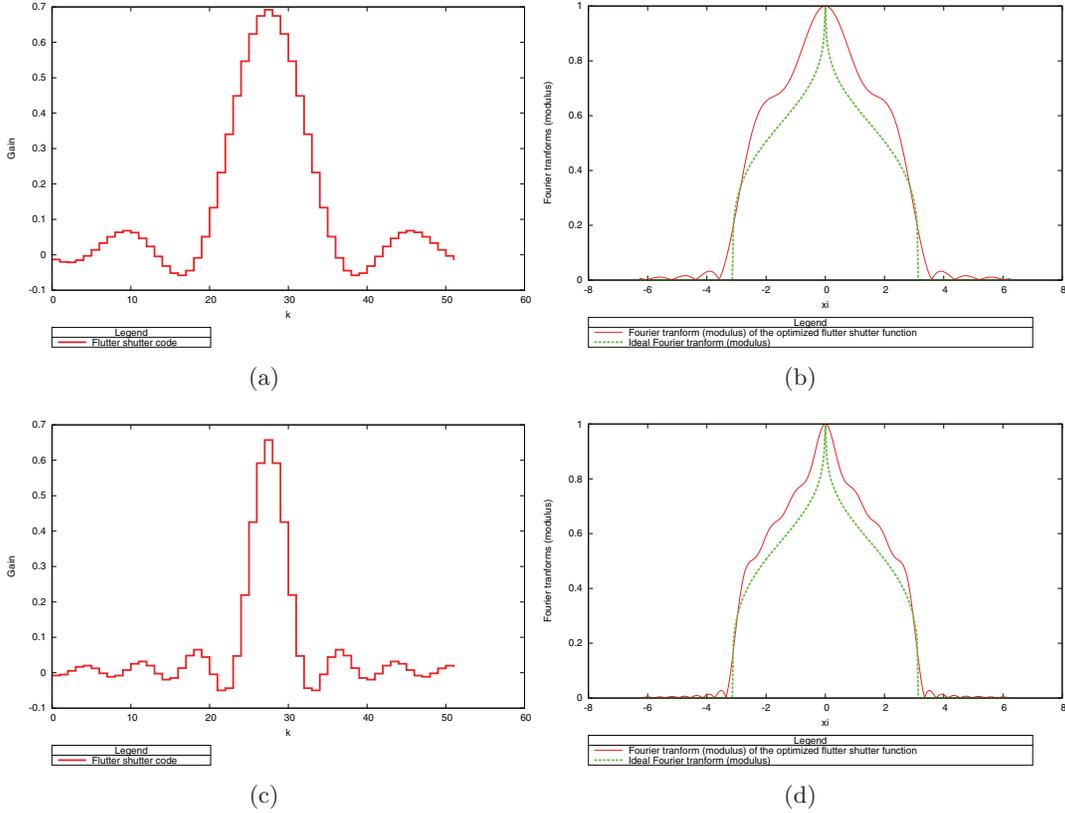


Figure 7. Codes α_k obtained assuming a uniform density for the velocities explicitly given in (6.2). Left: The flutter shutter code coefficients α_k , using an exposure time of 5 (a) or 10 (c) times larger than for the optimal snapshot. Right: The modulus of their corresponding Fourier transforms (red), and the Fourier transform of the optimal time-continuous flutter shutter gain function $\sqrt[4]{w(\xi)}$ defined in (3.4) (green).

Figure 8 provides the comparison with the optimal *snapshot* in terms of RMSE. The dotted blue curve provides the probability of the velocity according to the velocity distribution. The green line shows the average of $G(v)$ (defined in (5.2)), taking the velocity motion model ρ into account, as it is defined in (5.3).

Table 5 provides both the average gain, defined by (5.3), and its associated standard deviation, defined by (5.4), as in section 6.1.1. The gain is negligible.

6.1.3. Optimal codes, trimodal motion model. This section provides the optimal codes for a trimodal “traffic-like” velocity motion model explicitly given by

$$(6.3) \quad \rho(v) = \rho_0 \delta_0(v) + \frac{1 - \rho_0}{2} \delta_{15}(v) + \frac{1 - \rho_0}{2} \delta_{-15}(v).$$

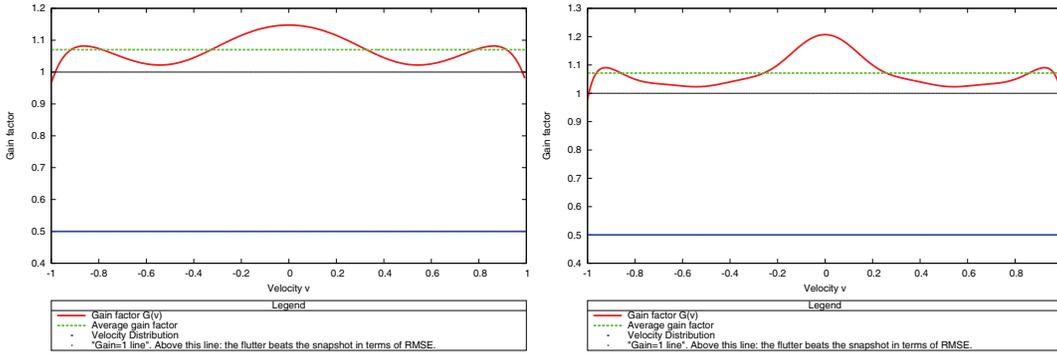


Figure 8. In red: The gain $G(v)$ in terms of RMSE (defined by (5.2)) of the optimal flutter shutter code with respect to the optimal snapshot for the uniform velocity distribution explicitly given in (6.2). Results are shown for exposure time factors of 5 (left) and 10 (right). The dotted blue curve represents the probability density $\rho(v) = \frac{1}{2}\mathbb{1}_{[-1,1]}(v)$ of the uniform velocity distribution. The green curves show the average gain μ as it is defined in (5.3).

Table 5

Average gain of the optimized flutter shutter compared to the snapshot, assuming a uniform density for the velocities explicitly given in (6.2). As could already be supposed from Figure 8, this gain is not significant.

Exposure time factor	5	10
Code length L	52	52
Average gain μ from (5.3)	1.0701	1.0715
Standard deviation σ from (5.4)	0.0416	0.0530

The setup is the same as in section 6.1.1, except for the exposure time factors c . We show optimal *flutter shutter codes* in the case where $\rho_0 = 0.99$. However, we provide the gains and standard deviations for several values of the parameter ρ_0 in Table 6.

Optimal *flutter shutter codes* are explicitly depicted in Figures 9(a) and 9(c). The Fourier transforms of the corresponding *flutter shutter gain functions* are given in Figures 9(b) and 9(d). For Figures 9(a) and 9(b) we have $c = 9$, while for Figures 9(c) and 9(d) we have $c = 25$.

Figure 10 provides the comparison with the optimal *snapshot* in terms of RMSE. The dotted blue curve provides the probability of the velocity according to the velocity distribution. The green line shows the average of $G(v)$ (defined in (5.2)), taking the velocity motion model ρ into account as it is defined in (5.3).

Table 6 provides both the average gain defined by (5.3) and its associated standard deviation defined by (5.4), as in section 6.1.1. Notice that the 1.17 asymptotic bound of [48] that optimizes the worst case, i.e., the maximal velocity, is largely beaten.

6.2. A reverse engineering of classic *flutter shutter codes*. This section provides the underlying velocity distribution ρ of classic *flutter shutter codes* of the literature. However, Algorithm 1 is applicable to any *flutter shutter code*. Algorithm 1, with the variant given by (4.3), is used. We normalize Δt in the definition of the *flutter shutter gain function* (see Table 3). This means that the velocities are expressed in pixels *per* Δt . Thus, the x -axis of Figures 11, 12, and 13 has the range $[-1, 1]$.

We proceed first to the reverse engineering of the Agrawal et al. *flutter shutter code* [38,

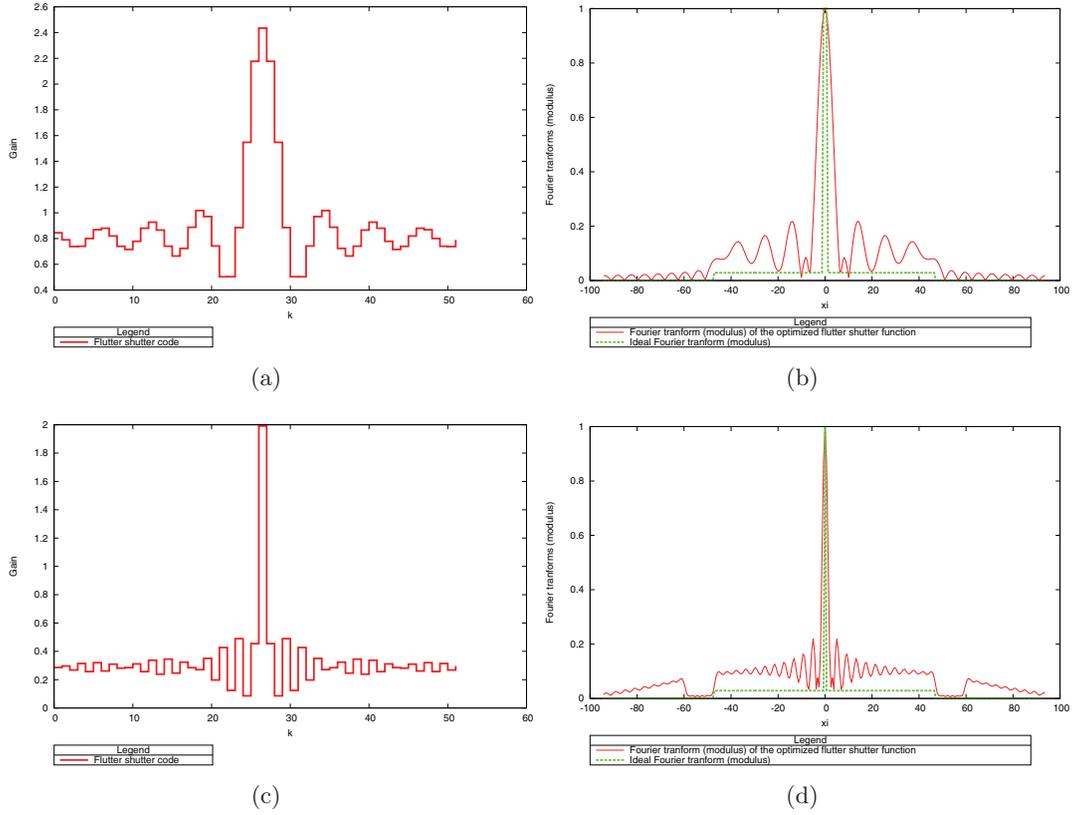


Figure 9. Codes obtained for a truncated Gaussian velocity density explicitly given in (6.3) with $\rho_0 = 0.99$. Left: The flutter shutter code coefficients α_k , using an exposure time of 9 (a) or 25 (c) times larger than for the optimal snapshot. Right: The modulus of their corresponding Fourier transform (red), and the Fourier transform of the optimal time-continuous flutter shutter gain function $\sqrt[4]{w(\xi)}$ defined in (3.4) (green).

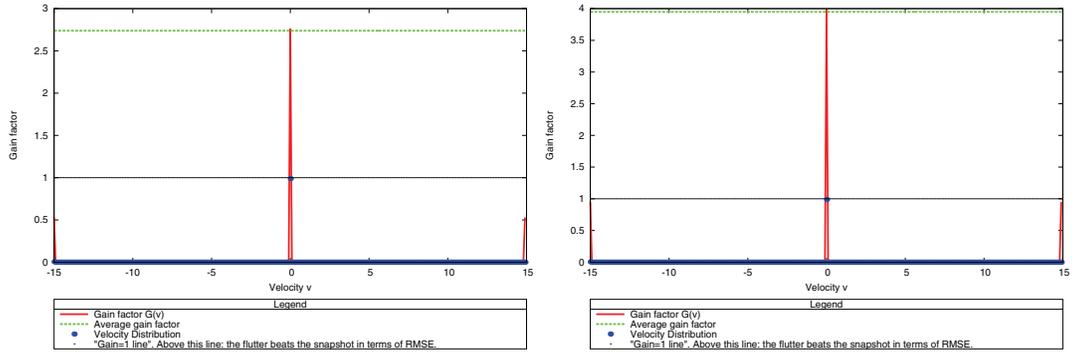


Figure 10. In red: The gain $G(v)$ in terms of RMSE (defined by (5.2)) of the optimal flutter shutter code with respect to the optimal snapshot for the trimodal velocity distribution. Results are shown for exposure time factors of 9 (left) and 25 (right). The dotted blue curve represents the probability density (6.3) with $\rho_0 = 0.99$ of the trimodal velocity distribution considered. The green curves show the average gain μ as defined in (5.3).

Table 6

Average gain of the optimized flutter shutter compared to the snapshot, assuming a trimodal velocity model given by (6.3). As guessed from Figure 10, when $\rho_0 = 0.99$, the gain is substantial and the increase is of approximately 230% compared to the asymptotic of [48]. The standard deviation can be nonnegligible, depending on the value of ρ_0 . Note that if $\rho_0 = 0$ or $\rho_0 = 1$, we retrieve the deterministic case, and the standard deviation would therefore be equal to zero.

Exposure time factor	9	25	25	25	25	25	25
Code length L	52	52	52	52	52	52	52
Value of the parameter ρ_0	0.99	0.99	0.9	0.8	0.7	0.6	0.5
Average gain (5.3)	2.7385	3.9501	2.9971	2.5841	2.2883	2.0073	1.7670
Standard deviation (5.4)	0.2220	0.3016	0.7966	0.8753	0.9221	0.8882	0.8125

p. 799] and patent application [39] in Figure 11(a). Note that the y -axis of this panel is log scaled. This distribution means that there is a high probability that the scene is still, and that more or less uniformly distributed velocity motions occur on a certain interval of velocities. However, this is an unlikely model for a camera motion, due to the strange fluctuations of the velocity distribution. The velocity distribution of another *flutter shutter code* of Agrawal and Xu [6, p. 7] is given in Figure 11(b). It is mainly concentrated on small velocities.

The reverse engineering of the Agrawal et al. *flutter shutter codes* published in [4, p. 2566] (resp., [3, p. 5]) is shown in Figure 12(a) (resp., Figure 12(b)).

Another example is the McCloskey code [27, p. 321], shown in Figure 13(a). The same scheme can be applied to the “standardized” *snapshot*, i.e., $\alpha(t) = \mathbb{1}_{[0,1]}(t)$, to estimate the underlying probability density of a classic camera. This example is given in Figure 13(b), where we deduce that it is optimal for relatively broad intervals centered at approximately $|v| = 1$. Among the velocity densities of Figures 11(b), 13, and 12, the velocity distribution of the *snapshot* shown in Figure 13(b) is the most convincing one.

7. Conclusion. Knowledge of a stochastic velocity model was shown to increase the expected RMSE gain of a *flutter shutter* with respect to an optimal *snapshot*. The use of this stochastic velocity model allows us to beat the 1.17 bound of [46, 48] that was established for known velocities. Indeed, the use of a stochastic velocity model permits us to optimize the average case, in terms of MSE, at minimal risk. This is in contrast with [48], which optimized only the worst case, i.e., in a practical situation the maximal velocity to guarantee the invertibility for every v such that $|v| \leq |v_{max}|$.

We proved a mathematical formula that allows us to perform a reverse engineering of several classic and patented *flutter shutter codes*. It computes the underlying probability density for which these codes are optimal.

Given any distribution for the expected velocities, the theory predicts, by closed formulae, the gain in terms of MSE of the optimal *flutter shutter* compared to the optimal *snapshot*. The combination of the forward, backward, and RMSE prediction provides a complete toolbox to optimize, analyze, and predict the gain of the *flutter shutter* with respect to its corresponding optimal *snapshot* in terms of RMSE. This toolbox allows us to decide whether the *flutter shutter* paradigm is useful for any given application. According to the results of this paper, for every given distribution $\rho(v)$ for the relative camera/scene velocity v , there exists an optimal *flutter shutter code*. This optimal *flutter shutter code* is invertible for all velocities v in the

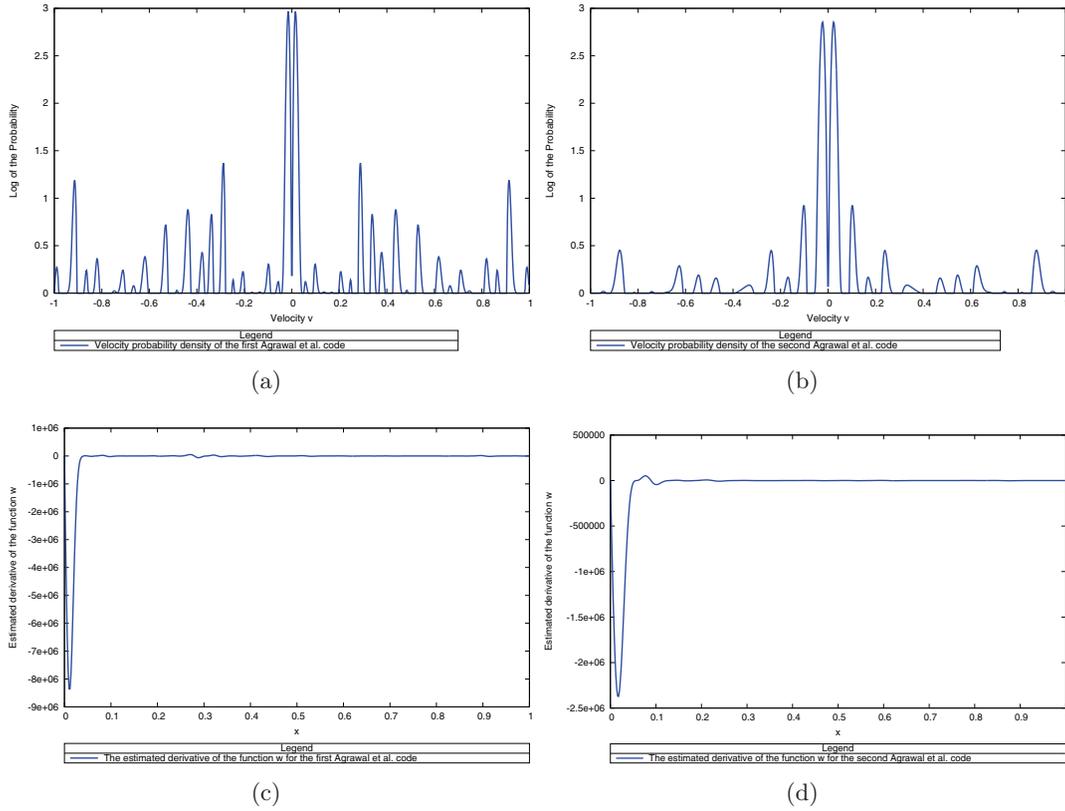


Figure 11. Top: The velocity probability densities ρ associated with the codes of Agrawal, Raskar, et al.; the x -axis is the velocity (in signed pixels per Δt), and the y -axis is the logarithm of the velocity probability densities ($\log(1 + \rho(v))$). (a) uses the code published in [38, p. 799] and patent application [39]; (b) uses the code published in [6, p. 7], which corresponds to an attempt to optimize both the MSE and the a posteriori velocity estimation. (c), (d) The estimation of the w' function. (Recall that w is defined in (3.4).) The x -axis represents x , and the y -axis the estimated $w'(x)$ from the flutter shutter code coefficients.

support of ρ . The gain of the optimal *flutter shutter* with respect to the optimal *snapshot* in terms of MSE can be significant, e.g., for Gaussian (resp., trimodal) velocity distribution considered in section 6.1.1 (resp., section 6.1.3). *A fortiori* this gain is very significant when compared to the 1.17 asymptotic bound that was proved in [46, 48] for fixed known velocity v . Yet, such gains can be attained only for very specific velocity distributions, e.g., the trimodal velocity distribution of section 6.1.3. For more generic cases like the Gaussian model, the RMSE gain of the *numerical flutter shutter* is a moderate 1.27 factor, while the exposure time is multiplied by a factor of 10 compared to the optimal *snapshot*. These RMSE gains are obtained under the following assumptions:

1. The convolution model is valid. In consequence, it must be assumed that the motion blur, i.e., the velocity, is constant everywhere in the image. This rules out many more complex motion blurs, e.g., camera shakes [53].
2. An accurate a posteriori estimation of the velocity can be obtained. Indeed, the RMSE computations are valid under the assumption that the deconvolution kernel is known.

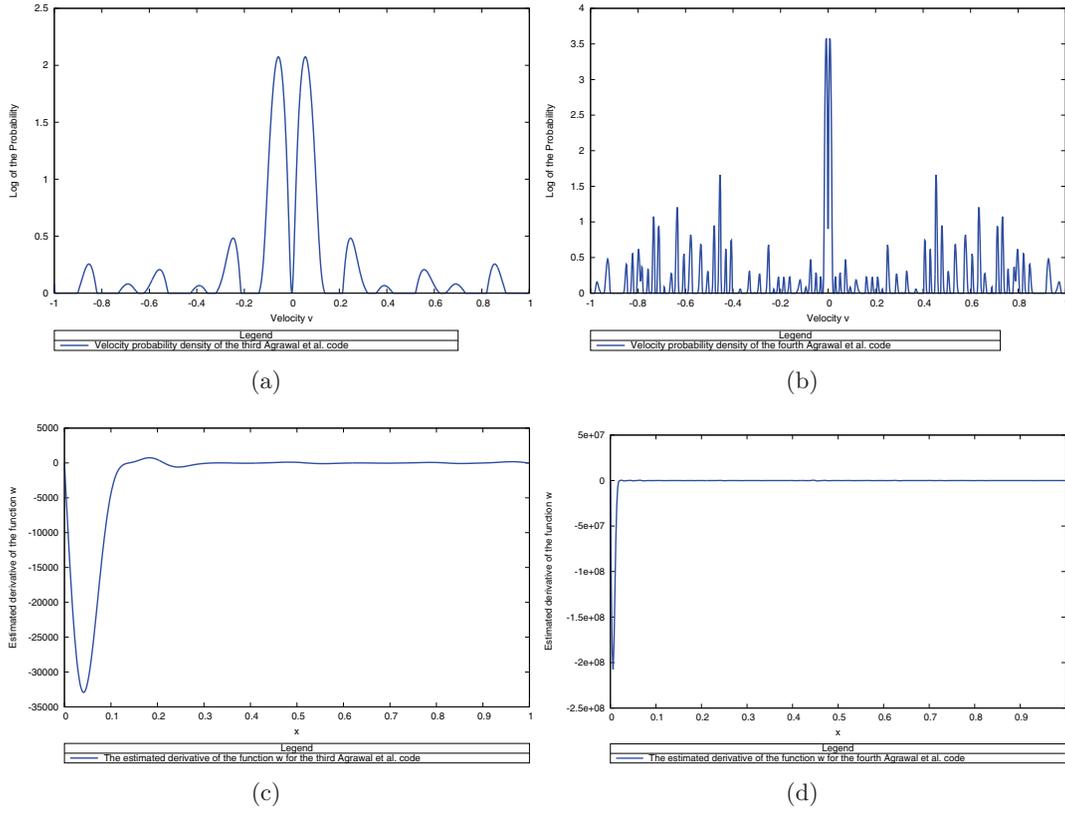


Figure 12. Top: The velocity probability densities ρ associated with the codes of Agrawal, Raskar, et al.; the x -axis is the velocity (in signed pixels per Δt), and the y -axis is the logarithm of the velocity probability densities ($\log(1 + \rho(v))$). (a) The code published in [4, p. 2566]; (b) the code published in [3, p. 5]. Bottom: The estimation of the w' function. (Recall that w is defined as in (3.4).) The x -axis represents x , and the y -axis the estimated $w'(x)$ from the flutter shutter code coefficients.

The RMSE gains of the *numerical flutter shutter* also yield upper performance bounds for the RMSE gains of the *analog flutter shutter* compared to the optimal *snapshot*. Indeed, the *numerical flutter shutter* always yields a higher gain [48] simply because it relaxes several constraints of the *analog flutter shutter* setup.

Practical conclusion. The above numerical results and practical considerations are restricted to locally or globally straight motions, and therefore to short exposures. As we mentioned in section 1, the *flutter shutter* setup imposes itself only under the condition that no shorter time exposure is possible, and/or that the size of the acquired data and/or its processing time must be so severely constrained that storing and fusing successive frames is simply not possible. This sends us back to the conception of special high-speed cameras with drastic memory constraints. We deduce that, under the assumption of a globally straight motion, the *flutter shutter* is the following:

- Useless unless we cannot keep all images. (Yet the *numerical flutter shutter* with a sinc code can be used successfully as a temporal filter, provided the maximal velocity is controlled [45, 49].)

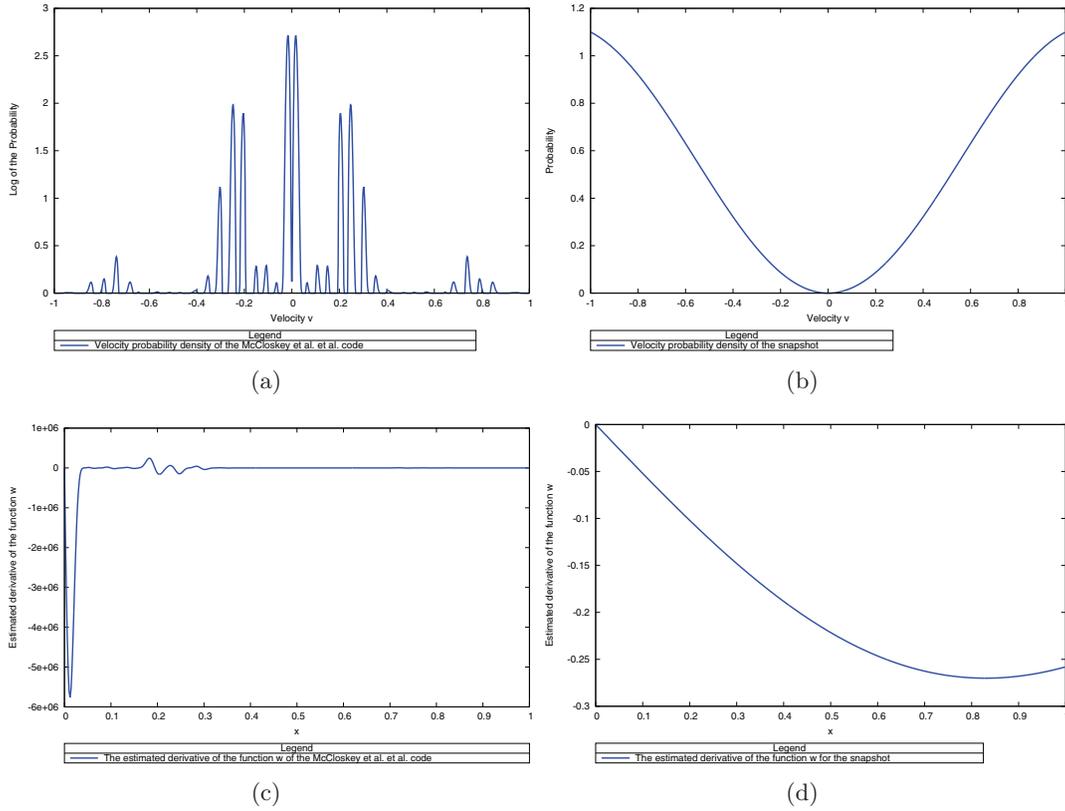


Figure 13. (a) The velocity probability density ρ associated with the McCloskey code [27, p. 321]; the x -axis is the velocity (in signed pixels per Δt), and the y -axis the logarithm of the distribution ($\log(1 + \rho(v))$). Figure (a) has a high probability of not moving and two small charges for two relatively small but nonzero velocities. (b) The probability density of velocities associated with a “standardized” snapshot with flutter shutter gain function $\mathbb{1}_{[0,1]}$; the x -axis is the velocity (in signed pixels per Δt), and the y -axis (not log scaled) is the corresponding probability density. This snapshot is optimized a priori for objects moving at velocity $|v| \approx 1$. This bimodal density is quite suitable for, e.g., a traffic surveillance camera. Bottom: The estimation of the w' function. (Recall that w is defined in (3.4).) The x -axis is x , and the y -axis the estimated $w'(x)$ from the flutter shutter code coefficients.

- Useful for a known uniform velocity. The numerical flutter shutter gains over the optimal snapshot by a 1.17 RMSE gain factor [48].
- More useful if one is able to model the velocity distribution. In that case the RMSE gain can increase more significant values in expectation. This setup requires nevertheless an accurate a posteriori velocity estimation.

Appendix A. Proof of Theorem 3.2. We choose, w.l.o.g., an unknown function $L^1(\mathbb{R}) \ni f := |\hat{\alpha}|^2$. To simplify the expressions we omit the ξ integration variable in the integrals which are all on \mathbb{R} . Thus, (3.5) can be rewritten as

$$(A.1) \quad E(f) = \left(\int f \right) \left(\int \frac{w}{f} \right),$$

where w defined in (3.4) is uniquely associated to the velocity probability density ρ . Hence, finding minimizer(s) of (3.5) among $L^2(\mathbb{R})$ functions is equivalent to finding minimizer(s) of (A.1) among all nonnegative functions that belong to $L^1(\mathbb{R})$. The proof is in two steps. We first prove that any minimizer f of (A.1) satisfies $f = C\sqrt{w}$ on the support of w , where C is an arbitrary positive constant. We then prove that if we extend it by zero outside this support, we obtain a minimizer of (A.1) in $L^1(\mathbb{R})$. This yields all minimizers α of (3.5) that belong to $L^2(\mathbb{R})$.

Let g be an arbitrary bounded perturbation with compact support included in the support of f . From (A.1) we deduce that the weak differential of E at f in the direction g satisfies

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

Since any function f minimizing (A.1) satisfies $E'(f)(g) = 0$ for every bounded perturbation g with support contained in the support of f , we obtain from (A.2) that

$$(A.3) \quad \int g \left[\left(\int \frac{w}{f} \right) - \left(\int f \right) \frac{w}{f^2} \right] = 0.$$

Therefore, any f minimizing (A.1) satisfies $f = C\sqrt{w}$ on the support of w , where $C \in (0, +\infty)$ can be chosen arbitrarily. (From its definition in (3.4), $w \geq 0$.) This concludes the first step of the proof.

We now prove that when extended by zero outside the support of w , f is a minimizer of (A.1) among all nonnegative functions in $L^1(\mathbb{R})$. It follows from (A.1) that $E(Cf) = E(f)$ for any $C \in (0, +\infty)$. Therefore, we can assume w.l.o.g. that f satisfies $\int f = \int \sqrt{w}$. In addition, from its definition, $L^1(\mathbb{R}) \ni f := |\hat{\alpha}|^2$ is nonnegative. Moreover, from its definition in (3.4), E satisfies $E(f+h) \geq E(f)$ for any nonnegative $h \in L^1(\mathbb{R})$ supported on $\mathbb{R} \setminus I$, where I denotes the support of w . Therefore, minimizing (A.1) on all nonnegative $f \in L^1(\mathbb{R})$ is equivalent to minimizing on $f \in L^1(\mathbb{R})$ the strictly convex functional $F(f) := \int \frac{w}{f}$ under the constraints $\int f = \int \sqrt{w}$, $f \geq 0$, $f = 0$ outside the support of w . Indeed, any minimizer of E is deduced, up to a positive multiplicative constant, from a solution of this constrained optimization. We shall now prove that $f := \sqrt{w}$ is the minimizer of F in $L^1(\mathbb{R})$ under the constraints $\int f = \int \sqrt{w}$, $f \geq 0$, $f = 0$, outside the support of w . From Remark 1 we have that $\sqrt{w} \geq 0$, that $\sqrt[4]{w} \in L^2(\mathbb{R})$, and therefore that $\sqrt{w} \in L^1(\mathbb{R})$. Hence, $f = \sqrt{w}$ satisfies the constraints $\int f = \int \sqrt{w}$, $f \geq 0$, $f = 0$ outside the support of w . Thus, this f is in the feasible set for F . In addition, from the Cauchy–Schwarz inequality, we have

$$(A.4) \quad \left(\int \sqrt{w} \right)^2 = \left(\int \frac{\sqrt{w}}{\sqrt{f}} \sqrt{f} \right)^2 \leq \left(\int \frac{w}{f} \right) \left(\int f \right)$$

for every f that is positive on the support of w . Equality in (A.4) is attained for $f = \sqrt{w}$. Therefore, \sqrt{w} is the unique minimizer of F under the aforementioned constraints. Thus, the minimizers of (A.1) among nonnegative functions that belong to $L^1(\mathbb{R})$ are of the form $C\sqrt{w}$ for some $C > 0$ on the support of w , and are zero outside this support. Since $f := |\hat{\alpha}|^2$, we deduce that α is a minimizer of (3.5) in $L^2(\mathbb{R})$ iff $\hat{\alpha}$ satisfies $|\hat{\alpha}| = C\sqrt[4]{w}$ for some $C > 0$ on the support of w , and is zero outside this support. This concludes the proof. ■

Appendix B. Proof of Theorem 3.3. We wish to find the minimizer(s), if they exist, of (3.5) among real-valued functions of the form of (3.6) that belong to $L^2(\mathbb{R})$. From (3.6), we deduce that this question is equivalent to finding an optimal sequence $(\alpha_k)_k$ in $\ell^2(\mathbb{Z})$.

The proof is in two steps. We first prove that $(\alpha_k)_k \in \ell^2(\mathbb{Z})$ is optimal iff it satisfies

$$\left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right|^2 = C \sqrt{\frac{w\left(\frac{\xi}{\Delta t}\right)}{\operatorname{sinc}^2\left(\frac{\xi}{2\pi}\right)}}$$

for some fixed $C > 0$ and for any $\xi \in [-\pi, \pi]$. We then prove that the sequence $(\alpha_k)_k$ defined by

$$\alpha_k = \frac{1}{2\pi} \int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)} \cos(ks)}{\sqrt{\operatorname{sinc}\left(\frac{\xi}{2\pi}\right)}} d\xi$$

defines an optimal real-valued $\alpha \in L^2(\mathbb{R})$ of the form of (3.6).

From (3.5) and Plancherel's identity (xx), we deduce that

$$(B.1) \quad E(\hat{\alpha}) = 2\pi \|\alpha\|_{L^2(\mathbb{R})}^2 \int_{\mathbb{R}} \frac{w(\xi)}{|\hat{\alpha}|^2(\xi)} d\xi.$$

From Remark 1 we have that the support of w is $[-\pi, |v_{max}|, \pi, |v_{max}|]$. Thus, from (B.1) we deduce that

$$(B.2) \quad E(\hat{\alpha}) = 2\pi \|\alpha\|_{L^2(\mathbb{R})}^2 \int_{-\pi|v_{max}|}^{\pi|v_{max}|} \frac{w(\xi)}{|\hat{\alpha}|^2(\xi)} d\xi.$$

From (3.6) we have

$$(B.3) \quad \|\alpha\|_{L^2(\mathbb{R})}^2 = \Delta t \sum_{k \in \mathbb{Z}} |\alpha_k|^2$$

and

$$(B.4) \quad \hat{\alpha}(\xi) = \Delta t \operatorname{sinc}\left(\frac{\xi \Delta t}{2\pi}\right) e^{-\frac{i\xi \Delta t}{2}} \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi \Delta t}$$

for any $\xi \in \mathbb{R}$. Hence, combining (B.2), (B.3), and (B.4), for any $\alpha \in L^2(\mathbb{R})$ of the form of (3.6), we find

$$(B.5) \quad E(\hat{\alpha}) = 2\pi \Delta t \left(\sum_{k \in \mathbb{Z}} |\alpha_k|^2 \right) \int_{-\pi|v_{max}|}^{\pi|v_{max}|} \frac{w(\xi)}{\left| \Delta t \operatorname{sinc}\left(\frac{\xi \Delta t}{2\pi}\right) e^{-\frac{i\xi \Delta t}{2}} \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi \Delta t} \right|^2} d\xi$$

$$(B.6) \quad = \frac{2\pi}{\Delta t^2} \left(\sum_{k \in \mathbb{Z}} |\alpha_k|^2 \right) \int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{w\left(\frac{\xi}{\Delta t}\right)}{\operatorname{sinc}^2\left(\frac{\xi}{2\pi}\right) \left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right|^2} d\xi.$$

Since, by assumption, we have $|v_{max}|\Delta t \leq 1$, we deduce that $[-\pi|v_{max}|\Delta t, \pi|v_{max}|\Delta t] \subset [-\pi, \pi]$. Note that $\operatorname{sinc}^2\left(\frac{\xi}{2\pi}\right) \neq 0$ for any $\xi \in [-\pi, \pi]$. In addition, we recall that the support of

$w\left(\frac{\cdot}{\Delta t}\right)$ is $[-\pi, |v_{max}|\Delta t, \pi, |v_{max}|\Delta t]$. In other words, $w\left(\frac{\cdot}{\Delta t}\right)$ is zero on $[-\pi, \pi] \setminus [-\pi, |v_{max}|\Delta t, \pi, |v_{max}|\Delta t]$. Hence, from (B.5)–(B.6), for any $\alpha \in L^2(\mathbb{R})$ of the form of (3.6) we have

$$(B.7) \quad E(\hat{\alpha}) = \frac{2\pi}{\Delta t^2} \left(\sum_{k \in \mathbb{Z}} |\alpha_k|^2 \right) \int_{-\pi}^{\pi} \frac{w\left(\frac{\xi}{\Delta t}\right)}{\text{sinc}^2\left(\frac{\xi}{2\pi}\right) \left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right|^2} d\xi.$$

For any $\alpha \in L^2(\mathbb{R})$ of the form of (3.6), as soon as $|v_{max}|\Delta t \leq 1$, the term $\sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi}$ that appears in (B.7) is the Fourier series (xxi) synthesis formula of some $f \in L^2(-\pi, \pi)$ function evaluated at $-\xi$. In other words, as soon as $|v_{max}|\Delta t \leq 1$, $\alpha_k = c_k(f)$ for any $k \in \mathbb{Z}$ for some $f \in L^2(-\pi, \pi)$. (This fact will be used later on to compute optimal $(\alpha_k)_k$.) From the Riesz–Fischer theorem (see, e.g., [54, p. 27]) and (B.7), we deduce that, as soon as $|v_{max}|\Delta t \leq 1$, minimizing (3.5) among $L^2(\mathbb{R})$ functions of the form (3.6) is equivalent to finding $f \in L^2(-\pi, \pi)$ that minimizes

$$(B.8) \quad E(f) = \frac{2\pi}{\Delta t^2} \left(\sum_{k \in \mathbb{Z}} |c_k(f)|^2 \right) \int_{-\pi}^{\pi} \frac{w\left(\frac{\xi}{\Delta t}\right)}{\text{sinc}^2\left(\frac{\xi}{2\pi}\right) |f(-\xi)|^2} d\xi$$

$$(B.9) \quad = \frac{2\pi}{\Delta t^2} \left(\sum_{k \in \mathbb{Z}} |c_k(f)|^2 \right) \int_{-\pi}^{\pi} \frac{w\left(\frac{\xi}{\Delta t}\right)}{\text{sinc}^2\left(\frac{\xi}{2\pi}\right) |f(\xi)|^2} d\xi,$$

where the last inequality is justified by the fact that w and sinc^2 are even functions. Hence, from Parseval’s identity (xxi) and (B.8)–(B.9), we deduce that, as soon as $|v_{max}|\Delta t \leq 1$, minimizing (3.5) among $L^2(\mathbb{R})$ functions of the form (3.6) is equivalent to finding $f \in L^2(-\pi, \pi)$ that minimizes

$$(B.10) \quad E(f) = \frac{1}{\Delta t^2} \left(\int_{-\pi}^{\pi} |f(\xi)|^2 d\xi \right) \int_{-\pi}^{\pi} \frac{w\left(\frac{\xi}{\Delta t}\right)}{\text{sinc}^2\left(\frac{\xi}{2\pi}\right) |f(\xi)|^2} d\xi.$$

We choose as unknown $L^1(-\pi, \pi) \ni g := |f|^2$. To simplify the expression, we omit the integration variable and the integration intervals, which are all $[-\pi, \pi]$, and the positive multiplicative constant $\frac{1}{\Delta t^2}$. Thus, from (B.10), we deduce that, as soon as $|v_{max}|\Delta t \leq 1$, minimizing (3.5) among functions of the form (3.6) is equivalent to finding a nonnegative g that belongs to $L^1(-\pi, \pi)$ and that minimizes

$$(B.11) \quad E(g) = \left(\int g \right) \left(\int \frac{\tilde{w}}{g} \right),$$

where $[-\pi, \pi] \ni \xi \mapsto \tilde{w}(\xi) := w\left(\frac{\xi}{\Delta t}\right) / \text{sinc}^2\left(\frac{\xi}{2\pi}\right)$. By the same calculations as in the proof of Theorem 3.2 (Appendix A), we deduce that any nonnegative $g \in L^1(-\pi, \pi)$ that minimizes (B.11) satisfies $g = C\sqrt{\tilde{w}}$, on the support of \tilde{w} , and $C > 0$ can be arbitrarily chosen. (This solution is nonnegative, belongs to $L^1(-\pi, \pi)$, and is therefore admissible.) The support of \tilde{w} is $[-\pi|v_{max}|\Delta t, \pi|v_{max}|\Delta t]$ and is therefore contained in $[-\pi, \pi]$. Indeed, by assumption we have that $|v_{max}|\Delta t \leq 1$. It remains to show that if we extend g by zero on $[-\pi, \pi] \setminus [-\pi|v_{max}|\Delta t, \pi|v_{max}|\Delta t]$, we obtain a solution. By the same arguments developed in the proof of Theorem 3.2, we see that minimizing $E(g)$ for $g \in L^1(-\pi, \pi)$ is equivalent to

minimizing the strictly convex functional $F(g) := \int \frac{\tilde{w}}{g}$ under the constraints $\int g = \int \sqrt{\tilde{w}}$, $g \geq 0$, $g = 0$, on $[-\pi, \pi] \setminus [-\pi|v_{max}|\Delta t, \pi|v_{max}|\Delta t]$. The fact that $g := \sqrt{\tilde{w}} \in L^1(\mathbb{R})$ is the unique solution to this problem follows by the very same arguments. It follows that $(\alpha_k)_k \in \ell^2(\mathbb{Z})$ is optimal iff the square modulus of its Fourier series synthesis (xxi) coincides with $\sqrt{\tilde{w}}$, up to a positive multiplicative constant, on $[-\pi, \pi]$. In other words, we deduce that $(\alpha_k)_k \in \ell^2(\mathbb{Z})$ is optimal iff

$$\left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right|^2 = C \sqrt{\frac{w\left(\frac{\xi}{\Delta t}\right)}{\text{sinc}^2\left(\frac{\xi}{2\pi}\right)}}$$

for some fixed $C > 0$ and for any $\xi \in [-\pi, \pi]$. This proves the first part of our theorem. It remains to compute an optimal real-valued sequence $(\alpha_k) \in \ell^2(\mathbb{Z})$.

We have that $\text{sinc}\left(\frac{\xi}{2\pi}\right) > 0$ for any $\xi \in [-\pi, \pi]$. We recall that $\sqrt{w} \in L^1(\mathbb{R})$. Therefore, the function

$$f : [-\pi, \pi] \ni \xi \mapsto \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}}{\sqrt{\text{sinc}\left(\frac{\xi}{2\pi}\right)}}$$

belongs to $L^2(-\pi, \pi)$. We can therefore consider its Fourier series (xxi). In addition, we recall that as soon as $|v_{max}|\Delta t \leq 1$, the sequence $(\alpha_k)_k$ corresponds to the Fourier series coefficient of the considered f . Thus, consider the real-valued sequence (α_k) given by

$$(B.12) \quad \alpha_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}}{\sqrt{\text{sinc}\left(\frac{\xi}{2\pi}\right)}} e^{-ik\xi} d\xi = \frac{1}{2\pi} \int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}}{\sqrt{\text{sinc}\left(\frac{\xi}{2\pi}\right)}} \cos(k\xi) d\xi.$$

Indeed, the support of $w\left(\frac{\xi}{\Delta t}\right)$ is $[-\pi|v_{max}|\Delta t, \pi|v_{max}|\Delta t]$, and $\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}/\text{sinc}\left(\frac{\xi}{2\pi}\right)$ is even. Therefore, for any $k \in \mathbb{Z}$, the coefficients α_k are real. In addition, we have the function $\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}/\text{sinc}\left(\frac{\xi}{2\pi}\right) \in L^2(-\pi, \pi)$. Thus, we deduce that $(\alpha_k)_k \in \ell^2(\mathbb{Z})$. From its definition (B.12) the sequence $(\alpha_k)_k$ satisfies

$$\left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right|^2 = \sqrt{\frac{w\left(\frac{\xi}{\Delta t}\right)}{\text{sinc}^2\left(\frac{\xi}{2\pi}\right)}}$$

for any $\xi \in [-\pi, \pi]$. Hence, the real-valued sequence $(\alpha_k)_{k \in \mathbb{Z}}$ defined by (B.12) is optimal. Thus, as soon as $|v_{max}|\Delta t \leq 1$, the function α of the form of (3.6), with the α_k defined by (B.12), belongs to $L^2(\mathbb{R})$, is optimal for (B.2), and is real-valued. This concludes our proof. ■

Appendix C. Proof of Theorem 4.1. The proof is in two parts. We first consider time-continuous *flutter shutter gain functions* and then treat the case of piecewise constant *flutter shutter gain functions*.

Let $\alpha \in L^2(\mathbb{R})$ be an optimal time-continuous (not of the form of (3.6)) *flutter shutter gain function* with respect to some velocity probability density ρ . Consider the function w associated with the probability density ρ by (3.4). From Remark 1, we have that $(0, +\infty) \ni \xi \mapsto w(\xi)$ is nonincreasing. By Theorem 3.2, we know that $|\hat{\alpha}| = \sqrt[4]{w}$ on the support of w

(up to a positive multiplicative constant) and 0 outside the support of w . It follows that any optimal $\alpha \in L^2(\mathbb{R})$ is such that $|\hat{\alpha}|$ is nonincreasing on $(0, +\infty)$.

We now prove formula (4.1). Notice that given w , the optimal density ρ can be assumed to be even. Indeed, if ρ is not even, (3.4) still holds true when replacing ρ by its symmetrized version $\frac{\rho(\cdot) + \rho(-\cdot)}{2}$. Thus, we can look for an even ρ and therefore simplify (3.4) into

$$(C.1) \quad w(\xi) = 2 \int_{\frac{|\xi|}{\pi}}^{+\infty} \frac{\rho(v)}{v} dv.$$

Since ρ is $L^1(\mathbb{R})$, this formula implies that w is absolutely continuous and that its derivative in the distributional sense on $\mathbb{R} \setminus \{0\}$ is

$$(C.2) \quad w'(\xi) = \text{sign}(\xi)w'(|\xi|) = 2\text{sign}(\xi) \left(\int_{\frac{|\xi|}{\pi}}^{+\infty} \frac{\rho(v)}{v} dv \right)' = \text{sign}(\xi) \left(\frac{1}{\pi} \int_{|\xi|}^{+\infty} \frac{\rho\left(\frac{v}{\pi}\right)}{\frac{v}{\pi}} dv \right)',$$

where the first equality is justified by the fact that we assumed that ρ is even, the second by (C.1), and the last by the change of variables $v \mapsto \frac{v}{\pi}$. Hence, from (C.2) we deduce that

$$w'(\xi) = -2\text{sign}(\xi) \frac{1}{\pi} \frac{\rho\left(\frac{|\xi|}{\pi}\right)}{\frac{|\xi|}{\pi}} = -2 \frac{\rho\left(\frac{|\xi|}{\pi}\right)}{\xi}.$$

Hence, we have $w'(\pi\xi) = -2 \frac{\rho\left(\frac{|\xi|}{\pi}\right)}{\pi\xi}$ and, since ρ is even by assumption, that $w'(\pi\xi) = -2 \frac{\rho(\xi)}{\pi\xi}$ on $\mathbb{R} \setminus \{0\}$, these equalities being understood in the distribution sense on $\mathbb{R} \setminus \{0\}$. It follows that $\rho(v)$, being a probability density function, is entirely determined by

$$(C.3) \quad \rho(v) = -\frac{\pi v}{2} w'(\pi v) \quad \text{for any } v \neq 0.$$

It remains to show that we can deduce ρ from α . By Theorem 3.2, any optimal $\alpha \in L^2(\mathbb{R})$ with respect to a velocity density ρ satisfies $|\hat{\alpha}|^4 = Cw$ for some positive constant C . Then (C.3) implies $\rho(v) = -\frac{vC}{2} (|\hat{\alpha}|^4)'(\pi v)$ for $v \neq 0$. The normalization factor C is chosen so that $\int \rho(v)dv = 1$. This proves the first part of our theorem. We now prove the second part of our theorem that concerns piecewise constant flutter shutter gain functions.

Let $\alpha \in L^2(\mathbb{R})$ be an optimal piecewise constant code of the form (3.6), and assume that $\Delta t |v_{max}| \leq 1$. (We recall that in this case ρ is supported on $[-|v_{max}|, |v_{max}|]$.) From (B.4) we have

$$(C.4) \quad |\hat{\alpha}(\xi)|^4 = \left| \Delta t \text{sinc} \left(\frac{\xi \Delta t}{2\pi} \right) \right|^4 \left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi \Delta t} \right|^4 \quad \text{for any } \xi \in \mathbb{R}.$$

By Theorem 3.3 the sequence $(\alpha_k)_k \in \ell^2(\mathbb{Z})$ defining α satisfies

$$\left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\xi} \right| = C \frac{\sqrt[4]{w\left(\frac{\xi}{\Delta t}\right)}}{\sqrt{\text{sinc}\left(\frac{\xi}{2\pi}\right)}}$$

for some fixed $C > 0$ and for any $\xi \in [-\pi, \pi]$. Therefore, we deduce that

$$(C.5) \quad \left| \sum_{k \in \mathbb{Z}} \alpha_k e^{-ik\Delta t \xi} \right|^4 = C \frac{w(\xi)}{\left| \text{sinc}\left(\frac{\xi\Delta t}{2\pi}\right) \right|^2}$$

for some fixed $C > 0$ and for any $\xi \in \left[-\frac{\pi}{\Delta t}, \frac{\pi}{\Delta t}\right]$. Combining (C.4) and (C.5), we obtain

$$(C.6) \quad |\hat{\alpha}(\xi)|^4 = C \left| \text{sinc}\left(\frac{\xi\Delta t}{2\pi}\right) \right|^2 w(\xi) \quad \text{and therefore} \quad \frac{|\hat{\alpha}(\xi)|^4}{\left| \text{sinc}\left(\frac{\xi\Delta t}{2\pi}\right) \right|^2} = C w(\xi)$$

for some $C > 0$ and for any $\xi \in \left[-\frac{\pi}{\Delta t}, \frac{\pi}{\Delta t}\right]$. Since $(0, +\infty) \ni \xi \mapsto w(\xi)$ is nonincreasing, from (C.6), we immediately obtain a necessary condition for an $\alpha \in L^2(\mathbb{R})$ of the form of (3.6) to be optimal. Indeed, an α of the form (3.6) is optimal only if

$$\left(0, \frac{\pi}{\Delta t}\right] \ni \xi \mapsto \frac{|\hat{\alpha}(\xi)|^4}{\left| \text{sinc}\left(\frac{\xi\Delta t}{2\pi}\right) \right|^2}$$

is nonincreasing. Furthermore, from (C.6), we obtain that

$$(C.7) \quad w'(\xi) = C \left(\frac{|\hat{\alpha}(\xi)|^4}{\left| \text{sinc}\left(\frac{\xi\Delta t}{2\pi}\right) \right|^2} \right)'(\xi)$$

for some $C > 0$ and for any $\xi \in \left[-\frac{\pi}{\Delta t}, \frac{\pi}{\Delta t}\right] \setminus \{0\}$, the equality in (C.7) being understood in the distribution sense. By (C.3) we have $\rho(v) = -\frac{v}{2}w'(\pi v)$ for $v \neq 0$. Therefore, from (C.7) we deduce that

$$(C.8) \quad \rho(v) = -\frac{vC}{2} \left(\frac{|\hat{\alpha}(\xi)|^4}{\left| \text{sinc}\left(\frac{\xi\Delta t}{2\pi}\right) \right|^2} \right)'(\pi v) \quad \text{for any } v \in \left[\frac{-1}{\Delta t}, \frac{1}{\Delta t}\right] \setminus \{0\}.$$

In addition, since by assumption we have $|v_{max}|\Delta t \leq 1$, we deduce that $\frac{1}{\Delta t} > |v_{max}|$ and therefore that $[-|v_{max}|, |v_{max}|] \subset \left[-\frac{1}{\Delta t}, \frac{1}{\Delta t}\right]$. Thus, since ρ is supported on $[-|v_{max}|, |v_{max}|]$, ρ is determined up to a positive multiplicative constant by (C.8). We again choose C so that $\int \rho = 1$. This concludes the proof. \blacksquare

Appendix D. Proof of Proposition 5.1. The proof is in three steps. We first prove that the energy defined by (5.1) is strictly convex. We then justify that this energy always has exactly one minimizer Δt^* . Lastly, we prove that the quantity $\Delta t^*|v_{max}|$ is constant when the velocity density ρ is uniform or a (truncated) Gaussian.

The function $\Delta t \mapsto \frac{\Delta t}{\sin^2 \frac{\xi v \Delta t}{2}}$ is strictly convex for $|\frac{\xi v \Delta t}{2}| < \pi$. It follows immediately that $E(\Delta t)$ is a strictly convex function on $I := \left[0, \frac{2}{|v_{max}|}\right]$. By a direct application of the monotone convergence theorem, we have $\lim_{\Delta t \rightarrow 0, \Delta t > 0} E(\Delta t) = \lim_{\Delta t \rightarrow \frac{2}{|v_{max}|}, \Delta t < \frac{2}{|v_{max}|}} E(\Delta t) = +\infty$. Furthermore, E is finite in the interior of I . Thus, E has exactly one minimizer Δt^* on I . It remains to show that if $\rho(v)$ is uniform or Gaussian, the optimal *snapshot* satisfies $\Delta t^*|v_{max}| = C$, where C is a positive constant. Consider $s > 0$ and $\rho_s(v) = s\rho(vs)$ a family of probability densities which comes from a compactly supported mother function

$\rho(v)\mathbb{1}_{[-|v_{max}|, |v_{max}|]}(v)$. Let Δt^* be the unique minimizer of (5.1). By an obvious change of variables,

$$E_s(\Delta t) := \iint_{-\pi}^{\pi} \frac{\xi^2}{\sin^2\left(\frac{\xi v \Delta t}{2}\right)} \frac{v^2 \Delta t}{4} \rho_s(v) d\xi dv = \frac{1}{s} \iint_{-\pi}^{\pi} \frac{\xi^2}{\sin^2\left(\frac{\xi v \Delta t}{2}\right)} \frac{v^2 \Delta t}{s} \rho(v) d\xi dv = \frac{1}{s} E\left(\frac{\Delta t}{s}\right).$$

Thus, $E_s(\Delta t)$ is minimized for $\frac{\Delta t}{s} = \Delta t^*$. In other words, $E_s(\Delta t)$ has $s\Delta t^*$ for a unique minimizer. Since E and E_s differ only by a $\frac{1}{s}$ positive multiplicative constant, we deduce that $s\Delta t^*$ is the unique minimizer of E . The rest of the proof follows since $\rho_s(v) = s\mathbb{1}_{[-\frac{1}{2}, \frac{1}{2}]}(vs)$ or $\rho_s(v) = s\mathbb{1}_{[-4,4]}(vs) \exp\left(\frac{-(vs)^2}{2}\right)$ (up to an irrelevant constant positive multiplicative constant for the optimization of E). The values $\Delta t^*|v_{max}| \approx 1.42$ (when ρ is uniform) and $\Delta t^*|v_{max}| \approx 1.94$ (when $\rho(v) \propto \mathbb{1}_{[-4\sigma, 4\sigma]} e^{-\frac{v^2}{2\sigma^2}}$) are obtained numerically. ■

Appendix E. Main notation and formulae.

- (i) $t \in \mathbb{R}$ time variable.
- (ii) $\Delta t > 0$ length of a time interval.
- (iii) $x \in \mathbb{R}$ spatial variable.
- (iv) $X \sim Y$ random variables X and Y have the same law.
- (v) $\mathbb{P}(A)$ probability of an event A .
- (vi) $\mathcal{P}(\lambda)$ Poisson random variable with intensity $\lambda > 0$. Thus, if $X \sim \mathcal{P}(\lambda)$, we have $\mathbb{P}(X = k) = \frac{\exp(-\lambda)\lambda^k}{k!}$.
- (vii) $\mathbb{E}(X)$ expected value of a random variable X .
- (viii) $\text{var}(X)$ variance of a random variable X .
- (ix) $f * g$ convolution of two functions $(f * g)(x) = \int_{-\infty}^{+\infty} f(y)g(x-y)dy$.
- (x) u ideal (noiseless) observable landscape just before sampling. Assumption: $u \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$, $[-\pi, \pi]$ band-limited.
- (xi) $\text{obs}(n)$, $n \in \mathbb{Z}$, observation of the landscape at a pixel supported on $[n - \frac{1}{2}, n + \frac{1}{2}]$.
- (xii) v relative velocity between the scene and the camera (unit: pixels per Δt seconds).
- (xiii) $\alpha(t)$ piecewise constant or time-continuous gain control function for the *analog flutter shutter* and *numerical flutter shutter* methods.
- (xiv) $\rho(v)$ probability distribution for the relative camera-scene velocities. Assumption: $\rho(v) = 0$ for any v such that $|v| > v_{max}$.
- (xv) $w(x) \geq 0$ weight function associated with the probability distribution ρ .
- (xvi) $\|f\|_{L^1(\mathbb{R})} = \int |f(x)|dx$, $\|f\|_{L^2(\mathbb{R})} = \sqrt{\int |f(x)|^2 dx}$.
- (xvii) $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x} = \frac{1}{2\pi} \mathcal{F}(\mathbb{1}_{[-\pi, \pi]})(x) = \mathcal{F}^{-1}(\mathbb{1}_{[-\pi, \pi]})(x)$.
- (xviii) $\mathbb{1}_{[a, b]}$ indicator function of an interval $[a, b]$.
- (xix) $\text{sign}(x) = 1$ if $x \geq 0$, and $\text{sign}(x) = -1$ if $x < 0$.
- (xx) Let $f, g \in L^1(\mathbb{R})$ or $L^2(\mathbb{R})$; then

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

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

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

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

- (xxi) Let $f \in L^1(-\pi, \pi)$ or $f \in L^2(\pi, \pi)$. The n th Fourier series coefficient of f is $c_n(f) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) e^{-int} dt$, and we have $f(t) = \sum_{n=-\infty}^{+\infty} c_n(f) e^{+int}$. Moreover, we have (Parseval) $\sum_{n=-\infty}^{+\infty} |c_n(f)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(t)|^2 dt$.

REFERENCES

- [1] A. ADAMS, D. E. JACOBS, J. DOLSON, M. TICO, K. PULLI, E. V. TALVALA, B. AJDIN, D. VAQUERO, H. LENSCH, M. HOROWITZ, S. H. PARK, N. GELFAND, J. BAEK, W. MATUSIK, AND M. LEVOY, *The Frankencamera: An experimental platform for computational photography*, ACM Trans. Graphics, 29 (2010), pp. 29:1–29:12.
- [2] A. AGRAWAL, M. GUPTA, A. VEERARAGHAVAN, AND S. G. NARASIMHAN, *Optimal coded sampling for temporal super-resolution*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 599–606.
- [3] A. AGRAWAL AND R. RASKAR, *Resolving objects at higher resolution from a single motion-blurred image*, in Proceedings of the (IEEE) Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
- [4] A. AGRAWAL AND R. RASKAR, *Optimal single image capture for motion deblurring*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 2560–2567.
- [5] A. AGRAWAL, A. VEERARAGHAVAN, AND R. RASKAR, *Reinterpretable imager: Towards variable post-capture space, angle and time resolution in photography*, in Computer Graphics Forum, Vol. 29, Wiley-Blackwell, New York, 2010, pp. 763–772.
- [6] A. AGRAWAL AND Y. XU, *Coded exposure deblurring: Optimized codes for PSF estimation and invertibility*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 2066–2073.
- [7] Y. BANDO AND T. NISHITA, *Towards digital refocusing from a single photograph*, in Proceedings of the IEEE Pacific Conference on Computer Graphics and Applications, 2007, pp. 363–372.
- [8] G. BORACCHI AND A. FOI, *Uniform motion blur in Poissonian noise: Blur/noise tradeoff*, IEEE Trans. Image Process., 20 (2011), pp. 592–598.
- [9] T. BUADES, Y. LOU, J.-M. MOREL, AND Z. TANG, *A note on multi-image denoising*, in Proceedings of the IEEE International Workshop on Local and Non-Local Approximation in Image Processing (LNLA), 2009, pp. 1–15.
- [10] T. S. CHO, A. LEVIN, F. DURAND, AND W. T. FREEMAN, *Motion blur removal with orthogonal parabolic exposures*, in Proceedings of the IEEE International Conference on Computational Photography (ICCP), 2010, pp. 1–8.
- [11] O. COSSAIRT, M. GUPTA, AND S. K. NAYAR, *When does computational imaging improve performance?*, IEEE Trans. Image Process., 22 (2013), pp. 447–458.
- [12] Y. DING, S. MCCLOSKEY, AND J. YU, *Analysis of motion blur with a flutter shutter camera for non-linear motion*, in Proceedings of the Springer-Verlag European Conference on Computer Vision (ECCV), 2010, pp. 15–30.
- [13] J. P. GODBAZ, M. J. CREE, AND A. A. DORRINGTON, *Extending AMCW lidar depth-of-field using a coded aperture*, in Proceedings of the Springer-Verlag Asian Conference on Computer Vision (ACCV), 2010, pp. 397–409.
- [14] J. GU, Y. HITOMI, T. MITSUNAGA, AND S. K. NAYAR, *Coded rolling shutter photography: Flexible space-time sampling*, in Proceedings of the IEEE International Conference on Computational Photography (ICCP), 2010, pp. 1–8.
- [15] M. GUPTA, A. AGRAWAL, A. VEERARAGHAVAN, AND S. G. NARASIMHAN, *Flexible voxels for motion-aware videography*, in Proceedings of the Springer-Verlag European Conference on Computer Vision (ECCV), 2010, pp. 100–114.
- [16] S. W. HASINOFF AND K. N. KUTULAKOS, *Light-efficient photography*, IEEE Trans. Pattern Anal. Machine Intell., 33 (2011), pp. 2203–2214.
- [17] L. HE, G. CUI, H. FENG, Z. XU, Q. LI, AND Y. CHEN, *The optimal code searching method with an improved criterion of coded exposure for remote sensing image restoration*, Optics Commun., 338

- (2015), pp. 540–550.
- [18] K. HUANG, J. HOU, AND J. ZHANG, *Optimal binary sequences for coded exposure photography using genetic algorithm*, in Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), 2014, pp. 287–291.
 - [19] J. JELINEK, *Designing the optimal shutter sequences for the flutter shutter imaging method*, in Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, Vol. 7701, 2010, p. 18.
 - [20] H. KOZUKA, *Image Sensor*, U.S. Patent 6,473,538, Oct. 29 2002.
 - [21] D. LANMAN, R. RASKAR, A. AGRAWAL, AND G. TAUBIN, *Shield fields: Modeling and capturing 3D occluders*, ACM Trans. Graphics, 27 (2008), pp. 131:1–131:10.
 - [22] A. LEVIN, R. FERGUS, F. DURAND, AND W. T. FREEMAN, *Image and depth from a conventional camera with a coded aperture*, ACM Trans. Graphics, 26 (2007), pp. 70:1–70:9.
 - [23] A. LEVIN, P. SAND, T. S. CHO, F. DURAND, AND W. T. FREEMAN, *Motion-invariant photography*, ACM Trans. Graphics, 27 (2008), pp. 71:1–71:9.
 - [24] F. LI, J. SUN, J. WANG, AND J. YU, *Dual-focus stereo imaging*, J. Electron. Imaging, 19 (2010), 043009.
 - [25] C. K. LIANG, T. H. LIN, B. Y. WONG, C. LIU, AND H. H. CHEN, *Programmable aperture photography: Multiplexed light field acquisition*, ACM Trans. Graphics, 27 (2008), pp. 55:1–55:10.
 - [26] M. MARTINELLO, T. E. BISHOP, AND P. FAVARO, *A Bayesian approach to shape from coded aperture*, in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2010, pp. 3521–3524.
 - [27] S. MCCLOSKEY, *Velocity-dependent shutter sequences for motion deblurring*, in Proceedings of the Springer-Verlag European Conference on Computer Vision (ECCV), 2010, pp. 309–322.
 - [28] S. MCCLOSKEY, *Acquisition System for Obtaining Sharp Barcode Images Despite Motion*, EP Patent 2,284,764, Feb. 16, 2011.
 - [29] S. MCCLOSKEY, *Temporally coded flash illumination for motion deblurring*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 683–690.
 - [30] S. MCCLOSKEY, W. AU, AND J. JELINEK, *Iris capture from moving subjects using a fluttering shutter*, in Proceedings of the IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), 2010, pp. 1–6.
 - [31] S. MCCLOSKEY, J. JELINEK, AND K. W. AU, *Method and System for Determining Shutter Fluttering Sequence*, U.S. Patent 12/421,296, Apr. 9 2009.
 - [32] S. MCCLOSKEY, K. MULDOON, AND S. VENKATESHA, *Motion invariance and custom blur from lens motion*, in Proceedings of the IEEE International Conference on Computational Photography (ICCP), 2011, pp. 1–8.
 - [33] L. MEI, X. CAI, AND W. LIU, *Defocus deblurring with a coded aperture*, in Proceedings of the IEEE International Conference on Progress in Informatics and Computing (PIC), Vol. 2, 2010, pp. 916–919.
 - [34] A. MOHAN, D. LANMAN, S. HIURA, AND R. RASKAR, *Image destabilization: Programmable defocus using lens and sensor motion*, in Proceedings of the IEEE International Conference on Computational Photography (ICCP), 2009, pp. 1–8.
 - [35] H. NAGAHARA, C. ZHOU, T. WATANABE, H. ISHIGURO, AND S. K. NAYAR, *Programmable aperture camera using LCoS*, in Proceedings of the Springer-Verlag European Conference on Computer Vision (ECCV), 2010, pp. 337–350.
 - [36] K. NAKAMURA, H. OHZU, AND I. UENO, *Image Sensor in Which Reading and Resetting Are Simultaneously Performed*, U.S. Patent 5,262,870, Nov. 16, 1993.
 - [37] R. RASKAR, *Method and Apparatus for Deblurring Images*, U.S. Patent 7,756,407, July 13, 2010.
 - [38] R. RASKAR, A. AGRAWAL, AND J. TUMBLIN, *Coded exposure photography: Motion deblurring using fluttered shutter*, ACM Trans. Graphics, 25 (2006), pp. 795–804.
 - [39] R. RASKAR, J. TUMBLIN, AND A. AGRAWAL, *Method for deblurring images using optimized temporal coding patterns*, U.S. Patent 7,580,620, Aug. 25, 2009.
 - [40] A. SARKER AND L. G. C. HAMEY, *Improved reconstruction of flutter shutter images for motion blur reduction*, in Proceedings of the IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2010, pp. 417–422.
 - [41] Y. W. TAI, N. KONG, S. LIN, AND S. Y. SHIN, *Coded exposure imaging for projective motion deblurring*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2408–2415.

- [42] Y. TENDERO, *The flutter shutter camera simulator*, IPOL J. Image Process. Online, 2 (2012), pp. 225–242.
- [43] Y. TENDERO, *Are Random Flutter Shutter Codes Good Enough?*, submitted; UCLA CAM REPORT 13-84, 2013.
- [44] Y. TENDERO, *The Flutter Shutter Code Calculator*, IPOL J. Image Process. Online, 5 (2015), pp. 234–256.
- [45] Y. TENDERO AND J.-M. MOREL, *An optimal blind temporal motion blur deconvolution filter*, IEEE Signal Process. Lett., 20 (2013), pp. 523–526.
- [46] Y. TENDERO AND J.-M. MOREL, *On the mathematical foundations of computational photography—Does the flutter shutter work better at night?*, J. Math. Imaging and Vision, 54 (2016), pp. 378–397.
- [47] Y. TENDERO, J.-M. MOREL, AND B. ROUGÉ, *A formalization of the flutter shutter*, J. Phys. Conf. Ser., 386 (2012), 012001.
- [48] Y. TENDERO, J.-M. MOREL, AND B. ROUGÉ, *The flutter shutter paradox*, SIAM J. Imaging Sci., 6 (2013), pp. 813–847.
- [49] Y. TENDERO AND S. OSHER, *Blind Uniform Motion Blur Deconvolution for Image Bursts and Video Sequences Based on Sensor Characteristics*, submitted; UCLA CAM REPORT 14-17, 2014.
- [50] Y. TENDERO AND S. OSHER, *On a mathematical theory of coded exposure*, Res. Math. Sci., (2015), to appear.
- [51] M. TRENTACOSTE, C. LAU, M. ROUF, R. L. MANTIUK, AND W. HEIDRICH, *Defocus techniques for camera dynamic range expansion*, in Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE), Vol. 7537, 2010, p. 16.
- [52] A. VEERARAGHAVAN, R. RASKAR, A. AGRAWAL, A. MOHAN, AND J. TUMBLIN, *Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing*, ACM Trans. Graphics, 26 (2007), pp. 69:1–69:12.
- [53] O. WHYTE, J. SIVIC, A. ZISSERMAN, AND J. PONCE, *Non-uniform deblurring for shaken images*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 491–498.
- [54] N. WIENER, *The Fourier Integral and Certain of Its Applications*, Cambridge Mathematical Library, Cambridge University Press, Cambridge, UK, 1988.
- [55] H. WU, M. RUBINSTEIN, E. SHIH, J. GUTTAG, F. DURAND, AND W. T. FREEMAN, *Eulerian video magnification for revealing subtle changes in the world*, ACM Trans. Graphics, 31 (2012), pp. 65:1–65:8.
- [56] W. XU AND S. MCCLOSKEY, *2D barcode localization and motion deblurring using a flutter shutter camera*, in Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV), 2011, pp. 159–165.