A formalization of the \textit{flutter shutter}

Yohann Tendero, Bernard Rougé, Jean-Michel Morel  
Centre de Mathématiques et de Leurs Applications (CMLA)  
École Normale Supérieure de Cachan - 61 av du Pdt Wilson 94235 Cachan Cedex France.  
E-mail: tendero@cmla.ens-cachan.fr

Abstract. Acquiring good quality images of moving objects by a digital camera remains a valid question. If the velocity of the photographed object is not known, it is virtually impossible to tune an optimal exposure time. For this reason the recent Agrawal et al. \textit{flutter shutter} apparatus has generated much interest. In this communication, we propose a mathematical formalization of a general \textit{flutter shutter} method, also permitting non-binary shutter sequences. Thanks to this formalization, the question of the optimal \textit{flutter shutter} code can be defined and solved. The method gives analytic formulas for the best attainable SNR for the restored image. It also gives a way to compute optimal flutter shutter codes.

1. Introduction

If a scene being photographed moves during the exposition process, or if the scene is steady and the camera moves, the resulting images can be degraded by motion blur. Thus, moving scenes require a “small” exposure time, and the number of sensed photons remains limited. The difficulty of motion blur is illustrated by its simplest example, the one dimensional uniform motion blur. The result is a convolution of the image with a one dimensional window shaped kernel. The support of the kernel increases linearly with the exposure time and the velocity of the motion. As soon as the exposure time exceeds a limit, the blur is no more invertible \cite{1}. The restoration process becomes an ill posed problem. An exciting alternative to this classic photography dilemma was proposed in \cite{2} where the authors suggest modifications in the acquisition process to get invertible blur kernels by using a \textit{flutter shutter}. If the shutter sequence is well chosen, invertibility is guaranteed for blurs with arbitrary size supports (see Figs 1, 2). Replacing the classic camera shutter by a \textit{flutter shutter}, it becomes possible to use any integration time. This also means that the exposure time on a given scene can become much longer: many more photons are therefore sensed. Thus, at first sight, the \textit{flutter shutter} looks like a magic solution that should equip all cameras. The question arises of whether it can indefinitely increase the SNR by increased exposure. As we shall see, the answer is that the \textit{flutter shutter} does not permit an indefinite increase of the SNR, no matter how long the exposure time is. This fact, that we will call the “\textit{flutter shutter paradox}”, will be translated into closed formulae for the optimal \textit{flutter shutter} and for the SNR of the deconvolved image. The theory will be complemented with simulated experiments that can be also executed on line at Image Processing On Line \cite{3}.

In section 2, we reformulate carefully the first steps of the image acquisition model using a physical Poisson model for the photons capture process. The \textit{flutter shutter} is formalized in section 3, where the SNR estimates are also given. This communication ends with an unexpected
mathematical answer to the question of the optimal flutter shutter code. (Flutter shutter codes have usually been optimized by random searches [4, 5, 6, 7, 8]). The proposed formalism and results also apply to the recent motion-invariant photography [9], where the authors suggest to accelerate the camera during the exposure, to get an invertible blur.

2. Image acquisition model
Formalizing the flutter shutter requires an accurate continuous stochastic model of the photon capture by a sensor array. Without loss of generality the formalization will be done in the case where the sensor array is 1D and where the photographed object is conceived as a “landscape” moving in a direction parallel to the sensor array. Let the sensor array is 1D and where the photographed object is conceived as a “landscape” capture by a sensor array. Without loss of generality the formalization will be done in the case of ∆t is a Poisson random variable 1] × [x − 1 2, x + 1 2] | X ∼ P means that a random variable X has law P, and * denotes the convolution ((f * g)(x) := ∫ −∞ ∞ f(y)g(x − y)dy). For sampling purposes we assume that the theoretical landscape l is seen through an optical system with a point spread function g.

Definition We call ideal landscape the deterministic function u = 1 [− 1 2, 1 2] * g * l, where g is the point spread function (PSF) of the optical system providing a cutoff frequency.

In other words, thanks to the convolution with g the acquisition system is able to sample u. We shall denote by u(x) the ideal (noiseless) pixel landscape value at a pixel centered at x, as it could only be obtained after infinite exposure. Notice that the landscape u contains in itself all spatial integrations required, from the PSF g and from the normalized pixel sensor. We assume in the sequel that u ∈ L1 ∩ L2(ℝ).

2.1. Sampling, interpolation
Since the optical kernel g provides a cutoff frequency, u is band-limited, namely 1 2 π,π u(x)e−ixξdx is supported in [−π, π]. It could therefore be sampled at unit rate. The discrete sensor observations, or samples, will be denoted by e(n) for n ∈ ℤ.

Given a discrete array observation e(n), n ∈ ℤ, its band limited interpolate e(x) x ∈ ℝ is defined by the Shannon-Whittaker interpolation as

\[ e(x) = \sum_{n \in \mathbb{Z}} e(n) \text{sinc}(x - n) \quad (\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}) \]  

(1)

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

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

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

(2)
Likewise, we call “spectral-averaged $SNR$”:

$$SNR_{\text{averaged}}(\tilde{u}_{\text{est}}) := \frac{\frac{1}{2\pi} \int \mathbb{E}[\tilde{u}_{\text{est}}(\xi) \mathbb{1}_{[-\pi,\pi]}(\xi)]d\xi}{\sqrt{\frac{1}{2\pi} \int \text{var}(\tilde{u}_{\text{est}}(\xi) \mathbb{1}_{[-\pi,\pi]}(\xi))d\xi}}.$$  \hspace{1cm} (3)

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

3. The numerical flutter shutter

The numerical flutter shutter method consists in gain modifications taking place after the acquisition by the sensor. This numerical set up of the flutter shutter is possible for example with CMOS sensors, or with a CCD matrix with adequately chosen gains at each sensor. The set up can also be described as a camera taking a burst of $N$ images with an exposure time $\Delta t$. The $k$-th image is multiplied, for $k \in 0, ..., N - 1$, by an $\alpha_k \in \mathbb{R}$ gain. Then all images are added up to obtain one observed image. The analog flutter shutter considered in [2] is a bit more restrictive: indeed the use of a physical shutter in front of the lens is only possible with positive gains. For a sake of conciseness in this communication, we will only state our results for the numerical flutter shutter, but homologue statements exist for the analog flutter shutter. An obvious objection to the utility of the numerical flutter shutter is that one could keep all images, or perform a multi-image denoising instead of adding them all. The set up is nevertheless relevant for Earth observation satellites, where an uncontrolled blur is caused by a drift in TDI’s (time differential integrators). In that case the resulting blur estimate could be eliminated by a flutter shutter, without any additional transmission burden if only the flutter shutter image (the sum) was transmitted. The $k$-th acquired elementary image at a pixel at position $n$ is a realization of $P \left( \int_{k \Delta t}^{(k+1) \Delta t} u(n - vt) dt \right)$. The flutter shutter observation is obtained by combining the $k$-th output with weight $\alpha_k$. Thus the flutter shutter output at a pixel centered at $n$ is

$$\text{obs}(n) \sim \sum_{k=0}^{N-1} \alpha_k P \left( \int_{k \Delta t}^{(k+1) \Delta t} u(n - vt) dt \right)$$  \hspace{1cm} (4)

where by construction $\text{obs}(n)$ are obtained for $n \in \mathbb{Z}$ and are independent. In the following it will be useful to associate with the flutter shutter its code defined as the vector $(\alpha_k)_{k=0, ..., N-1}$, and its flutter shutter function defined by $\alpha(t) = \alpha_k$ for $t \in [k, k+1]$. 

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

$$\text{obs}(n) \sim \sum_{k=0}^{N-1} \alpha_k P \left( \int_{k \Delta t}^{(k+1) \Delta t} u(n - vt) dt \right).$$  \hspace{1cm} (5)

We call flutter shutter its band limited interpolate $\text{obs}(x) \sim \sum_{n \in \mathbb{Z}} \text{obs}(n) \text{sinc}(x - n)$. We call flutter shutter function the function $\alpha(t) = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k,k+1]}(t)$.

**Theorem 3.1.** The observed samples of the flutter shutter are such that, for $n \in \mathbb{Z}$

$$\mathbb{E}(\text{obs}(n)) = \left( \frac{1}{v} \alpha \left( \frac{\cdot}{v \Delta t} \right) * u \right)(n) \text{ and } \text{var}(\text{obs}(n)) = \left( \frac{1}{v} \alpha^2 \left( \frac{\cdot}{v \Delta t} \right) * u \right)(n).$$  \hspace{1cm} (6)
Definition Assume that the estimated landscape obtained for \( n \) is \( \hat{\alpha}(\xi) \) and \( \alpha(\xi) \) is band limited, we can interpolate it using the band limited interpolation (1). The band limited interpolates of the ideal observation is

\[
e(x) = \sum_{n \in \mathbb{Z}} e(n) \text{sinc}(x - n).
\]

Then from (7) we have \( \hat{e}(\xi) = \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbbm{1}_{[-\pi,\pi]}(\xi) \). So the ideal deconvolved landscape \( d(x) \) obtained by combining the previous and \( \gamma(x) \) is \( \hat{d}(\xi) = \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbbm{1}_{[-\pi,\pi]}(\xi) / \Delta \alpha(\xi) \). We shall now adopt the same formulæ for the noisy case.

Definition Assume that the \textit{flutter shutter} with code \((\alpha_k)_{k=0, \ldots, N-1}\) is invertible. We call estimated landscape \( u_{\text{est,num}} \) of the \textit{flutter shutter} the function defined by (using the obtained \( \hat{e}(n) \) samples (4) instead of the ideal \( e(n) \) in \( \hat{d}(\xi) \)) \( \mathcal{F}(u_{\text{est,num}})(\xi) = \sum_{n \in \mathbb{Z}} obs(n) e^{-in\xi} \mathbbm{1}_{[-\pi,\pi]}(\xi) / \Delta \alpha(\xi) \).

\textbf{Theorem 3.2.} Given any flutter shutter function \( \alpha \), the flutter shutter has a spectral SNR (2) equal to \( \text{SNR}(\xi) = \| \hat{\alpha}(\xi) \|_2 \| \hat{\alpha}(\xi) \|_2 / \sqrt{\| \hat{\alpha}(\xi) \|_2 \| \hat{\alpha}(\xi) \|_2} \).

\textbf{Theorem 3.3.} Given a landscape \( u(xvt) \) moving at velocity \( v \), the ideal flutter shutter function is \( \alpha^\ast(t) = \text{sinc}(tv\Delta t) \).

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{image1.png}
\caption{Left: simulated noisy and blurry image acquired by a \textit{flutter shutter} camera with a 52 pixels blur support. Right: restored image. Those images were generated from a flutter shutter camera simulator, \url{http://dev.ipol.im/~tendero/ipol_demo/flutter_demo/}. It simulates a flutter shutter camera assuming a Poisson photon emission.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.4\textwidth]{image2.png}
\caption{On the left a \textit{flutter shutter} function \( \alpha \); x-axis \( k \), y-axis: gain. On the right its Fourier transform; x-axis \( \xi \), y-axis \( |\hat{\alpha}(\xi)| \). Notice that \( \hat{\alpha}(\xi) \neq 0 \forall \xi \in [-\pi, \pi] \).
\end{figure}
This means that the best $SNR_{\text{averaged}}$ of the recovered image is achieved by the $sinc$ function. Notice that this flutter shutter function has a constant Fourier transform on the support of $\hat{u}$ for any velocity $|\tilde{v}| \leq |v|$. This means that this flutter shutter code is (Figs. 3, 4) “self-deconvolving”. Being non piecewise-constant (coming from a code) this function is not directly implementable using a numerical flutter shutter strategy but it can be proved that a piecewise constant equivalent function can be used instead.

Figure 3. The flutter shutter gain function for the sinc-code (left). The Fourier transform (modulus) of the sinc – code (right), approximating the Fourier transform of the ideal gain function.

**Corollary 3.4.** The ideal flutter shutter has an averaged $SNR$ (3) equal to $SNR_{\text{averaged}} = \frac{1}{2\pi \sqrt{v}} \frac{\int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{\|u\|_L^1}}$.

4. Conclusion
These results mean that the $SNR$ of the flutter shutter remains bounded despite the use of an infinite exposure time accumulating many more photons than a short snapshot. Given a

Figure 4. Sinc code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 1.33$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change). The acquired image is “sharp”, it is no surprise since the sinc-code has a nearly constant Fourier transform thus, it does not alter any frequency.
reference velocity $v$ for the observed object, it can also be proved analytically that among snapshots (classic shutter functions of the form $\alpha(t) = \frac{1}{\Delta t}[0, \Delta t](t)$), the best choice is $\Delta t^* \approx \frac{1.0707}{|v|}$ (see Fig. 5). The ratio of SNRs between the best flutter shutter and the best snapshot is $\approx 1.17$. In other words the flutter shutter provides a slight SNR improvement, even when the motion is known. A paradoxical conclusion is this: for the agile numerical cameras of the future, the best camera exposure strategy will always be a flutter shutter.

Figure 5. This figure shows the RMSE curves for different snapshots kinds, on five test images (House, Alley, Boat, Cameraman, Peppers). On the $x$–axis, the blur support $(|v|\Delta t)$ in pixels, on the $y$–axis the corresponding RMSE. The curves confirm that, on average, the blur support for a standard camera should be of approximatively $\Delta t^* = 1.0909$ pixels. A larger support would lead to a better SNR on the observed image samples, but the deconvolution would entail a lower SNR on the deconvolved image. The best snapshot is a compromise between the number of photons caught during a time span $\Delta t$ and the deconvolution kernel. It gives a reference to compare all flutter shutter strategies in terms of SNR.

References
[1] Boracchi G and Foi A 2011 Image Processing, IEEE Transactions on 20 592–598
[2] Agrawal A and Raskar R 2007 Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on (IEEE) pp 1–8
[3] Tendero Y The flutter shutter camera simulator http://dev.ipol.im/~tendero/ipol_demo/flutter_demo/
[4] Agrawal A and Xu Y 2009 Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on (IEEE) pp 2066–2073
[5] Jelinek J 2010 Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series vol 7701 p 18
[6] Raskar R, Tumblin J and Agrawal A 2009 Method for deblurring images using optimized temporal coding patterns US Patent 7,580,620
[7] Raskar R 2010 Method and apparatus for deblurring images US Patent 7,756,407
[8] McCluskey S, Jelinek J and Au K 2009 Method and system for determining shutter fluttering sequence US Patent 12/421,296
[9] Levin A, Sand P, Cho T, Durand F and Freeman W 2008 ACM Transactions on Graphics (TOG) vol 27 (ACM) p 71