

# Optimizing flutter shutter to minimize camera blur

March 23 2016

---



A simulated blurry and noisy image is reconstructed by direct deconvolution.  
Credit: Yohann Tondero and Jean-Michel Morel.

Whether taking photos recreationally or professionally, photographers understandably want their snapshots to appear sharp and clear. Image clarity is dependent on exposure time, or the amount of time that a camera's sensor is exposed to light while a photograph is being taken. During this period, the shutter opens and the camera counts the number of photons emitted by the subject.

Because photographers typically have minimal control over scene

lighting, often the only way to heighten the signal-to noise-ratio (SNR), and thus a digital camera's sensitivity, is to integrate more photons and increase exposure time. Unfortunately, this also increases the chance of either the camera or the scene moving during the exposure process, resulting in motion blur that lessens photo quality. In contrast, a shortened exposure time could produce a grainy effect called noise. "If the time-exposure is long, there is a high risk of motion blur," say mathematicians Yohann Tanderö and Jean-Michel Morel. "If the time-exposure is short, the resulting image is noisy."

In a paper publishing this week in the *SIAM Journal on Imaging Sciences*, Tanderö and Morel develop a closed formula meant to reduce motion blur.

Flutter shutter, also known as coded exposure, is a camera application that de-blurs photos. "The flutter shutter was a great discovery; it seemed to promise a solution to one of the major problems of photography: the noise/blur dilemma," say Tanderö et al. "It allows for arbitrarily long exposure, without blur."

Flutter shutter causes a camera's shutter to open and close repeatedly based on a pseudo-random binary sequence, which increases exposure time. A shutter sequence, called a flutter shutter code, then reveals intervals where the photon flow experiences interruption. A successful flutter shutter code guarantees an invertible motion blur kernel that reverses severe blur, resulting from arbitrarily high velocities. Yet there is a limit: when the velocity of the camera or scene is constant, a flutter shutter is incapable of gaining more than a 1.17 factor in terms of root mean square error (RMSE), when compared to an optimal snapshot. This term is respectable, but a higher factor would be more sensitive and thus yield a clearer image.

Tanderö and Morel seek to enhance the use of flutter shutter in cameras

to relax this optimality bound. "We wanted to increase the SNR by more than 1.17," say the authors. "So we tried to reformulate the problem by noticing that the real problem was different: the velocity of the observed objects is unknown and is different from one scene to another. The scene might contain few and slow moving objects, many fast moving objects, etc. Yet the probability distribution of the objects' velocities is more steady and can often be observed and learned from the acquired images. This is the case, for example, for a surveillance camera which observes the same scene 24/7: the proportion of slow versus fast objects is relatively constant with the time."

While past studies have addressed ways to achieve invertible motion blur, most involve applications that are more complicated than the original flutter shutter. Tendero and Morel present a new closed formula that allows computation of optimal codes for any probability density of the expected scene velocities. This formula links optimal flutter shutter codes and velocity distributions.

"It is rewarding to find simple closed formulas for an applied mathematics problem," say Tendero et al. "Their proof is generally very instructive. Their use is easy and they bear their own intuition. But probably the best point is that our closed formula permits us to do a reverse engineering: given any flutter shutter proposed in an existing paper or patent (and there are many), the formalism we develop allows us to predict their underlying velocity distribution, namely the one they are optimal for and the associated gain in SNR."

The authors' formula makes it possible to associate optimal velocity distribution with an existing flutter shutter code. Using their stochastic velocity model, they surpass the previously-established 1.17 bound gain for known velocities, provided that the scene's velocity is known or able to be learned. The resulting flutter shutter code is invertible for all velocities.

Tendero and Morel see future potential uses for their research in multiple scenarios. "A consequence of our work is that the flutter shutter might have promising application when coupled with algorithms learning the velocities," say the authors. "Class of such algorithms is called optical flows and can be very efficient. This image acquisition chain can be applied to surveillance cameras, but also to camera-phones. In that case the [motion blur](#) is spatially variable and only a statistic of the velocities is really observable."

**More information:** A Theory of Optimal Flutter Shutter For Probabilistic Velocity Models, *SIAM Journal on Imaging Sciences* (to be published)

Provided by Society for Industrial and Applied Mathematics

Citation: Optimizing flutter shutter to minimize camera blur (2016, March 23) retrieved 2 May 2024 from <https://techxplore.com/news/2016-03-optimizing-flutter-shutter-minimize-camera.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--