HAL
open science

Mathematical theory of the Flutter Shutter : its paradoxes and their solution

Yohann Tendero

To cite this version:

Yohann Tendero. Mathematical theory of the Flutter Shutter : its paradoxes and their solution. General Mathematics [math.GM]. École normale supérieure de Cachan - ENS Cachan, 2012. English. NNT : 2012DENS0029 . tel-00752409

HAL Id: tel-00752409

https://theses.hal.science/tel-00752409v1

Submitted on 15 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ÉCOLE NORMALE SUPÉRIEURE DE CACHAN
Doctoral School of Practical Sciences

P H D T H E S I S

to obtain the title of

DOCTOR OF PHILOSOPHY
Specialty : MATHEMATICS

Defended by

Yohann TENDERO

MATHEMATICAL THEORY OF THE *FLUTTER SHUTTER*

Its Paradoxes and Their Solution

ENSC - 2012n°368

Advisors: Jean-Michel MOREL and Jérôme GILLES

Jury :

Reviewers :	Alessandro FOI	-	Tampere University of Technology, Tampere
	Stanley OSHER	-	University of California, Los Angeles
	Guillermo SAPIRO	-	University of Minnesota, Minneapolis
Advisors :	Jean-Michel MOREL	-	CMLA, École Normale Supérieure de Cachan
	Jérôme GILLES	-	University of California, Los Angeles
Examinators :	Frédéric CAO	-	DxO Labs
	Bernard ROUGÉ	-	CESBIO, Centre National d'Études Spatiales
	Véronique SERFATY	-	Direction Générale de l'Armement

June 22, 2012

Préface

Résumé
Cette thèse apporte des solutions théoriques et pratiques à deux problèmes soulevés par la photographie numérique en présence de mouvement, et par la photographie infrarouge. La photographie d'objets en mouvement semblait ne pouvoir se faire qu'avec des temps d'exposition très courts, jusqu'à ce que deux travaux révolutionnaires proposent deux nouveaux types de caméra permettant un temps d'exposition arbitraire. Le *flutter shutter* de Agrawal *et al.* [3] crée en effet un flou inversible, grâce à un obturateur aux séquences d'ouverture-fermeture bien choisies. Le *motion-invariant photography* de Levin *et al.* [69] obtient ce même effet avec une accélération constante de la caméra. Les deux méthodes suivent ainsi un nouveau paradigme, la *computational photography*, selon lequel les caméras sont repensées, car elles incluent un traitement numérique sophistiqué. Cette thèse propose une méthode pour évaluer la qualité image des nouvelles caméras. Le fil conducteur de l'analyse est donc l'évaluation du SNR (signal to noise ratio) de l'image obtenue après déconvolution. La théorie fournit des formules explicites pour le SNR, soulève deux paradoxes de ces caméras, et les résout. Elle permet d'obtenir le modèle de mouvement sous-jacent à chaque *flutter shutter*, notamment tous ceux qui sont brevetés. Une seconde partie plus brève aborde le problème de qualité principal en imagerie vidéo infrarouge, la non-uniformité. Il s'agit d'un bruit évolutif et structuré en colonnes causé par le capteur. La conclusion des travaux est qu'il est non seulement possible mais également efficace et robuste d'effectuer la correction sur une seule image. Cela permet de contourner le problème récurrent des *"ghost artifacts"* résultant d'une incohérence du traitement par rapport au modèle d'acquisition.

La théorie du *flutter shutter*

Une caméra numérique est un dispositif qui, en chaque pixel, compte les photons émis par le paysage (scène) observé durant un intervalle de temps Δt appelé temps d'exposition. A cause de la nature de l'émission de ces photons le nombre de photons compté est une variable aléatoire de Poisson. La différence entre la valeur idéale et la valeur réellement comptée par la caméra est appelée "*shot noise*" (bruit). Le rapport entre la moyenne de cette variable de Poisson et son écart type est appelé SNR. Il mesure la fluctuation relative du nombre de photons mesuré par le capteur. A (très) bas SNR le bruit est si fort que le paysage sous-jacent est à peine visible. Dans un système passif il n'est pas possible de "booster" cette émission de photons (par un flash) et le seul moyen d'augmenter le SNR est d'augmenter le temps d'exposition Δt. Si, le paysage et la caméra sont en mouvement relatif durant le temps d'intégration des photons il en résulte un flou de bougé (Fig. 1). Du point de vue mathématique un flou de bougé est une convolution du paysage observé par une fonction porte dont le support à la longueur du flou, c'est-à-dire la distance en pixels parcourue par la ligne de visée de la caméra durant le temps d'intégration. De fait une telle convolution n'est pas inversible en général

Figure 1 – A gauche: une image floue (et bruitée) acquise par une caméra *flutter shutter* numérique. Le flou a un support de 52 pixels. A droite: l'image déconvolée. Ces images ont été produites par un simulateur (publié [128]). Il simule l'image acquise à partir de l'émission (Poisson) de photons. Une telle déconvolution n'est pas possible sans utiliser une caméra *flutter shutter*.

(dès que le support du flou dépasse deux pixels) puisque la transformée de Fourier du noyau est une fonction *sinc*. Dès lors que, comme c'est le cas pour un satellite, la dérive de l'instrument est imposée et les moyens de calculs limités, le seul moyen de garantir l'inversibilité est de contraindre le temps d'intégration de sorte que le flou ne dépasse jamais deux pixels (c'est également le cas avec le dispositif *time delay integration*, TDI).

Récemment deux méthodes révolutionnaires d'acquisition ont été proposées. Elles permettent une exposition longue avec un flou de bougé, rendu déconvolable. Ces solutions permettent d'augmenter indéfiniment le temps d'intégration, le nombre de photons collectés, le SNR de l'image acquise. Toutes deux sont issues de la communauté de la "*computational photography*". La première méthode, le *flutter shutter* de Agrawal *et al.* [3] propose d'ouvrir et fermer le diaphragme de la caméra (interrompant le flux de photons) selon un code bien choisi. La seconde, la *motion-invariant photography* de Levin *et al.* [69] propose, paradoxalement, de bouger la caméra avec une accélération constante dans la direction de la vitesse v_0. Dans les deux cas la fonction de flou est changée et devient inversible, une seule image est transmise et la déconvolution est numérique. La question se pose alors de savoir si, *après* déconvolution, le SNR reste lui aussi arbitrairement grand. L'état de l'art ne répondait pas à cette question.

Nous donnons une formalisation mathématique [133] prouvant que ces méthodes fonctionnent effectivement et avons calculé le SNR de l'image déconvolée pour n'importe quelle configuration de caméra (standard, la *motion-invariant photography* et un *flutter shutter* quelconque). A vitesse v_0 fixée, cette étude [135] permet de calculer le temps d'exposition optimal pour une caméra standard. Ce meilleur cliché permet, par exemple, de calculer la longueur optimale (en nombre d'étages) d'un TDI, le critère étant le meilleur SNR pour l'image restaurée et non pour l'image acquise. Dans la littérature, l'optimisation d'un *flutter shutter* n'est faite que par des recherches aléatoires qui conduisent à des résultats moins bons qu'une caméra standard du fait du grand nombre de codes possibles. Il y a 2^{52} codes à tester dans le cas de Agrawal *et al.* mais un cas pratique conduit à un nombre bien plus grand encore. La conclusion de ce chapitre est qu'à la fois les codes publiés et brevetés [80, 98, 99] et la *motion-invariant photography* [70] sont, contrairement aux revendications des auteurs, en réalité moins bons en termes de SNR qu'une caméra standard bien utilisée lorsque la vitesse de l'objet photographié est connue. Ce fait peut être vérifié expérimentalement grâce à notre article [128] présentant une démonstration en ligne.

Nous avons donné deux généralisations (analogique et numérique) du *flutter shutter* dont la faisabilité technique a été étayée en nous appuyant sur des brevets dans chacun des cas. Il apparaît que le *flutter shutter* numérique est toujours

meilleur que l'analogique, permet plus de degrés de liberté et est plus facile à optimiser (dans la suite les résultats quantitatifs mentionnés ne concernent que le *flutter shutter* numérique). La conclusion de ce chapitre est que, quel que soit le *flutter shutter*, et contrairement à ce que laisse entendre la littérature, même à temps d'exposition infini le SNR reste fini. Cela signifie qu'augmenter le temps d'intégration peut, paradoxalement, conduire à une réduction du SNR de l'image restaurée, comme c'est le cas pour une caméra standard. Nous avons néanmoins montré qu'à temps d'intégration constant la caméra *flutter shutter* est toujours meilleure que la caméra standard (de 4% à v_0 fixé). Ce gain correspond non pas à un plus grand nombre de photons collectés mais à un gain provenant du noyau de déconvolution. Ceci constitue un premier paradoxe du *flutter shutter*. Nous avons optimisé, analytiquement, le *flutter shutter* pour obtenir le SNR maximum. Le gain est de 17% sur le SNR par rapport au cliché standard en supposant un temps d'intégration infini, ce qui est peu. C'est le second paradoxe. Le code optimal est auto-déconvolant et permet de déconvoler toutes les vitesses $|v| \leq |v_0|$. Cette optimisation correspond, dans une situation réelle, à une optimisation du pire cas.
Par la suite, nous sommes parvenus à contourner le deuxième paradoxe et à augmenter le rendement du *flutter shutter* au-delà de 17%, avec une solution stochastique au problème. En supposant que la vitesse v est inconnue mais que l'on connaît sa densité de probabilité (qui peut s'apprendre des images acquises au cours d'une phase de calibration de l'instrument) on peut calculer le *flutter shutter* optimal. Ce cadre de travail permet, par exemple, de traiter le cas où plusieurs objets bougent à des vitesses différentes, leurs surfaces relatives fournissant la densité de probabilité. Il permet d'optimiser non pas le pire des cas mais le cas moyen à risque minimal. Il permet de calculer [134] *analytiquement* la fonction d'obturation optimale et d'en déduire une approximation constante par morceaux utilisable par une caméra. Nous sommes donc en mesure de fournir [129], pour n'importe quelle densité de probabilité sur v, le meilleur code à appliquer (le meilleur *design* de caméra) pour un *flutter shutter* et aussi pour une caméra standard. Dans ce cas nous donnons le temps d'obturation optimal. Nous prédisons le gain en SNR du code d'obturation du *flutter shutter* optimisé. Toutefois, pour des modèles de vitesse plausibles le gain par rapport à une caméra standard utilisée à son maximum est modeste (25%, en moyenne pour une distribution Gaussienne et en multipliant le temps d'intégration par un facteur 10 par rapport au meilleur cliché). Le seul cas où le gain est intéressant est le cas d'une distribution de la forme $\rho(v) = (1-\epsilon)\delta_0(v) + \epsilon\delta_{v_0}(v)$. Le *flutter shutter* se rapproche alors du multi image avec recalage, pour un SNR moyen calculé sur toute l'image et en supposant v_0 grand et ϵ petit. Nous donnons également un moyen de calculer la densité de probabilité en fonction du "code" d'obturation du *flutter shutter*. Cela permet de procéder au "*reverse engineering*" de toutes les caméras *flutter shutter* "optimisée" et brevetées (Agrawal *et al.*, McCloskey *et al.*, etc.). Nous en déduisons que chaque code est optimal pour une certaine distribution de vitesse.
Une conclusion paradoxale serait ceci: face à une scène fixe avec des objets en mouvement rapide, le gain devient substantiel. Le *flutter shutter* garantit un SNR arbitrairement grand et il permet néanmoins d'obtenir une image nette des objets en mouvement. Toutefois, face à une scène en mouvement à vitesse connue l'apport du *flutter shutter* est modeste.

Restauration des images infrarouges

Dans une caméra infrarouge thermique non refroidie la fonction de transfert de chaque pixel (photo-site) est inconnue, différente et évolue dans le temps. Cela signifie que pour n photons comptés par le pixel on ne sait pas combien ont été réellement reçus à un instant donné. La différence entre les fonctions de transfert des pixels provoque la non-uniformité (Fig. 2). De plus pour des raisons technologiques (lecture du capteur) la non-uniformité n'est pas décorrélée et possède une structure en ligne ou en colonne. En effet, la réponse propre de chaque compteur d'électrons est différente, elle aussi. Il faut donc trouver, pour *chaque* pixel, une fonction qui n'est pas linéaire, et évolue dans le temps.

Le but de l'étude est de faire se rejoindre sur le plan de la qualité image les (coûteuses, lourdes et gloutonnes en énergie) caméras refroidies (pour lesquelles le facteur temps disparaît) et les caméras non refroidies. Une correction est si nécessaire que dans certaines caméras un diaphragme se ferme, interrompant donc l'acquisition, périodiquement (toutes les 10-30s) afin de procéder à une nouvelle calibration (*"one point NUC"*, *" two points NUC"*). Les solutions

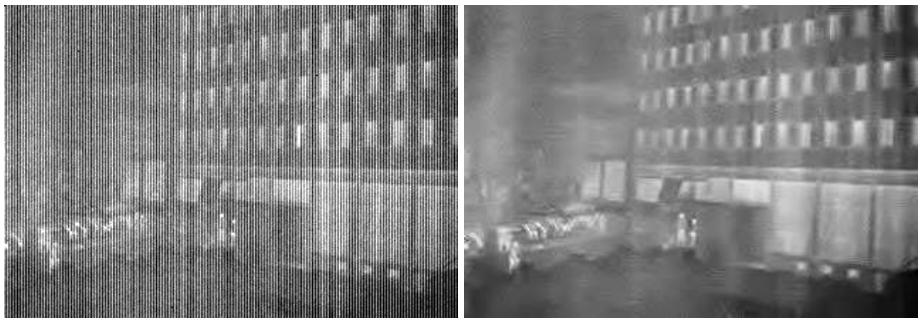

Figure 2 – Sur la gauche, une image raw prise par une caméra infrarouge (bande LWIR). La non-uniformité dans la réponse des capteurs provoque les stries. Sur la droite, la solution proposée (qui n'utilise que l'image de gauche) (publiée [130]).

algorithmiques actuelles ne sont pas satisfaisantes en termes d'amélioration de la qualité image. Pire encore, elles introduisent de nouveaux artéfacts: les *"ghost artifacts"*. En effet, depuis 10 ans l'état de l'art sur le sujet s'est focalisé sur une réponse multi images à ce problème. Les solutions de l'état de l'art sont soit stochastique soit correspondent, *in fine*, à la création d'un panorama. Les solutions stochastiques sont toutes basées sur l'hypothèse que [$\mathcal{H}$:] les histogrammes temporels des pixels sont égaux sur un intervalle de temps fixé. Ceci n'est bien sûr pas vrai en général (sauf si nous pouvions déplacer tous les pixels entre eux pour toutes les permutations possibles et si les fonctions des pixels étaient constantes dans le temps). Elles imposent à l'utilisateur des conditions particulières de mouvement de la caméra. L'utilisateur est forcé de ne cesser de balayer la scène pour assurer l'hypothèse [$\mathcal{H}$], ce qui requiert un grand nombre d'images. Elles estiment toutes la fonction du pixel en la supposant constante sur un intervalle de temps, ce qui est incompatible avec la recherche d'une fonction "à tout instant". De plus l'estimation est une approximation linéaire de la fonction du pixel. De fait, lorsque l'hypothèse [$\mathcal{H}$] est affaiblie (par exemple lorsque le paysage change : un véhicule arrive, etc.) les résidus de l'approximation linéaire constante par morceaux dans le temps se surimposent sur les nouvelles images ("en creux" voir, par exemple [42, 106]). Ces résidus perdurent du fait du grand nombre d'images utilisé et leur correction est difficile car on ne peut décider aisément si un changement dans l'observation provient d'une dérive de la non-uniformité ou d'un changement dans la scène. Les panoramas, eux, permettent de corriger la non linéarité du capteur. Toutefois, dans les images le "bruit de non-uniformité" (Fig. 2) domine. Dans ces conditions seul un recalage global (homographique) est envisageable. Ces algorithmes ne peuvent s'appliquer que s'il n'y a pas d'effet de perspective (scène vue de très loin) et n'apportent pas de solution satisfaisante. Pour toutes ces raisons les algorithmes de l'état de l'art ne sont pas du tout adaptés, ne produisent pas une qualité suffisante, et introduisent de nouveaux problèmes qu'il faut ensuite corriger [106].

Nous proposons une solution mono-image basée sur des changements de contrastes locaux. L'algorithme est basé sur [30] et permet de compenser une non-uniformité non linéaire, sans modèle, et automatiquement (zéro paramètre). Etant mono-image, elle garantit l'absence de *"ghost artifact"*. Nous l'avons encore améliorée en la rendant localement adaptative [130], cette partie est toujours automatique. Nous avons également défini une mesure de la qualité image, basée sur la $RMSE$ (root mean square error) permettant de s'affranchir des changements de contraste et reflétant mieux la qualité perceptuelle afin de pouvoir donner une base de comparaison fiable de tous ces algorithmes. Elle peut s'appliquer (et devrait dès lors que la chaîne peut introduire un léger changement de contraste) quel que soit le type d'algorithme afin de fournir une mesure quantitative plus pertinente. Nous pensons parvenir dans les prochains mois à une chaîne automatisée de correction de non-uniformité/débruitage (il faut estimer le bruit dans l'image) mono image et films tout en garantissant l'absence de *"ghost artifacts"* (robustesse).

Mots clés : *flutter shutter*, *motion-invariant photography*, cliché, flou de bougé, SNR, bruit de Poisson, optimisation, infrarouge, bruit spatial fixe, matrice de plan focal, correction de non-uniformité (NUC), débruitage.

Foreword

Abstract
This thesis provides theoretical and practical solutions to two problems raised by digital photography of moving scenes, and infrared photography. Until recently photographing moving objects could only be done using short exposure times. Yet, two recent groundbreaking works have proposed two new camera designs allowing arbitrary exposure times. The *flutter shutter* of Agrawal *et al.* [3] creates an invertible motion blur by using a clever shutter technique to interrupt the photon flux during the exposure time according to a well chosen binary sequence. The *motion-invariant photography* of Levin *et al.* [69] gets the same result by accelerating the camera at a constant rate. Both methods follow *computational photography* as a new paradigm. The conception of cameras is rethought, to include a sophisticated digital processing. This thesis proposes a method for evaluating the image quality of these new cameras. The *leitmotiv* of the analysis is the SNR (*signal to noise ratio*) of the image *after* deconvolution. It gives the efficiency of these new cameras design in terms of image quality. The theory provides explicit formulas for the SNR, raises two paradoxes of these cameras, and resolves them. It provides the underlying motion model of each *flutter shutter*, including patented ones. A shorter second part addresses the the main quality problem in infrared video imaging, the non-uniformity. This perturbation is a time-dependent noise caused by the infrared sensor, structured in columns. The conclusion of this work is that it is not only possible but also efficient and robust to perform the correction on a single image. This permits to ensure the absence of "ghost artifacts", a classic of the literature on the subject, coming from inadequate processing relative to the acquisition model.

The *flutter shutter* theory

A camera is a device counting at each pixel sensor, the number of photons emitted by the observed scene during an interval of time Δt called exposure time. (We neglect the randomness in the electron generation when photons arrive in the semiconductor.) Due to the nature of photon emission the number of photons counted is a Poisson random variable. Its mean would be the ideal pixel value. The difference between this ideal value and the actual value counted by the sensor is called (shot) noise. The ratio of its mean over its standard-deviation is called *signal to noise ratio* (SNR). It measures the relative fluctuation of the number of photons caught by the sensor. At (very) low SNR the noise is so strong compared to the underlying signal that it is almost impossible to distinguish the scene being observed from the noise. Therefore, from the beginning photography has been striving to achieve the highest possible SNR. In passive imaging systems there is no control over the scene itself. Thus no lighting is possible, to boost the photon emission. Hence, the only way to increase the SNR is to integrate more photons by increasing the exposure time Δt. If a scene being photographed moves during the exposition process, or if the scene is still and the camera moves, the resulting images

Figure 3 – Left: simulated noisy and blurry image acquired by a *flutter shutter* camera with a 52 pixels blur support. Right: restored image. Those images were generated from a *flutter shutter* camera simulator [128]. It simulates a *flutter shutter* camera assuming a Poisson photon emission. Such a deconvolution is not possible without a *flutter shutter*.

are degraded by motion blur. The difficulty of motion blur is illustrated by its simplest example, the one dimensional uniform motion blur (the relative velocity v_0 between the camera and the landscape is constant). The result of a too long exposure during the motion on the image is nothing but a convolution of the image with a one dimensional window shaped kernel. The support of the kernel increases linearly with the exposure time Δt and the velocity v_0 of the motion. If the exposure time is too long and the blur support exceeds two pixels, the blur is no more invertible and makes the restoration process to an ill posed problem.

As soon as, like for satellites, the motion is imposed (by its rotation around the Earth) and the computational capabilities very limited the only mean to ensure the invertibility is to constraint the exposure time Δt such that the blur support never exceeds two pixels (notice that it is also the case with the *time delay integration* device, TDI). Recently, two revolutionary techniques have been proposed to create invertible motion blurs of arbitrary length. These techniques permit to increase the exposure time Δt, the number of photons sensed and the SNR on the observed image arbitrarily while guaranteeing an invertible kernel. The Agrawal *et al.* *flutter shutter* apparatus [3, 69] suggests modifications in the acquisition process to get invertible blur kernels by using a clever shutter method. The authors propose to interrupt the flux of photons by opening and closing the shutter of the camera during the exposure time Δt (Fig. 3) according to a well chosen binary sequence. Paradoxically the Levin *et al.* *motion-invariant photography* suggests to accelerate the camera at a constant rate in the direction of v_0. In both cases the kernel is no more a window shaped function and is made invertible. For both apparatus only one image has to be transmitted. At first sight the *flutter shutter* looks like a magic solution that should equip all cameras. The question is to know whether or not the SNR *after* deconvolution remains arbitrarily high. This question was unanswered by the state of the art.

To study the *flutter shutter*, the first steps of the image formation is reformulated using a physical Poisson model for the photons capture process, including the obscurity noise. This model is necessary for the *flutter shutter* where all noise terms inherent to image sensing must be taken into account without any approximation. This study led us to formulate new questions, which can be termed "best snapshot theory". It is proven that for a known velocity v_0 the best snapshot has an exposure time Δt such that $|v_0|\Delta t \approx 1.0909$ (this best snapshot can be used to compute the optimal number of stages in a TDI device). It is verified mathematically and numerically that the *flutter shutter* actually works. The SNR of the deconvolved image is computed, for *any flutter shutter* function. It includes the standard camera, the *motion-invariant photography* and *any flutter shutter*. The study generalizes and analyzes the *flutter shutter* in digital

and analog implementations. The difference is that the *numerical flutter shutter* allows for negative gains, while the analog only allows for positive non piecewise constant gains. It is proven that the *numerical flutter shutter* beats the *analog flutter shutter* for the image quality (SNR) and is always more flexible to use. The technical aspects of feasibility for both proposed generalizations are supported by existing patents of imaging devices. The design of the sequence –a piecewise constant kernel– is crucial and classic literature [3] looks for a sequence that maximizes the modulus of the discrete Fourier transform by random search among the sequences of fixed integral. This is inaccurate as it neglects the blur induced by the constant part of the kernel. The "optimized" sequences are worse than the best snapshot and yield a lower SNR, due to the huge number of possible binary codes : 2^{52} in the case of Agrawal *et al.* but a practical case can lead to a much bigger number. This fact can be checked *online* using [128]. It is proven that even using the same time aperture the *flutter shutter* does always beat the standard camera, by 4%. This slight improvement comes from the deconvolution. It is proven, analytically, that for a fixed velocity v_0 the best *flutter shutter* comes from the Fourier series coefficient of a (zoomed) *sinc* function and that the SNR remains finite, no matter how long the exposure time. This optimal code is self-deconvolving, and is able to deconvolve any velocity $|v| \leq |v_0|$. It is proven that it increases the SNR by +17% compared to the best snapshot, even though the exposure time is infinite. This optimization is a worst case optimization.

Nevertheless, a better mouse trap was found, it increases the efficiency of the *flutter shutter* beyond the 17% bound by using a stochastic solution. It is proven that, on average, the SNR can increase significantly provided the probability density of the velocity v is *a priori* known (it is possible to estimate this probability density during a calibration phase). This framework is well suited to the case of multiple objects and/or velocities. Their relative surface provides the probability density. This corresponds to the optimization of the average case at minimal risk. Our solution permits to compute analytically the best *flutter shutter* function and to deduce the best code to use in the camera. Thus, given *any* probability density on v this thesis computes the best aperture strategy for a *flutter shutter*, a standard camera and compare their SNR. The gain, using a *numerical flutter shutter* is of 25% assuming a Gaussian distribution on v for an exposure time increased by a factor 10. In the case of a distribution of the form $\rho(v) = (1-\epsilon)\delta_0(v) + \epsilon\delta_{v_0}(v)$, the *flutter shutter* competes with a multi-image fusion scheme. It is proven that given *any flutter shutter* code it is possible to deduce its underlying probability density on v. This permits to proceed to a reverse engineering of all optimized and patented cameras (Agrawal *et al.*, McCloskey *et al.*, etc.). This thesis deduces that each *flutter shutter* code is optimal for some probability density on the observed velocities.

The conclusion is that the *flutter shutter* is useful, and even SNR-efficient if the observed objects are moving at high and unknown velocities. In this case the *flutter shutter* guarantees an arbitrarily high SNR and a sharp image. Nevertheless, if the velocity of the observed scene is known the gain in SNR is modest compared to a standard camera.

Restoration of infrared images

The standard readout technique of CCD devices works for each line (or row) independently and consists of transporting charges from pixels to a counter. Each pixel has its own transfer function response. Furthermore for each line the counter transfer function is different and in most cases the *non uniformity* (NU) presents some structured noise resulting in a row or line pattern in the sensed images. This noise is called *non uniformity* and comes from the differences between pixel transfer functions. For uncooled infrared cameras the difficulty is even increased as the detector response evolves quickly with time. This means that for an equal amount of photons counted by the camera the (true) number of photons sensed is unknown and drifts with time. Thus, we need to estimate for each pixel a non linear function[11], that evolves with time. The goal is to obtain the image quality of heavy, expensive and energy consuming cooled infrared cameras using a cheap uncooled camera. The *non uniformity* is a serious practical limitation to both civilian and military applications as it degrades the image quality severely (see Fig 4). A correction is so much needed, that in many uncooled infrared cameras a flap closes every 10-30 seconds to perform a partial calibration ("one point NUC", "two points NUC"). This interrupts the image flows, which can be calamitous. Therefore a good *non uniformity* algorithmic correction is a key

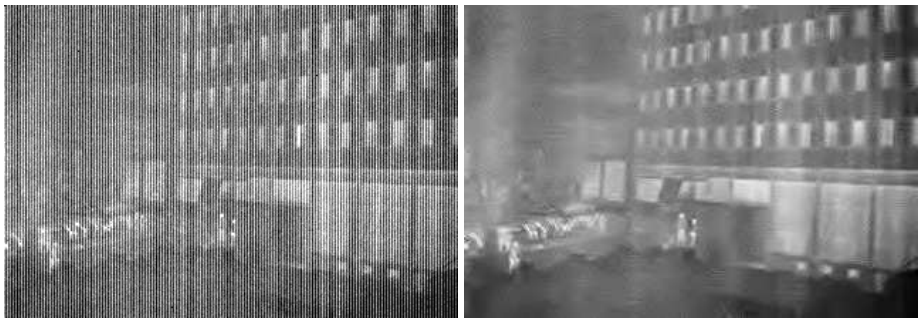

Figure 4 – On the left the RAW (input) image taken by an LWIR infrared camera. The *non uniformity* caused the vertical stripes. On the right the proposed solution, using only the image on the left.

factor in ensuring the best image quality and the robustness of the downstream applications.
The classic literature on the subject contains two kinds of algorithms, both working on movies. None of them give satisfactory results in terms of image quality. Even worse, some of them actually create new artifacts. The non uniformity correction methods are either stochastic, or equivalent to the creation of a panorama. The stochastic solutions [43, 46, 113, 115, 138, 139] are based on the hypothesis [$\mathcal{H}$:] all temporal pixel histograms should be equal on some time span. To ensure [$\mathcal{H}$:] they require particular conditions of observation and/or camera motion. The user is forced to sweep, non stop, many parts of the scene which requires the use of a large number of images. In order to perform the estimation (of the transfer functions) they assume piecewise constant functions through time. This contradicts the model essentially because of the large number of images needed. The estimation is a linear approximation. When the hypothesis [$\mathcal{H}$:] is no more valid (a vehicle arrives, etc.) residues of the correction as well as the previous landscape remains superimposed in the subsequent frames. Those are the "ghost artifacts". The usual method to avoid these artifacts is to restart the learning process. Nevertheless, the detection of scene changes is treacherous in presence of *non uniformity* because it is not possible to decide if a change in the observation comes from the *non uniformity* side –by the time drift– or from the scene itself.

The second kind of method uses a warping of the images and roughly creates a panorama [41, 160]. Since the noise dominates, only homographies are possible. This means that the scene must be seen at a very large distance.

In order to avoid the "*ghost artifacts*" and the image warping this thesis proposes to achieve the *non uniformity* correction in the image itself. The algorithm is based on [30] and applies local contrast changes. It is parameterless and can be tested online [132]. Also it can correct for non linearities of the *non uniformity* without any model of the *non uniformity*. The resulting images are quite neat (see Fig 4). This thesis proposes an image quality measure, based on the $RMSE$ (root-mean-square error) which permits to get rid of contrast changes and is best suited to the perceptual quality. This measure provides a reliable quantitative criterion the compare all those algorithms without any bias.

Keywords : *Flutter shutter*, *motion-invariant photography*, snapshot, motion blur, SNR, Poisson noise, optimization, infrared, fixed pattern noise, focal plane array, *non uniformity* correction (NUC), denoising.

Acknowledgement

I'd like to thank the Direction Générale de l'Armement for my PhD grant. This thesis has been also partially financed by the MISS project of Centre National d'Etudes Spatiales, the Office of Naval research under grant N00014-97-1-0839 and by the European Research Council, advanced grant "Twelve labours".

Je remercie tous ceux sans qui cette thèse ne serait pas ce qu'elle est : aussi bien par les discussions que j'ai eu la chance d'avoir avec eux, leurs suggestions ou contributions. Je pense en particulier à Bernard Rougé et Jean-François Aujol.

Je remercie également toutes les personnes que j'ai eu la chance de rencontrer lors de mon stage d'initiation à la recherche au sein de Thales Alénia Space: Guillaume, Frédéric, Marc, Aurélie, Natalie et Pierre.

Merci aussi à tous les habitants et ex-habitants de la "boite à thésards" (merci Loïc pour ce joli nom!), pour leur bonne humeur, les goûters et les Traditions : Adina, Aude, Ayman, Benjamin. Merci à Bruno pour avoir sû nous motiver pour aller au grand restaurant américain. Merci à Frédérique, Gabriele, Julie, Irène, Ives, José, Marc, Mathieu, Mauricio, Miguel, Rafael, Saad, Sammy. Thanks to Zhongwei Tang who came to town. Merci à Ives pour avoir été mon voisin pendant un an et demi. Merci aussi à Barbarra, Carlo, Claire, Enric, Lara et Rachel (bien peu de personnes peuvent se vanter d'avoir acheté une centrale avec des bananes). Merci à Nicolas C (cet homme possède le tapis de souris le plus classe du monde) et à Eric (il y aura des chipseu à mon pot) tant il est vrai qu'il est essentiel de ne pas confondre la coquetterie et la classe. Merci Morgan pour être d'une excellente compagnie dans nos Traditions. Nicolas L. (IPOL c'est lui) qui a déployé tous ses efforts pour que je puisse regarder Tron II malgré nos "assoupissements".

Je remercie Tristan Dagobert et Stéphane Landeau pour leur bonne humeur, leurs crépières et grâce à qui j'ai pu obtenir de belles images et tester les algorithmes en condition réelles.

Merci aussi à El Commandanté Neus, Sébastien, Rafael Grompone von Gïoi (le meilleur professeur d'Espagnol) et Mauricio (el estetoscopio esta en el piano). Le voyage, les gens que nous avons croisé, les lieux et le Sheriff sont inoubliables.

Merci aussi à Clothilde Melot, Marie-Christine Roubaud, Marina Talet et Bruno Torrésani.

On ne peut quitter le CMLA sans remercier tout le secrétariat pour leur disponibilité, leur travail formidable et leur bonne humeur. Merci donc à Micheline, Véronique, Virginie Sandra et Carine.

Thanks to Nick the Bear who struggled not to carpet bomb me. I know that in case of trouble I can apply for a grease monkey position. IK.

Merci aussi aux potes Fabien, Stef et Marilou, Maxime, Yanna, Hugues , Emilie, Flo et Nico.

Enfin un grand merci à Al, Françoise, Marjorie et Kadi BM pour leur soutien, leur aide. Je n'aurai pu y arriver sans eux.

Contents

Main Notations and Formulas **ix**

I The Theory and Practice of Invertible Motion Blurs **1**

I Introduction **3**
1 Related work 5
2 Overview 9

II Still Photography Theory **11**
1 Mathematical modeling 11
2 The still photography case 12
3 Sampling, interpolation 13
4 Noise measurement 14
5 Standard acquisition, the SNR with no blur 15
6 Image acquisition with a moving landscape 16
7 The multi-image fusion to improve the SNR 18

III The *Numerical Flutter Shutter* **21**
1 The *numerical flutter shutter* 21
2 *Flutter shutter* design: from continuous to discrete 28

IV The *Analog Flutter Shutter* **31**
1 The *analog flutter shutter* 31
2 Comparison of a piecewise constant *analog flutter shutter* with the *numerical flutter shutter* 34

V Examples of *Flutter Shutter* **37**
1 Snapshots 37
2 *Motion-invariant photography* 43

VI The *Flutter Shutter* Paradox **49**

VII A Stochastic Solution to the *Flutter Shutter Paradox* **53**
1 From motion to codes . . . 53
2 From codes to motion . . . 56
3 Best snapshot on average . . . 57

VIII Numerical Simulations **61**
1 All *flutter shutters* . . . 61
2 A reverse engineering of classic *flutter shutter* codes . . . 77
3 Simulations on optimized codes . . . 79
4 Estimating the kernel . . . 89

IX Discussion and Conclusion **93**

II Restoration of Infrared Images **95**

I *Non Uniformity* Correction **97**
1 Introduction . . . 97
2 Image acquisition model . . . 98
3 Related work . . . 100
4 The midway infrared correction . . . 102
5 The Adaptive and Denoising midway equalization algorithm (ADMIRE) . . . 105
6 Experiments . . . 108
7 Discussion and conclusion . . . 111

Annex **117**
1 Proof of $\hat{\alpha}_{MIP-ideal}(\xi) = \frac{1}{\sqrt{|a\xi|}} e^{-i\frac{\pi}{4} sign(\xi)}$ . . . 117

List of Tables

A.1 This table summarizes the results on *numerical* and *analog flutter shutters* (observed samples, inverse filter and SNR). 34

A.1 This table provides the relative $SNR^{spectral-averaged}$ of the *motion-invariant photography* compared to the best snapshot. 46

A.1 This table summarizes the results on the different *flutter shutter* strategies and their SNR. 51

A.2 This table provides the relative $SNR^{spectral-averaged}$ of all standard *flutter shutter* strategies compared to the best snapshot. 52

A.1 Quantitative ($RMSE$, $RMSE^{CI}$) comparison of different *flutter shutter* strategies. 61

A.2 The ratio of SNR, on average, between the optimized snapshot and the *flutter shutter* $R(v)$, Gaussian density for the velocities. 83

A.3 The ratio of SNR, on average, between the optimized snapshot and the *flutter shutter* $R(v)$, uniform density for the velocities. 87

A.4 The ratio of SNR, on average, between the optimized snapshot and the *flutter shutter* $R(v)$, handcrafted density for the velocities. 88

List of Figures

A.2 *Sinc* code : observed, deconvolved, residual noise (house image). 6

A.1 Acquisition system. 17

A.1 Best snapshot energy. 41
A.2 $RMSE$ curves for different snapshots kinds (varying the blur support $|v|\Delta t$). . . . 42
A.3 The *flutter shutter* gain function for the *motion-invariant photography* code and its Fourier transform (modulus). 47

A.1 Best snapshot on average energies. 59

A.1 The *flutter shutter* gain function for a snapshot and its Fourier transform (modulus). 68
A.2 The *flutter shutter* gain function for the accumulation code and its Fourier transform (modulus). 69
A.3 The Agrawal *et al.* *flutter shutter* gain function and its Fourier transform (modulus). 69
A.4 The *flutter shutter* gain function for a random code and its Fourier transform (modulus). 69
A.5 The *flutter shutter* gain function for a *motion-invariant photography* code and its Fourier transform (modulus). 70
A.6 The *flutter shutter* gain function for a the *sinc* code and its Fourier transform (modulus). 70
A.7 Test images. 71
A.8 Snapshot : observed, deconvolved, residual noise (boat image). 72
A.9 Accumulation : observed, deconvolved, residual noise (boat image). 72
A.10 Agrawal *et al.* code : observed, deconvolved, residual noise (boat image). 73
A.11 Random code : observed, deconvolved, residual noise (boat image). 73
A.12 *Motion-invariant photography* code : observed, deconvolved, residual noise (boat image). 74
A.13 *Sinc* code : observed, deconvolved, residual noise (boat image). 74
A.14 Snapshot : observed, deconvolved, residual noise (house image). 75
A.15 Accumulation : observed, deconvolved, residual noise (house image). 75
A.16 Agrawal *et al.* code : observed, deconvolved, residual noise (house image). 76

A.17 Random code : observed, deconvolved, residual noise (house image). 76
A.18 *Motion-invariant photography code* : observed, deconvolved, residual noise (house image). 77
A.19 *Sinc* code : observed, deconvolved, residual noise (house image). 78
A.20 Test images credits 78
A.21 The probability densities associated with Agrawal *et al.* codes: x-axis motion (in signed pixels), y-axis the *Log* of the probability. On the left: the code published in [3]. On the right: the code published in [6]. It corresponds to an attempt to optimize both the SNR and the PSF estimation. Notice that both probability densities are nonzero even for large motion blurs. 79
A.22 On the left: the probability density associated with the McCloskey code [78]: x-axis motion (in signed pixels), y-axis probability. On the right: the probability density of velocities associated with the best snapshot code, a "1.0909 pixel blur integration". On the x-axis, the velocities (in signed pixels), on the y-axis the corresponding probability density. This snapshot is optimized *a priori* for objects moving at velocity $|v|$. This bimodal density is natural for a traffic surveillance camera. 79
A.23 Optimized codes, Gaussian density for the velocities (exposure time factor 1 and 2). 84
A.24 Optimized codes, Gaussian density for the velocities (exposure time factor 5 and 10). 84
A.25 The ratio of SNR's between the optimized snapshot and the *flutter shutter* $R(v)$, Gaussian density for the velocities. 85
A.26 Optimized codes, uniform density for the velocities (exposure time factor 1 and 2). 86
A.27 Optimized codes, uniform density for the velocities (exposure time factor 5 and 10). 86
A.28 The ratio of SNR's between the optimized snapshot and the *flutter shutter* $R(v)$, uniform density for the velocities. 87
A.29 Optimized codes, handcrafted probability density. 88
A.30 The ratio of SNR's between the optimized snapshot and the *flutter shutter* $R(v)$, handcrafted density for the velocities. 89
A.31 Observed image (Agrawal *et al.* code. 90

A.2 The midway algorithm. 103
A.3 The proposed *non uniformity* correction algorithm on a non corrupted image. . . . 105
A.4 The proposed *non uniformity* correction algorithm, without denoising. 106
A.5 The need for a locally adaptive algorithm. 107
A.6 Simulated *non uniformity* (boat image). 109
A.7 Simulated *non uniformity* (house image). 109
A.8 Comparison with total variation based algorithm. 110
A.10 Result of the proposed *non uniformity* correction algorithm, infrared image. 111
A.11 Result of the proposed *non uniformity* correction algorithm, infrared image. 111

A.12 Result of the proposed *non uniformity* correction algorithm, infrared image. 112
A.13 Result of the proposed *non uniformity* correction algorithm, infrared image. 112
A.14 Result of the proposed *non uniformity* correction algorithm, infrared image. 113
A.15 Result of the proposed *non uniformity* correction algorithm, infrared image. 113
A.16 Result of the proposed *non uniformity* correction algorithm, hyperspectral image. . 114
A.17 Result of the proposed *non uniformity* correction algorithm, hyperspectral image. . 115

Main Notations and Formulas

(i) $t \in \mathbb{R}^+$ time variable

(ii) Δt length of a time interval

(iii) $x \in \mathbb{R}$ spatial variable

(iv) $X \sim Y$ means that the random variables X and Y have the same law

(v) $\mathbb{P}(A)$ probability of an event A

(vi) $\mathbb{E}X$ expected value of a random variable X

(vii) $var(X)$ variance of a random variable X

(viii) $f * g$ convolution of two $L^2(\mathbb{R})$ functions $(f * g)(x) = \int_{-\infty}^{+\infty} f(y)g(x-y)dy$

(ix) $l(t,x) > 0 \ \forall \ x \in \mathbb{R}^+ \times \mathbb{R}$ continuous landscape before passing through the optical system

(x) $\mathcal{P}(\lambda)$ Poisson random variable with intensity $\lambda > 0$

(xi) $\mathbf{P}_l$ bi-dimensional Poisson process on $\mathbb{R}^+ \times \mathbb{R}$ associated to the intensity field $l(t,x)$, $\mathbf{P}_l([t_1,t_2] \times [a,b]) \sim \mathcal{P}\left(\int_{t_1}^{t_2} \int_a^b l(t,x)dtdx\right)$

(xii) g *point-spread-function* of the optical system

(xiii) $u = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * l$ ideal observable landscape just before sampling. Assumption: $u \in L^1 \cap L^2$, band-limited

(xiv) $obs(n)$, $n \in \mathbb{Z}$ observation of the landscape at pixel n

(xv) $e(n) = \mathbb{E}(obs(n))$, $n \in \mathbb{Z}$ and $e(x)$ its band limited interpolation

(xvi) $\mathbf{P}_u$ Poisson process associated to the intensity field $u > 0 : = \mathbf{P}_u \sim \mathbf{P}_{\mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * l}$

(xvii) $\prod_a = \sum_{n \in \mathbb{Z}} \delta_{na}$ Dirac comb

(xviii) v relative velocity between the landscape and the camera (unit: pixels per second)

(xix) b apparent length of the support of the blur (unit: pixels), $Lv\Delta t = b$

(xx) $\alpha(t)$ piecewise constant or continuous gain control function for the *analog flutter shutter* & *numerical flutter shutter* methods

(xxi) $w(x) \geq 0$ weight function associated with the probability distribution of the velocity v

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(xxxii) $RMSE^{CI}\,(u, u_{est}) := RMSE\left(u_{mid(u,u_{est})}, u_{est_{mid(u,u_{est})}}\right)$ (Contrast invariant $RMSE$ (see 1.5))

Part I

The Theory and Practice of Invertible Motion Blurs

Chapter I

Introduction

Classic digital cameras are devices counting at each pixel sensor the number of photons emitted by the observed scene during an interval of time Δt called exposure time. Due to the nature of photon emission the counted number of photons is a Poisson random variable. Its mean would be the ideal pixel value. The difference between this ideal mean value and the actual value counted by the sensor is called (shot) noise. The ratio of the mean of the photon count over its standard-deviation is called Signal to Noise Ratio (SNR). At (very) low SNR the noise is so strong compared to the underlying signal that it is almost impossible to distinguish the scene being observed from the noise. Therefore, photography has been striving to achieve the highest possible SNR. In passive imaging systems, where there is no control over the scene lighting, the only way to increase the SNR is to integrate more photons by increasing the exposure time Δt.

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

Motion deblurring is the combination of two dependent problems a) the kind of kernel applied to the images which depends here on the motion b) the actual deblurring method, where the kernel may have to be estimated *a posteriori*, or not. Motion blur arises from multiples causes and is very common even for consumer level photography where it is partly compensated by optical, mechanical, or digital stabilizers. Lens based optical stabilizers use a floating sensor moving orthogonally to the optical axis. Vibrations are detected using accelerometer sensors estimating the camera motion in the x and y axis of the focal plane. Hence it cannot compensate for camera rotations. These techniques cannot compensate for motion blur of arbitrary length (support), since they are limited by mechanical and technical issues. In most cases the size of the blur support will increase proportionally to the exposure time. Thus they require a “small” exposure time despite the stabilization device. The difficulty of

motion blur is illustrated by its simplest example, the one dimensional uniform motion blur. The result of a too long exposure during the motion on the image is nothing but a convolution of the image with a one-dimensional window shaped kernel. The support of the kernel increases linearly with the exposure time and the velocity of the motion. As soon as the exposure time is too long, this blur is no more invertible, and the restoration problem is ill posed.

A revolutionary alternative to classic photography was proposed in [3, 4, 6, 98, 99] where the authors suggest modifications in the acquisition process to get invertible motion blur kernels by using a *flutter shutter*. These authors propose to use a binary shutter sequence interrupting the flux of incoming photons on well chosen time sub-intervals of the exposure time interval. If the shutter sequence is well chosen, invertibility is guaranteed for blurs with arbitrary size support. Hence, replacing the classic camera shutter by a *flutter shutter*, it becomes possible to use any integration time. This also means that the exposure time on a given scene can be much longer: many more photons are therefore sensed by the camera. Thus, the *flutter shutter* looks like a magic solution that should equip all cameras. Yet, does that mean that one can increase indefinitely the SNR by an increased exposure, at no cost from the motion blur side?

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

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

Among the kinds of possible set ups, the most flexible, adaptive to all kinds of motions, is the *numerical flutter shutter*, which allows for negative gains. It is proven that it also can realize the best SNR. One of the striking results of this mathematical analysis is the proof that, when the object velocity is *a priori* known, the best *numerical flutter shutter* code is given by the Fourier series coefficients of a (zoomed) *sinc* function (see Figs. A.1 and A.2). The proposed formalism also permits to compute by a closed formula the optimal aperture time for a classic snapshot, when the velocity of the photographed object is known (see Fig A.3). This snapshot theory allows us to match on an equal footing the new *flutter shutter* apparatus against a plain old camera. This comparison leads to what we call the two *flutter shutter paradoxes*. The first surprising result is that the *flutter shutter* always beats slightly a standard camera, even when using exactly the same exposure time. On the other hand, an infinite exposure time, accumulating many more photons than a classic snapshot, does not grant an infinite SNR. This rather disappointing fact that makes motion photography significantly different from the classic steady photography where, by increasing the aperture time, any SNR can be achieved.

Fortunately, a better mouse trap permits to override the first *flutter shutter paradox* by assuming

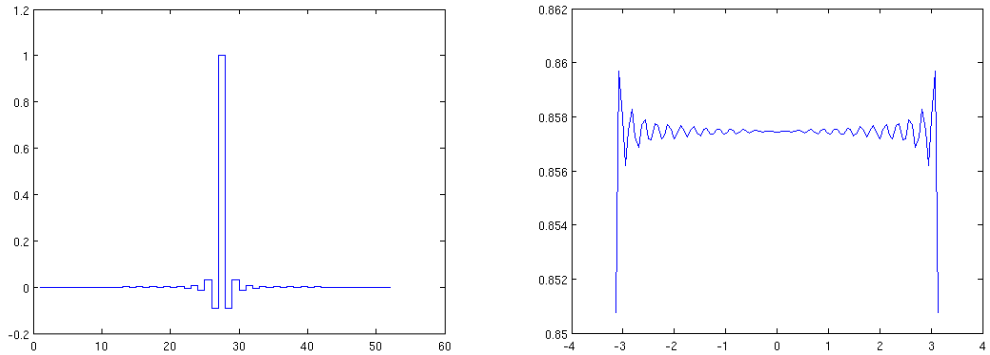

Figure A.1 – The *flutter shutter* gain function for the sinc-code (left). The Fourier transform (modulus) of the $sinc - code$ (right), approximating the Fourier transform of the ideal gain function.

that the observed objects adopt a known (or learned) random velocity distribution. Analytical formulas are proposed that link an optimal *flutter shutter* code (see Fig. A.4) with the probability density of the expected motion blurs. Furthermore, in presence of random motions, an adequate *flutter shutter* is proved to significantly increase the expected SNR (beyond the bound of the first *flutter shutter paradox*), with respect to an optimal snapshot. Not only this theory permits to formalize the design of optimized codes given random velocity model. Conversely, it also allows us to analyze *a posteriori any* existing *flutter shutter* strategy, and to perform a reverse engineering of existing patented codes.

1 Related work

Blind deconvolution techniques [17, 18, 39, 61, 66, 71, 119] aim at estimating the blur and recovering the sharp image directly from the blurred one. Deconvolution algorithms have been developed intensively, [24, 48, 54, 95, 108, 123, 151]. For example in [154, 157] the authors suggest a modification of the Richardson-Lucy method [74, 104] to control the artifacts of the restored image. Other priors have been investigated in [59, 77, 159]. In [67] Fergus *et al.* use natural image statistics to estimate the blur. In [7, 8, 22, 27, 28, 32, 50, 51, 53, 56, 57, 60, 103, 116–118, 121, 122, 145, 149] good results are shown for the blur estimation and/or deblurring problem. Using the compressive sensing framework, the question of the order of the pair image estimation/motion estimation for deconvolution is addressed in [52]. Nevertheless, the power spectrum of images acquired with a blur of more than two pixels contains several zero crossings. Thus, useful information for image quality is irreversibly lost. Hence, no matter how sophisticated the image reconstruction is, it is virtually impossible to recover a de-blurred image without strong hypotheses on the underlying landscape. Such strong hypotheses are unrealistic for most images. The results are therefore in practice poor [111]. In an attempt to transform the blur problem into a well posed problem the authors of [19, 21, 23, 100] proposed to use two photographs with different blurs instead of one. In [136, 156] the authors use a long exposure image and another

Figure A.2 – *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.

one, sharp but noisy, to deblur the first. In [6] the authors suggest to take several images with several exposure times so that the blur in each image is different. If the zeroes of each Fourier transform do not coincide then it is possible to deblur by picking non zero coefficients in each image. In [124] a similar hybrid scheme is used where an image at high resolution and long exposure is taken simultaneously with a burst of low resolution and short exposure. In [10] a Mumford-Shah like variational model is proposed to simultaneously estimate the blur and deblur in presence of multiple objects motion from videos.

In [110] the authors address the question of an automatic tuning of the exposure time to avoid overexposure in the case of still imaging. In [12] the authors treat the question of the optimal exposure time depending on the SNR of the restored image using a conventional camera. They consider the case of non invertible blurs with supports larger than two pixels, using a regularized deconvolution [33]. In [127] the authors use a full multi-image framework acquiring a bunch of sharp but noisy images and recovering a sharp image with increased SNR. For a review on multi image denoising the reader can refer to [15]. Conversely in [47] the authors reconstruct a movie from a single image using a temporally and spatially varying mask placed on the aperture. The mask helps encode the spatio-temporal information. In [35, 92, 125, 144, 150, 152] the authors use hybrid or complex camera systems. Unfortunately this kind of scheme may lead to other problems such as an expensive computational cost or hardware issues.

The simplest hardware set up seems to be proposed in [3, 4, 6] by Agrawal *et al.* The new acquisition process modulates the photon flux into the camera by opening and closing the camera shutter according to certain well chosen pseudo random binary codes. In the case of a uniform motion in front of the camera, the resulting blur kernel becomes invertible (there are no zeroes in its Fourier transform), however big the velocity is. The visual result of an image acquired by *flutter shutter* is

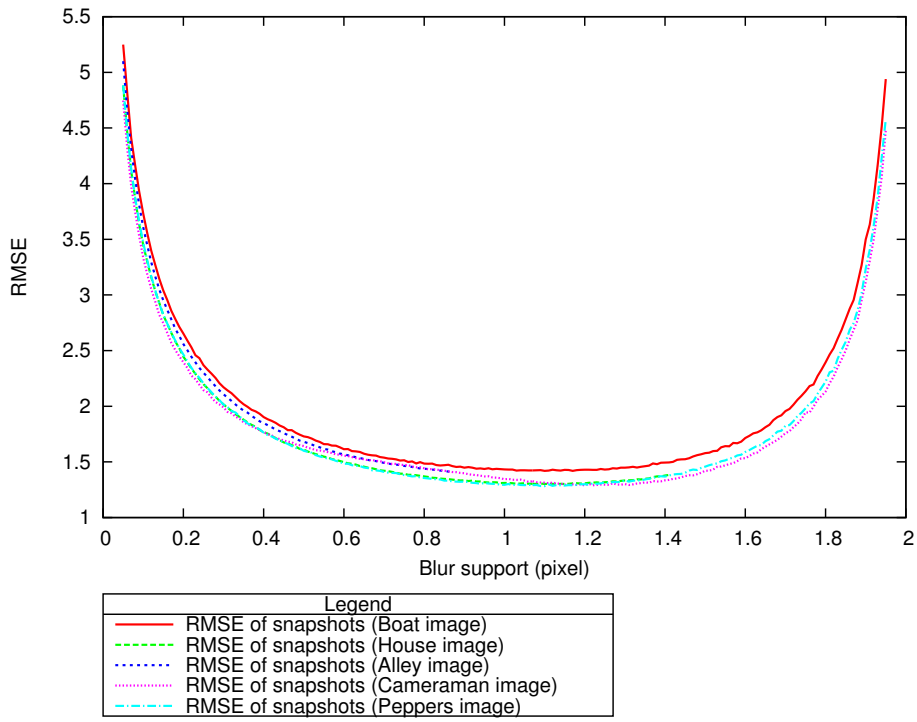

Figure A.3 – This figure shows the $RMSE$ curves for different snapshots kinds, on five test images (House, Alley, Boat, Cameraman, Peppers). On the $x-axis$, the blur support ($|v|\Delta t$) in pixels, on the $y-axis$ the corresponding $RMSE$. The curves confirm that, on average, the blur support for a standard camera should be of approximatively $\Delta t^* = 1.0909$ pixel. 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 Δt and the deconvolution kernel. It gives a reference to compare all *flutter shutter* strategies in terms of SNR.

close to a stroboscopic image, which can nonetheless give back a neat image by deconvolution. A compressive sensing *flutter shutter* camera was designed in [120] using random sequences where a blurry and low resolution image is acquired and processed to a neat and at high resolution image. Roughly speaking, the *flutter shutter* ensures that no information is lost by the motion blur; the compressed sensing technique deals with the increase of resolution. The compressed sensing technique is also used in [101] for spatio-temporal up-sampling. Alternatively the case of periodic events was investigated in [102]. In [63, 82, 90, 141] the authors use an active dynamic lighting pattern in place of the shutter to recreate a *flutter shutter* effect. The theory presented herewith works for this set up. In [79] the *flutter shutter* apparatus is applied to iris images and in [153] to bar-codes. In [78] the authors propose to optimize the binary *flutter shutter* code in function of the velocity of the scene. In [126] 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 [109] the authors treat the question of denoising an image taken by a *flutter shutter* camera, and also suggest an user assisted estimation of the blur. Their conclusion is that the denoising should be applied both

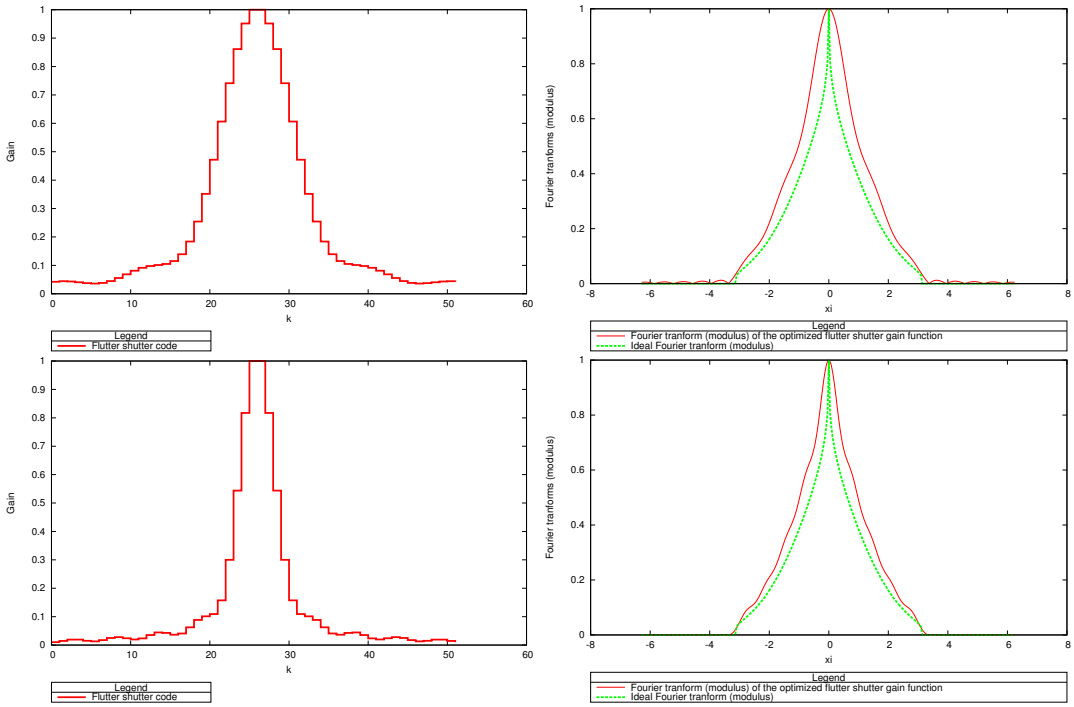

Figure A.4 – Codes obtained assuming a truncated Gaussian density for the velocities. On the left from top to bottom: the code for a truncated Gaussian velocity distribution measured ($x - axis$ k, $y - axis$ gain α_k) using an increase exposure time factor of 5 and 10. This means that the first code has an exposure time five times greater than the best snapshot on average, the second 10 times. On the right the corresponding Fourier transform (modulus) of the code (see III.18) and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ (VII.2) in bold. Those results permits to visualize the effect of the optimization. The conclusion is that nearly no more improvement can be expected from the convergence of the computed α_k coefficients of the code to the ideal $\sqrt[4]{w(\xi)}$ function. This is a consequence of Thm. 2.1 (chapter III).

before and after deconvolution. In [31] the authors treat the question of *a posteriori* motion estimation using a *flutter shutter*. In [40] a per pixel *flutter shutter* is used to build a camera that permits a post-capture balance between spatial and temporal resolutions of movies. A multi-camera equipped with *flutter shutters* is investigated in [2] and used to increase the frame rate of a single camera while having an increased amount of light captured compared to the equivalent hight-speed camera. A single camera equipped with a mask on the aperture and an array of light sources is used in [65] to construct the visual hull of an object (shape from silhouette). Another solution to get an invertible motion blur using only one image was found in [69] where Levin *et al.* suggested to move the camera in the direction of the motion during the exposure time. The authors use a constant acceleration motion in order to make the resulting kernel invertible and spatially invariant to the velocity. Hence an *a priori* knowledge of the motion direction is required. This approach has been generalized in [25] to the case of unknown directions, but it uses two images instead of one. In [81] the *motion-invariant photography* apparatus is implemented using the lens of the camera. In any cases, these approaches cause blur in static parts of the scene. Yet, thanks to the invertibility (well-posedness of the recovery problem),

in both cases, the sharp image can be recovered by a deconvolution. Notice that only one image is acquired and recovered at the end of the process. Alternatively in [9, 36, 68, 72, 76, 84, 88, 143] authors use a temporally fixed and spatially varying mask in order to estimate the depth, and/or refocus the out of focus part to get an always in focus (neat) image. In [45] the authors deal with the question of the optimal tradeoff between depth of field and exposure time. In [38] the authors take advantage of CMOS imaging sensors to implement a *coded rolling shutter* to trade vertical resolution for an increased dynamic range. The authors of [140] also suggest to use 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 "None of the possible combinations of aperture filter and deconvolution algorithm were able to consistently reduce the dynamic range of the captured image without excessively degrading image quality". Another computational camera is designed in [86] where the aperture is equipped with a mask and the sensor is moved at a constant velocity during the exposure. It is used to control the depth of field, create *bokeh* or a depth invariant blur size. Another camera prototype was designed in [73], where the authors suggest a programmable aperture (mask). It is also used for depth and digital refocusing. An interesting implementation, the *Frankencamera*, was proposed in [1]. It permits to "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". The authors demonstrate six computational photography applications. An even more complex scheme involving a fixed mask close to the sensor and dynamic one on the aperture is investigated in [5], where the authors explore the feasibility of post processing trade offs between spatial, angular and temporal resolutions. Finally reviews of computational photography can be found in [75, 96, 97, 161].

2 Overview

This thesis focuses on the various set ups permitting to acquire an image degraded by an invertible motion blur, namely the Agrawal *et al.* *flutter shutter* and the Levin *et al.* *motion-invariant photography*.

Chapter II section 1 proposes a general mathematical framework for image acquisition using a physical Poisson model for the photons capture process, including the obscurity noise. This model suits well our context since all noise terms inherent to image sensing are taken into account without any approximation. The model is detailed in a static context in section 2. Fourier-based SNR definitions are given in section 4, to take into account the deconvolution later on. The model is applied to the still photography in section 5. The non stationarity motion induced case is introduced in section 6 and the multi-images fusion is discussed in section 7.

In chapter III the mathematical model of chapter II is used to analyze the *numerical flutter shutter*, a digital implementation of the classic *flutter shutter* method. This set up is the most flexible, adaptive to all motion and allows for negative gains. The *numerical flutter shutter* does not reduce the number

of photons caught by the sensor and it is proven later on that it yields the best possible SNR. It is proven that it actually works and, for *any flutter shutter* gain function a formula providing the SNR of the neat deconvolved image is given. The *numerical flutter shutter* gain function is in principle piecewise constant nevertheless, it is useful for the theory to extend it to continuous gain functions. In section 2 a reverse formula permits to get back an equivalent piecewise constant *numerical flutter shutter*.

Chapter IV investigates classic analog implementation of the *flutter shutter*. This *analog flutter shutter* is a generalization of the original Agrawal *et al.* *flutter shutter* which allows for smoother, non binary, gain functions. For any *analog flutter shutter* apparatus, an explicit formula to measure directly the SNR of the deconvolved sharp image is given. Section 2 proves that the *numerical flutter shutter* SNR is always larger than the *analog flutter shutter* SNR with the same gain function.

A snapshot theory is developed in chapter V section 1. The standard camera apparatus is explored as a particular *flutter shutter* strategy. The SNR of the deconvolved image is calculated, for any standard acquisition strategy. The standard camera is optimized to get the best SNR possible, taking the deconvolution into account. This yields a precise definition of the best possible snapshot in presence of known motion. This best snapshot is used later on as a reference in terms of SNR. In section 2 the Levin *et al.* *motion-invariant photography* is proven to be a particular case of the general *analog flutter shutter* theory. The SNR of the *motion-invariant photography* apparatus is computed and compared with the other *flutter shutter* strategies. This section also proposes to implement the *motion-invariant photography* kernel using a *numerical flutter shutter*. This permits to generalize the *motion-invariant photography* method to the case where the direction of the relative velocity v is not *a priori* known.

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

Chapter VII proposes a solution to the first *flutter shutter paradox* theorem provided the probability density of the observed velocities is known. Section 1 gives analytical formulae that link an optimal *flutter shutter* code with *any* probability density of the expected velocities. A backward analysis, computing the probability density of *any* (patented) code is given in section 2. This framework is also applied to the standard camera in section 3.

All results are illustrated in chapter VIII. Section 1 shows simulations of several *flutter shutter* strategies, including the Agrawal *et al.* *flutter shutter* code and the Levin *et al.* *motion-invariant photography*. A reverse engineering of classic *flutter shutter* codes is performed in section 2. Section 3 provides optimized codes and comparisons with the best snapshot on average, for centered-Gaussian, uniform (with different sdt-dev σ and ranges) and an handcrafted distribution of the velocity.

Chapter II

Still Photography Theory

Abstract : *This chapter starts by modeling the stochastic photon capture by a light sensor, given that the photon flux is a Poisson space-dependent emission. It takes into account both the classic shot noise and the obscurity noise. To cope with the fact that the image noise is colored after deconvolution, a "spectral" definition is used for the signal to noise ratio (SNR). The SNR is computed without approximation, from the proposed model. The modeling will treat in the same formalism all possible types of flutter shutter, including an analog model, a digital model, the classic Agrawal et al. flutter shutter, and the Levin et al. motion-invariant photography.*

1 Mathematical modeling

This section presents a continuous stochastic model of photons captured by a sensor array. The model applies to a standard image acquisition on still or moving landscapes, provided the motion is uniform and stationary. Without loss of generality (w.l.o.g.) the formalization will be done in the case where the sensor array is 1D and the landscape moves in the direction given by the sensor.

Let $\mathbf{P}_l : \mathbb{R}^+ \times \mathbb{R}$ be a bi-dimensional Poisson process of intensity $l(t,x)$, $\forall (t,x) \in \mathbb{R}^+ \times \mathbb{R}$ (here l is called landscape, t and x are the time and spatial positions, respectively). This means that for every a, b, t_1, t_2 (with $a < b$ and $0 \leq t_1 < t_2$) $\mathbf{P}_l([t_1, t_2] \times [a, b])$ is a Poisson random variable with intensity $\int_{t_1}^{t_2} \int_a^b l(t,x)dxdt$. The theoretical observation of a pixel sensor (photon counter) of unit length centered at x during the time span $[0, \Delta t]$ is a Poisson random variable

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

where $[x - \frac{1}{2}, x + \frac{1}{2}]$ represents the normalized sensor unit, and $X \sim P$ means that a random variable X has law P. In other terms the probability to observe k photons coming from the landscape l seen

at the position x on the time interval $[0, \Delta t]$ and using a normalized sensor is

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

2 The still photography case

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

Then the theoretical observation of a pixel sensor (photon counter) of unit length centered at x using an exposure time of Δt is a Poisson random variable

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

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

Definition We call ideal landscape the deterministic function

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

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

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

Definition (Ideal acquisition system.) The image acquired by the *ideal* acquisition system, before sampling, corresponds to samples of the Poisson process $\mathbf{P}_l$. The intensity u (ideal landscape value) is related to the landscape l by (II.1) and is band limited.

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

Definition (Real acquisition system with noise included in the landscape.) A more realistic acquisition system adds a landscape independent noise also known as dark noise (or obscurity noise or thermal noise). Assuming that this noise has variance η (II.2) entails

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

Independently of their thermal cause, thermal photon emission is Poissonian. Furthermore, at each pixel sensor the number of thermal photons caught is independent of the number of photons caught coming from the landscape u. Therefore, the sum of the numbers of thermal and landscape photons hitting the sensor is a Poisson random variable whose intensity is the sum of the thermal and landscape intensities. Hence, all computations using the "noisy" landscape $u + \eta$ remain formally the same as with the noiseless ideal acquisition system defined in (II.2). The estimator considered is unbiased. Nevertheless this means that, in the sequel $u+\eta$ is estimated, not u. However, notice that u and $u+\eta$ only differ by a constant which only affects the mean of the image and does not change the perceptual quality. As a matter of fact it only impacts the null frequency of the SNR function as it is defined by Boracchi *et al.* in [12] and does not change the 'spectral-averaged" SNR of section 4 compared to estimating u. Therefore, w.l.o.g. we will assume that u already contains the obscurity noise in itself. Notice that η being a constant, $u + \eta$ and u have the same cut off frequency. We will also always assume in the sequel that $u \in L^1 \cap L^2(\mathbb{R})$. This assumption will be necessary to apply some of the mathematical formulas, but represents no artificial restriction on the acquisition physical model. Indeed, first, the average photon emission is always bounded. Second, taking a large enough support, we can always suppose w.l.o.g. that the landscape has bounded support and that the acquisition time is large but not infinite. Thus we can assume that the noise is zero at infinity. Under these conditions $u \in L^1 \cap L^2$.

3 Sampling, interpolation

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

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

$$e(x) = \sum_{n \in \mathbb{Z}} e(n)\mathrm{sinc}(x - n) \qquad (\text{ see in } \textbf{(xxix)} \text{ the definition of } \textit{sinc}.) \tag{II.4}$$

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

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

4 Noise measurement

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

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

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

$$SNR(\mathbb{u}_{est}(x)) := \frac{|\mathbb{E}\,\mathbb{u}_{est}(x)|}{\sqrt{var(\mathbb{u}_{est}(x))}}. \tag{II.7}$$

Likewise, we call "spectral SNR" of $\mathbb{u}_{est}$ the frequency dependent ratio defined by

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

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

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

Lets show first that this definition makes sense. In the sequel, the considered estimator is unbiased. This implies that $\mathbb{E}\hat{\mathbb{u}}_{est}(\xi) = \hat{\mathbb{u}}(\xi)$. Furthermore, since $u \in L^1(\mathbb{R})$ by Riemann-Lebesgue's we get that $\hat{u}(\xi)$ is continuous, so is $|\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)| = |\hat{\mathbb{u}}(\xi)|$. Hence $|\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)|$ is integrable on the interval $[-\pi,\pi]$. The variance term $var(\hat{\mathbb{u}}_{est}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi))$ is positive. Thus, $\int var(\hat{\mathbb{u}}_{est}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi))d\xi$ converge in $\bar{\mathbb{R}}_+$. It would be infinite if some frequency interval is lost and, in this case the $SNR^{spectral-averaged}(\hat{\mathbb{u}}_{est}) = 0$. This means that the above definition is correct, at least in a mathematical sense. Now, lets turn to its meaning.

Proposition 4.1. *Given $\hat{\mathbb{u}}_{est}(\xi)$ an unbiased estimator of $\hat{u}(\xi)$ then*

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

Proof. The mean squared error (MSE) can be decomposed into bias and variance terms. Since the estimator is unbiased the bias term is zero. Its variance term is equal to $\int var(\mathbb{u}_{est}(\xi))d\xi =$

$\int \mathbb{E}|\mathbb{u}_{est}(\xi)|^2 d\xi + \int |\mathbb{E}\mathbb{u}_{est}(\xi)|^2 d\xi$. Furthermore, on one hand we have

$$\int \mathbb{E}|\hat{\mathbb{u}}_{est}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)|^2 d\xi = \mathbb{E}\left(\int |\hat{\mathbb{u}}_{est}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)|^2 d\xi\right) \text{ (by Fubini for positive functions)}$$
$$= 2\pi\mathbb{E}\left(\int |\mathbb{u}_{est}(x)|^2 dx\right) = 2\pi \int \mathbb{E}|\mathbb{u}_{est}(x)|^2 dx \text{ (by Fubini, again).}$$

On the other hand, we have

$$\int |\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)|^2 d\xi = \int |\hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)|^2 d\xi \text{ (unbiased estimator)}$$
$$= 2\pi \int |u(x)|^2 dx = 2\pi \int |\mathbb{E}\mathbb{u}_{est}(x)|^2 dx.$$

Consequently,

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

Notice that, the naive average of the SNR function defined in [12], would not reflect the $RMSE$. Worse, it would not vanish if some frequency interval is lost. Such information loss implies that the *flutter shutter* is not invertible. The invertibility of the *flutter shutter* is discussed and the inverse filter is defined page 25. □

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

5 Standard acquisition, the SNR with no blur

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

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

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

Theorem 5.1. (Fundamental theorem of photography)
In the case of a still image, the SNR satisfies $SNR(x) = \sqrt{u(x)L\Delta t}$, where $L\Delta t$ is the total exposure time. It is therefore proportional to the square root of both the exposure time and the light intensity.

Remark In a passive optical system we have no control over the landscape light emission $u(x)$. No

lighting is possible, to boost the photon emission. Thus the only secure way to increase the SNR is to increase the exposure time $L\Delta t$.

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

$$\mathbb{u}_{est}(x) \sim \frac{\mathcal{P}\left(L\Delta t u(x)\right)}{L\Delta t}.$$

Then $\mathbb{E}(\mathbb{u}_{est}(x)) = u(x)$ and $\mathbb{u}_{est}(x)$ is an unbiased estimator for $u(x)$. The variance of $\mathbb{u}_{est}(x)$ is

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

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

6 Image acquisition with a moving landscape

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

Thus from now on we assume $l(t,x) = l(x - tv(t))$, and mainly $v(t) \equiv v$ where l is the underlying stationary landscape model (section 2). Hence all the former discussion made on the acquisition system, sampling and interpolation holds. This section focuses on the mathematical model for standard photography, where there is either a motion blur in a single continuous exposure, or the acquisition of an image burst (several consecutive snapshots with shorter exposure). We shall give the conditions under which the exposure is short enough to get a quality photograph, and evaluate the SNR in all three cases: snapshots, longer exposure with motion blur, and image burst made of short enough snapshots.

6.1 Standard motion blur

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

Theorem 6.1. *The standard motion blur is equivalent to an image obtained by a convolution of the ideal landscape u by a fixed (window shaped) kernel $\mathbb{1}_{[0,b]}$ where b is the blur length, equal to $Lv\Delta t$.*

Proof. The ideal landscape u is moving in the camera frame at a speed v (counted in pixels per second)

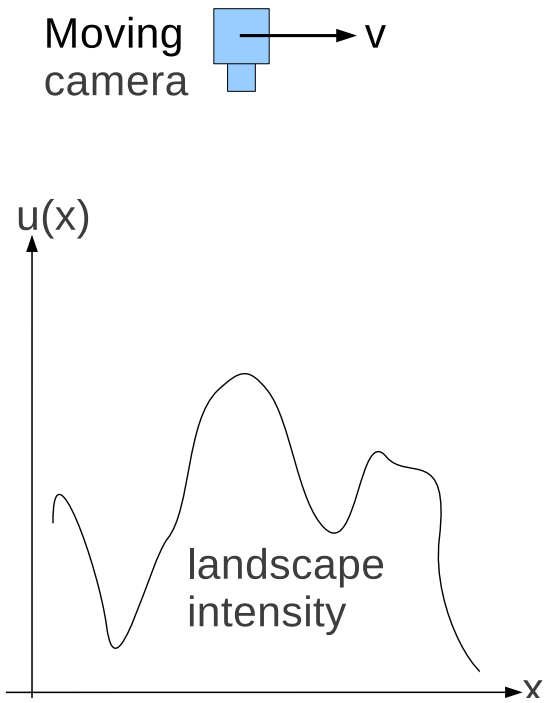

Figure A.1 – Acquisition system: the camera plane moves parallel to the focal plane in fixed direction, with scalar speed $v(t)$. $x - axis$: position, $y - axis$: intensity of photon emission by the landscape at position x. The exposure time $L\Delta t$ is assumed short enough, or the camera elevation large enough to ensure that there is no apparent deformation of the landscape due to its varying height.

and using (II.3) we get that the acquired image at position x can be *any* realization of

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

□

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

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

is said negligible and it is actually invertible.

7 The multi-image fusion to improve the SNR

The multi-image acquisition strategy consists in taking a burst of L images using a Δt exposure time. The exposure time Δt is taken small enough, so that the elementary blur of each one of the L images is negligible (see chapter III section 1.3). Each image is registered and added to the former ones to obtain a single accumulated image. See [15] for an analysis of the feasibility of this technique. We shall assume here that the registration operation is error-free, so that the operation amounts to an increased exposure time with a stationary landscape.

Theorem 7.1. *The multi-image acquisition strategy using L images increases the SNR by a factor of $\sqrt{L}$ compared to a standard snapshot.*

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

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

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

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

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

$$\sum_{k=0}^{L-1} \mathcal{P}\left(\frac{1}{v}(\mathbb{1}_{[0,v\Delta t]} * u)(x)\right) \sim \mathcal{P}\left(L\frac{1}{v}(\mathbb{1}_{[0,v\Delta t]} * u)(x)\right). \tag{II.11}$$

(This last equivalence uses the independence of the Poisson random variables, justified by the disjoint time intervals). Following the same scheme used in (Thm. 1.1) from (II.11) we get the estimated

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

whose variance is L times smaller. Thus the L factor remains in all equations. Finally from (V.1, V.6) we deduce that the SNR is increased by a $\sqrt{L}$ factor. □

In other words the use of a multi-image fusion with a sufficient L permits to achieve *any* SNR. This result will contrast with the first *flutter shutter* paradox that an infinite time exposure does not grant an infinite SNR. This is the object of the next chapters.

Chapter III

The *Numerical Flutter Shutter*

Abstract : *This chapter defines a digital implementation of the flutter shutter compatible with recent technologies. This set up is the most flexible, adaptive to all motion and allows for negative gains. Furthermore, the numerical flutter shutter does not reduce the number of photons caught by the sensor and it is proven later on that it yields the best possible SNR. It is investigated using the model of photon acquisition of chapter II. A closed formula is obtained giving explicitly the SNR of the sharp deconvolved image. The numerical flutter shutter gain function is in principle piecewise constant. Nevertheless, it is useful for the theory to extend it to continuous gain functions. Anyway, a reverse formula permits to get back an equivalent piecewise constant numerical flutter shutter from any continuous gain function.*

After having treated the classic image acquisition strategies, we are now in a position to treat the various *flutter shutter* strategies and to compare them to the classic ones. Two things are at stake: first, to prove that the various *flutter shutters* actually work, and second to evaluate the SNR of the resulting image and to compare it to the SNR of classic strategies. The hope would be that the *flutter shutter* retains the very interesting feature of the multi image denoising, namely an increase of the SNR by a factor proportional to $\sqrt{L\Delta t}$, the total exposure time. We shall see that this is not so.

1 The *numerical flutter shutter*

The *numerical flutter shutter* method consists in a numerical sensor gain modification taking place *after* the acquisition by the sensor. Roughly speaking the camera takes a burst of L images using an exposure time Δt. The k-th image is multiplied, for $k \in 0, ..., L-1$, by an $\alpha_k \in \mathbb{R}$ gain. Then all images are added to obtain *one* observed image, the *flutter shutter*. The exposure time Δt must be small enough so the blur of each image is negligible (definition in chapter V section 1). This technique is similar to the multi-image acquisition strategy but does not use any registration technique. According to, for example, [64, 89] an image sensor can have a duty ratio of nearly 100% (the duty ratio is the ratio of light integration time over readout, storage, reset times - that is the percentage of useful time). It means that a sensor can integrate light without interruption. Thus, the *numerical flutter shutter*,

as it is described below without "dead time" between two consecutive gains a_k is perfectly reasonable from a technological viewpoint. Nevertheless, it seems that its interest is limited: why not keeping all images instead of adding them all? One of the obvious reasons is compression, particularly for Earth observation satellites. In that case the motion blur due to a drift in satellite trajectory estimate could be eliminated by a *flutter shutter*, without any additional transmission (or computational) burden if only the *flutter shutter* image (the sum) is transmitted. The k-th acquired elementary image at a pixel at position n is a realization of

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

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

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

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

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

$$obs(n) \sim \sum_{k=0}^{L-1} \alpha_k \mathcal{P}\left(\int_{k\Delta t}^{(k+1)\Delta t} u(n-vt)dt\right). \tag{III.2}$$

We call *numerical flutter shutter* its band limited interpolate

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

We call *flutter shutter function* the function

$$\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t). \tag{III.3}$$

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

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

3. $\alpha_0 = 1$ and $\alpha_k = 0\ \forall k \in \{1, ..., L-1\}$ (standard snapshot)

1.1 The motionless case $v = 0$

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

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

Thus

$$\mathbb{E}\left(obs(x)\right) = \sum_{k=0}^{L-1} \alpha_k \Delta t u(x) \tag{III.4}$$

and

$$var\left(obs(x)\right) = \sum_{k=0}^{L-1} \alpha_k^2 \Delta t u(x). \tag{III.5}$$

From (III.4) we deduce that the estimated landscape $\mathbb{u}_{est}$ is obtained using

$$\mathbb{u}_{est}(x) \sim \frac{obs(x)}{\sum_{k=0}^{L-1} \alpha_k \Delta t}. \tag{III.6}$$

Let $r \sim \mathbb{u}_{est} - u$ be the residual error then from (III.4-III.5) we get

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

Hence $\mathcal{F}(\mathbb{u}_{est})$ is an unbiased estimator for $\hat{u}$. Moreover, from (III.5, III.6) we deduce

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

Thus, by (II.7), $SNR(\mathbb{u}_{est}) = \sqrt{u(x)\Delta t \frac{(\sum_{k=0}^{L-1} \alpha_k)^2}{\sum_{k=0}^{L-1} \alpha_k^2}}$. Since by Jensen's inequality applied to the convex $x \mapsto x^2$ function we always have $\frac{1}{L}(\sum_{k=0}^{L-1} \alpha_k)^2 \leq \sum_{k=0}^{L-1} \alpha_k^2$, the (numerical) *flutter shutter* is less favorable than a mere accumulation. To give an example, in the case of a Agrawal *et al.* code where $\alpha_k = 0$ or 1 and the number of non null α_k is $L/2$. Then the SNR is $\sqrt{u(x)\Delta t \frac{(L/2)^2}{L/2}} = \sqrt{\frac{u(x)\Delta t L}{2}}$, to be compared to the pure accumulation case where $\alpha_k = 1$ and the SNR is $\sqrt{u(x)\Delta t \frac{L^2}{L}} = \sqrt{u(x)\Delta t L}$.

Thus the Agrawal *et al.* *flutter shutter* entails in the motionless case a $\sqrt{2}$ loss factor for the SNR. We now pass to the general more interesting case, namely when there is a motion.

1.2 *Flutter shutter* in presence of motion

Theorem 1.1. *The observed samples of the numerical flutter shutter are such that, for $n \in \mathbb{Z}$*

$$\mathbb{E}\left(obs(n)\right) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n) \tag{III.7}$$

and

$$var(obs(n)) = \left(\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right) * u\right)(n). \tag{III.8}$$

Proof. From the *numerical flutter shutter* samples definition (III.2),

$$\mathbb{E}\left(obs(x)\right) = \sum_{k=0}^{L-1} \alpha_k \int_{k\Delta t}^{(k+1)\Delta t} u(x - vt)dt = \int_0^{L\Delta t} \alpha(s)u(x - vs)ds \tag{III.9}$$

where we recall that

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

Thus,

$$\mathbb{E}\left(obs(x)\right) = \int_0^{Lv\Delta t} \frac{1}{v}\alpha(\frac{y}{v})u(x - y)dy = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(x). \tag{III.10}$$

Similarly from (III.1)

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

□

Notice that $obs(x)$ is not necessarily a Poisson random variable. However, if all α_k are equal to 0 or 1 then by (III.9)

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

because we are adding independent Poisson random variables. On the other hand, if for some k, $\alpha_k \notin \{0, 1\}$, then if $X \sim \mathcal{P}(\lambda)$ then $\alpha_k X$ is *not* a Poisson random variable.

Ideal *numerical flutter shutter*

The formulas of Theorem 1.1 giving the samples obtained by *flutter shutter* appear not to depend on the particular form of α as a piecewise constant function on a finite set of intervals with length Δt. As a matter of fact, we can envisage *any* function $\alpha \in L^2(\mathbb{R})$ to be *numerical flutter shutter*, for an ideally controlled camera where at each instant the gain $\alpha(t)$ is changed. This leads to the following definition and corollary.

Corollary 1.2. *We call* continuous numerical flutter shutter *any band-limited and bounded gain function $\alpha \in L^2(\mathbb{R})$. Then the formulas* (III.7) *and* (III.8) *of Theorem 1.1 are still valid.*

Proof. By assumption the observed ideal landscape u belongs to $L^1 \cap L^2(\mathbb{R})$. We recall that $L^1 * L^2 \subset L^2$ and $L^2 \cap L^2 \subset C_0(\mathbb{R})$ (the set of continuous functions on $\mathbb{R}$ tending to 0 at infinity). Furthermore, α being band limited is continuous. Being also bounded, the expectation and variance functions of (III.7) and (III.8) are continuous and therefore well defined at any point. It remains to show that these formulas are valid for a general gain function. Consider for this an approximation of $\alpha(t)$ by a finite *numerical flutter shutter* code $(\alpha_k)_k$ such that $(\alpha_k)_k$ tends to α in L^1, L^2 and L^∞, (that is, uniformly) when $k \to \infty$. The formulas (III.7) and (III.8) are valid for $(\alpha_k)_k$, and the corresponding formulas for α are deduced by passing to the limit. □

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

1.3 The inverse filter of a *numerical flutter shutter*

Step 1: the noiseless case Let us examine first the discrete noiseless case, when $obs(n) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$ and $obs(n)$ is obtained for $n \in \mathbb{Z}$ but being band limited, can be interpolated to $obs(x)$, for any $x \in \mathbb{R}$. Then

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

By hypothesis (see chapter II section 2) we assumed that $\hat{u}(\xi) = 0$ for $|\xi| > \pi$. Hence for the invertibility we must only require that $|\hat{\alpha}(\xi v)| > 0$ for $\xi \in [-\pi, \pi]$.

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

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

$$\hat{\gamma}(\xi) = \frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}. \tag{III.11}$$

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

We shall as logical define the recovered landscape from noisy data by the formulae that would be valid for noiseless data. Assume that we observe $e(n) = \mathbb{E}(obs(n))$ for $n \in \mathbb{Z}$ and wish to compute $\hat{e}(\xi)$ from the discrete observed $(e(n))_{n\in\mathbb{Z}}$. Since $e(x)$ is band limited, we can interpolate it using the band limited interpolation (II.4). The band limited interpolate of the ideal observation is

$$e(x) = \sum_{n\in\mathbb{Z}} e(n)\mathrm{sinc}(x-n). \tag{III.12}$$

Then from (III.12) we have

$$\hat{e}(\xi) = \sum_{n\in\mathbb{Z}} e(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{III.13}$$

So the ideal deconvolved landscape $d(x)$ obtained by combining (III.11) and (III.13) is

$$\hat{d}(\xi) = \frac{\sum_{n\in\mathbb{Z}} e(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}. \tag{III.14}$$

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

Flutter shutter landscape recovery in the real noisy case

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

$$\mathcal{F}(\mathbb{u}_{est,num})(\xi) = \frac{\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)}{\hat{\alpha}(\xi v)}, \tag{III.15}$$

where the observed $obs(n)$ samples (III.1) are used instead of the ideal $e(n)$ in (III.14).

Theorem 1.3. *The numerical flutter shutter has a spectral SNR (II.8) equal to*

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

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

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

and its variance is

$$var(\mathcal{F}(\mathbb{u}_{est,num}(\xi))) = \frac{||\alpha||_{L^2}^2 ||u||_{L^1} \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}. \tag{III.16}$$

Proof. By (III.8, III.15),

$$\begin{aligned}
&var(\mathcal{F}(\mathbb{u}_{est,num})(\xi)) \\
&= \frac{var\left(\sum_{n\in\mathbb{Z}} obs(n) e^{-in\xi} \mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{|\hat{\alpha}(\xi v)|^2} \\
&= \frac{\sum_{n\in\mathbb{Z}} var(obs(n)) |e^{-in\xi} \mathbb{1}_{[-\pi,\pi]}(\xi)|^2}{|\hat{\alpha}(\xi v)|^2} \\
&= \frac{\sum_{n\in\mathbb{Z}} \left(\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right) * u\right)(n) \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} \\
&= \frac{||\frac{1}{v}\alpha^2\left(\frac{\cdot}{v}\right) * u||_{L^1} \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} \\
&= \frac{\frac{1}{v}||\alpha^2\left(\frac{\cdot}{v}\right)||_{L^1} ||u||_{L^1} \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} = \frac{\frac{1}{v}||\alpha\left(\frac{\cdot}{v}\right)||_{L^2}^2 ||u||_{L^1} \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} \\
&= \frac{\frac{v}{v}||\alpha||_{L^2}^2 ||u||_{L^1} \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2} = \frac{||\alpha||_{L^2}^2 ||u||_{L^1} \mathbb{1}_{[-\pi,\pi](\xi)}}{|\hat{\alpha}(\xi v)|^2}.
\end{aligned} \tag{III.17}$$

In this proof the crucial point is the use of the Poisson summation formula **(xxx)** in equation (III.17). Following the same scheme and starting from (Thm. III.7) $\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,num}(\xi))(\xi)$ can be computed by using (for the derivation of the third line) the second Poisson formula **(xxx)**, and the fact that u is band limited with $\hat{u}$ supported on $[-\pi,\pi]$:

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

From (III.16) and using the definition of the spectral SNR (II.8), we obtain

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

□

Remark From (III.16) we also deduce that $var(\mathcal{F}(\mathbb{u}_{est,num})(\xi))$ is invariant by changing α into $\lambda\alpha$ for $\lambda \neq 0$ (rescaling): as could be expected, the *flutter shutter* code is defined up to a multiplicative constant.

Remark Going back to the case where α is a discrete *numerical flutter shutter*, we can see a necessary condition on Δt for its invertibility. Indeed, from (III.3) follows that

$$\hat{\alpha}(\xi) = \sum_{k=0}^{L-1} \alpha_k \mathcal{F}(\mathbb{1}_{[k\Delta t,(k+1)\Delta t[})(\xi) = \Delta t \text{sinc}(\frac{\xi\Delta t}{2\pi})e^{\frac{-i\xi\Delta t}{2}} \sum_{k=0}^{L-1} \alpha_k e^{-ik\xi\Delta t}. \tag{III.18}$$

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

2 *Flutter shutter* design: from continuous to discrete

Even if the above theory deals with continuous and discrete codes as well, in practice any continuous *flutter shutter* code found by some abstract optimization must eventually be realized as a feasible device. Thus it must be replaced by a piecewise constant one on intervals of length Δt. Assume that we have designed a continuous *flutter shutter* function $\beta \in L^2(\mathbb{R})$, invertible for all velocities below $|v|$, which means $\hat{\beta}(\xi v) \neq 0$ for $\xi \in [-\pi,\pi]$. The values of $\hat{\beta}(v\xi)$ outside $[-\pi,\pi]$ do not matter for our scopes, the filter and inverse filter being always applied to band-limited functions. Thus, we can always assume that $\hat{\beta}(v\xi)$ is zero outside $[-\pi,\pi]$. Our goal is to deduce from β a numerical *flutter shutter* code α which coincides with β at velocity v on the spectrum of u. In other terms, we want $\hat{\alpha}(v\xi) = \hat{\beta}(v\xi)$ for $\xi \in [-\pi,\pi]$. Under that condition, the observed signal $obs(n) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$ by α or β will be identical. Furthermore, from Thm. III.7 we will have

$$\mathbb{E}\left(\hat{o}bs(\xi)\right) = \hat{\alpha}(v\xi)\hat{u}(\xi) = \hat{\alpha}(v\xi)\mathbb{1}_{[-\pi,\pi]}(\xi)\hat{u}(\xi) = \hat{\beta}(v\xi)\hat{u}(\xi),$$

meaning that the expectation of spectrum of the observed signal is unchanged (but not necessarily its variance).

The question is to find an equivalent code function $\alpha(t) = \sum_{k\in\mathbb{Z}} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$, as defined by (III.3), but not necessarily compactly supported. Our requirement is that $\hat{\alpha}(\xi v) = \hat{\beta}(\xi v)$ on $[-\pi,\pi]$.

By (III.18), a *numerical flutter shutter* has the general form (where we allow for an infinite code $(\alpha_k)_{k\in\mathbb{Z}}$), $\hat{\alpha}(v\xi) = \Delta t \mathrm{sinc}(\frac{v\Delta t\xi}{2\pi}) e^{\frac{-iv\Delta t\xi}{2}} \sum_k \alpha_k e^{-ikv\Delta t\xi}$. We want $\hat{\alpha}(v\xi) = \hat{\beta}(v\xi)$ for $\xi \in [-\pi, \pi]$, which is equivalent to having for $\xi \in [\pi, \pi]$,

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

and therefore

$$\frac{\hat{\beta}(v\xi) e^{i\frac{v\Delta t\xi}{2}}}{\Delta t \mathrm{sinc}(\frac{v\Delta t\xi}{2\pi})} \mathbb{1}_{[-\pi,\pi]} = \sum_{k\in\mathbb{Z}} \alpha_k e^{-ikv\Delta t\xi} \mathbb{1}_{[\pi,\pi]}. \tag{III.19}$$

The left member of this equation belongs to $L^2([-\pi, \pi])$ provided the sinc in the denominator does not vanish, which is true if $\Delta t|v| < 2$. The above formula appears to be the Fourier series decomposition of the left hand member on the Fourier basis on the interval $[\frac{-T}{2}, \frac{T}{2}]$ satisfying $\frac{2\pi}{T} = \Delta t|v|$, which gives $T = \frac{2\pi}{\Delta t|v|}$. Moreover we assume that $|v|\Delta t \leq 1$. Indeed this supplementary condition is mandatory for the temporal sampling of the left hand member of (III.19) and get $\frac{T}{2} > \pi$. Thus, if $|v|\Delta t \leq 1$ $[\frac{-T}{2}, \frac{T}{2}]$ contains $[-\pi, \pi]$, implying that (III.19) is correct, and that $(\alpha_k)_{k\in\mathbb{Z}} \in l^2(\mathbb{Z})$, are the Fourier series coefficients (provided $|v|\Delta t \leq 1$)

$$\alpha_k = \frac{\Delta t|v|}{2\pi} \int_{-\frac{\pi}{\Delta t v}}^{\frac{\pi}{\Delta t v}} \frac{\hat{\beta}(v\xi) e^{i\frac{v\Delta t\xi}{2}}}{\mathrm{sinc}(\frac{v\Delta t\xi}{2\pi})} \mathbb{1}_{[-\pi,\pi]} e^{ikv\Delta t\xi} d\xi. \tag{III.20}$$

Thus,

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

This proves the following theorem.

Theorem 2.1. *Let $\beta \in L^2(\mathbb{R})$ be a band-limited time convolution kernel satisfying $\hat{\beta}(v\xi) \neq 0$ for $\xi \in [-\pi, \pi]$, in other terms invertible on all band-limited functions and for all velocities below $|v|$. If $|v|\Delta t \leq 1$, there exists an invertible flutter shutter code function*

$$\alpha(t) = \sum_{k\in\mathbb{Z}} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t) \tag{III.21}$$

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

flutter shutter are explicitly given by

$$\alpha_k = \frac{1}{2\pi} \int_{-\pi|v|\Delta t}^{\pi|v|\Delta t} \frac{\hat{\beta}(\frac{\xi}{\Delta t}) e^{i\frac{\xi}{2}}}{sinc(\frac{\xi}{2\pi})} e^{ik\xi} d\xi. \tag{III.22}$$

The question arises of whether the discrete *numerical flutter shutter* α function yields a $PSNR$ (peak signal to noise ratio) as good as the original β. According to the formula giving the SNR in Theorem 1.3, we simply have to compare $||\alpha||_{L^2}$ and $||\beta||_{L^2}$. More precisely, the ratio $\frac{||\beta||_{L^2}}{||\alpha||_{L^2}}$ gives the multiplication factor of the SNR obtained with β to get the SNR of the restored image using the discrete filter α. But by assumption, we have $\hat{\beta}$ supported on $[-\pi|v|, \pi|v|]$ $||\beta||^2_{L^2(\mathbb{R})} = \frac{1}{2\pi}||\hat{\beta}||^2_{L^2(\mathbb{R})} = \int_{-\pi v}^{\pi v} |\hat{\beta}(\xi)|^2 d\xi$. On the other hand by construction, $\hat{\alpha} = \hat{\beta}$ on $[-\pi v, \pi v]$. It follows that $||\alpha^2||_{L^2(\mathbb{R})} \geq ||\beta^2||_{L^2(\mathbb{R})}$.

Thus, we have also proved:

Corollary 2.2. *Let β be a continuous numerical flutter shutter. Then its discrete equivalent numerical flutter shutter has a smaller or equal spectral SNR.*

Chapter IV

The *Analog Flutter Shutter*

This chapter investigates the classic analog implementation of the flutter shutter using the model of photon acquisition of chapter II. This analog flutter shutter is a generalization of the original Agrawal et al. flutter shutter which allows for smoother, non binary, gain functions. For any analog flutter shutter apparatus, an explicit formula to measure directly the SNR of the deconvolved sharp image is given. The chapter ends with a comparison of the SNR of analog and numerical flutter shutter.

1 The *analog flutter shutter*

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

Definition (*Analog flutter shutter* function.)
Let $\alpha(t) \in [0, 1]$ be the gain used at time t. We call *analog flutter shutter function* any *positive* function

$\alpha \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$.

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

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

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

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

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

We call *analog flutter shutter* its band limited interpolate

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

Theorem 1.1. *The observed samples of the analog flutter shutter are such that, for $n \in \mathbb{Z}$*

$$\mathbb{E}(obs(x)) = \frac{1}{v}(\alpha\left(\frac{\cdot}{v}\right) * u)(x) \tag{IV.2}$$

and

$$var(obs(x)) = \frac{1}{v}(\alpha\left(\frac{\cdot}{v}\right) * u)(x). \tag{IV.3}$$

Proof. Directly from the *analog flutter shutter* samples definition (IV.1). □

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

Theorem 1.2. *The analog flutter shutter method has a spectral SNR equal to*

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

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

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

and the variance is

$$var(\mathcal{F}(\mathbb{u}_{est,ana}(\xi))) = \frac{||\alpha||_{L^1}||u||_{L^1}}{|\hat{\alpha}|^2(\xi v)}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{IV.4}$$

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

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

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

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

Therefore,

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

□

A brief summary pointing out the differences between the *analog* and *flutter shutter* is given in Tab.A.1. The *analog flutter shutter* controls the percentage of photons allowed to travel to the sensor, therefore only positive functions are implementable. It decreases the number of photons sensed thus tends to decrease the resulting SNR. On the other hand the *numerical flutter shutter* requires piecewise constant *flutter shutter* functions. Consequently, if a *flutter shutter* function is positive and piecewise constant (implementable on both cameras) the *numerical flutter shutter* should always be chosen as it leads to a better SNR of the reconstructed image. The question of choice of the *flutter shutter* type in the general case is answered in section 2 below.

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

Table A.1 – This table summarizes the results on *numerical* and *analog flutter shutters*. The first column describes the structure of the *numerical flutter shutter*, the second describes the *analog flutter shutter*. The first line indicates which kind of *flutter shutter functions* $\alpha(t)$ are implementable with respect to the *flutter shutter* type. The second (resp. the third) gives the expected value (resp. variance) of the (observed) *flutter shutter*. The fourth shows the inverse filter to be applied to the *flutter shutter* in order to deconvolve. The fifth (resp. the sixth) gives the expected value (resp. variance) of the deconvolved. Given any *flutter shutter function* $\alpha(t)$ the last one gives the spectral SNR of both methods. Provided that a *flutter shutter function* $\alpha(t)$ is implementable on both kinds of *flutter shutter* the spectral SNR of the *analog flutter shutter* is lower than the spectral SNR of the *numerical flutter shutter* (see Thm. 2.1).

2 Comparison of a piecewise constant *analog flutter shutter* with the *numerical flutter shutter*

The question arises of whether it is better to apply an *analog flutter shutter*, or the equivalent *numerical flutter shutter* with exactly the same code $0 \leq \alpha \leq 1$. (From the technological viewpoint, an *analog flutter shutter* could be easily implemented with a classic CCD, and a numerical one with a CMOS). A first observation in favor of the *numerical flutter shutter* is given in the next lemma.

Lemma 2.1. *The variance of the analog flutter shutter observation (IV.3) is larger or equal to the variance of the numerical flutter shutter observation, (III.8), with equality when* $\forall\ k\ \alpha_k = 0$ *or* 1.

Proof. Since $0 \leq \alpha(t) \leq 1$ (because it is a proportion of incoming photons allowed to travel through

the sensor) we have $\alpha(\frac{t}{v}) \geq \alpha^2(\frac{t}{v})$. Hence $(\alpha\left(\frac{\cdot}{v}\right) * u)(x) \geq (\alpha^2\left(\frac{\cdot}{v}\right) * u)(x)$ (because $u \geq 0$). Using (IV.3) and (III.8) concludes the proof. □

Since the expected value of (IV.2) is equal to the expected value of the *numerical flutter shutter* (see Thm. III.7) the inverse filter is equal to the inverse filter of the *numerical flutter shutter* (III.11). The next result is a decider for the *numerical flutter shutter* (when it is possible to implement it with the same code as an *analog flutter shutter*, meaning that α is piecewise constant.)

Theorem 2.2. *Let $0 \leq \alpha \leq 1$ be a piecewise constant code function for the analog flutter shutter. Then the spectral SNR of the analog flutter shutter method is smaller or equal to the spectral SNR of the numerical flutter shutter with the same code.*

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

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

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

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

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

This result also implies that the variance of the estimated landscape (IV.4) using an *analog flutter shutter* method $var(\mathcal{F}(\mathbb{u}_{est,ana}))$ is larger or equal to $var(\mathcal{F}(\mathbb{u}_{est,num}))$ (III.16) using a *numerical flutter shutter* method when α is positive and piecewise constant.

Chapter V

Examples of *Flutter Shutter*

This chapter explores the standard camera apparatus as a particular flutter shutter strategy. The SNR of the deconvolved image is calculated, for any standard acquisition strategy. The standard camera is optimized to get the best SNR possible, taking the deconvolution into account. It is proven that given the relative velocity v, the best snapshot consists in the use of an exposure time Δt such that the blur support is $|v|\Delta t \approx 1.0909$ (pixel). This can be termed as the "snapshot" theory. This snapshot theory provides a reference to compare all flutter shutters and the standard camera in terms of SNR. It is proven that the Levin et al. motion-invariant photography method is a particular case of analog flutter shutter. Furthermore, it is proven that the motion-invariant photography apparatus is indeed invertible and a formula providing its SNR is given. This chapter also proposes to implement the motion-invariant photography kernel using a numerical flutter shutter. This permits to generalize the motion-invariant photography method to the case where the direction of the relative velocity v is not a priori known. The chapter ends with a summary of all usual flutter shutter strategies and their impact on the SNR of the recovered sharp image.

1 Snapshots

The goal of this section is to provide a thorough definition and analysis of the classic flutterless photography. The above results shall be used in the sequel to compare the *flutter shutter* with classic cameras. Uniform motion blurs using a standard camera have been studied nicely, for example in [12].

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

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

where (V.1) is known only for $x \in \mathbb{Z}$. In short, a snapshot is nothing but a *flutter shutter* (analog or

numerical) with code $\alpha(t) = \mathbb{1}_{[0,\Delta t](t)}$. Thus

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

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

Definition Given v, we call "standard snapshot" the use of an integration time (Δt) such that $|v|\Delta t < 2$. We call *snapshot samples* at position n of the landscape u at velocity v the random variables

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

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

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

By definition of the standard snapshot, Δt is small enough so (V.2) has no zero on $[-\pi, \pi]$ (the support of $\hat{u}$). Thus (V.1, V.2) lead to the definition of the inverse filter γ satisfying

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

implying

$$\hat{\gamma}(\xi) = \frac{v\mathbb{1}_{[-\pi,\pi]}(\xi)}{2\frac{sin(\frac{\xi v\Delta t}{2})}{\xi}e^{-i\xi\frac{v\Delta t}{2}}}. \tag{V.3}$$

The following definition and estimation of variance and SNR are direct applications of the same quantities for the numerical (or analog) *flutter shutter*:

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

$$\mathcal{F}(\mathbb{u}_{est,sna})(\xi) = \hat{\gamma}(\xi)\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi). \tag{V.4}$$

Theorem 1.1. *For a standard snapshot using an exposure time of Δt the spectral SNR (II.8) is*

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

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

$$\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,sna})(\xi))) = \hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi); \tag{V.5}$$

and the variance is

$$var(\mathcal{F}(\mathbb{u}_{est,sna}(\xi))) = \frac{||u||_{L^1}\mathbb{1}_{[-\pi,\pi]}(\xi)}{\Delta t\left|2\frac{sin(\frac{\xi v\Delta t}{2})}{\xi v\Delta t}\right|^2}. \tag{V.6}$$

Proof. Since $\alpha(t) = \mathbb{1}_{[0,\Delta t]}$, $\|\alpha\|_{L^2}^2 = \Delta t$ and $\hat{\alpha}(\xi) = \frac{2\sin(\frac{\Delta t\xi}{2})}{\xi}e^{-i\frac{\Delta t\xi}{2}}$, these formulas are direct applications of Theorem 1.3 (chapter III). □

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

Theorem 1.2. (Best exposure time for landscape recovery)
Consider a landscape $u(x - vt)$ *moving at velocity* v*. Then for a snapshot the* $SNR^{spectral-averaged}$ *(II.9) is maximized when* $|v|\Delta t^* \approx 1.0909$ *pixel and is equal to*

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

Proof. From (V.6) the energy (variance of the noise) to be minimized in order to guarantee the best $SNR^{spectral-averaged}$ *after* deconvolution is

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

Then

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

This derivative vanishes when $b^* = v\Delta t^* \approx 1.0909$ pixel (see Fig. A.1). Then (II.9, V.5, V.6) entail

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

$$\approx \frac{\sqrt{\frac{1.0909}{2\pi v}}\int_{-\pi}^{\pi}|\hat{u}(\xi)|d\xi}{\sqrt{||u||_{L^1}\int_{-\pi}^{\pi}\frac{d\xi}{\left|\frac{sin(\frac{1.0909\xi}{2})}{\frac{1.0909\xi}{2}}\right|^2}}} \approx \frac{0.1359}{\sqrt{v}}\frac{\int_{-\pi}^{\pi}|\hat{u}(\xi)|d\xi}{\sqrt{||u||_{L^1}}}.$$

□

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

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

The practical validity of this definition is illustrated on Fig. A.2 where, the $RMSE$ is calculated of example tests varying the exposure time. Notice that since the estimator is unbiased, minimizing the variance is equivalent to minimizing the $RMSE$ of the deconvolved image, by the same arguments developed in Chap. II section 4. As predicted by the theory on average on the example tests the best blur support is of approximately 1.0909 pixel. From Thm. 1.1 we get that the actual value of the $RMSE$ is a function of u, v and Δt, explaining the different curves observed.

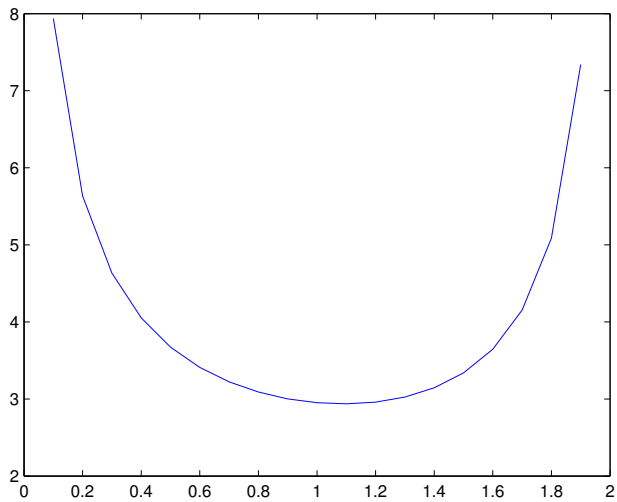

Figure A.1 – This figure shows the square root of the energy $E(\Delta t)$ of Thm. 1.2. The energy $E(\Delta t)$ measures the variance of the Poisson noise *after* deconvolution, for *any* snapshot, in function of the exposure time Δt. Since our estimator is unbiased, minimizing the variance is equivalent to minimizing the $RMSE$ of the deconvolved image, by the same arguments developed in Chap. II section 4. Therefore, we ought to minimize $E(\Delta t)$ in order to guarantee the best $SNR^{spectral-averaged}$, and the smallest $RMSE$ taking the deconvolution into account: x-axis blur ($|v|\Delta t$) in pixel, y-axis the standard deviation of the noise taking the deconvolution into consideration. The minimum is reached for a blur of approximately 1.0909 pixel. Without loss of generality, by the arguments developed in the proof of Thm. 1.2, the curve has been drawn for $v = 1$.

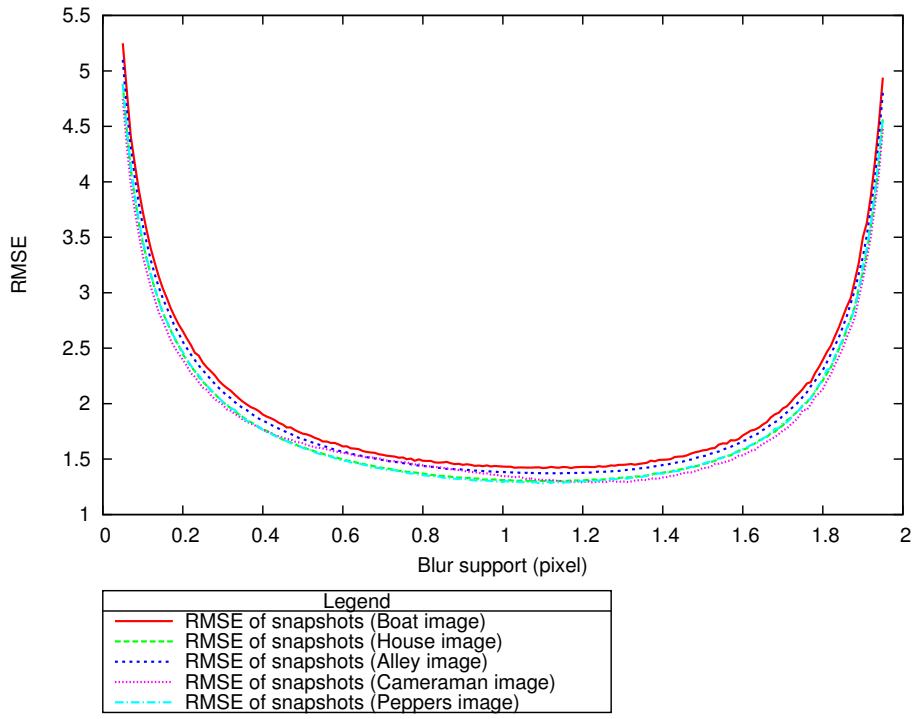

Figure A.2 – This figure shows the $RMSE$ curves for different snapshots kinds, on five test images (House, Alley, Boat, Cameraman, Peppers). On the $x-axis$, the blur support ($|v|\Delta t$) in pixels, on the $y-axis$ the corresponding $RMSE$. Notice that, some of these curves are so close to each other that they superimpose. Without loss of generality, by the arguments developed in the proof of Thm. 1.2, the comparison is made with a fixed $v = 1$. From the proof of Thm. 1.2 we get that the best snapshot only depends on the blur support $|v|\Delta t$. However, from Thm. 1.1 for a fixed landscape u we get that the value of this SNR is a function of the two variables Δt and $v\Delta t$. This is no surprise looking at (V.1). Furthermore, since our estimator is unbiased, minimizing the variance is equivalent to minimizing the $RMSE$ of the deconvolved image, by the same arguments developed in Chap. II section 4. Therefore, the curves confirm that, on average on multiple images, the blur support for a standard camera should be of approximatively $\Delta t^* = 1.0909$ pixel. Moreover, from Thm. 1.2 we get that for a fixed v and Δt the value of the $RMSE$ depends on the landscape u, explaining the differences between curves. Theses curves also show that our SNR definition is indeed proportional to the $RMSE$ of the deconvolved image. A larger support would lead to a better SNR on the observed image samples, but the deconvolution would entail a lower SNR (and a bigger $RMSE$) on the deconvolved image. The best snapshot is a compromise between the number of photons caught during a time span Δt and the deconvolution kernel. It gives a reference to compare all *flutter shutter* strategies in terms of SNR.

2 *Motion-invariant photography*

We shall prove that the *motion-invariant photography* method proposed in [69, 70] is equivalent to an *analog flutter shutter* method using a specific *flutter shutter function*. Thus, we are able to compute its SNR and compare it with the other *flutter shutter* methods. This fact comes as a surprise, the gain in a *flutter shutter* being controlled, while with the *motion-invariant photography* method the shutter remains fully open during the whole aperture time. Thus, its gain remains constant and equal to one on the normal time scale. Nevertheless, a time renormalization gives a variable gain. The *motion-invariant photography* apparatus consists in *moving the camera at a constant acceleration in the direction of v while the landscape is moving at a constant velocity v*. Thus, to use the *motion-invariant photography* method the direction of v must be known *a priori*. Furthermore, this means that the apparent relative velocity (between the camera and the landscape) is $v(t) = -at - v$. The *motion-invariant photography* was discovered by searching among all camera motions one providing the same kernel for all velocities v of the landscape. With the formalism proposed in the former chapters the observed value of the *motion-invariant photography* using a finite aperture time T on the centered time interval $[\frac{-T}{2}, \frac{T}{2}]$ is a realization of

$$
\begin{aligned}
obs(x) &\sim \mathcal{P}\left(\int_{\frac{-T}{2}}^{\frac{T}{2}} u(x - \frac{a}{2}t^2 - v.t)dt\right) \qquad \text{(V.7)}\\
&\sim \mathcal{P}\left(\int_{\frac{-T}{2}}^{\frac{T}{2}} u\left(x - \frac{a}{2}(t+\frac{v}{a})^2 + \frac{v^2}{2a}\right)dt\right) \sim \mathcal{P}\left(\int_{\frac{-T}{2}+\frac{v}{a}}^{\frac{T}{2}+\frac{v}{a}} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt\right)\\
&\sim \mathcal{P}\left(\int_{\frac{-T}{2}+\frac{v}{a}}^{0} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt + \int_{0}^{\frac{T}{2}+\frac{v}{a}} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt\right)\\
&\sim \mathcal{P}\left(-\int_{\frac{T}{2}-\frac{v}{a}}^{0} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt + \int_{0}^{\frac{T}{2}+\frac{v}{a}} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt\right)\\
&\sim \mathcal{P}\left(\int_{0}^{\frac{T}{2}-\frac{v}{a}} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt + \int_{0}^{\frac{T}{2}+\frac{v}{a}} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt\right)\\
&\text{which does not depends of the sign of } \tfrac{v}{a} \text{ (as the two integrals switch)}\\
&\sim \mathcal{P}\left(\int_{0}^{\frac{T}{2}-|\frac{v}{a}|} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt + \int_{0}^{\frac{T}{2}+|\frac{v}{a}|} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt\right)\\
&\sim \mathcal{P}\left(\int_{\frac{T}{2}-|\frac{v}{a}|}^{\frac{T}{2}+|\frac{v}{a}|} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt + 2\int_{0}^{\frac{T}{2}-|\frac{v}{a}|} u\left(x - \frac{a}{2}t^2 + \frac{v^2}{2a}\right)dt\right)\\
&\text{assuming a long enough exposure time such that } \tfrac{T}{2} - |\tfrac{v}{a}| \geq 0\\
&\sim \mathcal{P}\left(\int_{(\frac{T}{2}-|\frac{v}{a}|)^2}^{(\frac{T}{2}+|\frac{v}{a}|)^2} \frac{1}{2\sqrt{t}} u\left(x - at + \frac{v^2}{2a}\right)dt + 2\int_{0}^{(\frac{T}{2}-|\frac{v}{a}|)^2} \frac{1}{2\sqrt{t}} u\left(x - at + \frac{v^2}{2a}\right)dt\right)\\
&\sim \mathcal{P}\left(\int_{a(\frac{T}{2}-|\frac{v}{a}|)^2}^{a(\frac{T}{2}+|\frac{v}{a}|)^2} \frac{1}{a}\frac{1}{2\sqrt{\frac{t}{a}}} u\left(x - \frac{t}{a}a + \frac{v^2}{2a}\right)dt + 2\int_{0}^{a(\frac{T}{2}-|\frac{v}{a}|)^2} \frac{1}{a}\frac{1}{2\sqrt{\frac{t}{a}}} u\left(x - \frac{t}{a}a + \frac{v^2}{2a}\right)dt\right)
\end{aligned}
$$

$$\sim \mathcal{P}\left(\int_{a(\frac{T}{2}-|\frac{v}{a}|)^2}^{a(\frac{T}{2}+|\frac{v}{a}|)^2} \frac{1}{2\sqrt{at}} u\left(x-t+\frac{v^2}{2a}\right) dt + 2\int_0^{a(\frac{T}{2}-|\frac{v}{a}|)^2} \frac{1}{2\sqrt{at}} u\left(x-t+\frac{v^2}{2a}\right) dt\right)$$

$$\sim \mathcal{P}\left(\int_{a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a}}^{a(\frac{T}{2}+|\frac{v}{a}|)^2-\frac{v^2}{2a}} \frac{1}{2\sqrt{a(t+\frac{v^2}{2a})}} u\left(x-t\right) dt + 2\int_{-\frac{v^2}{2a}}^{a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a}} \frac{1}{2\sqrt{a(t+\frac{v^2}{2a})}} u\left(x-t\right) dt\right)$$

$$\sim \mathcal{P}\left(\int_{-\infty}^{\infty} \underbrace{\left(\underbrace{\frac{\mathbb{1}_{]a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a},a(\frac{T}{2}+|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{2\sqrt{a(t+\frac{v^2}{2a})}}}_{A} + \underbrace{\frac{\mathbb{1}_{]-\frac{v^2}{2a},a(\frac{T}{2}-|\frac{v}{a}|)^2-\frac{v^2}{2a}[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}}_{B}\right)}_{\alpha_{MIP}(t)} u(x-t)dt\right). \tag{V.8}$$

The denominator of B is arbitrarily close to 0. Thus, $\alpha_{MIP}(t)$ can become larger than one. This means that, in general, it could not be realized *stricto sensu* by an *analog flutter shutter*, where the relative motion v of landscape and camera would be uniform. Fortunately, the above formula shows that the *motion-invariant photography* apparatus is mathematically equivalent to an *analog flutter shutter* and can be analyzed in terms of SNR like any other *flutter shutter*. It is not a *numerical flutter shutter*, since the *flutter shutter function* modifies directly the intensity of the Poisson random variables. This means that the observed samples of the *motion-invariant photography* are always Poisson random variables. The claim raised in [69, 70] that the method is motion invariant comes from the fact that the kernel only depends of $|\frac{v}{a}|$. Thus, if $|a|$ is large enough, the relative variations of $|\frac{v}{a}|$ toward v are small. Under that assumption the kernel $\alpha_{MIP}(t)$ is indeed nearly invariant with respect to the velocity v. Notice that when $T \to +\infty$, the "A" part of $\alpha_{MIP}(t)$ tends to 0, since $a(\frac{T}{2}-|\frac{v}{a}|)^2 \to sign(a)\infty$.

Theorem 2.1. *The motion-invariant photography using a finite aperture time T is equivalent to an analog flutter shutter with a flutter function equal to*

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

Proof. This is a direct consequence of (V.7)-(V.8) and chapter IV. □

The question arises of whether or not the kernel

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

is indeed invertible for all band-limited functions whose Fourier transform lies on $[-\pi,\pi]$. This finite aperture scheme is a technically feasible approximation of the ideal *motion-invariant photography*

using an infinite aperture time with an accelerating camera. Let the aperture time $T \to +\infty$, then provided $a > 0$, $\alpha_{MIP}(t) \to \frac{\mathbb{1}_{]-\frac{v^2}{2a},+\infty[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}$ in L^1_{loc} and in the tempered distribution sense. Thus, skipping the time translation, we get the ideal *motion-invariant photography* "flutter" function $\alpha_{MIP-ideal}(t) := \frac{\mathbb{1}_{]0,+\infty[}(t)}{\sqrt{at}}$. In order to analyze the bounds of the *motion-invariant photography*, its Fourier transform $\hat{\alpha}_{MIP-ideal}(\xi) = \int_0^{+\infty} \frac{1}{\sqrt{at}} e^{-it\xi} dt$ is computed in Annex 1. Note that when $a < 0$, $\alpha_{MIP}(t) \to \frac{\mathbb{1}_{]-\frac{v^2}{2a},-\infty[}(t)}{\sqrt{a(t+\frac{v^2}{2a})}}$ whose Fourier transform is $\hat{\alpha}_{MIP-ideal}(-\xi)$. Thus *asymptotically* the choice of the direction of the acceleration has no influence on the invertibility of the *motion-invariant photography*.

Lemma 2.2. *(Invertibility of the motion-invariant photography method.)*
Using a large enough aperture time the motion-invariant photography kernel is invertible, whatever the sign of a.

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

Lemma 2.3. *(Efficiency of the ideal motion-invariant photography method.)*
When $T \to +\infty$ the ideal motion-invariant photography method has a spectral SNR

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

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

Proof. We have, when $T \to \infty$, $\alpha_{MIP} \to \alpha_{MIP-ideal}$ thus at $\xi = 0$ $\hat{\alpha}_{MIP-ideal}(0) = ||\alpha_{MIP-ideal}||_{L^1}$ since $\alpha_{MIP-ideal}(t) \geq 0$, which proves, by Thm. 1.2, that $SNR^{spectral}(0) = \lim_{x\to\infty} \frac{x}{\sqrt{x}} \frac{|\hat{u}(0)|}{\sqrt{||u||_{L1}}} = \frac{|\hat{u}(0)|}{\sqrt{||u||_{L1}}}\infty$. Let now $\xi \neq 0$. Then

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

(since $||\alpha_{MIP-ideal}||_{L^1} = \infty$, and $|\hat{\alpha}_{MIP-ideal}(\xi)| < \infty$). This entails, again by Thm. 1.2, that $SNR^{spectral}(\xi) = 0$. The last result comes from the fact that the variance is infinite on a set $[-\pi, \pi] \setminus \{0\}$, thus $SNR^{spectral-averaged} = 0$. □

Lemma 2.3 means that the *motion-invariant photography* behaves like a standard camera using an infinite time exposure : only the null frequency is preserved. Indeed an invertible kernel does not guarantee a good SNR *after* deconvolution (except on the unreal case where the acquired samples are noiseless). A convolution against a kernel $\alpha(t)$ having a small but non zero $|\hat{\alpha}(\xi)|$ on the support of $\hat{u}(\xi)$ (for example a Gaussian with a large standard deviation) would lead to the same result. The *motion-invariant photography* is therefore a perfect example of the *flutter shutter paradox* (Cor. 1.6 chapter VI). To sense many more photons does not necessarily imply a better SNR after deconvolution. The authors of [69] wrote as a drawback of the *flutter shutter* that it was losing half the photons while the *motion-invariant photography* kept them all: "(about the Agrawal *et al.* *flutter shutter*) ...the amount of recorded light is halved. Because of the loss of light, the vertical budget is reduced from $2T$ to T for each ω_x". The number of acquired photons can be arbitrarily large using a *flutter shutter* or a *motion-invariant photography* apparatus. Nevertheless the SNR of the image obtained *after* deconvolution is lower than the SNR (see chapter VIII) of the best snapshot acquiring little photons (comparatively) and despite the fact that the snapshot "spends energy outside the slope wedge and thus does not make a full usage of the vertical $\hat{k}_{\omega_x}$ budget" [69]. We now turn to practical aspects of the *motion-invariant photography*. For obvious practical reasons it is not possible to accelerate infinitely the camera. Thus $\hat{\alpha}_{MIP}$, using a finite aperture time, is nonetheless an approximation of $\hat{\alpha}_{MIP-ideal}$. It may seem surprising, at first sight, that the finite aperture approximation has a better SNR than the ideal one. This comes from the fact that, for a finite time aperture, α_{MIP} belongs to $L^1(\mathbb{R})$. Its Fourier transform may have zeros but, for finite and large enough T's they are *outside* $[-\pi, \pi]$, the support of $\hat{u}$. This fact is illustrated below, where the value $SNR^{spectral-averaged}$ is given for a variety of choices for a and T. To compare, on an equal footing, all *flutter shutters* and the *motion-invariant*

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

Table A.1 – This table provides the relative $SNR^{spectral-averaged}$ compared to the best snapshot. A number greater than one means an increase of the SNR, less than one a loss. This fact illustrates the asymptotic result on the *motion-invariant photography* (lemma 2.3) : the bigger T is the worse the results become (the noisier the deconvolved is).

photography we propose to find a piecewise constant *flutter shutter code* approximating $\alpha_{MIP-ideal}$, using the framework of Thm. 2.1. This permits to override a drawback of the method. Indeed, a *flutter shutter* implementation will work *without* the *a priori* knowledge of the direction of the velocity v of the landscape. It is a bit clumsy to directly approximate α_{MIP}, since it already is an approximation of the ideal $\alpha_{MIP-ideal}$ *motion-invariant photography* function and would result inevitably in a lower SNR (an therefore an unfair comparison). To do so, we remark that $\hat{\alpha}_{MIP-ideal}$ does not belong

to $L^1(\mathbb{R})$ nor to $L^2(\mathbb{R})$ and to avoid this pitfall we change it for $\hat{\alpha}_{MIP-ideal}(\xi)\mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$. Indeed frequencies outside the interval $[-\pi|v|,\pi|v|]$ are of no interest for our scope since u is band limited on $[-\pi,\pi]$, the expected value of the observation being $\mathbb{E}\left(obs(n)\right) = \left(\frac{1}{v}\alpha\left(\frac{\cdot}{v}\right) * u\right)(n)$ (see Thm. 1.1).

This change does not ensure that $\hat{\alpha}_{MIP-ideal}(\xi)\mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$ belongs to $L^2(\mathbb{R})$. Nevertheless we can at least compute its Fourier expansion (as the Fourier expansion of an $L^1(\mathbb{R})$ function). Now, we are in position to apply Thm. 2.1 (with $\hat{\alpha}_{MIP-ideal} = \frac{1}{\sqrt{|a\xi|}}e^{-i\frac{\pi}{4}sign(\xi)}\mathbb{1}_{[-\pi|v|,\pi|v|]}(\xi)$) and to compute the code. Being Hermitian this function provides a real code (i.e. coming from a real function in the space domain). The only loss incurred in applying Thm. 2.1 with a function that does not belong to $L^2(\mathbb{R})$ is the goodness of the convergence, which is reduced to a tempered distribution convergence. Nonetheless, by the localization principle and Riemann-Lebesgue theorem, we also have at the very least a pointwise convergence everywhere, except in 0, of the Fourier expansion. Thus, it is no surprise that the proposed numerical approximation (which is a trigonometric polynomial, thus $C^\infty(\mathbb{R})$) works well and is indeed invertible for all band limited functions such that $\hat{u}$ is supported on $[-\pi,\pi]$. The obtained code (w.l.o.g for $v = 1$ and $\Delta t = 1$) and its Fourier transform are shown Fig. A.3, and will be compared advantageously to the Agrawal *et al.* code later on (chapter VIII). The proposed implementation of *motion-invariant photography* using a *numerical flutter shutter*, is simpler from a technical point of view, since it does not require to control the camera motion itself. This permits to compute SNR's for *any* finite code and compare the *motion-invariant photography* like any other *flutter shutter* set up. The above formalism, paradoxes, and comparison also applies

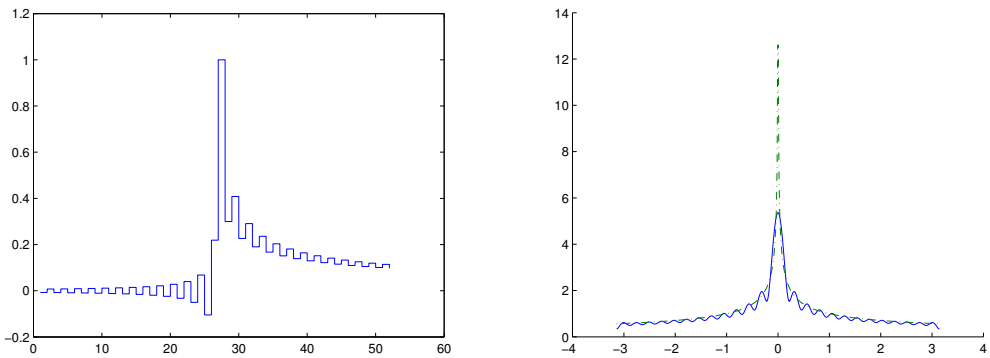

Figure A.3 – Left: the *flutter shutter* gain function for the *motion-invariant photography* code (w.l.o.g. for $v = 1$), $x - axis$: k, $y - axis$: the gain α_k. On the right: the Fourier transform (modulus) of the *motion-invariant photography* code (in bold) and of the ideal *motion-invariant photography* function $\hat{\alpha}_{MIP-ideal}$ (dashed dots line style). As predicted the proposed approximation does not vanish on $[-\pi,\pi]$. Thus the convolution of a band-limited function by the *motion-invariant photography* code is invertible, $x - axis$: frequency ξ, $y - axis$: $|\hat{\alpha}(\xi)|$.

to the "Motion blur removal with orthogonal parabolic exposures" [25], a recent extension of the *motion-invariant photography* using two images, namely two orthogonal motion invariant apparatus. Indeed it is equivalent to the use of two *flutter shutters* and, in that case, a fair comparison shall also involve the acquisition of two *flutter shutter* images. Surprisingly, the *numerical flutter shutter*

permits to approximate this ideal function with a finitely supported *flutter shutter* function (that is, a finite code) while avoiding an unrealistic infinite acceleration. It also permits to get rid of the exigence of an *a priori* knowledge of the direction of v. In consequence, like any other *flutter shutter* the coded-*motion-invariant photography* will work for any direction of v. Finally, it increases the efficiency of the method compared to the classic one involving an accelerating camera. Indeed it permits to control the Fourier transform and to concentrate it easily on the support of $\hat{u}(\xi)$ i.e. where the information is (contrarily to $\hat{\alpha}_{mip-ideal}$ which is supported on the whole $\mathbb{R}$). Predictive results are shown in Tab. A.2 and simulations in chapter VIII.

Chapter VI

The *Flutter Shutter* Paradox

This chapter proves that the best aperture strategy is indeed a flutter shutter, not a single aperture. It is proven that given the relative velocity v the strategy guaranteeing the best SNR of any *flutter shutter comes from the Fourier series coefficients of a (zoomed) sinc function. This means that the best flutter shutter is not realizable by an analog flutter shutter apparatus. It is proven that the SNR of the recovered sharp image remains finite, regardless of the aperture time. It is proven that the flutter shutter does always beat the standard camera, even though the aperture time remains the same. Compared to the best snapshot the best numerical flutter shutter camera increases the SNR by* 17%.

Theorem 1.4. (Ideal *flutter shutter* function)
Consider a landscape $u(x - vt)$ moving at velocity v. Then an optimal continuous numerical flutter shutter gain function maximizing the average spectral SNR (II.9) is equal to $\alpha^*(t) = sinc(tv)$.

Proof. Among all gain control functions $\alpha(t)$ one giving the best $SNR^{spectral-averaged}$ (II.9) is given by minimizing the averaged variance of $\hat{\mathfrak{u}}_{est}$ (III.16),

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

where the inequality is Jensen's inequality applied to the strictly convex function $x > 0 \mapsto \frac{1}{x}$. Because of this strict convexity, the equality occurs when $|\hat{\alpha}(\xi)|^2 \equiv 1$ on $[-\pi v, \pi v]$, up to an irrelevant multiplicative constant for a *numerical flutter shutter* (see chapter III Lemma 1.3). Thus, an optimal *numerical flutter shutter* function is $\alpha^*(t) = \text{sinc}(tv)$ (up to a normalization constant). □

Notice that the proposed optimal *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 "self-deconvolving". Being non positive this ideal gain control function is *not* implementable using an *analog flutter shutter*, and being non piecewise-constant is *not directly* implementable using a *numerical flutter shutter* strategy. However a piecewise constant approximation can be used in a *numerical flutter shutter* strategy

with Thm. 2.1 (chapter III) as soon as $|v|\Delta t \leq 1$, and it is enough to let $\Delta t \to 0$ to approximate the optimal flutter shutter.

Corollary 1.5. (Upper bound on the SNR)
Consider a landscape $u(x-vt)$ moving at velocity v. The ideal numerical flutter shutter strategy using $\alpha^*(t) = sinc(tv\Delta t)$ has a spectral SNR (II.8) equal to

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

Moreover the averaged spectral SNR (II.9) is equal to

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

Proof. By Thm. 1.4, an optimal flutter shutter strategy satisfies $|\hat{\alpha}^*(\xi)| = \mathbb{1}_{[-\pi v,\pi v]}$. Using Parseval's formula we deduce that $||\alpha^*||_{L^2}^2 = v$. Then from chapter III Thm. 1.3 we deduce that $SNR^{spectral}(\xi) = \frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\sqrt{v}} \frac{|\hat{u}(\xi)|}{\sqrt{||u||_{L^1}}}$. It follows that $SNR^{spectral-averaged} = \frac{1}{2\pi\sqrt{v}} \frac{\int_{-\pi}^{\pi} |\hat{u}(\xi)| d\xi}{\sqrt{||u||_{L^1}}}$. □

Corollary 1.6. (The *flutter shutter paradox*)
The use of a flutter shutter strategy increasing the total exposure time does not permit to achieve an arbitrary SNR. Consider a landscape $u(x-vt)$ moving at velocity v. Then the $SNR^{spectral-average}$ of any flutter shutter strategy is bounded independently of the total exposure time. In other words increasing the exposure time has a limited effect on the SNR.

Proof. This is a direct consequence of Cor. 1.5. Indeed, the exposure time is the (infinite) length of the support of α^*, but the SNR of the restored image is nevertheless finite.

Moreover, $SNR(analog\ flutter) \leq SNR(numerical\ flutter)$ for *any analog flutter shutter* function and $SNR(numerical\ flutter) \leq SNR(best\ numerical\ flutter) < \infty$ by Cor. 1.5. Thus,

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

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

□

Corollary 1.7. (Efficiency of the numerical *flutter shutter*)
Consider a landscape $u(x-vt)$ moving at velocity v. Then the ratio R of $SNR^{spectral-average}$ between the ideal flutter shutter and the best snapshot with exposure time equal to Δt^* is equal to $R = \frac{SNR^{spectral-average}(flutter,ideal)}{SNR^{spectral-average}(snapshot)} \approx \frac{\frac{1}{2\pi}}{0.1359} \approx 1.171$.

Proof. This is a direct consequence of Thm. 1.2 (chapter V) and Cor. 1.5. □

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

Corollary 1.8. (Deconvolution gain) *For a landscape $u(x - vt)$ moving at velocity v, consider its best classic snapshot with exposure time equal to Δt^* and the truncated flutter shutter strategy using $\alpha = \alpha^* \mathbb{1}_{[-\frac{\Delta t^*}{2}, \frac{\Delta t}{2}]}$. Then the spectral SNR, $SNR^{spectral-average}$ is larger for this restricted flutter shutter than for the best snapshot. The ratio of the SNRs is approximately* 1.04.

Proof. This is a mere numerical estimation using Thms. 1.2 (chapter V) and 1.4 (chapter VI). □

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

The tabular Tab. A.1 summarizes the results on the various *flutter shutter* strategies explored, focusing on the resulting SNR.

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

Table A.1 – This table summarizes the results on the different *flutter shutter* strategies and their SNR. On the first column the types of flutter are indicated. The second and last row give the optimal *flutter shutter* function in two categories: the best simple snapshot, and the best numerical *flutter shutter*. The best *numerical flutter shutter* is a sinc and has a finite SNR in spite of using an infinite exposure time. This is the *flutter shutter* paradox of Cor. 1.6. M.I.P. stands for *motion-invariant photography*. The second column shows the form of the α function. Agrawal *et al.* code is analog discrete, with $\alpha_k \in \{0, 1\}$. The last column gives the SNR's of each method in its more adequate presentation: the first line shows that the accumulation is the best strategy, the only one able to increase indefinitely the SNR. The second and last lines compare the average spectral SNR's for the best snapshot and the best *numerical flutter shutter* (av. stands for average). The SNR gain with the *numerical flutter shutter* with respect to the best snapshot is only approximately 1.171. The spectral SNR formulas for the *analog* and *numerical flutter shutter* are similar but distinct. The *analog* involves the L^1 norm of α and the *numerical* the L^2 norm.

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

Table A.2 – This table provides the relative $SNR^{spectral-averaged}$ of all standard *flutter shutter* strategies compared to the best snapshot. A number greater than one means an increase of the SNR, less than one a loss. All recent methods does have a good SNR on the acquired image (because the support of their *flutter shutter* function is big and the number of photons caught much higher compared to the snapshot), but it is all lost by the deconvolution if the kernel is not well chosen. Those results are coherent with the numerical simulations of chapter VIII.

Chapter VII

A Stochastic Solution to the *Flutter Shutter Paradox*

This chapter sets a better mouse trap to override the flutter shutter paradox of Cor. 1.6 chapter VI and increase the efficiency beyond the 17% *bound, by assuming that the observed objects adopt a known (or learned) random velocity distribution. This set up is best suited to analyze and optimize the case of multiple objects and/or velocities in the observed scene. Analytical formulas are proposed that link an optimal flutter shutter code with the probability densities of the expected velocities. Not only this theory permits to formalize the design of optimized codes given a random velocity model. Conversely, it allows us to analyze a posteriori* any *existing flutter shutter strategy, and to perform a reverse engineering of existing patented codes. This theory also applies to standard cameras and permits to compute the best snapshot given* any *velocity probability density.*

When the motion is perfectly known then the best *flutter shutter function* is a *sinc* as proved in chapter VI Thm. 1.4. However as stated in chapter VI Cor. 1.7 the use of this *sinc* function is not a breakthrough compared to the best snapshot since the SNR is increased by small factor. Nevertheless, if the motion is not well-known then without some *flutter shutter* method the only way to ensure a sharp image for a broad range of velocities v is to use a very small exposure time leading to a poor SNR. It is then possible to achieve better results *on average*, and to ensure anyway always a sharp image. To optimize the SNR gain, however, we must know the probability distribution of the velocities. Fixing the norm of the code and maximizing its DFT (discrete Fourier transform) as done in [4, 6, 55] does not lead to optimal codes in a strong sense.

1 From motion to codes

Let $\rho(v)$ be a probability density for $v \in \mathbb{R}$ (which is equivalent to a probability density over the blur length), such that $\rho(v) = 0$, $\forall |v| > v_{max}$. Notice that without the knowledge of the maximum speed v_{max} it is virtually impossible to guarantee $|v|\Delta t < 2$ (see chapter III section 1.3) for v in the whole support of ρ. We assume that the probability density ρ has been estimated during a calibration phase,

(e.g.) by taking sample snapshots without *flutter shutter*, and using them to compute the actual blur density function by tracking zeros in the image DFT. In this section we address the question of finding the best flutter function (the best code) α, yielding an average minimal noise for the restored image. The noise level is measured by the variance of the restored image. Thus we look for a *flutter shutter* code maximizing $SNR^{spectral-average}$, which amounts by (III.16) to minimize

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

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

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

As stated previously if v is known the optimization of E_v is easy. For $\rho(v) = \delta_{v_0}$ the best code is *not* a snapshot. The optimal *flutter shutter* for fixed v maintains a white noise after deconvolution. This is achieved with a sinc code whose spectral support is uniform over $[-\pi|v|, \pi|v|]$. Now, when v is a random variable whose probability density is known, the question is to find again the optimal α_k coefficients. There are several obvious possible causes for a variable velocity:

1. When the scene is not planar, the speed of objects will change according to their distance;

2. When the camera shakes or moves (with non-constant and or unknown amplitudes);

3. In Agrawal's examples, the background is static, the observed vehicles move.

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

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

where we used the Fubini theorem. Then from (VII.1) defining

$$w(\xi) := \int_{\mathbb{R}} \frac{\rho(v)\mathbb{1}_{[-|v|\pi,|v|\pi]}(\xi)}{|v|} dv, \tag{VII.2}$$

this minimization boils down to minimizing the functional

$$E(\hat{\alpha}) = ||\hat{\alpha}||_2^2 \int_{-\infty}^{\infty} \frac{w(\xi)}{|\hat{\alpha}|^2(\xi)} d\xi. \tag{VII.3}$$

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

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

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

Theorem 1.1. *The optimal functions α^* minimizing $E(\hat{\alpha})$ satisfy $|\hat{\alpha}^*| = \sqrt[4]{w}$ on the support of w. If we extend $\hat{\alpha}^*$ by zero outside this support, we obtain bounded continuous codes belonging to $L^2(\mathbb{R})$.*

Proof. We choose as unknown function $f(\xi) := |\alpha|^2(\xi)$. To simplify expressions we omit the ξ integration variable in the integrals which are all on $\mathbb{R}$. We have by (VII.3), $E(f) = (\int f) \int \frac{w(\xi)}{f(\xi)} d\xi$. Thus its weak differential in the direction of every bounded perturbation g which has compact support in the support of f satisfies

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

A minimal f therefore satisfies for every bounded g with support contained in the support of f, $E'(f)(g) = 0$ which yields

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

and therefore $f = C\sqrt{w}$ on the support of w. Being nonnegative this solution is admissible and we deduce $|\hat{\alpha}(\xi)| = \sqrt[4]{w}$ on the support of w, as announced. It remains to show that when extended by zero out the support, this is our solution. To this aim, we notice that the minimization of $E(f)$ is equivalent to minimizing the strictly convex functional $F(f) := \int \frac{w(\xi)}{f(\xi)}$ under the constraints $\int f = 1$, $f \geq 0$, $f = 0$ outside the support of w. The solution to this problem, if it exists, is therefore unique. Thus, it remains to show that $f := \frac{w^{\frac{1}{2}}}{\int w^{\frac{1}{2}}}$ is our solution. This is an easy consequence of Cauchy-Schwartz inequality. Indeed, for every f that is positive on the support of w,

$$\int w^{\frac{1}{2}} = \int \frac{w^{\frac{1}{2}}}{f^{\frac{1}{2}}} f^{\frac{1}{2}} \leq \left(\int \frac{w}{f} \right)^{\frac{1}{2}} \left(\int f \right)^{\frac{1}{2}}.$$

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

The only thing left is, given L, to compute the coefficients α_k, $k \in \{0, ..., L-1\}$ to approximate the above solution.

Corollary 1.2. (Computation of the α_k coefficients)
Let $\rho(v)\mathbb{1}_{[-|v_{max}|,|v_{max}|]}(v)$ be a compactly supported probability density on the velocities v and Δt be such that $|v_{max}|\Delta t \leq 1$. Then the (piecewise constant) flutter shutter function defined by $\alpha(t) = \sum_{\mathbb{N}} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$ with $\alpha_k = \frac{1}{2\pi} \int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w(\frac{s}{\Delta t})} cos(\frac{2(k+1)s)}{2}}{sinc(\frac{s}{2\pi})} ds$ is the best quadratic approximation of the ideal flutter shutter function namely $\mathcal{F}^{-1}(\sqrt[4]{w})(t)$.

Proof. The goal is here to construct $\alpha(t)$ a piecewise constant function such that (from Thm. III.7) : $\hat{\alpha}(\xi v) = \sqrt[4]{w(\xi v)}$ for $\forall \xi \in \mathbb{1}_{[-\pi,\pi]}$ and $v \in support(\rho)$. Thus we can apply Thm. 2.1 (with $|v_{max}|$ instead of $|v|$ and $\beta = \sqrt[4]{w}$). Now,

$$\alpha_k = \frac{1}{2\pi}\int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w(\frac{s}{\Delta t})}}{\text{sinc}(\frac{s}{2\pi})e^{\frac{-is}{2}}} e^{iks} ds = \frac{1}{2\pi}\int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w(\frac{s}{\Delta t})}}{\text{sinc}(\frac{s}{2\pi})} e^{\frac{i(2k+1)}{2}} ds$$
$$= \frac{1}{2\pi}\int_{-\pi|v_{max}|\Delta t}^{\pi|v_{max}|\Delta t} \frac{\sqrt[4]{w(\frac{s}{\Delta t})} cos(\frac{2(k+1)s)}{2}s)}{\text{sinc}(\frac{s}{2\pi})} ds \text{ since } w \text{ is symmetric with respect to zero.}$$

This concludes the proof. □

As wished the a_k coefficients of the optimized *flutter shutter* code are real but not necessarily positive (nor the *ideal flutter shutter* function $\mathcal{F}^{-1}(\sqrt[4]{w})(t)$) implying that the code is not usable with an *analog flutter shutter* camera. Nevertheless in practice the use of a finite number a_k does not guarantee that $\hat{\alpha}(\xi) \neq 0$ for $\xi \in [-\pi|v_{max}|\Delta t, \pi|v_{max}|\Delta t]$.

2 From codes to motion

By the formulas of the preceding section, given a code $(\alpha_0, ..., \alpha_{L-1}) \in \mathbb{R}^L$, we are now able to estimate its underlying velocity probability distribution.

Theorem 2.1. *Let $\alpha(t)$ be a flutter shutter function. Then, it is optimal with respect to some velocity probability distribution $\rho(v)$ if and only if the function $\xi \in \mathbb{R}^+ \to |\hat{\alpha}(\xi)|$ is non increasing. Moreover, if $\xi \in \mathbb{R}^+ \to |\hat{\alpha}(\xi)|$ is non increasing then $\rho(v) = -\frac{1}{2}vw'(\pi v)$, $v \neq 0$.*

Proof. We first establish how the function $\rho(v)$ can be deduced from the function w (and therefore eventually from α). We first notice that given w, if there is an associated function ρ, it can be assumed even. Indeed, if it is not, (VII.2) is still true when replacing ρ by its symmetrized $\frac{\rho(v)+\rho(-v)}{2}$. Thus we can look for an even ρ and simplify (VII.2) into

$$w(\xi) = 2\int_0^{+\infty} \frac{\rho(v)\mathbb{1}_{[-v\pi,v\pi]}(\xi)}{v} dv = w(-\xi)$$

Thus, for $\xi \neq 0$ we shall have

$$w'(\xi) = sign(\xi)w'(|\xi|) = 2sign(\xi)\left(\int_0^{+\infty} \frac{\rho(v)\mathbb{1}_{[-v\pi,v\pi]}(|\xi|)}{v} dv\right)'$$
$$= 2sign(\xi)\left(\int_{\frac{|\xi|}{\pi}}^{+\infty} \frac{\rho(v)}{v} dv\right)' = -2sign(\xi)\left(\int_{+\infty}^{\frac{|\xi|}{\pi}} \frac{\rho(v)}{v} dv\right)'$$
$$= -2sign(\xi)\frac{\rho(\frac{|\xi|}{\pi})}{\frac{|\xi|}{\pi}} = -2\frac{\rho(\frac{|\xi|}{\pi})}{\frac{\xi}{\pi}}.$$

Hence, we have $w'(\pi\xi) = -2\frac{\rho(\xi)}{\xi}$ If follows that $\rho(v)$ is entirely determined by

$$\rho(v) = -\frac{1}{2}vw'(\pi v), \text{ for } v \neq 0. \tag{VII.4}$$

This formula has two uses. First, if $\xi \in \mathbb{R}^+ \to w'(\xi) = (|\hat{\alpha}|^4)'(\xi)$ is positive on a set with positive measure, then ρ cannot be a probability density. Thus, the considered code is not optimal for any velocity distribution if $|\hat{\alpha}|$ is increasing somewhere. If instead $|\hat{\alpha}|$ is nonincreasing on $\mathbb{R}^+$, (VII.4) gives the direct algorithm calculating ρ detailed in Algorithm 1. □

input : a *flutter shutter code* $(\alpha_k)_{k\in\{0,\cdots,L-1\}}$, Δt the time step of the *flutter shutter*
output: underlying probability density ρ for which the code is optimal

1. compute $\hat{\alpha}(\xi)$ $(= \Delta t sinc(\frac{\xi\Delta t}{2\pi})e^{-i\xi\Delta t}\sum_{k=0}^{L-1}\alpha_k e^{-ik\xi\Delta t}$, see Table A.1)
2. compute w the weight (VII.2) function by $\tilde{w}(\xi) = |\hat{\alpha}(\xi)|^4$
3. compute $\rho(v) = -\frac{1}{2}vw'(\pi v)$, $v \neq 0$

Algorithm 1: Computing the velocity distribution associated with a given code.

Remark As we shall see most classic codes do not strictly satisfy the conditions of Theorem 2.1. Fortunately, the measure of the set where for these codes $|\hat{\alpha}|$ is increasing is small, and this occurs on intervals where $\hat{\alpha}$ is also small. Thus, we can apply Algorithm 1 by modifying only slightly α (or w) by replacing (VII.4) by

$$\rho(v) = -\frac{1}{2}vw'(\pi v)\mathbb{1}_{-vw'(\pi v)\geq 0}, \text{ for } v \neq 0. \tag{VII.5}$$

This algorithm will be applied in chapter VIII section 2 to several classic patented codes to give their underlying $\rho(v)$.

3 Best snapshot on average

The goal of this subsection is to define an optimized snapshot using the framework of section 1. Again, provided $\rho(v)$ a probability density on v we shall minimize (from section V Thm. 1.2)

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

which yields to

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

Notice that $\Delta t \leq \frac{2}{|v_{max}|}$. The zero of $E'(\Delta)$ is computed numerically.

Theorem 3.1. (Blurs on average invariance)
Given $s > 0$ and $\rho_s(v) = s\rho(vs)$ a family of probability densities which comes from a compactly supported mother function $\rho(v)\mathbb{1}_{[-|v_{max}|,|v_{max}|]}(v)$, the optimal snapshot on average using the Δt^* associated to $\rho_s(v)$ then the best blur on average, namely $s\Delta t^*$ remains constant.

Proof. Let Δt^* be such that

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

is minimal and consider

$$\begin{aligned} E_s(\Delta t) &= \int\int_{-\pi}^{\pi} \frac{\xi^2}{sin^2(\frac{\xi v \Delta t}{2})} \frac{v^2 \Delta t}{4} \rho_s(v) d\xi dv = s \int\int_{-\pi}^{\pi} \frac{\xi^2}{sin^2(\frac{\xi v \Delta t}{2})} \frac{v^2 \Delta t}{4} \rho(vs) d\xi dv \\ &= \int\int_{-\pi}^{\pi} \frac{\xi^2}{sin^2(\frac{\xi v}{2}\frac{\Delta t}{s})} \frac{v^2 \Delta t}{s^2} \rho(v) d\xi dv = \frac{1}{s} \int\int_{-\pi}^{\pi} \frac{\xi^2}{sin^2(\frac{\xi v}{2}\frac{\Delta t}{s})} \frac{v^2 \Delta t}{s} \rho(v) d\xi dv = \frac{1}{s} E(\frac{\Delta t}{s}). \end{aligned}$$

Thus $E_s(\Delta t)$ is minimized for $\frac{\Delta t}{s} = \Delta t^*$. □

Corollary 3.2. *The best blur on average remains constant to (approximatively)* 1.44 *for uniformly distributed velocities and constant to (approximatively)* 1.96 *for truncated Gaussian distributed velocities.*

Proof. Immediate since $\rho_s(v) = s\mathbb{1}_{[\frac{-1}{2},\frac{1}{2}]}(vs)$ or $\rho_s(v) = s\mathbb{1}_{[-4,4]}(vs)e^{\frac{-(vs)^2}{2}}$ (up to an irrelevant constant factor for the optimization of E). □

Definition (Best snapshot on average.)
Given a moving landscape $u(x - vt)$ moving at velocity v with a probability of density $\rho(v)$, we call best snapshot on average the use of the exposure time Δt^* minimizing $E(\Delta t) = \int\int_{-\pi}^{\pi} \frac{\xi^2}{sin^2(\frac{\xi v \Delta t}{2})} \frac{v^2 \Delta t}{4} \rho(v) d\xi dv$, the variance of the noise on average. When $\rho(v)$ is uniform over $[-|v_{max}|, |v_{max}|]$ then $\Delta t^* |v_{max}|$ is constant ($\Delta t^* |v_{max}| \approx 1.44$). Similarly when $\rho(v)$ is a truncated Gaussian, (ie up to a constant renormalization, $\rho(v) = \mathbb{1}_{[-4\sigma,4\sigma]} e^{\frac{-v^2}{2\sigma^2}}$) then $\Delta t^* |v_{max}|$ is also constant ($\Delta t^* |v_{max}| \approx 1.96$).

Those results means that for a standard camera, *assuming* a uniform or a truncated Gaussian motion model the exposure time should be tuned such that the blur length never exceed the $\Delta t^* v_{max}$ constant dependent of the motion model (see Fig. A.1).

Theorem 3.3. (Scale invariance of the ideal Fourier transform of the *flutter shutter functions*)
Given $s > 0$ and $\rho_s(v) = s\rho(vs)$ a family of probability densities which comes from a compactly

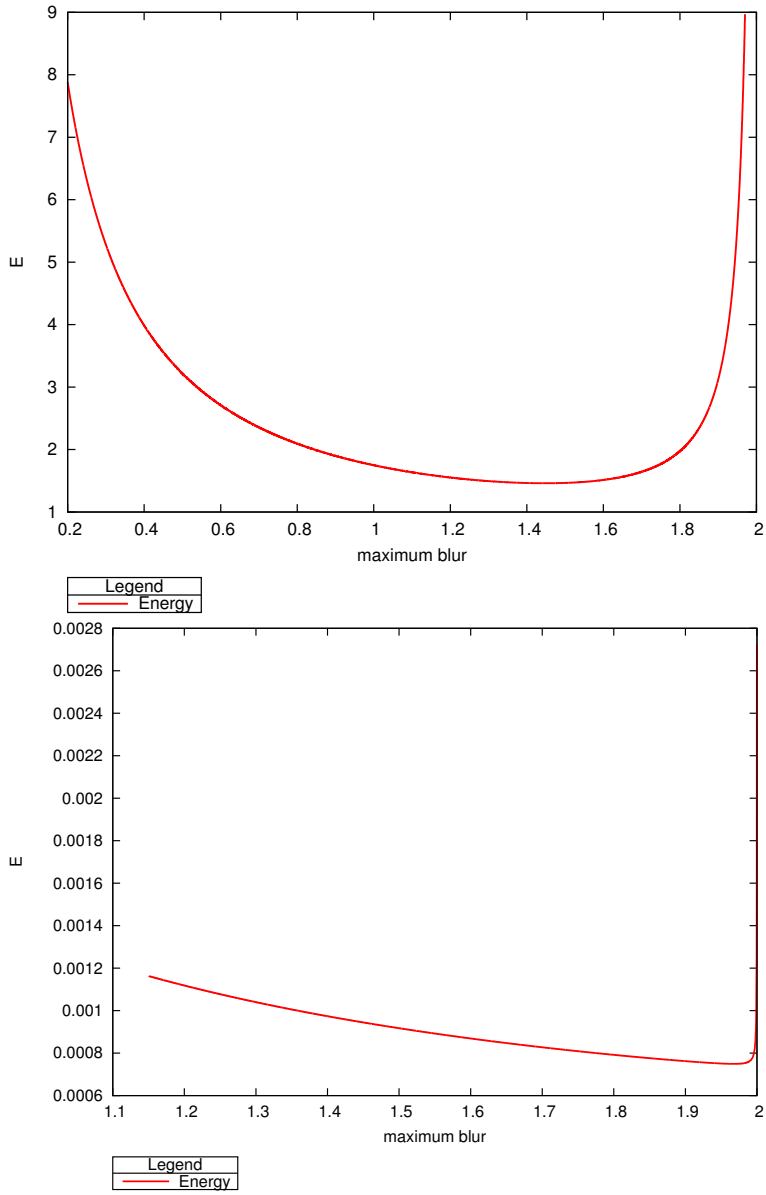

Figure A.1 – The energies E -proportional to the deviation of the noise (on average)- for different snapshots: x-axis maximum blur ($v_{max}\Delta t$) in pixel, y-axis the value of E. Top: for a uniform motion model on the velocities. The minimum is reached for a maximum blur of approximately 1.44 pixel. Bottom : for a truncated Gaussian motion model on the velocities. The minimum is reached for a maximum blur of approximately 1.96 pixel. It is worth to take more risk, as the probability of higher velocities is much smaller for the Gaussian model. As previously done, the cure is shown for $v_{max} = 1$ w.l.o.g.

supported mother function $\rho(v)\mathbb{1}_{[-|v_{max}|,|v_{max}|]}(v)$*, the optimal snapshot on average using the* Δt^* *associated, a fixed exposure time factor* c *and a code length* L *then the ideal Fourier transform of the flutter function* $\sqrt[4]{w(\xi)}$ *(with* $\Delta t = \frac{C\Delta t^*}{L}$*) is independent of the parameter* s*.*

Proof. From (VII.2) we have

$$w(\xi) = \int \frac{\rho_s(v)}{|v|}\mathbb{1}_{[-|v|\Delta t\pi,|v|\Delta t\pi]}(\xi)dv = s\int \frac{\rho(v)}{|v|}\mathbb{1}_{[-|v|\frac{\Delta t}{s}\pi,|v|\frac{\Delta t}{s}\pi]}(\xi)dv.$$

Then Thm. 3.1 entails that $\frac{\Delta t}{s} = \Delta t^*$ at the optimum, and *numerical flutter shutter* functions are defined up to a constant multiplication (see chapter III section 1.3). This result means that there is much less ideal function that could be expected at first sight (and thus less codes). □

Corollary 3.4. *The ideal Fourier transform of the flutter shutter function remains constant for uniformly distributed velocities and truncated Gaussian distributed velocities.*

Proof. Immediate since $\rho_s(v) = s\mathbb{1}_{[\frac{-1}{2},\frac{1}{2}]}(vs)$ or $\rho_s(v) = s\mathbb{1}_{[-4,4]}(vs)e^{\frac{-(vs)^2}{2}}$ (up to an irrelevant constant factor for w). □

Those results means that assuming a uniform or a truncated Gaussian motion model, a fixed code length L and an exposure time factor compared to the best snapshot on average there is only one Fourier transform of the *flutter shutter function* $\hat{\alpha}(\xi) = \sqrt[4]{w(\xi)}$ independent of the model parameter (the range or the std-dev). The function to approximate being constant in this set up; codes shall be compared varying the exposure time factor.

Chapter VIII

Numerical Simulations

In this chapter, the first part is dedicated to the simulation of various flutter shutter strategies, where we compare them in terms of $RMSE$. The results are coherent with the predictive results using the Fourier based SNR criterion. The flutter shutter camera simulator simulates the observed samples of flutter shutter cameras directly from the Poisson photon emission. Then, the reverse engineering of classic flutter shutter strategies is proposed. The last section is dedicated to the numerical computation and comparison of optimized flutter shutter codes (see chapter VII) for Gaussian, uniform and a realistic handcrafted densities of probability $\rho(v)$ on the observed velocities v.

1 All *flutter shutters*

The purpose of this section is to compare experimentally different acquisition strategies : snapshot, *flutter shutter* using the Agrawal *et al.* code [3], a random uniform over $[-1, 1]$ code, the *motion-invariant photography* code (M.I.P, see section chapter V 2) and the *sinc* code (see section VI Thm. 1.4). All strategies are compared using the $RMSE$, the contrast invariant $RMSE$ ($RMSE^{CI}$) and the visual image quality. A benchmark of acquisition strategies is given (Tab. A.1) to illustrate the *flutter shutter* paradox of Cor. 1.6 section VI.

Code type:	Snapshot	Agrawal *et al.* code	Random code	M.I.P. code	*Sinc* code
$RMSE$	1.47	2.54	2.25	2.31	1.46
$RMSE^{CI}$	1.42	2.83	2.49	2.19	1.42

Table A.1 – Quantitative ($RMSE$, $RMSE^{CI}$) comparison of different strategies for fixed velocity $v = 1$ (so the blur support is of 52 pixels, except for the snapshot) on the *boat* test image. The rand code performs better than the Agrawal *et al.* or the Levin *et al.* M.I.P. code. Indeed the rand code is (on average) closer to the optimum *sinc* code. Unsurprisingly, the best $RMSE$ is obtained using the *sinc* code. But it beats only slightly the snapshot, as predicted by the *flutter shutter paradox* of Cor. 1.6 section VI. The Levin *et al.* *motion-invariant photography* and the Agrawal *et al.* code of *flutter shutter* are perfects examples of the *flutter shutter paradox*. More acquired photons does not necessarily mean a better SNR in the deconvolved image ("a photon can kill another photon!").

1.1 Algorithm

There exist two different types of *flutter shutter* depending on whether the gain modification takes place before (*analog flutter shutter*) or after the photons hit the sensor (*numerical flutter shutter*). For an *analog flutter shutter* the gain is defined as the proportion of incoming photons that are allowed to travel to the pixel sensor. Thus only positive (actually in $[0, 1]$) gains are feasible. The *numerical flutter shutter* camera, instead, takes a burst of L images using an exposure time of Δt . Then the $k-th$ image is multiplied by a gain $\alpha_{\in}\mathbb{R}$ and added to the previous one to obtain the observed image. Consequently, only one image is stored and transmitted. This implies that the observed value $obs(n)$ at pixel n is always a Poisson random variable for the *analog flutter shutter*, but not for the *numerical flutter shutter*. In the following the sequence of gains used on the camera is called "*flutter shutter* code" and is defined as the vector $(\alpha_k)_{k=0,\cdots,L-1}$. Given a code the *flutter shutter* function is defined by $\alpha(t) = \alpha_k$ for $t \in [k\Delta t, (k+1)\Delta t[$, $\alpha(t) = 0$ otherwise.

1.2 Short description

The algorithms will be first described in a continuous, and then in a discrete (see section 1.6) framework. Roughly the algorithm consists of four steps: 1) Simulate the ideal noiseless observed image, 2) Simulate Poisson (photonic) noise (see section 1.3) to obtain the observed image 3) Estimate the landscape by deconvolution 4) Compute the error, namely the Root-Mean-Squared-Error ($RMSE$) and a contrast invariant $RMSE$ (see section 1.5). The implementation in C++ is detailed in the implementation section 1.6. For color images each component is processed independently, the $RMSE$ is averaged over all components.

The *analog flutter shutter*

The simulation data are: an image $u(x)$, a code (fluttering sequence or gain) $\alpha(t) \geq 0$, a velocity v, and an SNR level (and using a normalized $\Delta t = 1$). Let $\hat{u}(\xi) = \int_{-\infty}^{\infty} u(x)e^{-ix\xi}dx$ be the Fourier transform of u. We assume in the following that u is band limited: $\hat{u}(\xi) = 0,\ \forall|\xi| > \pi$. The *flutter shutter function*

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

is designed so that

$$\hat{\alpha}(\xi v) = \Delta t \frac{2sin(\frac{\xi v \Delta t}{2})}{\xi v \Delta t} \sum_{k=0}^{L-1} \alpha_k e^{i\xi v \Delta t \frac{2k+1}{2}} \neq 0\ \forall \xi \in [-\pi, \pi],$$

therefore it is invertible on the support of $\hat{u}(\xi)$.

1. Compute the ideal noiseless observed pixel value:

   at pixel x the ideal noiseless observed pixel value is $\left(\frac{1}{v}u * \alpha(\frac{\cdot}{v})\right)(x)$ computed using $\mathcal{F}^{-1}\left(\hat{u}(\xi)\hat{\alpha}(v\xi)\right)(x)$.

2. Simulate the observed pixel value:

   the observed pixel value at pixel x is a Poisson random variable with intensity $\lambda(x) = \left(\frac{1}{v}u * \alpha(\frac{\cdot}{v})\right)(x)$ computed using $\lambda(x) = \mathcal{F}^{-1}\left(\hat{u}(\xi)\hat{\alpha}(v\xi)\right)(x)$ by definition of the *analog flutter shutter*. Let $o(x)$ be a realization of $Poisson(\lambda(x))$ (see section 1.3 below).

3. Estimate the landscape pixel value by deconvolution:

   the estimated landscape from the observed pixel value is then obtained by a deconvolution filter, $\mathbb{u}_{est}(\xi) = obs * \gamma$, where γ is the inverse filter satisfying $\left(\frac{1}{v}\alpha(\frac{\cdot}{v}) * \gamma = \delta(x)\right)$, computed by $\hat{u}_{est}(\xi) = \hat{o}(\xi)\hat{\gamma}(\xi) = \frac{\hat{o}(\xi)}{\hat{\alpha}(v\xi)}$.

4. Compute error using $RMSE$ and contrast invariant $RMSE^{CI}$

   Straightforward with section 1.5.

   A detailed numerical implementation is given in section 1.6.

The *numerical flutter shutter*

For the *numerical flutter shutter*, the only difference is that there is no positivity constraint on $\alpha(t)$. But the simulation algorithm is slightly different:

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

is designed so that

$$\hat{\alpha}(\xi v) = \Delta t \frac{2sin(\frac{\xi v \Delta t}{2})}{\xi v \Delta t} \sum_{k=0}^{L-1} \alpha_k e^{i\xi v \Delta t \frac{2k+1}{2}} \neq 0 \ \forall \xi \in [-\pi, \pi],$$

therefore invertible on the support of $\hat{u}(\xi)$.

1. Compute the ideal noiseless elementary pixel value:

   at pixel x the elementary noiseless pixel value is $e_k(x) = \left(\frac{1}{v}u * \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(\frac{\cdot}{v})\right)(x)$ computed using $\mathcal{F}^{-1}\left(\frac{1}{v}\hat{u}(\xi)\hat{\mathbb{1}}_{[k\Delta t,(k+1)\Delta t[}(\xi v)\right)(x)$.

2. Compute the simulated observed pixel value

   By definition of the *numerical flutter shutter*, the observed pixel value is realization of the Poisson mixture $obs(x) = \sum_{k=0}^{L-1} \alpha_k Poisson(e_k(x))$ (see sections 1.3, 1.4 below).

3. Estimate the landscape pixel value by deconvolution

   The estimated landscape from the observed pixel value is then obtained by deconvolution $\mathbb{u}_{est}(\xi) = o * \gamma$, where γ is the inverse filter satisfying $\left(\frac{1}{v}\alpha(\frac{\cdot}{v}) * \gamma = \delta(x)\right)$, computed by $\hat{u}_{est}(\xi) = \hat{o}(\xi)\hat{\gamma}(\xi) = \frac{\hat{o}(\xi)}{\hat{\alpha}(v\xi)}$.

4. Compute error using $RMSE$ and contrast invariant $RMSE^{CI}$

   Straightforward with section 1.5.

A detailed numerical implementation is available in section 1.6.

1.3 Simulation of a Poisson random variable X with intensity λ

The usual algorithm is [62]:

- If ($\lambda \leq 50$) then
  - Let $g = exp(-\lambda)$; $em = -1$; $t = 1$; boolean $rejected = true$;
  - While ($rejected$) do
    1. $em = em + 1$;
    2. $t = t.rand$; (where $rand$ is a uniform on $[0, 1]$ random generator)
    3. If ($t <= g$) then : $X = em$; $rejected = true$; endif;
  - endwhile;
- Else : simulate a Gaussian random variable X with mean and variance equal to λ (Box & Muller [13] used here), and round it.

1.4 Signal to Noise Ratio selection

The goal is to find a renormalization factor λ^2 such that the random variable X defined by $X \sim \frac{1}{\lambda^2} Poisson(\lambda^2 u(x))$ has a $SNR(X) = k$ when $u(x) = 100$. It corresponds to tuning an averaged number of photons for a medium brightness value (100). If $\lambda^2 = \frac{k^2}{100}$ then $SNR(X) = k$.

1.5 Contrast invariant $RMSE$

In many cases reconstruction errors inherent to a method can be quantified using the Root-Mean-Squared-Error $RMSE\,(u, u_{est}) := \sqrt{\frac{\int_D |u(x)-u_{est}(x)|^2 dx}{measure(D)}}$. However, when a small contrast change occurs between the original and the processed image, the $RMSE$ can become substantial, while the images remain perceptually indistinguishable. For example take an image $u(m,n)$ defined over a sub-domain $D \subset \mathbb{Z}^2$, and another one $u_{est} = u + 10$. Then $RMSE(u, u_{est}) = 10$ is large but does not reflect the quality of the reconstruction $\mathbb{u}_{est}$. Comparatively, a convolution with for example a Gaussian can give a smaller $RMSE$ while making considerable damage. This bias is avoided by normalizing the images before computing the $RMSE$. The principle of the normalization is that two images related to each other by a contrast change are perceptually equivalent. Their distance should reflect this fact and be zero. The midway equalization [29] is best suited for that purpose, because it equalizes the image histogram to a "midway" histogram depending on both images. By the midway operation both images undergo a minimal distortion and adopt exactly the same histogram. Thus we shall define the contrast invariant $RMSE$ ($RMSE^{CI}$) by

$$RMSE^{CI} = RMSE(u_{est_{mid(u,u_{est})}}, u_{mid(u,u_{est})})$$

where $mid(u, u_{est})(= mid(u_{est}, u))$ is the midway histogram between $[u$ and $\mathbb{u}_{est}$. $u_{mid(u,u_{est})}$ is the image u specified on the $mid(u, u_{est})$ histogram (having an histogram equal to $mid(u, u_{est})$) and $u_{est_{mid(u,u_{est})}}$ is u_{est} specified on $mid(u_{est}, u)$.

1.6 Implementation

The described algorithm has been implemented in C++, the code (resp. its documentation) are available at https://edit.ipol.im/edit/algo/mrt_flutter_shutter/srcflutter_1.tar.gz(resp. https://edit.ipol.im/edit/algo/mrt_flutter_shutter/srcdocflutter.tar.gz). Given an image $u(m,n)$ defined for $m \in \{1, \cdots, M\}$ and $n \in \{1, \cdots, L\}$, a code $(\alpha_k)_{k=0,\cdots,L-1}$, a velocity v, and an SNR level (and using a normalized $\Delta t = 1$), the *analog flutter shutter* camera and its restoration process are simulated as described below. For color images each component is processed independently, the $RMSE$ is averaged over all components. Assuming without loss of generality that the blur is in direction of the image lines, the following algorithm is repeated for each component.

1.7 *Analog flutter shutter*

1. Step 1:

(a) compute $\tilde{u}(m,n)$ the $2D-DFT$ of u: for $m=-\frac{M}{2},\cdots,\frac{M}{2}-1$ and $n=-\frac{N}{2},\cdots,\frac{N}{2}-1$:

$$\tilde{u}(m,n)=\frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}u(k,l)\omega_M^{-km}\omega_N^{-nl}$$

where $\omega_N=\exp\left(\frac{2i\pi}{N}\right)$.

(b) Compute the motion kernel generated by the *flutter shutter* as

i. for each $m=-\frac{M}{2},\cdots,\frac{M}{2}-1$;

ii. for each $n=-\frac{N}{2},\cdots,\frac{N}{2}-1$;

iii. $a(m,n)=\Delta t\frac{sin(\frac{\pi v\Delta tn}{N})}{\frac{\pi v\Delta tn}{N}}\sum_{k=0}^{L-1}\alpha_k e^{-i(\frac{2\pi v\Delta tn}{N})(k+0.5)}$;

(c) compute the product of $\tilde{u}(m,n)$ and $a(m,n)$;

(d) compute the inverse DFT of the previous, store it in $e(m,n)$;

(here $e(m,n)$ is a coefficient of the ideal noiseless image observed, up to the periodization effect.)

(e) crop the result to avoid the periodization effect.

2. Step 2: for each (m,n) simulate the Poisson random variable with intensity $e(m,n)$ and the desired SNR using sections 1.3, 1.4, store it in $o(m,n)$.

(Here $o(m,n)$ contains a simulation of the observed image.)

3. Step 3:

(a) use classic mirror symmetry among the columns obtain $u_s(m,n)$;

(b) compute the $2D-DFT$ of u_s;

(c) compute the motion kernel like in Step 2;

(d) divide the $2D-DFT$ of u_s by the motion kernel;

(e) compute the inverse $2D-DFT$ of the previous;

(f) crop to remove the mirror symmetry;

(here the last operation gives a simulation of the restored knowing $o(m,n)$ and the code.)

4. Step 4: compute the $RMSE$ and $RMSE^{CI}$ after cropping to avoid border effects.

1.8 *Numerical flutter shutter*

1. Step 1:

(a) compute $\tilde{u}(m,n)$ the $2D-DFT$ of u: for $m=-\frac{M}{2},\cdots,\frac{M}{2}-1$ and $n=-\frac{N}{2},\cdots,\frac{N}{2}-1$: $\tilde{u}(m,n)=\frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1}u(k,l)\omega_M^{-km}\omega_N^{-nl}$ where $\omega_N=\exp\left(\frac{2i\pi}{N}\right)$;

(b) compute the elementary noiseless observations $e_k(m,n)$;

i. for each $m = -\frac{M}{2}, \cdots, \frac{M}{2} - 1$;

ii. for each $n = -\frac{N}{2}, \cdots, \frac{N}{2} - 1$;

iii. $c_k(m,n) = \Delta t \frac{sin(\frac{\pi v \Delta t n}{N})}{\frac{\pi v \Delta t n}{N}} e^{-i(\frac{2\pi v \Delta t n}{N})(k+0.5)}$;

(c) compute the product of $\tilde{u}(m,n)$ and $c_k(m,n)$;

(d) compute the inverse $2D - DFT$ of the previous, store it in $e_k(m,n)$;
(here $e_k(m,n)$ contains the ideal noiseless observed up to the periodization effect.)

(e) crop the result to avoid periodization effect.

2. Step 2:

(a) for each (m,n) simulate the Poisson random variable with intensity $e_k(m,n)$ and the desired SNR using section 1.3, 1.4, store it in $o_k(m,n)$;

(b) for each (m,n) the observed $o(m,n) = \sum_{k=0}^{L-1} \alpha_k e_k(m,n)$;
(here $o(m,n)$ contains a simulation of the observed image.)

3. Steps 3-4: identical to the *analog flutter shutter*.

1.9 Usual codes

For comparison purposes the length L of all codes is 52 like in [3, 6, 98, 99]. These strategies are: 1) Snapshot 2) Accumulation 3) Agrawal *et al.* 4) rand code 5) *Sinc* code. The first two codes are standard shutter strategies in classic cameras corresponding to the use of the (1,0,...,0) code (resp. (1,...,1) code).

Agrawal *et al.* code is (1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1) published in [3] (p7) and patented [99].

The rand code is (0.5491, -0.4903, -0.6919, -0.5125, -0.9079, 0.3560, -0.6357, 0.0275, 0.9568, 0.8670, 0.4087, 0.4097, 0.8659, 0.3344, 0.9198, -0.8389, 0.7476, 0.9808, 0.1366, -0.7247, 0.9320, 0.8069, -0.5848, 0.9493, -0.1682, 0.9533, -0.2173, -0.5834, 0.7483, 0.4785, 0.2266, 0.9764, -0.4708, 0.6723, 0.8312, 0.0084, 0.6892, -0.5245, -0.8651, 0.3417, 0.5183, 0.1317, -0.1301, 0.3432, -0.3262, 0.4367, -0.9771, -0.2120, 0.1160, 0.6648, 0.9446, 0.0590) generated from a uniform distribution over $[-1, 1]$.

The M.I.P. (in the sequel M.I.P. stands for *motion-invariant photography* code is (-0.0072629, 0.0075565, -0.0078763, 0.0082229, -0.0086028, 0.0090181, -0.0094769, 0.0099834, -0.010549, 0.01118, -0.011893, 0.012702, -0.01363, 0.014703, -0.015961, 0.017451, -0.01925, 0.021458, -0.024239, 0.027842,

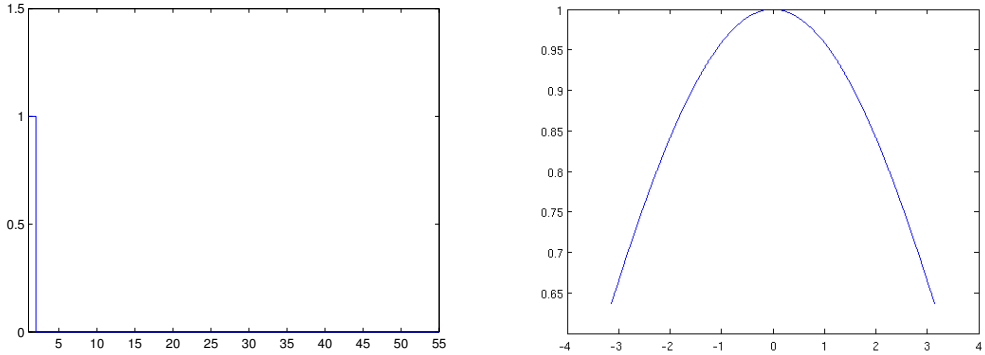

Figure A.1 – The *flutter shutter* gain function for a snapshot (left). The Fourier transform (modulus) of a snapshot (right).

-0.032697, 0.03958, -0.050083, 0.067993, -0.10493, 0.21912,1, 0.29949, 0.40839, 0.22649, 0.29058, 0.18999, 0.23588, 0.16715, 0.20305, 0.15111, 0.18066, 0.13904, 0.16419, 0.12953, 0.15143, 0.12177, 0.14119, 0.11529, 0.13274, 0.10976, 0.12561, 0.10498, 0.11949, 0.10078, 0.11418, 0.097058). This is the best L^2 approximation of the ideal *motion-invariant photography* function function. More precisely, given v, it is a discretization of the $\alpha_{MIP-ideal}(t) = \frac{\mathbb{1}_{]0,\infty[}(t)}{\sqrt{t}}$ function with cutoff frequency equal to πv (given here for $|v|$=1).

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

This is the best L^2 approximation of the ideal gain function. More precisely, given v, it is a discretization of the $sinc(x) = \frac{sin(\pi x)}{\pi x}$ function with cutoff frequency equal to πv (given here for $|v|$=1).

The next figures display the codes on the left, and on the right their Fourier transforms (modulus). From top to bottom: the snapshot (Fig. A.1), the standard blur (Fig. A.2), Agrawal *et al.* code (Fig. A.3), the rand code (Fig. A.4) and the optimal *sinc* code (Fig. A.6). Notice that being non positive the (ideal) *sinc* code cannot be used with an *analog flutter shutter* camera.

1.10 Experiments

Different strategies are here compared on the two following images using the *numerical flutter shutter*, which gives a better SNR than an *analog flutter shutter*. Without loss of generality all results are

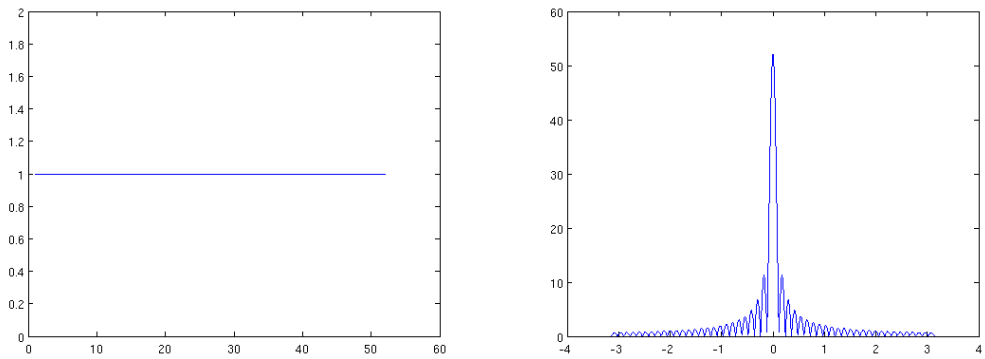

Figure A.2 – The *flutter shutter* gain function for the accumulation (left). The Fourier transform (modulus) of the accumulation (right), invertible only when $Lv\Delta t < 2$.

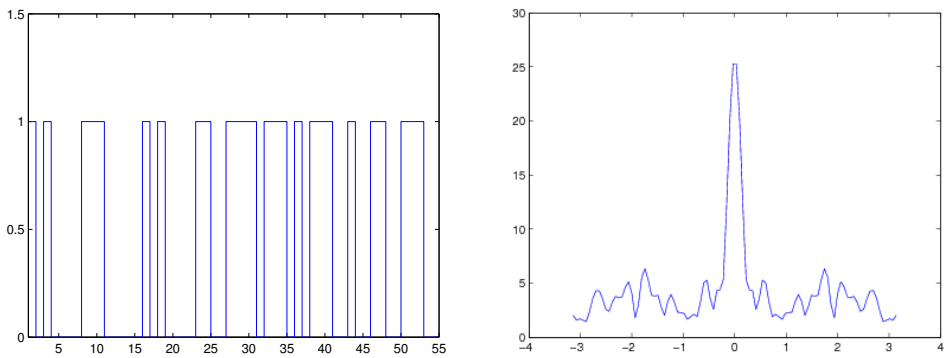

Figure A.3 – The binary *flutter shutter* gain function for the optimized Agrawal *et al.* code (left). The Fourier transform (modulus) of the Agrawal *et al.* code (right).

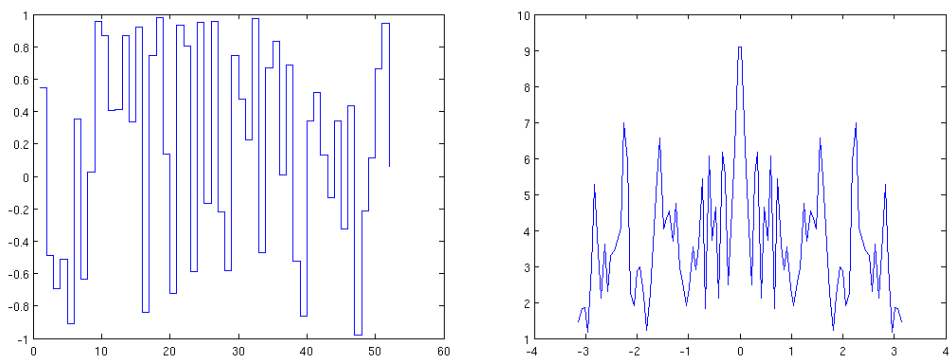

Figure A.4 – The *flutter shutter* gain function function for the rand code (left). The Fourier transform (modulus) of the rand code (right). We shall see in the sequel that despite the lack of optimization this code performs better than the Agrawal *et al.* code.

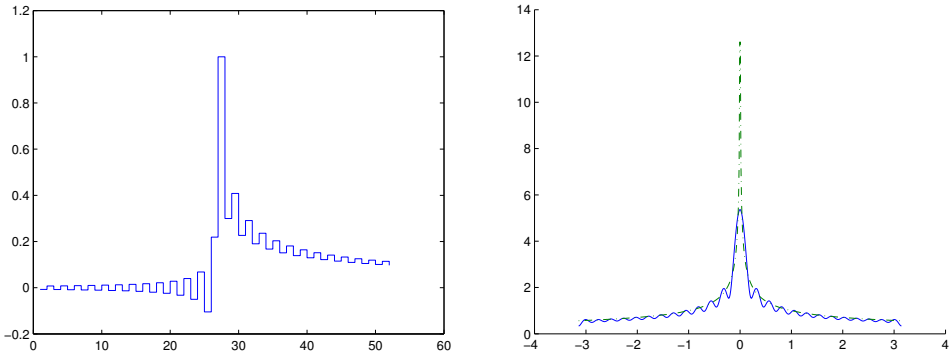

Figure A.5 – The *flutter shutter* gain function for a the M.I.P. (*motion-invariant photography*) code (left). On the right: the Fourier transforms (modulus) of the *motion-invariant photography* code (in bold) and of the ideal *motion-invariant photography* function $\hat{\alpha}_{MIP-ideal}$ (dash dots line style). As predicted the proposed approximation is close to the optimum. As it is stated in [69] this apparatus performs better than the Agrawal *et al.* code. However, it may be noticed that the both the ideal *motion-invariant photography* function and its piecewise constant approximation are far from the ideal *flutter shutter* function coming from a *sinc* (see A.6). Thus, the SNR of the recovered image is small compared to the best snapshot (see 1.10). This fact shall not surprise the reader, the constant acceleration apparatus was found by searching the best strategy among camera motions. Thus, the degrees of freedom of the *motion-invariant photography* are smaller than the *numerical flutter shutter*.

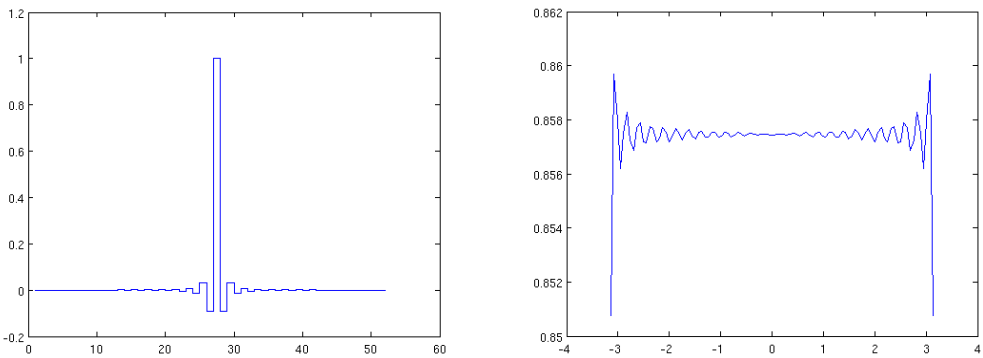

Figure A.6 – The *flutter shutter* gain function for the sinc-code (left). The Fourier transform (modulus) of the $sinc - code$ (right), approximating the Fourier transform of the ideal gain function.

Figure A.7 – The boat test image (left). The house test image (right)

given here for a normalized velocity $v = 1$ and a SNR equal to 100 for the grey-level 100. On the boat image on the left of Fig. A.7 are applied successively: a snapshot (Fig. A.8), a classic motion blur (Fig. A.9), Agrawal *et al.* code (Fig. A.10), a rand code (Fig. A.11), the *motion-invariant photography* code (Fig. A.12), and the optimal numerical *sinc* code (Fig. A.13).

On the house image on the right of Fig. A.7 are applied successively: a snapshot (Fig. A.14), a classic motion blur (Fig. A.15), Agrawal *et al.* code (Fig. A.16), a rand code (Fig. A.17), and the *motion-invariant photography* code (Fig. A.18), and the optimal numerical flutter *sinc* code (Fig. A.19).

Figure A.8 – Snapshot : observed image (left). The blur interval length is equal to 1 pixel here. Reconstructed image ($RMSE = 1.47$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

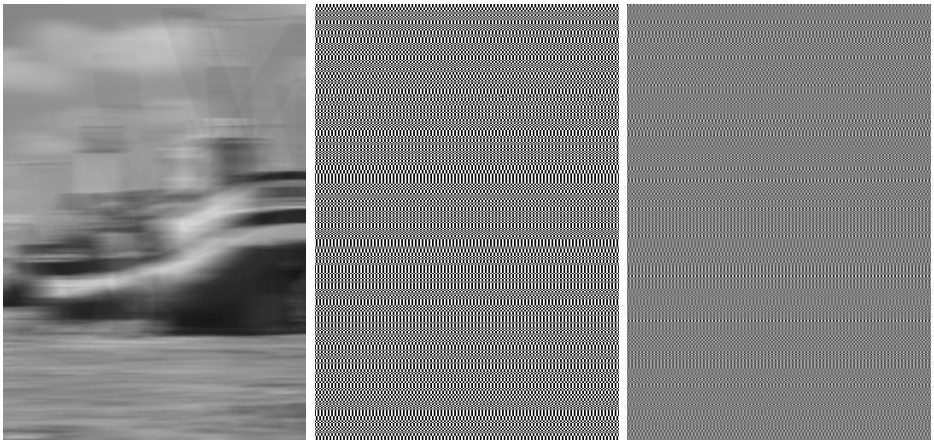

Figure A.9 – Accumulation : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.10 – Agrawal *et al.* code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 2.54$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.11 – Rand code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 2.25$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.12 – M.I.P. code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 2.31$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.13 – *Sinc* code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 1.46$) (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.

Figure A.14 – Snapshot : observed image (left). The blur interval length is equal to 1 pixel here. Reconstructed image ($RMSE = 1.34$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

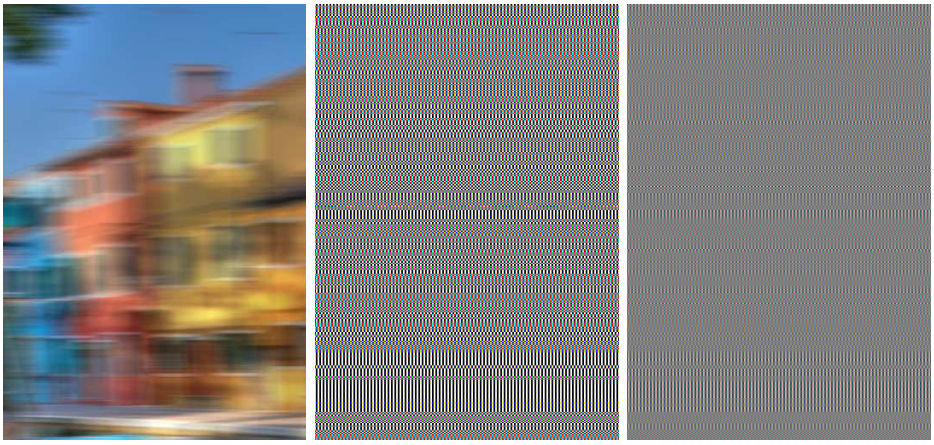

Figure A.15 – Accumulation : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.16 – Agrawal *et al.* code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 2.34$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.17 – Rand code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 2.01$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

Figure A.18 – M.I.P. code : observed image (left). The blur interval length is equal to 52 pixels here. Reconstructed image ($RMSE = 2.11$) (middle). Residual noise (difference between ground truth and reconstructed, dynamic normalized on $[0, 255]$ by an affine contrast change).

2 A reverse engineering of classic *flutter shutter* codes

Applying Algorithm 1 (page 57) with the variant given by (VII.5), to the Agrawal *et al.* code gives (using a normalized $\Delta t = 1$) the probability density of Fig A.21. This distribution can be understood in two different ways. It either means that there is a high probability that the scene is still and then otherwise uniformly distributed motions occur on a certain interval of velocities. But this is an unlikely model for a camera motion. Instead, it is a quite natural model given the kind of motions considered in the Agrawal *et al.* codes. Indeed the velocity histogram of Fig A.21 considers that there is a fixed background with a large area and then some moving objects whose velocity probability distribution is indeed close to uniform in a broad interval. All proposed codes [3, 6, 80, 98, 99] are equally optimal for specific probability densities of velocities. Another example is the the McCloskey code [78], treated in Fig A.22.

The same scheme can be applied to the standard snapshot code $(1, 0, ..., 0)$ to estimate the underlying probability density of Fig A.22.

Notice that we cannot apply such a scheme on $\alpha_{MIP-ideal}(\xi)$. Indeed $w(\xi) = |\hat{\alpha}_{MIP-ideal}|^4 = \frac{1}{x^2}$ thus $-\frac{1}{2}xw'(x) = 2\frac{1}{x^2}$ which is not $L^1(\mathbb{R})$. This means that the *motion-invariant photography* does not comes from a probability density. The same result holds for (the band limited version used to be able to compute a code) $\alpha_{MIP-ideal}(\xi)\mathbb{1}_{[-\pi|v_{max}|\Delta t,\pi|v_{max}|\Delta t]}(\xi)$, because $\frac{1}{x^2}$ is not locally integrable near zero either. This means that to each code of finite length L will correspond a different probability density. These densities does not converge to a probability density when $L \to \infty$.

Figure A.19 – *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.

Figure A.20 – Image credits: The boat (standard test image). The house http://www.flickr.com/photos/setaou/2162752903/Hervé Bry, Flickr CC-BY-NC-SA)

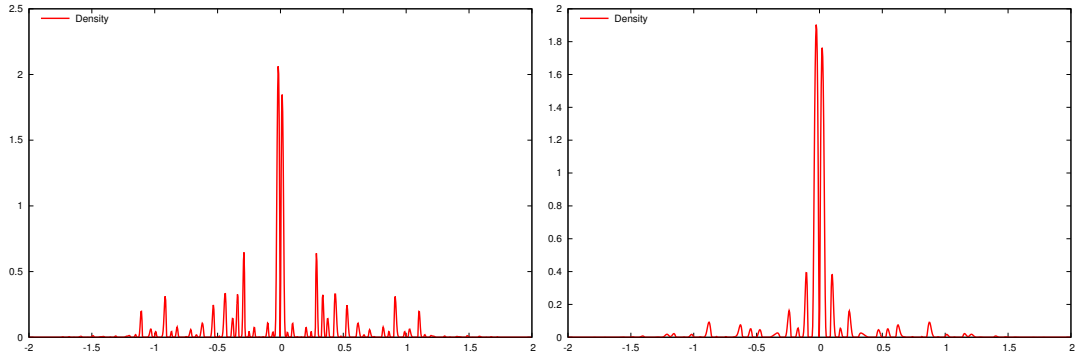

Figure A.21 – The probability densities associated with Agrawal *et al.* codes: x-axis motion (in signed pixels), y-axis the *Log* of the probability $(Log(1+\rho(v)))$. On the left: the code published in [3]. On the right: the code published in [6]. It corresponds to an attempt to optimize both the SNR and the PSF estimation. Notice that both probability densities are nonzero even for large motion blurs.

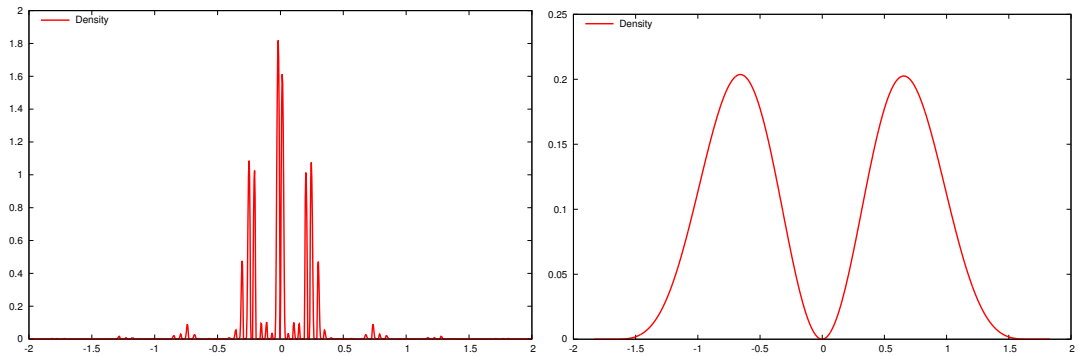

Figure A.22 – On the left: the probability density associated with the McCloskey code [78]: x-axis motion (in signed pixels), y-axis probability *Log* scaled $(Log(1+\rho(v)))$. On the right: the probability density of velocities associated with the best snapshot code, a "1.0909 pixel blur integration". On the x-axis, the velocities (in signed pixels), on the y-axis the corresponding probability density. This snapshot is optimized *a priori* for objects moving at velocity $|v|$. This bimodal density is natural for a traffic surveillance camera.

3 Simulations on optimized codes

The goal of this section is to explore numerically several natural models of motion, give the corresponding optimized codes and to compare their efficiency to the best snapshot on average. This corresponds to the comparison on an equal footing of two alternative solutions: the snapshot and the *numerical flutter shutter*. Both technical solutions are optimized on average (see chapter VII) given a model $\rho(v)$ on the velocities. We begin in section 3.1 by providing the algorithm computing optimal *flutter shutter codes*. The comparison between the *numerical flutter shutter* and the snapshot is made in terms of the ratio $R(v)$ of their $SNR^{spectral-averaged}$ (II.9) as done in Cor. 1.7 (chapter VI). Indeed from (III.16) and (V.6) we have

$$R(v) = \frac{SNR^{spectral-average}(flutter)}{SNR^{spectral-average}(snapshot)} = \sqrt{\frac{\int_{-\pi}^{\pi} \frac{1}{\Delta t^* |2 \frac{sin(\frac{\xi v \Delta t^*}{2})}{\xi v \Delta t^*}|^2} d\xi}{\int_{-\pi}^{\pi} \frac{||\alpha||_{L^2}^2}{|\hat{\alpha}(\xi v)|^2} d\xi}} \tag{VIII.1}$$

where v is in the support of ρ, Δt^* is the exposure time of the best snapshot on average defined in chapter VII section 3, and $\Delta t (\neq \Delta t^*$ in general) is the time-step of the *flutter shutter*. The average gain of the *flutter shutter* in terms of SNR is defined by

$$\int R(v)\rho(v)dv \tag{VIII.2}$$

and the associated standard deviation (risk) is

$$\sqrt{\int |R(v) - \int R(u)\rho(u)du|^2 \rho(v)dv} \tag{VIII.3}$$

Notice that the parameters of a *flutter shutter* are: L the length of the code $(\alpha_k)_{k\in\{0,\cdots,L-1\}}$, the motion model $\rho(v)$, the time step Δt. The parameter for the best snapshot is only the motion model $\rho(v)$. The best snapshot provides Δt^* the optimal aperture time for a standard camera. To ease the comparison with the Agrawal *et al.* code, all experiments are made with $L = 52$ which is the length of this code. Three motion models are considered: truncated Gaussian, uniform and an handcrafted trimodal motion model. The time step of the *flutter shutter* is chosen such that the total exposure time of the *flutter shutter* device $L\Delta t$ is an integer factor c of Δt^*, $L\Delta t = c\Delta t^*$. This permits an easy comparison of the considered *flutter shutter* with the theoretical potential gain $\sqrt{c}$ coming from an increased time aperture. In the examples thereafter $c = 1$, $C = 2$, $c = 5$ and $c = 10$. Thus, the *flutter shutter* camera integrates five (resp. ten) times longer than the optimal snapshot. In other words the support of the *flutter shutter function* $\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t]}(t)$ is five (resp. ten) times larger than the support of the snapshot function $\mathbb{1}_{[0,\Delta t]}(t)$.

All the presented codes of Figs. A.23, A.24, A.26, A.27 and A.29 are the best possible choice for α_k such that $\alpha(t)$ with $\Delta t = \frac{c\Delta t^*}{L}$ approximates the ideal $\alpha^*(t)$ (defined in chapter VII Thm. 1.1). From these results we shall compare the two strategies using a finite exposure time (mandatory to have a practical solution). We have seen that if the velocity is known, namely $\rho(v) = \delta_v$, the best code comes from a zoomed *sinc* function. Moreover, the best snapshot has an exposure time Δt tuned so that $|v|\Delta t \approx 1.0909$ (see chapter V) and the *flutter shutter* increases the SNR by a 1.17 factor approximately (see chapter VI). Notice that, from chapter VII Thm. 3.3, *ceteris paribus* a scale change of the motion model results in a scale change of the $w(\xi)$ function and a zoom of the code. From chapter VII Thm. 3.1 an equivalent result holds for the best snapshot. Thus, without loss of generality comparisons can be made on normalized motion models. The weight function w (defined

in (VII.2)) being concave, one can deduce that the modulus of the Fourier transform of the *ideal flutter shutter function* (section VII Thm. 1.1) is also concave. Thus all Fourier transforms of optimized codes (explicitly given in Figs. A.24, A.27 and A.29) inherit this low-pass behavior -more or less peaked at the null frequency- depending on the motion model ρ and on c, the exposure time factor assumed for the optimization.

3.1 Algorithm

The algorithm producing optimal codes and comparisons with the snapshot is first described in a continuous version in Algorithm 2. It computes integrals, numerical details for integral computation are given in 3. Roughly the algorithm of the demo consists of six steps, the input parameters are 1) a motion model ρ (Gaussian $\mathcal{N}(0,\sigma)$ truncated at 4σ, uniform or handcrafted), a code length L, an exposure time factor c. The exposure time factor permits to set the support of the *flutter shutter function* as a multiple of the support of the best snapshot.

Numerical evaluation of an integral by the Simpson method

Let f be the function to integrate on an interval $[a,b]$. Assume that $[a,b]$ is split in n even subintervals. The Simpson method consists [16] in the approximation

$$\int_a^b f(x)dx \approx \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{\frac{n}{2}-1} f(x_{2j}) + 4\sum_{j=1}^{\frac{n}{2}} f(x_{2j-1}) + f(x_n)\right)$$

where $h = \frac{b-a}{n}$ and $x_j = a + jh$ for $j \in 0, \cdot, n-1$ (thus $x_0 = a$ and $x_n = b$). This yields to the following pseudo code given in Algorithm 3, provided $a < b$, a function f to integrate on the interval $[a,b]$, and a precision parameter ϵ.

3.2 Optimal *flutter shutter codes*

The goal of this section is to explore numerically several natural motion models, to give the corresponding optimized codes and to compare their efficiency to the best snapshot. This corresponds to the comparison on an equal footing of two alternative solutions: the snapshot and the *numerical flutter shutter*.

Optimal codes, Gaussian motion model

The optimal codes for a truncated Gaussian $\mathcal{N}(0,\frac{1}{4})$ motion model ($\rho(v) = \mathbb{1}_{[-1,1]}e^{\frac{-v^2}{2(\frac{1}{4})^2}}$) are explicitly given on the left side of Fig A.24. The Fourier transforms of the corresponding *flutter shutter functions* are given on the right side of Fig A.24. On Fig A.24 between the two plots at the top and the two at the bottom, the discretization step Δt of the *flutter shutter functions* $\alpha(t)$ changes and the support of

input : motion model ρ (to be chose among a list), code length L, exposure time factor c
output: Text files containing: the code coefficients $(a_k)_{k\in\{-\frac{L}{2},\cdots,\frac{L}{2}-1\}}$, Fourier transform of α, ideal Fourier transform w, relative efficiency $R(v)$, relative efficiency on average and associated risk.

1. compute Δt^* the exposure time of the best snapshot on average

$$\Delta t^* = argmin_{\Delta t}\left(\int\int_{-\pi}^{\pi}\frac{\xi^2}{sin^2(\frac{\xi v\Delta t}{2})}\frac{v^2\Delta t}{4}\rho(v)d\xi dv\right)$$

   the minimum is calculated by scanning on values of $\Delta t \in [0.001|v_{max}|, 1.999|v_{max}|]$ at a precision of $0.001|v_{max}|$. The numerical evaluation of the integral is detailed in Algorithm 3, the precision parameter is fixed at $\epsilon = 0.001$. This is implemented in *best_snapshot.cpp*;

2. compute $\Delta t = \frac{c\Delta t^*}{L}$, the time step of the *flutter shutter*;

3. compute the code coefficients $\alpha_k = \frac{1}{2\pi}\int_{-\pi|v_{max}|}^{\pi|v_{max}|}\frac{\sqrt[4]{w(s)}cos(2(k+1)s)}{sinc(\frac{s}{2\pi})}ds$ for $k \in \{-round(\frac{L}{2}), \cdots, roudn(\frac{L}{2})-1\}$ where $w(\xi) := \int_{\mathbb{R}}\frac{\rho(v)\mathbb{1}_{[-|v|\pi,|v|\pi]}(\xi)}{|v|}dv$. The numerical evaluation of the integral is detailed in Algorithm 3, the precision parameter is fixed at $\epsilon = 0.001$. Depending on the model this step is implemented in *flutter_optimizer_{motion model name}.cpp* files;

4. write the code $(a_k)_k$, the (modulus) of the Fourier transform of the *flutter shutter gain function* and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ in the corresponding text files;

5. write the relative efficiency $R(v) = \frac{SNR(flutter)}{SNR(snapshot)} = \sqrt{\frac{\int_{-\pi}^{\pi}\frac{1}{\Delta t^*|2\frac{sin(\frac{\xi v\Delta t^*}{2})}{\xi v\Delta t^*}|^2}d\xi}{\int_{-\pi}^{\pi}\frac{||\alpha||_{L^2}^2}{|\hat{\alpha}(\xi v)|^2}d\xi}}$ in the appropriate text file. Integrals are evaluated by Riemann sums on 1000 points. This is implemented in *code_comparison.cpp*;

6. write the relative efficiency $\int R(v)\rho(v)dv$ on average and the associated risk $\sqrt{\int|R(v)-\int R(u)\rho(u)du|^2\rho(v)dv}$. Integrals are evaluated by Riemann sums on 1000 points. This is implemented in *code_comparison.cpp*.

Algorithm 2: Pseudo-code of the algorithm computing optimized code.

the *flutter shutter function* doubles. The green curves showing $\sqrt[4]{w(\xi)}$ remain the same as the motion model ρ is unchanged. Notice that the approximation is slightly better for the larger exposure factor $c = 10$ (bigger support of $\alpha(t)$). Indeed, since $w(\xi)$ is compactly supported, its ideal *continuous flutter shutter* has an infinite support.

The Fig A.25 provides the comparison with the snapshot in terms of SNR, for any velocity v in the support of the motion model $\rho(v)$. On the left side, for an exposure factor $c = 5$ and on the right side for $c = 10$. The red curves show the efficiency $R(v)$ function defined by (VIII.1). The efficiency function $R(v)$ directly measures the performance of the considered *flutter shutter* camera

input : a, b, ϵ (precision), function f
output: tn the numerical approximation of $\int_a^b f(x)dx$
initialization: $h \leftarrow \frac{b-a}{2}$;
$s1 \leftarrow f(a) + f(b)$;
$s2 \leftarrow 0$;
$s4 \leftarrow f(a+h)$;
$tn \leftarrow h\frac{s1+4s4}{3}$;
$zh \leftarrow 2$;
while $|ta - tn| \leq \epsilon|tn|$ **do**
  $ta \leftarrow tn$;
  $zh \leftarrow 2zh$;
  $h \leftarrow \frac{h}{2}$;
  $s2 \leftarrow s2 + s4$;
  $s4 \leftarrow 0$;
  $j \leftarrow 1$;
  while $j > zh$ **do**
    $s4 \leftarrow s4 + f(a + jh)$;
    $j \leftarrow j + 2$;
  end
  $tn \leftarrow h\frac{s1+2s2+4s4}{3}$.
end
output : tn

Algorithm 3: Integral evaluation by the Simpson method.

design in terms of reduction of the $RMSE$ compared to the best snapshot. The dotted blue curve provides the motion model density $\rho(v)$. The optimization permits to concentrate the gain in SNR on most probable velocities, while for higher but less likely velocities v the optimized *flutter shutter* performs worse than the snapshot. The green line shows the average of $R(v)$ taking the motion model ρ into account (defined by (VIII.2)). Notice that the optimal asymptotic bound of chapter VI Cor. 1.7 is beaten, by approximately 50%. Table A.3 provides both the relative efficiency (VIII.2) and its associated risk defined by (VIII.3). It permits to measure "how risky" the optimization on average is.

Exposure time factor	1	2	5	10
Code length L	52	52	52	52
Average gain (VIII.2)	0.9899	1.1737	1.2554	1.2705
Std-dev (VIII.3)	0.0003	0.0062	0.0290	0.0392

Table A.2 – Average gain of the optimized *flutter shutter* compared to the snapshot, assuming a truncated Gaussian density for the velocities. As guessed from Fig A.28 the gain is substantial and the increase is approximately 50% higher than the asymptotic 1.17 bound given in chapter VI Cor. 1.7 for the particular case of $\rho(v) = \delta_{v_0}(v)$.

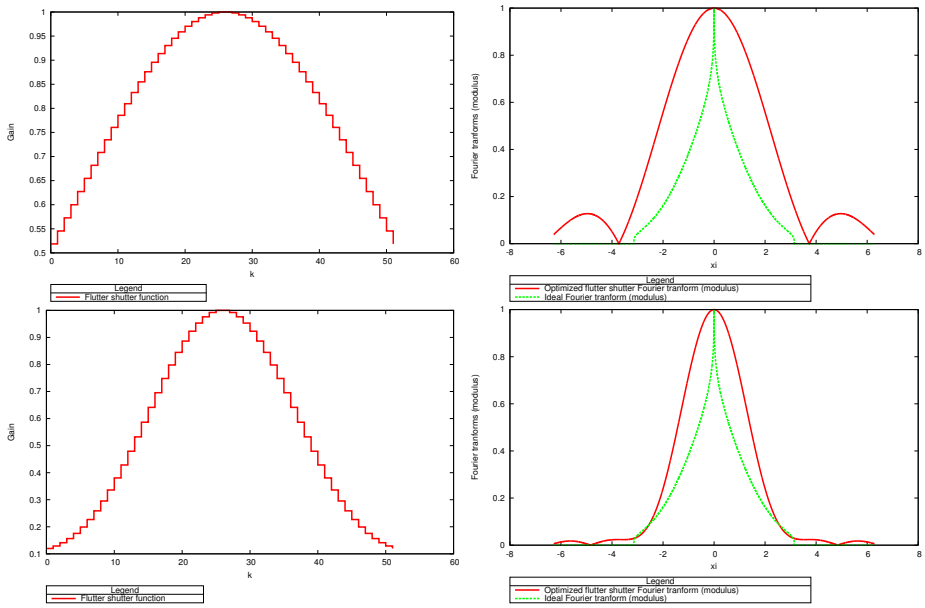

Figure A.23 – Codes obtained for a truncated Gaussian velocity density. On the left, top and bottom: the code α_k, using an exposure time 1 and 2 times larger than for the best snapshot. On the right (in red) the modulus of their corresponding Fourier transform, and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ (VII.2) in 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 function*.

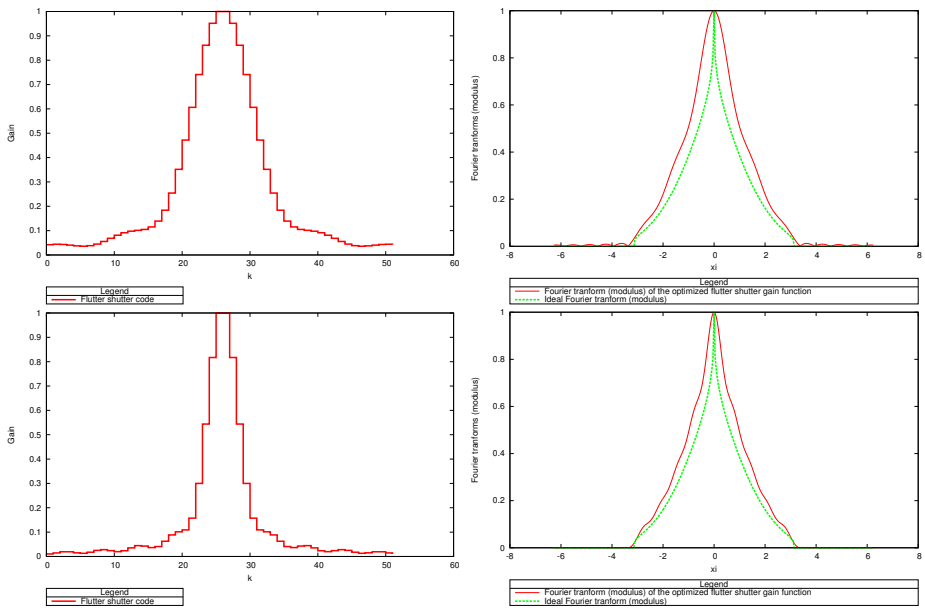

Figure A.24 – Codes obtained for a truncated Gaussian velocity density. On the left, top and bottom: the code α_k, using an exposure time 5 and 10 times larger than for the best snapshot. On the right (in red) the modulus of their corresponding Fourier transform, and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ (VII.2) in 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 function*.

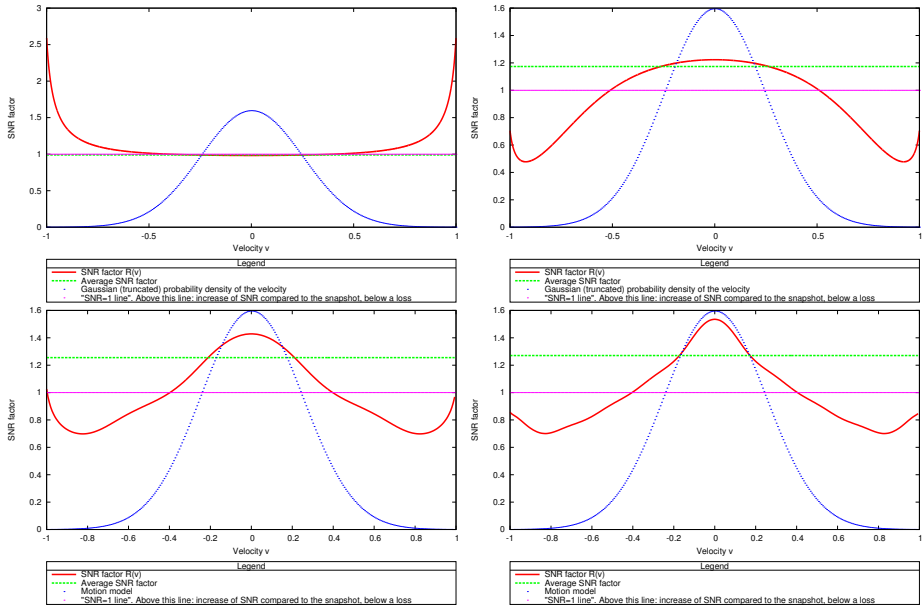

Figure A.25 – In red: the ratio of the SNR $R(v)$ of the optimal *flutter shutter* (truncated Gaussian motion model) to the optimal snapshot, following formula (VIII.1) for exposure times 1, 2, 5 and 10. The dotted blue curve represents the probability density $\rho(v)$ of the considered motion model. In green, its average taking the motion model $\rho(v)$ into account. Notice that the optimization permits to concentrate the gain on most probable velocities, as expected. For less likely velocities, the best *flutter shutter* can perform worse that the snapshot. Nevertheless, on average the gain is substantial and the optimal bound of chapter VI Cor. 1.7 is beaten.

Optimal codes, uniform motion model

The optimal codes for an uniform $\mathcal{U}[-1,1]$ motion model are explicitly given on the left side of Fig A.27. The Fourier transforms of the corresponding *flutter shutter functions* given on the right side of Fig A.27. On Fig A.27 between the two plots at the top and the two at the bottom, the discretization step Δt of the *flutter shutter gain function* $\alpha(t)$ changes. The green curves showing the ideal Fourier transform $\sqrt[4]{w(\xi)}$ remain the same as the motion model ρ is unchanged. Notice that the approximation is slightly better for the larger exposure factor $c = 10$ (bigger support of $\alpha(t)$). The Fig A.28 provides the comparison with the snapshot in terms of SNR for any velocity v in the support of the motion model $\rho(v)$. On the left side, for an exposure factor $c = 5$ and on the right for $c = 10$. The red curves show the efficiency $R(v)$ function defined by (VIII.1). The efficiency function $R(v)$ directly measures the performance of the considered *flutter shutter* camera design in terms of reduction of the $RMSE$ compared to the best snapshot.

The dotted blue curve provides the motion model density $\rho(v)$. The green line shows the average efficiency taking the motion model ρ into account (defined by (VIII.2)). Table A.3 provides both the relative efficiency (VIII.2) and its associated risk defined by (VIII.3).

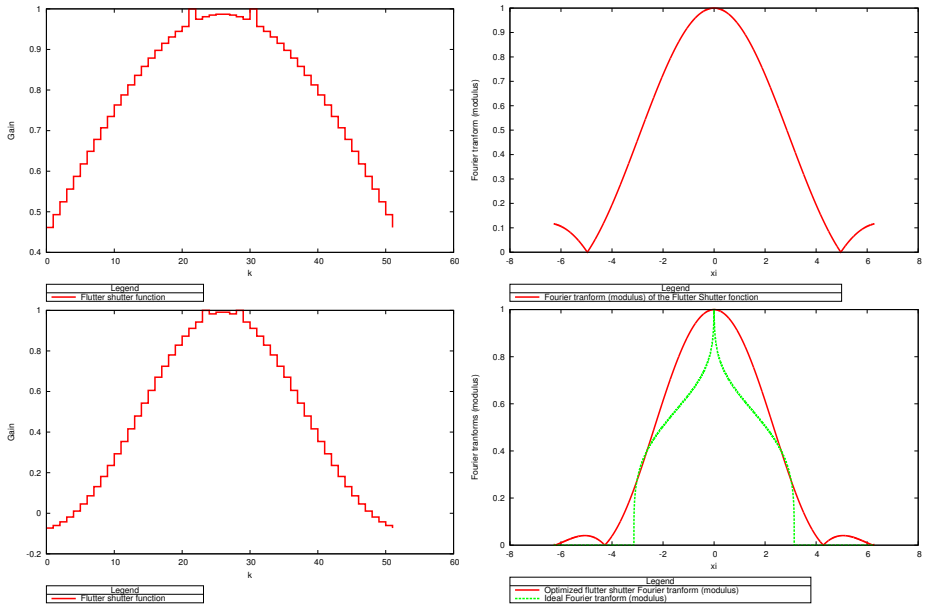

Figure A.26 – Codes α_k obtained assuming a uniform density for the velocities. . On the left from top to bottom: the code α_k for a uniform velocity distribution using an exposure time 1 and 2 times larger than for the best snapshot. On the right the corresponding Fourier transform (modulus) of the code in red and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ (VII.2) in 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 function*.

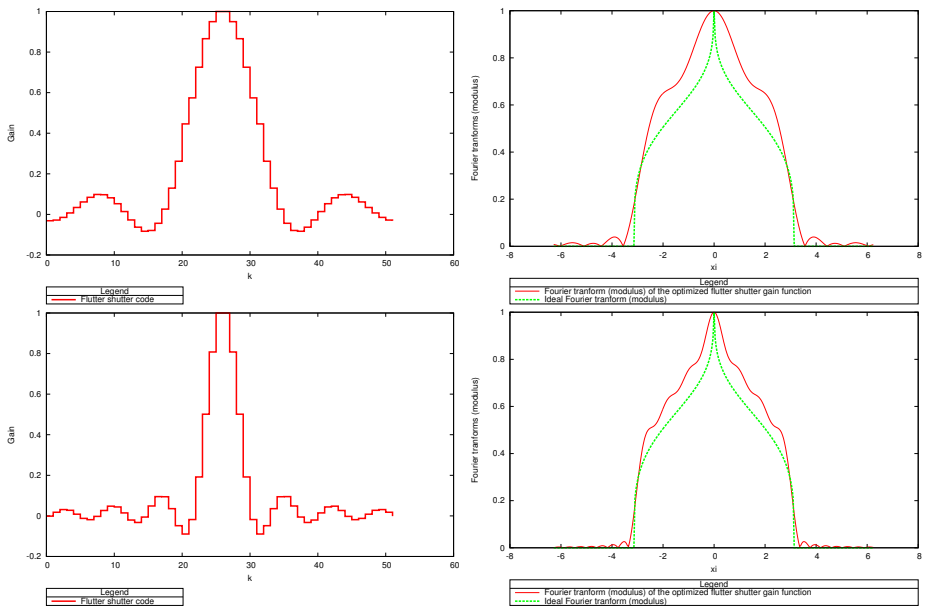

Figure A.27 – Codes α_k obtained assuming a uniform density for the velocities. . On the left from top to bottom: the code α_k for a uniform velocity distribution using an exposure time 5 and 10 times larger than for the best snapshot. On the right the corresponding Fourier transform (modulus) of the code in red and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ (VII.2) in 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 function*.

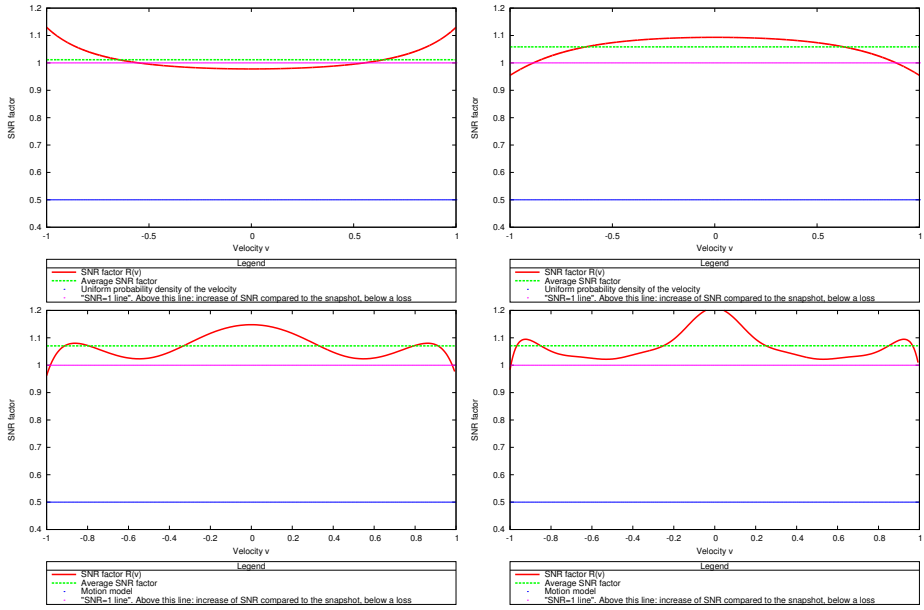

Figure A.28 – The ratio of SNR's between the optimized snapshot and the *flutter shutter* $R(v)$ (VIII.1), in red. From top left to bottom right for an increased exposure time of 1, 2, 5 and 10. The dotted blue line represents the probability density $\rho(v)$ of the motion model. In green, its average taking the motion model $\rho(v)$ into account. Unfortunately the gain remains quite small.

Exposure time factor	1	2	5	10
Code length L	52	52	52	52
Average gain (VIII.2)	1.01162	1.0584	1.0798	1.0846
Std-dev (VIII.3)	0.0015	0.00142	0.0036	0.0036

Table A.3 – Average gain of the optimized *flutter shutter* compared to the snapshot, assuming a uniform density for the velocities. For the exposure time factor of 1, the gain comes from a better deconvolution kernel. As could already be guessed from Fig A.28, this gain is not significant.

Optimal codes, handcrafted motion model

The optimal codes for the trimodal motion model $\rho(v) = 0.99\delta_0(v) + 0.005\delta_{15}(v) + 0.005\delta_{-15}(v)$ are explicitly given on the left side of Fig A.29. This model may seem far fetched but can be relevant for a camera observing a highway. Notice that the optimal *flutter shutter* assuming $\rho(v) = 0.99\delta_0(v) + 0.005\delta_{15}(v) + 0.005\delta_{-15}(v)$ and $\rho(v) = 0.99\delta_0(v) + 0.01\delta_{15}(v)$ are equals. The Fourier transforms of the corresponding *flutter shutter functions* are given on the right side of Fig A.29. On Fig A.29 between the two plots at the top and the two at the bottom, the discretization step Δt of the *flutter shutter function* $\alpha(t)$ changes. The green curves showing $\sqrt[4]{w(\xi)}$ remain the same as the motion model ρ is unchanged. Notice that the approximation is slightly better for the larger exposure factor $c = 10$ (bigger support of $\alpha(t)$). Indeed, since $w(\xi)$ is compactly supported, its ideal *continuous flutter shutter* has an infinite support. The Fig A.30 provides the comparison with the snapshot in terms of SNR, for any velocity v in the support of the motion model $\rho(v)$. On the left side, for an exposure factor

$c = 9$ and on the right side for $c = 25$. The red curves show the efficiency $R(v)$ function defined by (VIII.1). The efficiency function $R(v)$ directly measures the performance of the considered *flutter shutter* camera design in terms of reduction of the $RMSE$ compared to the best snapshot. The blue dots show the motion model density $\rho(v)$. The optimization permits to concentrate the gain in SNR on most probable velocities, while for higher but less likely velocities v the optimized *flutter shutter* performs worse than the snapshot. The green line shows the average of $R(v)$ taking the motion model ρ into account (defined by (VIII.2)). Notice that the optimal asymptotic bound of chapter VI Cor. 1.7 is beaten. The gain is comparable to the theoretical gain coming from an increased exposure and a still landscape ($\sqrt{9}$ and $\sqrt{25}$). Table A.4 provides both the relative efficiency (VIII.2) and its associated risk defined by (VIII.3). It permits to measure "how risky" the optimization on average is.

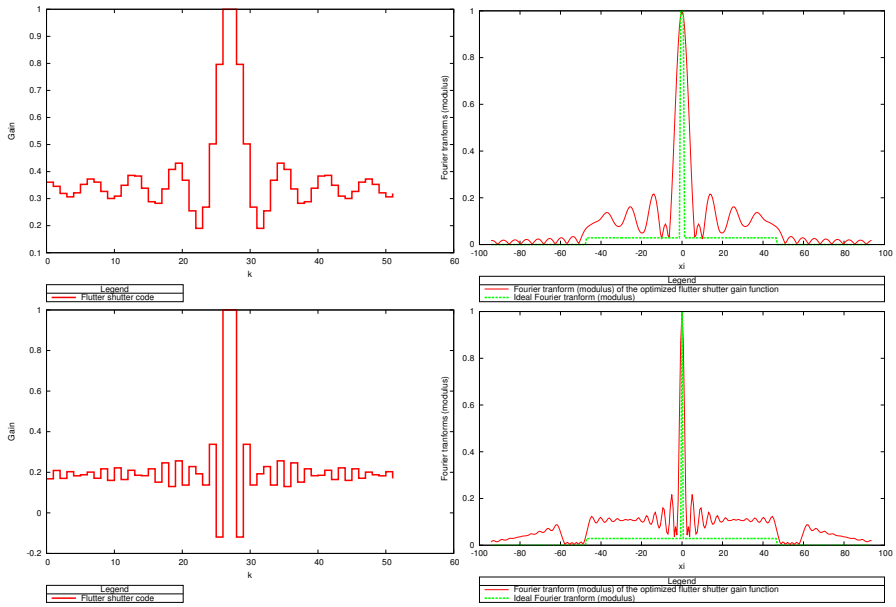

Figure A.29 – Codes obtained assuming a trimodal density for the velocities of the form $\rho(v) = 0.99\delta_0(v) + 0.005\delta_{15}(v) + 0.005\delta_{-15}(v)$. On the left from top to bottom: the code for a trimodal velocity distribution measured (x-axis: k, y-axis: the gain α_k) using an increased exposure time factor of 9 and 25. On the right the corresponding Fourier transform (modulus) of the code in red and the ideal Fourier transform $\sqrt[4]{w(\xi)}$ (VII.2) in green. These results permit to visualize the effect of the optimization.

Exposure time factor	9	25
Code length L	52	52
Average gain (VIII.2)	2.7360	3.8404
Std-dev (VIII.3)	0.0437	0.0766

Table A.4 – Average gain of the optimized *flutter shutter* compared to the snapshot, assuming the handcrafted density for the velocities. The gain is substantial, compared to the asymptotic 1.17 factor given in chapter VI Cor. 1.7 for the particular case of $\rho(v) = \delta_{v_0}(v)$. Notice that in both cases, the gain is close to the theoretical gain coming from an increase exposure time $\sqrt{9}$ and $\sqrt{25}$.

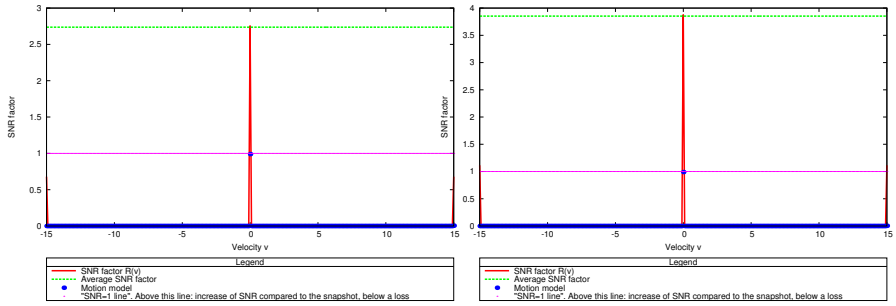

Figure A.30 – The ratio of SNR's between the optimized snapshot and the *flutter shutter* $R(v)$ (VIII.1), in red. On the left : an increased exposure time of 9, on the right an increased exposure time of 25. The blue dots represents the motion model $\rho(v)$ (the Dirac mass. In green, the mean of $R(v)$ (assuming $\rho(v) = 0.99\delta_0(v) + 0.005\delta_{15}(v) + 0.005\delta_{-15}(v)$). Actually this constant emphasizes the effect of the *flutter shutter*: it is not only possible to integrate more photons but it is more efficient. As we can see the *flutter shutter* clearly outperforms the best snapshot.

4 Estimating the kernel

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

The other way is to estimate v from the observed image itself. This is an apparently ill posed problem, since any observed pattern in the image or in its Fourier transform could come either from the landscape or from the code. Nevertheless, codes like Agrawal's, because of the oscillating pattern of their Fourier transform, leave a distinguishable trace in the Fourier transform of the observed image (Fig. A.31).

Omitting the noise term, we are observing the landscape u convolved with $\alpha_{v*}(t) = \frac{1}{v*}\alpha(t/v*)$ for an unknown $v*$. The oscillating pattern of the Fourier transform modulus of the observed image most likely belongs to the α kernel, and not to the landscape (see Fig. A.31). We shall use this observation to extract the α kernel. Notice that even for a *snapshot* an estimation of the blur is needed before deconvolution, and not easier.

The log-modulus of the observed image is

$$log(|\hat{o}bs(\xi)|) = log\left(|\frac{1}{v*}\hat{\alpha}(t/v*)|(\xi)\right) + log\left(|\hat{u}(\xi)|\right).$$

Let us compute $log(|\hat{o}bs(\xi)|) - log\left(|\frac{1}{v}\hat{\alpha}(t/v)|(\xi)\right)$ for a set of values for v. For most of the v values, the result will present all sorts of oscillating patterns, because we are subtracting phase-shifted oscillations (see Fig A.32). On the other hand for v close to v^* the oscillations cancel (see Fig A.32). Thus, for a wrong v, $Log(|\hat{o}bs(\xi)|)$ will have a large total variation. Enforcing the total variation to be finite

using a convolution with a small Gaussian (which also performs a denoising) leads to the following definition of v_{est} estimator of v

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

4.1 Numerical implementation

1. For each line on the observed image in the direction of the blur:
   (a) compute $||\nabla \left((log(|\hat{o}bs|) - log(|\frac{1}{v}\hat{\alpha}(t/v)|) * G \right) (\xi)||_{L^1}$, for different $v \in support(\rho)$;
   (b) keep the one that minimizes the previous;
2. average the estimator of each line to get v_{est}.

4.2 Example : Agrawal's code

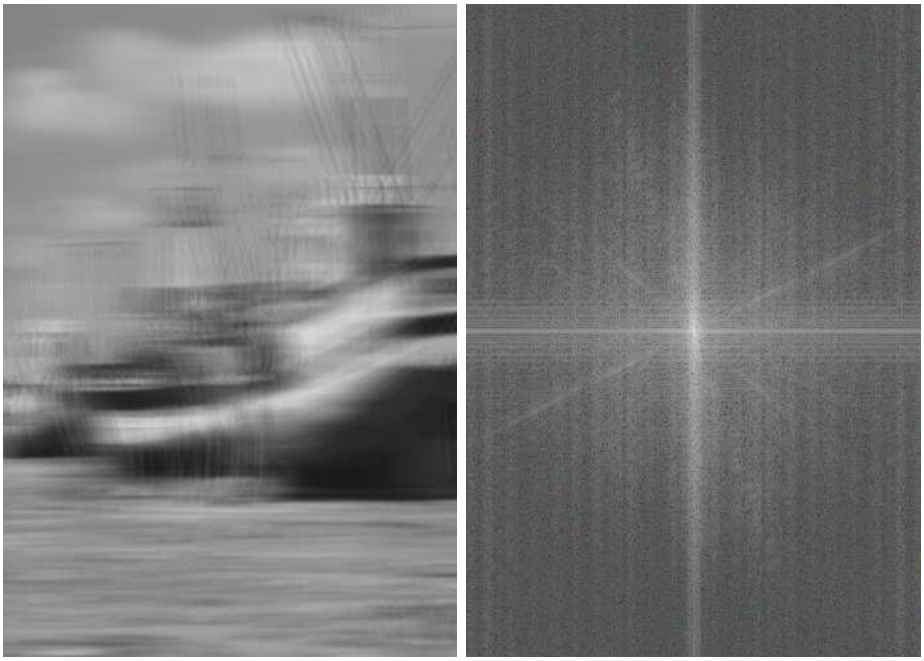

Figure A.31 – Acquired image with the Agrawal *et al.* code and the (log) modulus of its Fourier transform. Notice the vertical pattern trace of the *flutter shutter* code.

Unfortunately optimal codes have to be smooth, hence the previous scheme is not robust in practice. However codes like Agrawal's work. Moreover, trivial constant codes equal to one permit to guess the velocity v and the direction analyzing the Fourier transform of such an acquired image. Hence the reasonable scheme (particularly for aerial or satellite imaging) would be to build simultaneously two

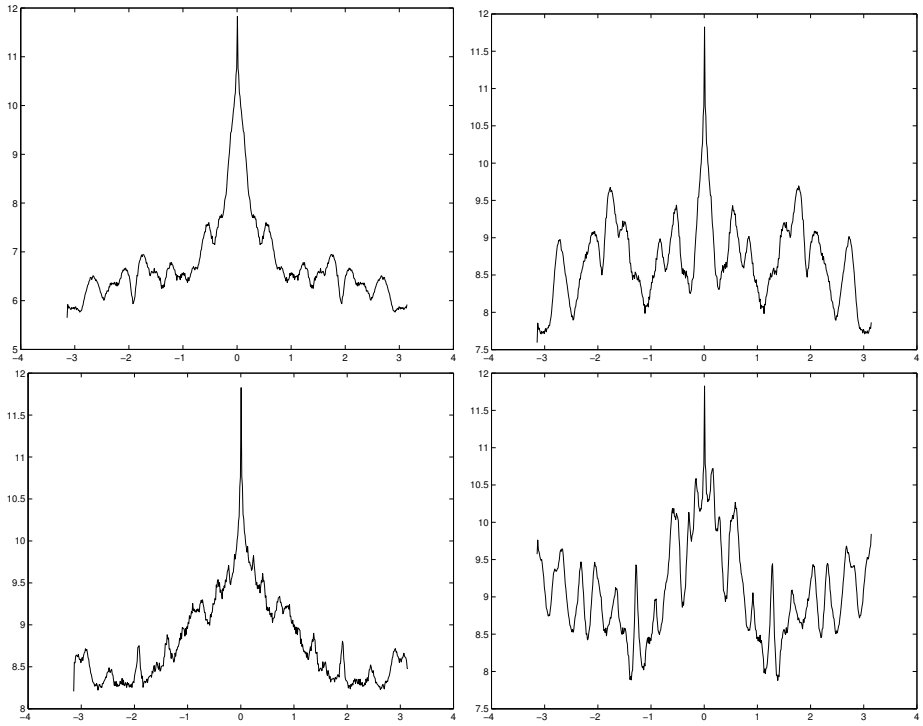

Figure A.32 – The observed image is simulated for a velocity $v = 1$. The graphs display the log of the sum of the (modulus of the) Fourier transform in the direction of the blur minus the log of the (modulus) Fourier transform of the code applied for several velocities v ($log(|\hat{o}bs|) - log(|\frac{1}{v}\hat{\alpha}(t/v)|)$). From top to bottom and left to right for $v = 0$ (TV-norm=31.28), $v = 0.5$ (TV-norm=33), $v = 1$ (TV-norm=25), $v = 1.5$ (TV-norm=44). The estimated velocity is 1, as the velocity which minimizes the TV-norm of $log(|\hat{o}bs|) - log(|\frac{1}{v}\hat{\alpha}(t/v)|)$. In other words deconvolving with wrong kernels leads to an oddly oscillating (modulus) of the Fourier transform which is not credible in a natural image.

images, one with, and one without a *Flutter-Shutter*, use the standard image to estimate accurately v, then use this value of v and to deconvolve the *flutter shutter* image.

Chapter IX

Discussion and Conclusion

This thesis has started by modeling the stochastic photon acquisition of a moving landscape by a light sensor. The model intrinsically contains noise terms due to the Poisson photon emission. This model permits to formalize and analyze a general *flutter shutter* theory which includes the standard photography, the original Agrawal *et al. flutter shutter*, two suggested generalizations of the *flutter shutter* and the Levin *et al. motion-invariant photography*. A formula providing directly the SNR of the sharp recovered images has been given, for all these methods. It also permits to prove what we called the *flutter shutter paradoxes*. A well optimized *flutter shutter* does always beat the traditional camera, even using the same aperture time. And, for an infinite exposure time accumulating many more photons than a snapshot the SNR remains finite (contrarily to the classic still photography). Two kinds of *flutter shutter* setups have been considered: an *analog flutter shutter* and a *numerical flutter shutter* permitting smoother, negative gain-control-functions and leading to the best SNR of the restored images. It also appeared that the *motion-invariant photography* is a particular case of an *analog flutter shutter*. The *motion-invariant photography* has been generalized to the case of unknown velocity direction by using a *numerical flutter shutter*. It is proven that knowing the velocity the best *flutter shutter* code comes from the Fourier series coefficients of a (zoomed) *sinc* function. The SNR raise is of 17% compared to the best snapshot leading to a poor efficiency of such an acquisition system, even if the exposure time is infinite. Also, the thesis assumes that images are acquired at an infinite numerical precision. Of course, taking the limited dynamic range, saturation or quantization effects in account could only degrade this factor. Then, a better mouse trap was set up to increase the efficiency of the *flutter shutter* beyond the 17% bound of one of the *flutter shutter paradoxes*. It was proven that, on average, the SNR can increase significantly provided the probability density of the observed velocities is known *a priori*, or learned if the camera code can be parametrized thereafter. Optimized snapshots have been considered leading to the definition of best blur. It gives the best aperture time to use in a standard camera and can be used, for example, to compute the ideal number of stages of the *time delay and integration* device commonly used in push broom satellites. A reverse engineering of all classic *flutter shutter* codes was performed, it provides the underlying probability density for which they are optimal. The conclusion is that the *flutter shutter* is useful in presence of (unknown) velocities, and can even become SNR efficient depending

on the probability density of the observed velocities.

Limitations, future work

This study assumes that the *analog flutter shutter* is an ideal mathematical object. Therefore, the time responses and/or imperfection of the temporal mask required to implement this apparatus are neglected implying that the provided results are optimistic. Most of the analysis and all the experiments are limited to the case of uniform translational motion. Space-variant blur has strong practical relevance, like camera shake blur, as it behaves rather differently than the uniform blur and could be considered in a future work. Furthermore, the applied nature of the problem calls for experimental validation using a real *flutter shutter* camera.

Part II

Restoration of Infrared Images

Chapter I

Non Uniformity Correction

This chapter proposes a new way to correct for the non uniformity and the noise. The method was initially designed for uncooled infrared-type images but also applies to other kind of camera devices. The method works on static images, needs no registration, no camera motion and no model for the non uniformity. The proposed method uses a hybrid scheme including an automatic locally-adaptive contrast adjustment and a state-of-the-art image denoising method. It permits to correct for a fully non-linear non uniformity and the noise efficiently using only one image. We compared it with total variation on real raw and simulated non uniformity infrared images. The strength of this approach lies in its simplicity, and on its low computational cost. It needs no test-pattern or calibration and produces no "ghost-artifact".

1 Introduction

Infrared imaging has proved to be a very efficient tool in a wide range of industry, medical, and military applications. Infrared cameras are used to measure temperatures, signatures, to perform detection, etc. However, the performance of the imaging system is strongly affected by the random spatial response of each pixel sensor. Under the same illumination the readout of each sensor is different. It leads to a structured noise resulting in a row or line pattern in the images (depending on the readout system). This "noise" is called fixed pattern noise and produces "*non uniformity*" in the observed images. These differences between sensor readout are due to imprecisions in the fabrication process, among other issues [11] and are stronger at longer wavelength such as in infrared imaging [112]. The readout of a pixel sensor is a non linear [11] function of the incoming luminance. The *non uniformity* is a serious practical limitation to both civilian and military applications - as it severely degrades image quality [85] (see Fig. A.1). For uncooled infrared cameras the problem is even worse because the detector response evolves quickly with time. Therefore the correction cannot be done once and for all by the manufacturer. It also means that we need to estimate, for each pixel, a function with little or no model at all and using few or one image to aim a good correction. Indeed the use of numerous images to achieve the correction leads to artifacts –those are called "ghosts artifacts" and are challenging to remove [42, 94, 107, 146, 162, 163]– because of the sensor drift

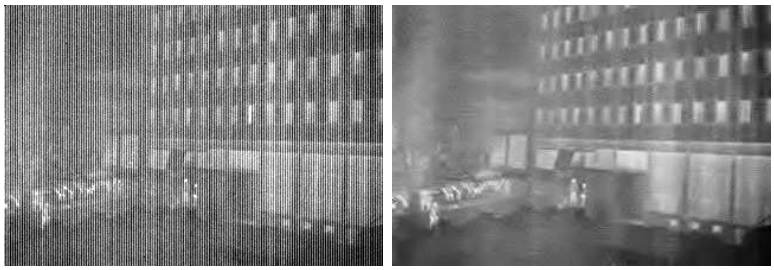

Figure A.1 – On the left : an image (RAW) taken with an infrared camera. The *non uniformity* is so strong that it is hard to distinguish between the noise and the underlying landscape. On such an image performing an identification, matching pattern, etc. is almost hopeless. On the right the same image corrected with the proposed method.

[105, 112, 113]. In other words, for each pixel, the correction at times t_1 and $t_2 \neq t_1$ are different. A correction is so much needed, that in many uncooled infrared cameras a flap closes every 30 seconds to perform a partial calibration [49, 58]. This interrupts the image flow, which is problematic for many applications. Thus, for uncooled infrared cameras a periodic update of the *non uniformity* correction is required.
A good *non uniformity* algorithmic correction is a key factor in ensuring the best image quality and the robustness of the downstream applications such as pattern recognition, image registration, etc.

A review of existing techniques is proposed in section 3. A single image, fully automatic, *non uniformity* correction algorithm is detailed in section 4 and generalized to a locally adaptive variant in section 5. It shows that motion compensation or accumulation algorithms are not necessary to achieve a good image quality. The proposed method can compensate for a fully non linear *non uniformity*, without any parametric model on the *non uniformity* side. It does not require motion, or motion compensation, does not need a test pattern or calibration and does not produce any "ghost artifact". A state of the art denoising algorithm is modified to suit our context.The proposed method is illustrated in section 6 on simulated non linear *non uniformity* in Figs. A.6-A.7, compared with a total variation based method (this method is described in section 6.1) in section 6.2 and evaluated on real raw LWIR images from thermal infrared cameras and **thermal infrared** hyperspectral images in section 6.3.

2 Image acquisition model

An imaging sensor is a device that collects photons and converts them into charges. The standard readout technique of sensors works for each row (or line) independently and consists of transporting charges from the pixels to a counter (which produces the numerical value to be read). Each pixel has its own (and unknown) transfer function response. Furthermore, for each column the counter transfer function is different. The function resulting of the whole chain sensor-counter is not linear [11]. In the

sequel we will assume without loss of generality (w.l.o.g.) that the *non uniformity* comes only from the sensor part. It is the difference between (transfer) functions that produces the *non uniformity* and leads to a structured noise resulting in a row or line pattern in the images. This model applies as soon as the *readout* is parallel among columns. This is the *always* the case with the LWIR arrays or the **thermal infrared hyperspectral** camera considered here. The perturbation model is

$$o(i,j,t) = \phi_{(i,j,t)}\left(u(i,j,t) + \eta(i,j,t)\right), \ \forall (i,j,t) \in \{1,...,N\} \times \{1,...,M\} \times \mathbb{R}_+, \tag{I.1}$$

where

- $(i,j,t) \in \{1,...,N\} \times \{1,...,M\} \times \mathbb{R}_+$ is the pixel at position (i,j) and time $t \geq 0$;
- $u(i,j,t)$ is the ideal (noiseless) landscape value;
- $\eta(i,j,t)$ is a random photon noise;
- $\phi_{(i,j,t)} : \{0,...,255\} \mapsto \{0,...,255\}$ is the contrast change (transfer function) of the pixel sensor at position (i,j) and time $t \geq 0$;
- $o(i,j,t)$ is the observed value at position (i,j) and time $t \geq 0$.

Omitting the noise, $\phi_{(i,j,t)}(x)$ represents the readout of the pixel (i,j) at time t for some incident radiance x. At time t the transfer function of the pixel (i,j) and the transfer function of the counter are contained in $\phi_{(i,j,t)}$ (w.l.o.g). At each pixel sensor (i,j) and time t the photon (Poisson) noise $\eta(i,j,t)$ is sensed (and added to the ideal landscape pixel value $u(i,j,t)$), thus it also undergoes the contrast change function $\phi_{(i,j,t)}$. This means that correcting for the *non uniformity* also whitens the noise and makes it easier to sweep. Consequently, that if a denoising algorithm has to be applied, it must be *after* the *non uniformity* correction (a fact that will be used in section 5). Indeed white noises are easier to remove. The proposed model is consistent with the approximated linear model of the classic literature (in terms of noise). Furthermore, notice that the hypothesis on the domains and images $\{0,...,255\}$ of the contrast changes $\phi_{(i,j,t)}$ is no restriction (but only convenient to use). Consequently, (I.1) models thoroughly the whole acquisition process including the noise that is also modified by the non-uniformity of the sensor array. It permits to deal with a realistic *non linear* sensor response [11] contrarily to the classic linear (gain/offset) approximation used in the literature. The goal of a non-uniformity correction algorithm is to compensate for the *local* contrast changes induced by the $\phi_{(i,j,t)}$ which means to apply some $\tilde{\phi}_{(i,j,t)}$ such that $\tilde{\phi}_{(i,j,t)}\left(\phi_{(i,j,t)}\right) = g(x)$ for $(i,j,t) \in \{1,...,N\} \times \{1,...,M\} \times \mathbb{R}_+$. Notice that, in general, it is useless to ask for $g = Id$ since users, screens (gamma correction), etc. usually tune the contrast of the images at ease.

3 Related work

To get rid of the non-uniformity many techniques have been developed over the years. It is possible to classify them into two main kinds:

- Calibration based techniques consist in an equalization of the response to an uniform black body source of radiations. They are not convenient for real time applications, since they force to interrupt the image flow. (This calibration is usually automatic, a shutter closing in front of the lens periodically). They usually assume that $\phi_{(i,j,t)}(x) = x + b_{(i,j)}$ $\forall t \in [t_1, t_2[$ (so called one point NUC, one black body) or that $\phi_{(i,j)}(x) = a_{(i,j)}x + b_{(i,j)}$ $\forall t \in [t_1, t_2[$ is linear (two points NUC using two black bodies). Thus they assumes piecewise constant $\phi_{(i,j,t)}$ on the time interval $[t_1, t_2[$. A new correction is performed every $t_2 - t_1$ (because of the sensor drift). Of course the two points NUC [34] performs better.

- Scene based techniques, involving motion compensation or temporal accumulation. Such methods are complex and require certain observation conditions (motion). They usually assume linear $\phi_{(i,j)}$.

In the sequel, we will focus on scene based techniques as calibration based techniques require to interrupt the camera which is calamitous in practice. Numerous algorithms have been reported in the literature to remove the fixed pattern noise caused by the lack of a cross-pixels sensor equalization. Some algorithms estimate the sensor parameters while, equivalently, others attempt at recovering the "true" landscape value $u(i, j, t)$. These algorithms process a sequence of images $(o(i, j, t))_{t\in 1,...,L}$, not a single frame. Thus they are subject to the creation of "ghost artifacts", the reason is discussed below. Most of them use a simplified (linear) model for the transfer function of the pixel sensor:

$$o(i, j, t) = a_{(i,j,t)} \left(u(i, j, t) + \eta(i, j, t)\right) + b_{(i,j,t)}, \ \forall (i, j, t) \in \{1, ..., N\} \times \{1, ..., M\} \times \mathbb{R}_+ \tag{I.2}$$

There are methods like [43] suggesting to equalize the mean and standard deviation (*stddev*) *through time* of each pixel sensor by a linear transform. Such algorithms rely on the data diversity found in most of the video sequences with some degree of motion. The key idea is

[$\mathcal{H}$:] If all pixel sensors have seen the same landscape, they should have (at least) the same mean and same standard deviation, namely

$$mean_{t\in\{1,...L\}} (o(i, j, t)) = C_m \ \forall \ (i, j) \in \{1, ..., N\} \times \{1, ..., M\} \tag{I.3}$$

$$stddev_{t\in\{1,...L\}} (o(i, j, t)) = C_{std} \ \forall \ (i, j) \in \{1, ..., N\} \times \{1, ..., M\}. \tag{I.4}$$

To summarize the authors suggest to adjust the sensor readout using a linear transform to enforce the equalities (I.3-I.4) above. But this is only possible if there is a long camera sequence with enough

motion where each sensor sweeps many different parts of the scene. Indeed a small window leads to little or no correction at all since it weakens [$\mathcal{H}$]. A subtle consequence of using long sequences is that the sensor (the contrast changes $\phi_{(i,j,t)}$) is assumed to be constant over the whole sequence. If the user cannot sufficiently move the camera, the convergence will be slow. This means that it is probable that between the beginning and the end of the sequence the $\phi_{(i,j,t)}$ will have changed because of the sensor drift. It leads to an inaccurate estimation of $\phi_{(i,j,t)}$: two pixels that have not seen the same radiance should not satisfy (I.3) or (I.4). Moreover since the sensor is not linear, there will be some residuals. These residuals may be negligible, except when the scene changes suddenly: the approximation is wrong, because it is based on past observations. It will take time to update the estimation, depending on L. The residues of the correction as well as the previous landscape will remain superimposed in the subsequent frames. This situation creates the "ghost artifacts". The usual way to avoid these "ghost artifacts" is to restart the learning process (to forget some past data) if a new scene appears. Nevertheless, the detection of scene changes may be treacherous particularly if it occurs in a small portion of the image (a new vehicle, etc.) since it may be masked by the *non uniformity* (see Fig. A.1).
A variant, like for instance [93], adjusts the minimum and the maximum of the readout values, assuming the time histograms observed in each sensor are equal over a long enough time sequence:

$$mean_{t\in\{1,...L\}}(o(i,j,t)) = C_1 \ \forall \ (i,j) \in \{1,...,N\} \times \{1,...,M\} \tag{I.5}$$

$$stddev_{t\in\{1,...L\}}(o(i,j,t)) = C_2 \ \forall \ (i,j) \in \{1,...,N\} \times \{1,...,M\}. \tag{I.6}$$

This last method is called Constant Range [138]. As pointed out by several authors [42] the length L of the sequence is a crucial factor of success here. There is no way to tune L *a priori* and two problems may arise:

- If L is too small and the estimation is wrong because all sensors have not seen the same landscape ([$\mathcal{H}$] is wrong);
- If L is too large and because of the approximation bias and time drift of the sensor behavior, the previous images may appear as superimposed in the last ones. We retrieve the previously cited "ghost artifact" effect.

There is a way to avoid the "ghost artifacts" [42], which consists in a reset of the estimation when the scene changes too much. For example, in [42] the authors use a simple threshold to perform scene change detection (but the level of this threshold is not easy to tune in general). In [106] the author state that "slow global motion and strong edges in the scene are the main causes" of the *non uniformity*. Indeed non consistent with [$\mathcal{H}$:] motions and/or bad length L of the sequence used to enforce (I.3-I.4 or I.5-I.6) lead to "ghost artifacts" because of the sensor drift. But, it is only more visible near edges.

Indeed the higher dynamic near edges weakens the linear approximation of the correction. The idea of treating edges separately also appears, for example, in [158]. To sum up, all these algorithms require a long exposition time with a varying scene and a serious (and sometimes involuntary) camera motion.

There are numerous implementations and studies [37, 44, 46, 91, 94, 114, 138, 147, 148] for these two major algorithms. A recursive filter [43] estimates the parameters of the linear function which approximates the $\phi_{(i,j,t)}$, or a Kalman filter is preferred [137]. Other authors [115, 139] propose a neural network based algorithm. Several other variants can be found in [106]. The registration based algorithms [142] consider often only translations [20] (but homographies should be used instead, at least on a static scene). Creating a panorama has been proposed [41] to obtain a ground truth, and to use it as a calibration pattern. However, as pointed out [160], in presence of the structured fixed pattern noise occurring in most infrared cameras, panoramas won't lead to a good result. Indeed, a mean act as a low pass filter. Thus, low frequencies of the *non uniformity* will remain in the produced images.
Recently in [146] the authors minimize the total variation of the produced images. They also assume a linear model for the *non uniformity*. It generalizes [87] but works on image sequences. Thus, it is also subject to "ghost artifacts". In the following, the presented algorithm works on a single image. Furthermore, in the sequel we will omit the time dependence t of the *non uniformity* ($\phi_{(i,j,t)} := \phi_{(i,j)}$).

4 The midway infrared correction

The goal of this section is to give the background (section 4.1) and details (see sections 4.2,4.3) of the *non uniformity* correction of infrared images by midway equalization (MIRE) [131] (see [132] for an on line use and implementation in $C++$).

4.1 The midway histogram equalization method

The midway algorithm was designed initially to correct for gain differences between cameras [29]. It permits to compare two images taken with different cameras more easily after their histograms have been equalized. This algorithm was later extended to flicker correction [30]. The midway equalization achieves much better and smoother results than giving flat (uniform) histograms to images. The idea of the midway equalization is to replace the (arbitrarily chosen) uniform histogram of classic uniform equalization by an histogram depending on the input images. It is optimal in the sense of the Wasserstein (transport) distance; the midway histogram being at equal distance of the histograms of input images. The midway equalization is defined and explained below.
Consider two cumulative histograms H_1, H_2 of two images. The midway cumulative histogram of the corrected image is simply

$$Hmid^{-1} := \frac{H_1^{-1} + H_2^{-1}}{2},$$

and this average can be extended to an arbitrary number of images (and to non constant weights). A precise definition of the pseudo inverse is given in section 4.3. Once the midway histogram is computed, a monotone contrast change is applied to images to specify $Hmid$ as their common histograms. Thus, all images get the midway histogram, which is the best compromise between all histograms (see Fig A.2).

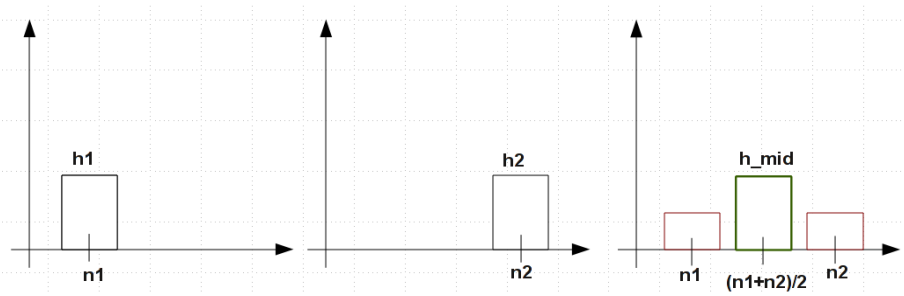

Figure A.2 – Two histograms h_1, h_2 (left side) and the corresponding midway histogram h_mid (on the right), compared to the direct histogram average, which would create two modes (centered at $n1$ and $n2$) and is therefore wrong. The uniform equalization would destroy the grey level dynamic and create artifacts. It is not a good candidate to get good quality images.

4.2 The idea

Since many infrared correction algorithms actually propose to equalize the temporal histograms of each pixel sensor, the midway is quite adapted to get a better result than a simple equalization. Equalization can be based on the fact that single columns (or lines, depending of the readout system) carry enough information by themselves for an equalization. The images being continuous, the difference between two adjacent columns is statistically small, implying that two neighboring histograms should be nearly equal. This hypothesis here is similar to the temporal one [$\mathcal{H}$] but is better suited to the decision to carry the equalization inside the image itself. It does not require any additional *hypothesis* on the *non uniformity* (linearity, etc.). In other words, the proposition is to transport the histogram of each column (or line) to the midway of histograms of neighboring columns (resp. lines). In presence of strong fixed-pattern-noise (FPN) it will be useful to perform this *sliding midway method* over a little more than two columns, because the FPN is not independent in general.

4.3 The midway infrared equalization algorithm (MIRE)

We give here the numerical details to implement the midway infrared equalization algorithm. It is fully automatic, it compensates for non linear *non uniformity*. Thus, it suits well an infrared image denoising chain (as a preprocessing for example). Assume in the sequel that the equalization is performed among the columns of a discrete (8-bits w.l.o.g.) image $o(i,j) \in \{0,...,255\}$ $\forall (i,j) \in \{1,...,N\} \times \{1,...,M\}$.

The "midway infrared equalization (MIRE)" algorithm proceeds as follows

For each column $j \in \{1, \cdots, M\}$:

1. compute the cumulative histogram H_j of each column c_j

$$\begin{array}{rcccl} H_j & : & \{0, \cdots, 255\} & \longrightarrow & [0,1] \\ & & l & \longmapsto & H_j(l) = \frac{1}{N}\sum_{k=0}^{l}\sum_{i=1}^{N}\mathbf{1}_{\{o(i,j)=k\}}, \end{array}$$

2. for each column c_j compute a local midway histogram $\tilde{H}_j^{-1} := \sum_{\substack{g \\ k\in(-n,\cdots,n)}}(k)H_{k+j}^{-1}$ using Gaussian weights $g(k) = g_s(k) = \frac{1}{s\sqrt{2\pi}}e^{\frac{-k^2}{2s^2}}$ with standard deviation s, the window size is $2n$ where $n = floor(4s)$ and,

$$\begin{array}{rcccl} H_j^{-1} & : & [0,1] & \longrightarrow & \{0, \cdots, 255\} \\ & & z & \longmapsto & H_j^{-1}(z) = min\{l \in \{0, \cdots, 255\} \mid H_j(l) \geq z\}, \end{array}$$

3. specify the histogram of the column c_j onto this midway histogram $\tilde{H}_j$

   $d(i,j) = \tilde{H}_j^{-1}\left(H_j(o(i,j))\right) \ \forall i \in \{1, \cdots, N\}$.

4. Output image: $d(i,j)$.

Since we work on images separately the method is not affected by motions or scene changes. This completely avoids "ghost artifacts" [42, 93] and any problem caused by the calibration parameters drifting over time. The above steps $1-3$ will be summarized in a single formula in the final algorithm below, without recurring to inverse histograms. An algorithm selecting s is given in the next section.

Automatically Fitting the Perfect Parameter

The *non-uniformity* leads to an increased total-variation norm. Hence, following the idea of [87], the smoothest image is also the one with little or no *non-uniformity* at all. So the simplest way to find the good (s^*) parameter automatically is :

$s^* = argmin_s ||I_s||_{TV-line}$ where I_s is the image processed by MIRE with the parameter s. The discrete total variation is defined by $||I||_{TV-line} = \sum_{i,j} |(\nabla I)_{i,j}|$ with $(\nabla I)_{i,j} = \left(I_{i,j+1} - I_{i,j}\right)$.

The optimization can be done by scanning a broad range of s. Choose a s_step and a s_max ($s_step = 0.5$ and $s_max = 8$ by default for the demo). Start with $s = 0$, repeat : 1) process the image 2) increase s of s_step, then stop when $s > s_max$.

The following ensures a certain safety of the proposed method, a fact that is confirmed by the experience of Fig A.3.

Theorem 1. If H_i $i \in \{-n, ..., n\}$ are $2n+1$ cumulative histograms of the same landscape seen by $2n+1$ different columns of the sensor, and $H_{mid}^{-1} = \sum_{j=-n}^{n} \frac{H_j^{-1}}{2n+1}$ then :

$$||H_{mid} - H_{true}||_2 \leq max_{i \in \{-n,...,n\}} (||H_i - H_{true}||_2)$$

Moreover if the H_i $\forall i \in \{-n, ..., n\}$ from the $2n+1$ columns of the sensor are *i.i.d. and centered* on H_{true} then

$$||H_{mid} - H_{true}||_2 \underset{n \to \infty}{\to} 0$$

Figure A.3 – On the left : an uncorrupted test image (boat). On the middle : the result of the MIRE algorithm ($s^* = 0$); the produced image is the same. This experiment was done using [132]. On the right : the result of the locally adaptive variant of MIRE described in section 5.1 ($s^* = 0$ everywhere in the image). As predicted the algorithm does not make the image worse or create artifacts (safety check). Results on real raw images corrupted with *non uniformity* are detailed in section 6.

5 The Adaptive and Denoising midway equalization algorithm (ADMIRE)

In this section we describe the novelties proposed to our previous work (see section 4 and [131, 132]). It consists in a modification of the MIRE (see section 4.3) to make it locally adaptive to the image. The need for a locally adaptive scheme is illustrated below (see Fig. A.5). This modification is detailed in section 5.1. The result of this locally adaptive scheme is nevertheless corrupted by the noise (which may be strong as in Fig. A.4). Thus, we shall also embed a denoising scheme in order to increase the signal to noise ratio.

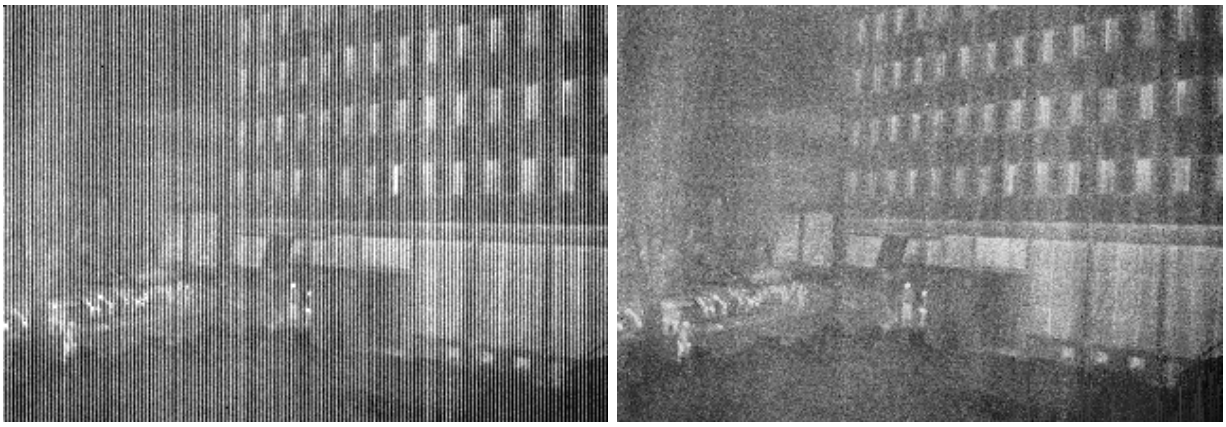

Figure A.4 – On the left : a real raw image produced by an infrared camera. On the right : the result of the locally adaptive MIRE algorithm (see section 5.1). It is strongly corrupted by noise.

5.1 The adaptive midway equalization algorithm

There is no real justification (except simplicity) to keep a constant parameter s over the whole image. Indeed the image can present different contents and structures in different areas. Thus focusing on different parts of the image the best s parameter may be different. A fact that is illustrated in Fig. A.5. Thus, we propose to adapt the s parameter of the MIRE algorithm locally. The proposed algorithm is :

1. For each s in $s_min : s_max$ (see section 4.3) process the image by MIRE;
2. Decompose all images in patches (8×8 patches always used here);
3. For each patch keep the one with smallest $TV - line$ (the best one as in section 4.3);
4. Average and aggregate all patches to get the *non uniformity* corrected image.

5.2 The denoising step : anisotropic DCT threshold on overlapping patches

Since the processed images seem to be corrupted by a strong noise we propose to adapt [26] to perform a good denoising. Notice that from the model detailed in section 2 we deduce that any denoising should be applied *after* the *non uniformity* correction as it whitened the noise. Indeed a perfect *non uniformity* correction would lead to a white noise in the restored image. A fact that can be checked using a "reverse engineering" of [83]. Thus we propose to modify the DCT threshold denoising algorithm of [26] which performs a DCT threshold on sliding and overlapping patches. It is well suited with the previous step of section 5 which also uses patches. Moreover [26] is unaffected[1] by

[1]The non local means denoising [14] has issues on some images (depending on the amount of residues of the *non uniformity*). Indeed residues of *non uniformity* interfere with patch distances (even after a column by column normalization of patches by their variance.)

Figure A.5 – On the left : a real raw image produced by an infrared camera. Middle : the result of the algorithm with the parameter $s = 1$. On the right with $s = 7.5$. For example, focusing on a zone in the middle of the image the image with $s = 7.5$ is nicer but, the zone below poles is bad. On the other hand in the image processed with $s = 1$ the zone in the middle is still corrupted by the *non uniformity*. Thus, a fixed s parameter for the whole image will not lead to the best quality possible everywhere.

residues of *non uniformity*. We kept 8×8 patches as in [26] for the examples given below in section 6. The modification consists of the use of two different thresholds in the direction of the *non uniformity* and the orthogonal direction. Indeed the coefficients in the direction of the lines should be bigger (for an image corrupted by columns *non uniformity* noise as in Fig. A.1) to denoise more. Indeed this direction is more corrupted by residues of *non uniformity*. It leads to an anisotropic filtering of the patches. To sum up, we suggest to perform an anisotropic DCT threshold on overlapping patches as a final denoising step. Provided two thresholds T_i and T_j (denoising strength) in the i (lines being indexed by i) and j direction of the image the algorithm is :

1. Decompose the image into sliding patches;
2. For each patch :
   (a) Compute 2D-DCTII transform of the patch;
   (b) Threshold the DCTII coefficients, with a threshold equal to T_j in the j direction and T_i (lines being indexed by i) everywhere else;
   (c) Calculate inverse 2D-DCT transform of the patch;
   (d) (normalize by a factor of $\frac{1}{4patch_size*patch_size}$);
3. Average and aggregate all patches to get the denoised image.

5.3 Implementation

The implementation is easy and was done with $C{+}{+}$. To avoid border effects we used a reflection of the image across borders. A $C++$ source code of the MIRE algorithm is available in [132], it allows on line experiments. The demonstration performs the MIRE algorithm, permits to see and download the result and shows the s^* parameter computed by the algorithm. For the denoising step we refer to

[155], where a $C++$ source code is available (as well as an on line demonstration). The overall chain of ADMIRE is :

1. Perform the adaptive midway equalization algorithm described in section 5.1;
2. Perform the anisotropic DCT threshold on overlapping patches described in section 5.2.

Of course a temporal extension (to videos) of the proposed method avoiding flicker (and "ghost artifacts") is possible, using a temporal midway [30].

6 Experiments

Simulations of Fig. A.6-A.7 are made using a nonlinear randomly generated model of NU. Results are quantified in terms of $RMSE$ and $RMSE^{CI}$ (the contrast invariant $RMSE$ reflects more the intrinsic quality of the image). They confirm the guess of visual improvement in quality. The comparison with the total variation based method is given in Figs. A.8-A.9. The proposed algorithm outperforms it. The algorithm was run on real raw images, Figs. A.10-A.15, thus for theses experiments it is not possible to compute a $RMSE$ (as the groundtruth is unknown). The proposed algorithm outperforms classic literature methods in real situations on real (non simulated) *non uniformity* while using only one image.

6.1 Total variation based method

Let $o(i,j)\ \forall(i,j) \in \{1,...,N\} \times \{1,...,M\}$ be the observed image. The $TV-line$ based method [87] looks for a constant $k(j)$ to add to each column. So

$$||o(i,j) + k(j)||_{TV-line}$$

is as small as possible. This boils down to the minimization of $\sum_i |o(i,j+1) + \delta(j) - o(i,j)|$ for each column j (notice that this sum involves only the column j and its neighboring column $j+1$). Then $k(j+1) = k(j) + \delta(j)$, where $k(0) = c$ chosen so that the resulting image has the same mean as the observed image o. In practice this can be done by :

1. Keep the first column intact ($j = 0$);
2. For each $j \in \{1,...,M\}$;
   (a) Minimize $\sum_{i \in \{1,...,N\}} |o(i,j+1) + \delta(j) - o(i,j)|$, by trying all possible $\delta(j)$ constants (using the quantification of the image);
3. Add a constant to the whole image so the output has the same mean as o.

Figure A.6 – On the left : an image with a strong simulated non linear *non uniformity.* On the right : the image processed by the proposed algorithm. $RMSE = 9.6629$, $RMSE^{CI} = 5.7314$.

Figure A.7 – On the left : an image with a strong simulated non linear *non uniformity.* On the right : the image processed by the proposed algorithm. $RMSE = 8.7100$, $RMSE^{CI} = 7.3411$.

6.2 Comparative experiments

The comparative experiments with the total variation were processed using *Megawave* [2] (resthline module [87]). Experiments on real raw infrared images are shown in Figs. A.8-A.9. The denoising step of ADMIRE has been deactivated for these experiments to allow a fair comparison with the total variation (TV) based method (which does not denoise). ADMIRE always shows a significant improvement on the TV based method and the final visual quality is very satisfactory.

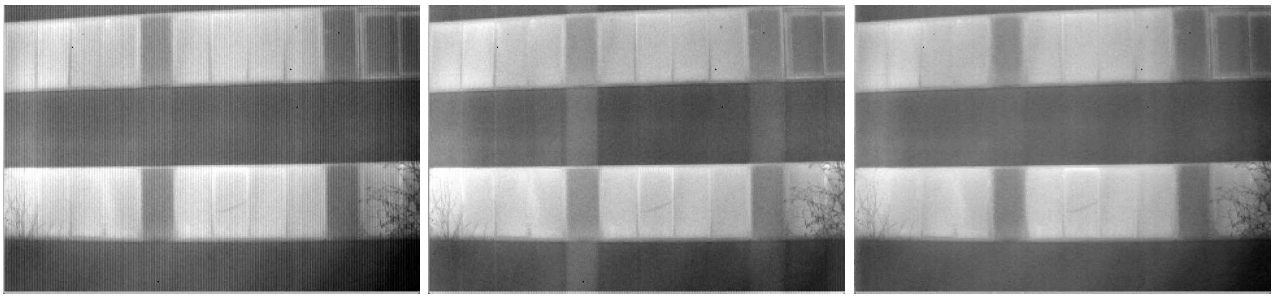

Figure A.8 – On the left : a real raw image of a building taken by an infrared camera. Middle : the total variation based method. Notice the artifacts created on the concrete stripes. The concrete stripes are at constant temperature thus, they should keep a constant grey level. On the right the proposed method (we deactivate the *DCT* denoising to provide a fair comparison).

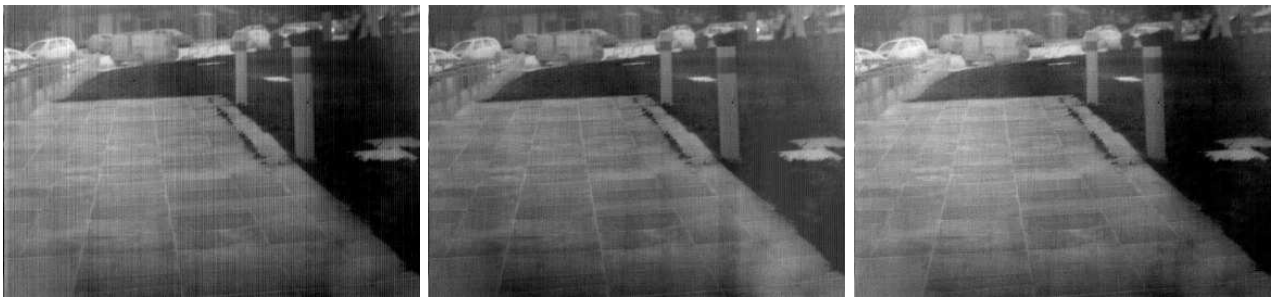

Figure A.9 – On the left : a real raw image of an outdoor scene taken by an infrared camera. Middle : the total variation based method. Notice the artifacts created below the poles. On the right the proposed method (we deactivate the *DCT* denoising to provide a fair comparison).

6.3 Experiments on real raw images

The subsequent experiments permit to visualize the result of the proposed method on numerous real raw LWIR infrared and **thermal infrared** hyperspectral images. We used different types of landscape to visualize the effect of the proposed *non uniformity* correction on edges, textures, and at different level of time exposure (noise). The conclusion is that the quality is quite satisfactory (see Figs.A.10-A.17).

[2] *Megawave* is available at megawave.cmla.ens-cachan.fr/

Figure A.10 – On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

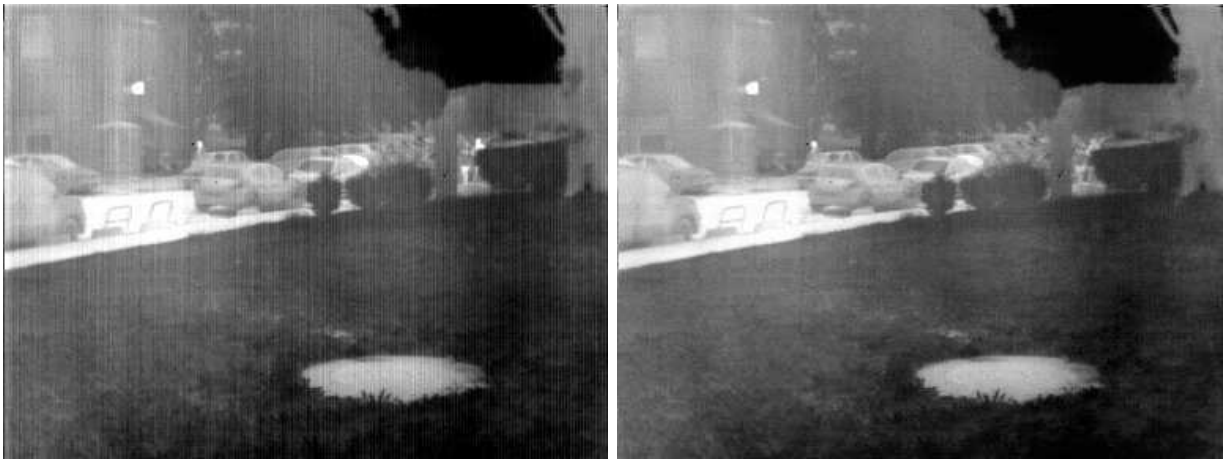

Figure A.11 – On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

7 Discussion and conclusion

In this part we started by modeling the image formation chain including the *non uniformity* and the Poisson (shot) noise terms. From this model we deduced the correct algorithm (chain) to apply to an image in order to perform a good *non uniformity* correction. We developed an image processing chain to correct for the *non uniformity* and the noise. A single image locally adaptive *non uniformity* correction was designed. It can compensate for fully non linear *non uniformity*, without any parametric model on the *non uniformity* side. It does not require motion, or motion compensation, does not need a test pattern or calibration and does not produce any "ghost artifact". A state of the art denoising algorithm was modified according to the model to obtain a *non uniformity* correction chain. Evaluations using both simulated and real raw images from infrared and **thermal infrared**

Figure A.12 – On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

Figure A.13 – On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

hyperspectral cameras show that the approach performs an efficient *non uniformity* correction in terms of $RMSE$, contrast invariant $RMSE^{CI}$ and visual image quality. Comparisons were made with a total variation based method. The conclusion is that a single image, ghost-less and non linear *non uniformity* correction is not only possible but robust and quite efficient.

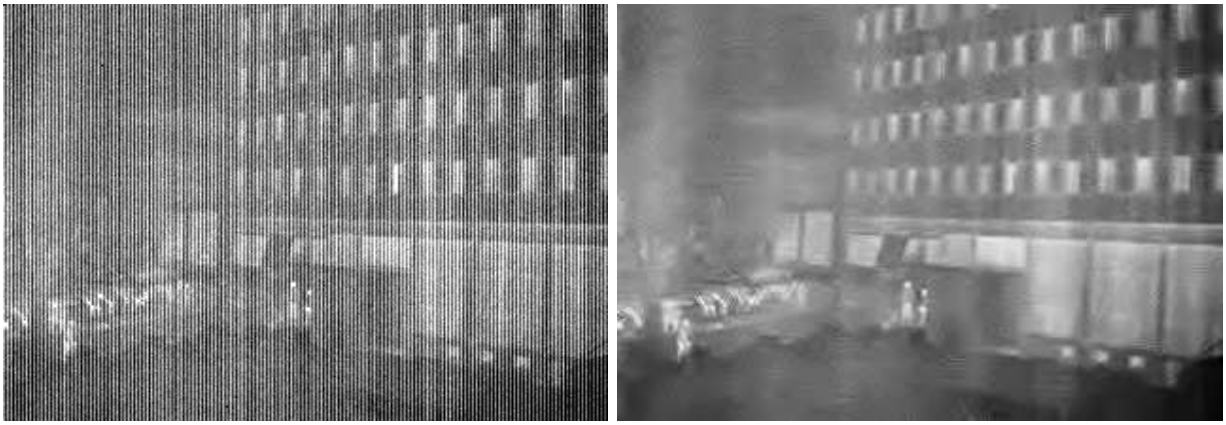

Figure A.14 – On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

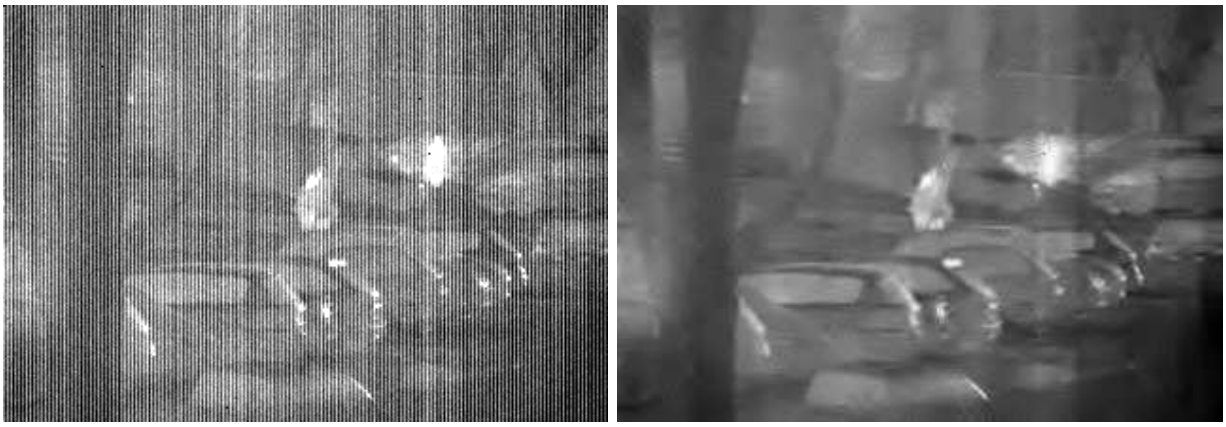

Figure A.15 – On the left : a real raw image taken by an infrared camera. On the right the result of ADMIRE.

Figure A.16 – On the left : a real raw image taken by an hyperspectral camera. On the right the result of ADMIRE.

Figure A.17 – On the left : a real raw image taken by an hyperspectral camera. On the right the result of ADMIRE.

Annex

1 Proof of $\hat{\alpha}_{MIP-ideal}(\xi) = \frac{1}{\sqrt{|a\xi|}} e^{-i\frac{\pi}{4}sign(\xi)}$

First we remark that $\alpha_{MIP-ideal}$ is $L^1_{loc}(\mathbb{R})$, then for $u = \rho e^{i\theta} \in \mathbb{C}$, $\mathcal{R}e(u) > 0$ (which enforces the absolute convergence of the following integral) we have $\int_0^{+\infty} \frac{1}{\sqrt{at}} e^{-ut} dt = \lim_{R\to+\infty} \int_0^{\frac{R}{u}} \frac{1}{u} \frac{1}{\sqrt{a\frac{z}{u}}} e^{-z} dz = \lim_{R\to+\infty} \int_0^{\frac{R}{\rho}e^{-i\theta}} \frac{1}{u} \frac{1}{\sqrt{a\frac{z}{u}}} e^{-z} dz = \frac{1}{\sqrt{au}} \int_0^{+\infty} \frac{1}{\sqrt{t}} e^{-t} dt$ with $\sqrt{u} = e^{\frac{1}{2}(Ln(|u|)+iArg(u))}$. Let $R > 0$, (and write $u = \rho e^{i\theta}$,) $u \mapsto \frac{e^{-u}}{\sqrt{u}}$ is holomorphic with one simple pole at $z = 0$ on the whole half complex plane $\mathcal{R}e(z) \geq 0$. The residue is $\lim_{z\to 0}(z-0)\frac{e^{-z}}{\sqrt{z}} = 0$. Thus, we have on the loop $[0, \frac{R}{\rho}e^{-i\theta}], [\frac{R}{\rho}e^{-i\theta}, \frac{R}{\rho}]$ and $[\frac{R}{\rho}, 0]$

$$\int_{[0,\frac{R}{\rho}e^{-i\theta}]} \frac{e^{-z}}{\sqrt{z}} dz + \int_{[\frac{R}{\rho}e^{-i\theta},\frac{R}{\rho}]} \frac{e^{-z}}{\sqrt{z}} dz + \int_{[\frac{R}{\rho},0]} \frac{e^{-t}}{\sqrt{t}} dt = 0, \tag{1}$$

which implies that $\int_{[0,\frac{R}{\rho}e^{-i\theta}]} \frac{e^{-z}}{\sqrt{z}} dz = -\int_{[\frac{R}{\rho}e^{-i\theta},\frac{R}{\rho}]} \frac{e^{-z}}{\sqrt{z}} dz - \int_{[\frac{R}{\rho},0]} \frac{e^{-t}}{\sqrt{t}} dt$. Now,

$$\lim_{R\to+\infty} \int_{[\frac{R}{\rho}e^{-i\theta},\frac{R}{\rho}]} \frac{e^{-z}}{\sqrt{z}} dz = \lim_{R\to+\infty} \int_{[e^{-i\theta},1]} \frac{R}{\rho} \frac{e^{-\frac{Rz}{\rho}}}{\sqrt{\frac{Rz}{\rho}}} dz = \lim_{n\to+\infty} \int_{[e^{-i\theta},1]} \sqrt{\frac{n}{\rho}} \frac{e^{-\frac{nz}{\rho}}}{\sqrt{z}} dz = 0.$$

Indeed, since $\mathcal{R}e(u) > 0$ we have $\theta \in]\frac{-\pi}{2}, \frac{\pi}{2}[$ thus $[e^{-i\theta}, 1] \subset \{z \in \mathbb{C}, \mathcal{R}e(z) > 0\}$. Then for $z \in [e^{-i\theta}, 1]$ and $n \in \mathbb{N}$ let $f_n(z) = \sqrt{\frac{n}{\rho}} \frac{e^{-\frac{nz}{\rho}}}{\sqrt{z}}$. We have, for $z \in [e^{-i\theta}, 1]$, $|f_n(z)| = |\sqrt{\frac{n}{\rho}} \frac{e^{-\frac{n\rho_z(\cos(\theta_z)+i\sin(\theta_z))}{\rho}}}{\sqrt{z}}| = |\sqrt{\frac{n}{\rho}} \frac{e^{-\frac{n\rho_z\cos(\theta_z)}{\rho}}}{\sqrt{z}}| \leq |\sqrt{\frac{n}{\rho}} \frac{e^{-\frac{n\rho_z}{\rho}}}{\sqrt{z}}| \to 0$ when $n \to \infty$ since $\mathcal{R}e(z) > 0$ entails $\cos(\theta_z) \in]0, 1]$. Moreover $|f_n(z)| \leq |\sqrt{\frac{n}{\rho}} \frac{e^{-\frac{n}{\rho}\frac{\sqrt{2}}{2}}}{\sqrt{z}}|$ ($\rho_z \geq \frac{\sqrt{2}}{2}$ from $z \in [e^{-i\theta}, 1]$) and $|f_n(z)| \leq \frac{\sqrt{2}}{\sqrt{z}} := f(z)$ which is $L^1([e^{-i\theta}, 1])$. Which yields, by using Lebesgue's dominated convergence theorem,

$$\lim_{n\to+\infty} \int_{[\frac{n}{\rho}e^{-i\theta},\frac{n}{\rho}]} \frac{e^{-z}}{\sqrt{z}} dz = 0.$$

Thus (from 1), $\lim_{n\to+\infty}\int_{[0,\frac{n}{\rho}e^{-i\theta}]}\frac{e^{-z}}{\sqrt{z}}dz = \lim_{n\to+\infty} -\int_{[\frac{n}{\rho},0]}\frac{e^{-t}}{\sqrt{t}}dt = \int_{[0,+\infty]}\frac{e^{-t}}{\sqrt{t}}dt$. This entails that for $\mathcal{R}e(u) > 0$, $\int_0^{+\infty}\frac{1}{\sqrt{at}}e^{-ut}dt = \frac{1}{\sqrt{au}}\int_0^{+\infty}\frac{1}{\sqrt{t}}e^{-t}dt$. By passing to the limit, which is easy by splitting the integral, we get the wished Fourier transform

$$\hat{\alpha}_{MIP-ideal}(\xi) = e^{\frac{-1}{2}(Ln(|a\xi|)+iArg(a\xi))}\int_0^{\infty}\frac{1}{\sqrt{t}}e^{-t}dt = \frac{1}{\sqrt{|a\xi|}}e^{-i\frac{\pi}{4}sign(\xi)}\int_0^{+\infty}\frac{1}{\sqrt{t}}e^{-t}dt,$$

where $sign(\xi) = 1$ if $\xi \geq 0$ and 0 elsewhere.

Bibliography

[1] A. Adams, D.E. Jacobs, J. Dolson, M. Tico, K. Pulli, E.V. Talvala, B. Ajdin, D. Vaquero, H. Lensch, M. Horowitz, et al. The Frankencamera: an experimental platform for computational photography. *ACM Transactions on Graphics (TOG)*, 29(4):29, 2010. 9

[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)*, pages 599–606, 2010. 8

[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)*, pages 1–8, 2007. 1, 2, 5, 6, 7, vi, 4, 61, 67, 77, 79

[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)*, pages 2560–2567, 2009. 4, 6, 53

[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*, volume 29, pages 763–772. Wiley-Blackwell, 2010. 9

[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)*, pages 2066–2073, 2009. vi, 4, 6, 53, 67, 77, 79

[7] C. Ancuti, C.O. Ancuti, and P. Bekaert. Deblurring by matching. In *Computer Graphics Forum*, volume 28, pages 619–628. Wiley Online Library, 2009. 5

[8] M. Backes, T. Chen, M. Duermuth, H. Lensch, and M. Welk. Tempest in a teapot: Compromising reflections revisited. In *Proceedings of the IEEE Symposium on Security and Privacy-Volume*, pages 315–327, 2009. 5

[9] Y. Bando and T. Nishita. Towards digital refocusing from a single photograph. In *Proceedings of the IEEE Pacific Conference on Computer Graphics and Applications*, pages 363–372, 2007. 9

[10] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Proceedings of the IEEE International Conference on Computer Vision (ICIP)*, pages 1–8. 6

[11] Lucien M. Biberman, editor. *Electro-Optical Imaging : system performance and modeling*. SPIE Press, october 2000. 7, 97, 98, 99

[12] G. Boracchi and A. Foi. Uniform Motion Blur in Poissonian Noise: Blur/Noise Tradeoff. *IEEE Transactions on Image Processing*, 20(2):592–598, 2011. 4, 6, 13, 14, 15, 37

[13] G.E.P. Box and M.E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958. 64

[14] A. Buades, B. Coll, and J.M. Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65, 2005. 106

[15] T. Buades, Y. Lou, JM 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)*, pages 1–15, 2009. 6, 18

[16] Richard Burden and J. Faires. *Numerical Analysis Seventh Edition*. Brooks Cole, 2001. 81

[17] J.F. Cai, J. Hui, L. Chaoqiang, and S. Zuowei. Blind motion deblurring using multiple images. *Journal of computational physics*, 228(14):5057–5071, 2009. 5

[18] J.F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 104–111, 2009. 5

[19] J.F. Cai, H. Ji, C. Liu, and Z. Shen. High-quality curvelet-based motion deblurring from an image pair. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1566–1573, 2009. 5

[20] S. Cain, E. Armstrong, and B. Yasuda. Joint estimation of image, shifts, and nonuniformities from IR images. In *Infrared Information Symposium (IRIS) on Passive Sensors*, 1997. 102

[21] W.G. Chen, N. Nandhakumar, and W.N. Martin. Image motion estimation from motion smear-a new computational model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):412–425, 1996. 5

[22] S. Cho and S. Lee. Fast motion deblurring. In *ACM Transactions on Graphics (TOG)*, volume 28, page 145, 2009. 5

[23] S. Cho, Y. Matsushita, and S. Lee. Removing non-uniform motion blur from images. In *Proceedings of the IEEE International Conference on Computer Vision (ICIP)*, pages 1–8, 2007. 5

[24] T. Cho, C. Zitnick, N. Joshi, S. Kang, R. Szeliski, and W. Freeman. Image restoration by matching gradient distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1–1, 2011. 5

[25] 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)*, pages 1–8, 2010. 8, 47

[26] R.R. Coifman and D.L. Donoho. Translation-Invariant De-Noising. *Wavelets and statistics*, 103:125, 1995. 106, 107

[27] S. Dai and Y. Wu. Motion from blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 5

[28] S. Dai and Y. Wu. Removing partial blur in a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2544–2551, 2009. 5

[29] J. Delon. Midway image equalization. *Journal of Mathematical Imaging and Vision*, 21(2):119–134, 2004. 65, 102

[30] J. Delon. Movie and video scale-time equalization application to flicker reduction. *IEEE Transactions on Image Processing*, 15(1):241–248, 2006. 4, 8, 102, 108

[31] 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)*, pages 15–30, 2010. 8

[32] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 787–794, 2006. 5

[33] A. Foi, S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian. A spatially adaptive Poissonian image deblurring. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 1, pages I–925, 2005. 6

[34] A. Friedenberg and I. Goldblatt. Nonuniformity two-point linear correction errors in infrared focal plane arrays. *Optical Engineering*, 37:1251, 1998. 100

[35] M.E. Gehm, R. John, D.J. Brady, R.M. Willett, and T.J. Schulz. Single-shot compressive spectral imaging with a dual-disperser architecture. *Optics Express*, 15(21):14013–14027, 2007. 6

[36] J.P. Godbaz, M.J. Cree, and A.A. Dorrington. Extending amcw lidar depth-of-field using a coded aperture. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 397–409, 2010. 9

[37] P. Goyal. NUC algorithm by calculating the corresponding statistics of the decomposed signal. *International Journal on Computer Science and Technology*, 1(2), 2010. 102

[38] J. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar. Coded rolling shutter photography: Flexible space-time sampling. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2010. 9

[39] A. Gupta, N. Joshi, C.L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proceedings of the (Springer-Verlag) European conference on Computer vision (ECCV)*, pages 171–184, 2010. 5

[40] 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)*, pages 100–114, 2010. 8

[41] R.C. Hardie, M.M. Hayat, E. Armstrong, and B. Yasuda. Scene-based nonuniformity correction with video sequences and registration. *Applied Optics*, 39(8):1241–1250, 2000. 8, 102

[42] J.G. Harris and Y.M. Chiang. Minimizing the ghosting artifact in scene-based nonuniformity correction. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3377, pages 106–113, 1998. 4, 97, 101, 104

[43] J.G. Harris and Y.M. Chiang. Nonuniformity correction of infrared image sequences using the constant-statistics constraint. *IEEE Transactions on Image Processing*, 8(8):1148–1151, 1999. 8, 100, 102

[44] R. Hart and Thales Optronics Ltd O. Thomas. A study of non-uniformity correction methods for staring array IR detectors. 1st EMRS DCT Technical Conference, Edinburgh, 2004. 102

[45] S.W. Hasinoff and K.N. Kutulakos. Light-efficient photography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2203–2214, 2011. 9

[46] M.M. Hayat, S.N. Torres, S. Cain, and E.E. Armstrong. Model-based real-time nonuniformity correction in focal plane array detectors. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3377, pages 122–132, 1998. 8, 102

[47] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, S.K. Nayar, T. Kobayashi, O. Cossairt, D. Miau, SK Nayar, C. Zhou, et al. Video from a single coded exposure photograph using a learned over-complete dictionary. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Nov 2011. 6

[48] T. Hou, S. Wang, and H. Qin. Image deconvolution with multi-stage convex relaxation and its perceptual evaluation. *IEEE Transactions on Image Processing*, (99):1–1, 2010. 5

[49] J.S. Houchin and K.A. Parulski. Method and apparatus for pixel non-uniformity correction, September 10 1991. US Patent 5,047,861. 98

[50] P. Hsu and B.Y. Chen. Blurred image detection and classification. In *Proceedings of the 14th international conference on Advances in multimedia modeling*, pages 277–286. Springer-Verlag, 2008. 5

[51] N. Jacobs, S. Schuh, and R. Pless. On unusual pixel shapes and image motion. Technical report, Computer Science and Engineering, Washington University in St. Louis, MO, USA, 2009. 5

[52] N. Jacobs, S. Schuh, and R. Pless. Compressive sensing and differential image-motion estimation. In *Proceedings of the IEEE Conference on Acoustics Speech and Signal Processing (ACASSP)*, pages 718–721, 2010. 5

[53] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Estimation of blur and noise parameters in remote sensing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, page 3580, 2002. 5

[54] P.A. Jansson and M. Richardson. Deconvolution of images and spectra. *Optical Engineering*, 36:3224, 1997. 5

[55] J. Jelinek. Designing the optimal shutter sequences for the flutter shutter imaging method. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7701, page 18, 2010. 53

[56] H. Ji and C. Liu. Motion blur identification from image gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 5

[57] J. Jia. Single image motion deblurring using transparency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 5

[58] W. Jin, C. Liu, and J. Xiu. Infrared nonuniformity correction and radiometric calibration technology using U-shaped blackbody. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8194, page 819405, 2011. 98

[59] N. Joshi, S.B. Kang, C.L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics (TOG)*, 29(4):30, 2010. 5

[60] N. Joshi, R. Szeliski, and D.J. Kriegman. PSF estimation using sharp edge prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 5

[61] T. Kenig, Z. Kam, and A. Feuer. Blind image deconvolution using machine learning for three-dimensional microscopy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2191–2204, 2010. 5

[62] Donald E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms*. Addison-Wesley, 1969. 64

[63] S.J. Koppal, S. Yamazaki, and S.G. Narasimhan. Exploiting DLP Illumination Dithering for Reconstruction and Photography of High-Speed Scenes. *International Journal of Computer Vision (IJCV)*, pages 1–20, 2011. 7

[64] H. Kozuka. Image sensor, October 29 2002. US Patent 6,473,538. 21

[65] D. Lanman, R. Raskar, A. Agrawal, and G. Taubin. Shield fields: modeling and capturing 3D occluders. *ACM Transactions on Graphics (TOG)*, 27(5):131, 2008. 8

[66] A. Levin. Blind motion deblurring using image statistics. *Advances in Neural Information Processing Systems*, 19:841, 2007. 5

[67] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Deconvolution using natural image priors. *ACM Transactions on Graphics (TOG)*, 26(3):0–2, 2007. 5

[68] A. Levin, R. Fergus, F. Durand, and W.T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (TOG)*, 26(3):70–es, 2007. 9

[69] A. Levin, P. Sand, T.S. Cho, F. Durand, and W.T. Freeman. Motion-invariant photography. In *ACM Transactions on Graphics (TOG)*, volume 27, page 71, 2008. 1, 2, 5, 6, 8, 43, 44, 46, 70

[70] A. Levin, P. Sand, T.S. Cho, F. Durand, and W.T. Freeman. Method and apparatus for motion invariant imaging, October 1 2009. US Patent 20,090,244,300. 2, 43, 44

[71] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2657–2664, 2011. 5

[72] F. Li, J. Sun, J. Wang, and J. Yu. Dual-focus stereo imaging. *Journal of Electronic Imaging*, 19(4), 2010. 9

[73] C.K. Liang, T.H. Lin, B.Y. Wong, C. Liu, and H.H. Chen. Programmable aperture photography: multiplexed light field acquisition. In *ACM Transactions on Graphics (TOG)*, volume 27, page 55, 2008. 9

[74] L.B. Lucy. An iterative technique for the rectification of observed distributions. *The astronomical journal*, 79:745, 1974. 5

[75] R. Lukac. *Computational Photography: Methods and Applications*, volume 2. CRC Press, 2010. 9

[76] 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)*, pages 3521–3524, 2010. 9

[77] M. Martinello and P. Favaro. Single image blind deconvolution with higher-order texture statistics. *International Semina on Video Processing and Computational Video*, pages 124–151, 2011. 5

[78] S. McCloskey. Velocity-dependent shutter sequences for motion deblurring. In *Proceedings of the (Springer-Verlag) European conference on Computer vision (ECCV)*, pages 309–322, 2010. vi, 7, 77, 79

[79] 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)*, pages 1–6, 2010. 7

[80] S. McCloskey, J. Jelinek, and K.W. Au. Method and system for determining shutter fluttering sequence, April 9 2009. US Patent 12/421,296. 2, 77

[81] 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)*, pages 1–8, 2011. 8

[82] Scott McCloskey. Temporally coded flash illumination for motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 683–690, 2011. 7

[83] O.J. Medina, J.E. Pezoa, and S.N. Torres. A frequency domain model for the spatial fixed-pattern noise in infrared focal plane arrays. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 8155, page 50, 2011. 106

[84] 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)*, volume 2, pages 916–919, 2010. 9

[85] A.F. Milton, F.R. Barone, and M.R. Kruer. Influence of nonuniformity on infrared focal plane array performance. *Optical Engineering*, 24(5):855–862, 1985. 97

[86] 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)*, pages 1–8, 2009. 9

[87] L. Moisan. Resthline. Computer program, MegaWave2 Modulus, 2007. 102, 104, 108, 110

[88] 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)*, pages 337–350, 2010. 9

[89] K. Nakamura, H. Ohzu, and I. Ueno. Image sensor in which reading and resetting are simultaneously performed, November 16 1993. US Patent 5,262,870. 21

[90] S.G. Narasimhan, S.J. Koppal, and S. Yamazaki. Temporal dithering of illumination for fast active vision. In *Proceedings of the (Springer-Verlag) European conference on Computer vision (ECCV)*, pages 830–844, 2008. 7

[91] P.M. Narendra. Reference-free nonuniformity compensation for IR imaging arrays. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 252, pages 10–17, 1980. 102

[92] S.K. Nayar and M. Ben-Ezra. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, 2004. 6

[93] J. Pezoa, S. Torres, J. Córdova, and R. Reeves. An enhancement to the constant range method for nonuniformity correction of infrared image sequences. *Progress in Pattern Recognition, Image Analysis and Applications*, pages 259–279, 2004. 101, 104

[94] W. Qian, Q. Chen, J. Bai, and G. Gu. Adaptive convergence nonuniformity correction algorithm. *Applied Optics*, 50(1):1–10, 2011. 97, 102

[95] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICIP)*, volume 2, pages 1048–1054, 2005. 5

[96] R. Raskar. Less is more: coded computational photography. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 1–12, 2007. 9

[97] R. Raskar. Computational photography: Epsilon to coded photography. In *Emerging Trends in Visual Computing: LIX Fall Colloquium, ETVC 2008, Palaiseau, France, November 18-20, 2008, Revised Selected and Invited Papers*, volume 5416, page 238. Springer-Verlag New York Inc, 2009. 9

[98] R. Raskar. Method and apparatus for deblurring images, July 13 2010. US Patent 7,756,407. 2, 4, 67, 77

[99] R. Raskar, J. Tumblin, and A. Agrawal. Method for deblurring images using optimized temporal coding patterns, August 25 2009. US Patent 7,580,620. 2, 4, 67, 77

[100] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3):311–317, 2005. 5

[101] D. Reddy, A. Veeraraghavan, and R. Chellappa. P2C2: programmable pixel compressive camera for high speed imaging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 329–336, 2011. 7

[102] D. Reddy, A. Veeraraghavan, and R. Raskar. Coded strobing photography for high-speed periodic events. In *Imaging Systems*. Optical Society of America, 2010. 7

[103] C. Ren, W. Chen, I. Shen, et al. Three-stage motion deblurring from a video. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 53–62, 2007. 5

[104] W.H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972. 5

[105] O. Riou, S. Berrebi, and P. Bremond. Nonuniformity correction and thermal drift compensation of thermal infrared camera. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5405, pages 294–302, 2004. 98

[106] A. Rossi, M. Diani, and G. Corsini. A comparison of deghosting techniques in adaptive nonuniformity correction for IR focal-plane array systems. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2010. 4, 101, 102

[107] A. Rossi, M. Diani, and G. Corsini. Temporal statistics de-ghosting for adaptive non-uniformity correction in infrared focal plane arrays. *Electronics letters*, 46(5):348–349, 2010. 97

[108] P.D. Samarasinghe, R.A. Kennedy, and H. Li. On non-blind image restoration. In *Proceedings of the IEEE International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–7, 2009. 5

[109] 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)*, pages 417–422, 2010. 7

[110] C. Sascha, J. Helios, et al. Automatic level control for video cameras towards hdr techniques. *Journal on Image and Video Processing (EURASIP)*, 2010, 2011. 6

[111] S. Schuon and K. Diepold. Comparison of motion de-blur algorithms and real world deployment. *Acta Astronautica*, 64(11-12):1050–1065, 2009. 5

[112] D.A. Scribner, M.R. Kruer, and J.M. Killiany. Infrared focal plane array technology. *Proceedings of the IEEE*, 79(1):66–85, 1991. 97, 98

[113] D.A. Scribner, K.A. Sarkady, J.T. Caulfield, M.R. Kruer, and G. Katz. Nonuniformity correction for staring IR focal plane arrays using scene-based techniques. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1308, pages 224–233, 1990. 8, 98

[114] D.A. Scribner, K.A. Sarkady, M.R. Kruer, J.T. Caulfield, J.D. Hunt, M. Colbert, and M. Descour. Adaptive retina-like preprocessing for imaging detector arrays. In *Proceedings of the IEEE Conference on Neural Networks*, pages 1955–1960, 1993. 102

[115] D.A. Scribner, K.A. Sarkady, M.R. Kruer, J.T. Caulfield, JD Hunt, and C. Herman. Adaptive nonuniformity correction for IR focal-plane arrays using neural networks. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1541, pages 100–109, 1991. 8, 102

[116] A. Sellent, M. Eisemann, and M. Magnor. Calculating motion fields from images with two different exposure times. Technical report, Tech. rep., Computer Graphics Lab, Technical University of Braunschweig, 5, 2008. 5

[117] A. Sellent, M. Eisemann, and M. Magnor. Motion field and occlusion time estimation via alternate exposure flow. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, 2009. 5

[118] K. Sengupta and F. Porikli. Geometric Sequence (GS) imaging with Bayesian smoothing for optical and capacitive imaging sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 90–97, 2009. 5

[119] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics (TOG)*, volume 27, page 73, 2008. 5

[120] G. Shi, D. Gao, D. Liu, and L. Wang. High resolution image reconstruction: a new imager via movable random exposure. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1177–1180, 2009. 7

[121] M. Shi and J.Y. Zheng. A slit scanning depth of route panorama from stationary blur. In *Proceedings of the IEEE on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1047–1054, 2005. 5

[122] C. Song, H. Zhao, W. Jing, and H. Zhu. PSO based motion deblurring for single image. In *Proceedings of the (ACM) Annual Conference on Genetic and Evolutionary Computation*, pages 85–92, 2011. 5

[123] Y. Tai, P. Tan, and M. Brown. Richardson-Lucy deblurring for scenes under projective motion path. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (99):1–1, 2010. 5

[124] Y.W. Tai, H. Du, M.S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. 6

[125] Y.W. Tai, H. Du, M.S. Brown, and S. Lin. Correction of spatially varying image and video motion blur using a hybrid camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1012–1028, 2010. 6

[126] 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)*, pages 2408–2415, 2010. 7

[127] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis. Synthetic shutter speed imaging. In *Computer Graphics Forum*, volume 26, pages 591–598. Wiley Online Library, 2007. 6

[128] Y. Tendero. The Flutter Shutter camera simulator. *Submitted to Image Processing On Line*, pages 1–27, 2012. 2, 6, 7

[129] Y. Tendero. The Flutter Shutter optimizer. *Submitted to Image Processing On Line*, pages 1–25, 2012. 3

[130] Y. Tendero and J. Gilles. ADMIRE: a locally adaptive single-image, non-uniformity correction and denoising algorithm: application to uncooled IR camera. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume to appear, pages 1–18. SPIE, 2012. 4

[131] Y. Tendero, J. Gilles, S. Landeau, and JM Morel. Efficient single image non-uniformity correction algorithm. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7834, page 10, 2010. 102, 105

[132] Y. Tendero, S. Landeau, and J. Gilles. Non-uniformity correction of infrared images by midway equalization. *Submitted to Image Processing On Line*, pages 1–23, 2012. 8, 102, 105, 107

[133] Y. Tendero, J-M Morel, and B. Rougé. A formalization of the flutter shutter. In *Proceedings of the Conference on New Computational Methods for Inverse Problems (NCMIP)*, Cachan, France, 2012. 2

[134] Y. Tendero, J-M Morel, and B. Rougé. How to optimize a Flutter Shutter? *Submitted to International Journal On Computer Vision*, pages 1–31, 2012. 3

[135] Y. Tendero, J-M Morel, and B. Rougé. The Flutter Shutter paradox. *Submittted to SIAM Journal on Imaging Sciences*, pages 1–33, 2012. 2

[136] M. Tico and K. Pulli. Image enhancement method via blur and noisy image fusion. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1521–1524, 2009. 5

[137] S.N. Torres and M.M. Hayat. Kalman filtering for adaptive nonuniformity correction in infrared focal-plane arrays. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 20(3):470, 2003. 102

[138] S.N. Torres, E.M. Vera, R.A. Reeves, and S.K. Sobarzo. Scene-based non-uniformity correction method using constant range: Performance and analysis. In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics, IX: 224–229*, 2002. 8, 101, 102

[139] S.N. Torres, E.M. Vera, R.A. Reeves, and S.K. Sobarzo. Adaptive scene-based nonuniformity correction method for infrared-focal plane arrays. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5076, pages 130–139, 2003. 8, 102

[140] M. Trentacoste, C. Lau, M. Rouf, R. l Mantiuk, and W. Heidrich. Defocus techniques for camera dynamic range expansion. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 7537, page 16, 2010. 9

[141] R. Tsai et al. Pulsed control of camera flash, June 14 2011. US Patent 7,962,031. 7

[142] S. Tzimopoulou-Fricke and A.H. Lettington. Scene-based techniques for nonuniformity correction of infrared focal plane arrays. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3436, pages 172–183, 1998. 102

[143] 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 Transactions on Graphics (TOG)*, 26(3):69, 2007. 9

[144] A. Veeraraghavan, D. Reddy, and R. Raskar. Coded strobing photography: Compressive sensing of high speed periodic events. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):671–686, 2011. 6

[145] V.S. Veeravasarapu and J. Sivaswamy. Motion deblurring as optimisation. In *Proceedings of the ACM Indian Conference on Computer Vision, Graphics and Image Processing*, pages 267–273, 2010. 5

[146] E. Vera, P. Meza, and S. Torres. Total variation adaptive scene-based nonuniformity correction. In *Imaging Systems*. Optical Society of America, 2010. 97, 102

[147] E. Vera and S. Torres. Fast adaptive nonuniformity correction for infrared focal-plane array detectors. *EURASIP Journal on Applied Signal Processing*, 13:1994–2004, 2005. 102

[148] R.E. Vera and I.S. Torres. Ghosting reduction in adaptive nonuniformity correction of infrared focal-plane array image sequences. In *Proceedings of the IEEE Conference on Image Processing (ICIP)*, volume 2, pages II–1001, 2003. 102

[149] 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)*, pages 491–498, 2010. 5

[150] B. Wilburn, N. Joshi, V. Vaish, E.V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 765–776, 2005. 6

[151] S. Xiang, G. Meng, Y. Wang, C. Pan, and C. Zhang. Image deblurring with matrix regression and gradient evolution. *Pattern Recognition*, 2011. 5

[152] S. Xu, H. Liang, D. Tu, and G. Li. A deblurring technique for large scale motion blur images using a hybrid camera. In *Proceedings of the International Congress on Image and Signal Processing (CISP)*, volume 2, pages 806–810, 2010. 6

[153] 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)*, pages 159–165, 2011. 7

[154] W. Yongpan, F. Huajun, X. Zhihai, L. Qi, and D. Chaoyue. An improved richardson-lucy algorithm based on local prior. *Optics & Laser Technology*, 42(5):845–849, 2010. 5

[155] G. Yu and G. Sapiro. DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line*, 2011. 108

[156] L. Yuan, J. Sun, L. Quan, and H.Y. Shum. Image deblurring with blurred/noisy image pairs. In *ACM Transactions on Graphics (TOG)*, volume 26, page 1, 2007. 5

[157] L. Yuan, J. Sun, L. Quan, and H.Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. In *ACM Transactions on Graphics (TOG)*, volume 27, page 74, 2008. 5

[158] T. Zhang and Y. Shi. Edge-directed adaptive nonuniformity correction for staring infrared focal plane arrays. *Optical Engineering*, 45:016402, 2006. 102

[159] J. Zhao, H. Feng, Z. Xu, and Q. Li. An improved image restoration approach using adaptive local constraint. *Optik-International Journal for Light and Electron Optics*, 2011. 5

[160] W. Zhao and C. Zhang. Efficient scene-based nonuniformity correction and enhancement. In *Proceedings of the IEEE Conference on Image Processing (ICIP)*, pages 2873–2876, 2006. 8, 102

[161] C. Zhou and S. Nayar. Computational cameras: Convergence of optics and processing. *IEEE Transactions on Image Processing*, (99):1–1, 2011. 9

[162] C. Zuo, Q. Chen, G. Gu, and W. Qian. New temporal high-pass filter nonuniformity correction based on bilateral filter. *Optical Review*, 18(2):197–202, 2011. 97

[163] C. Zuo, Q. Chen, G. Gu, and X. Sui. Scene-based nonuniformity correction algorithm based on interframe registration. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 28(6):1164–1176, 2011. 97