The Flutter Shutter

Overview

- The Fourier transform and the sinc function
- Poisson random variable
- Steady acquisition model
- Fundamental principle of photography
- The Raskar Flutter Shutter
- Acquisition model of a moving landscape
- The numerical Flutter Shutter
- The Flutter Shutter paradox

- Let $f, g \in L^1(\mathbb{R})$ or $L^2(\mathbb{R})$

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

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

- Parseval

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

- Band limited : $\hat{u}(\xi)$ supported on $[-\pi, \pi]$

- Let $f, g \in L^1(\mathbb{R})$ or $L^2(\mathbb{R})$

- (convolution) $(f * g)(x) = \int_{-\infty}^{+\infty} f(y)g(x-y)dy$

- The previous definitions entail

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

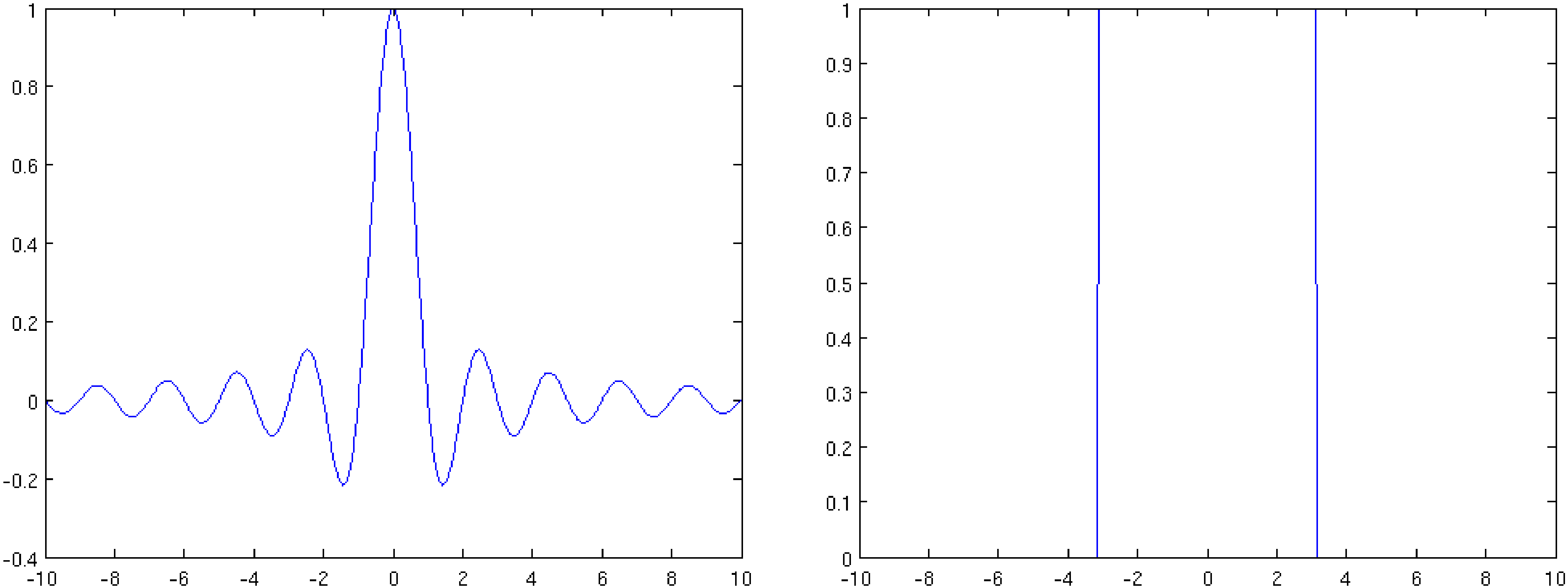
- The sinc function is $sinc(x) = \frac{sin(\pi x)}{\pi x}$

- Whose Fourier transform is $\mathbb{1}_{[-\pi,\pi]}(\xi)$

- Acts as a Dirac for band-limited functions :

  if $\hat{u}(\xi)$ supported on $[-\pi, \pi]$

  then $(u * sinc)\,(x) = u(x)\ \forall x \in \mathbb{R}$



The sinc function (left) and it's Fourier transform (right).

(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 $\sum_{n\in\mathbb{Z}} f(n) = \hat{f}(0) = ||f||_{L^1}$.

- X Poisson random variable

$$X \sim \mathcal{P}(\lambda) \qquad \lambda > 0 \text{ intensity}$$

$$\mathbb{P}(X = k) = e^{-\lambda}\frac{\lambda^k}{k!},\ k \in \mathbb{N}$$

$$\mathbb{E}(X) = \lambda \qquad var(X) = \lambda$$

- Simulation of $X \sim Poisson(\lambda)$ (D. Knuth)

**If $(\lambda \leq 50)$ then

Let $g = exp(-\lambda)$; em=-1; t =1; boolean rejected=true;

While (rejected) DO

em=em+1;

t=t.rand; (where rand is an uniform on [0,1] random generator)

If $(t > g)$ then : X=em; rejected=true; endif;

endwhile;

**Else : simulate X a Gaussian random variable with mean and variance equal to λ

- Simulation of $X \sim \mathcal{N}(\lambda, \lambda)$ (polar Box&Muller)

$$X = \sqrt{(-2log(rand)}.cos(2\pi rand)$$

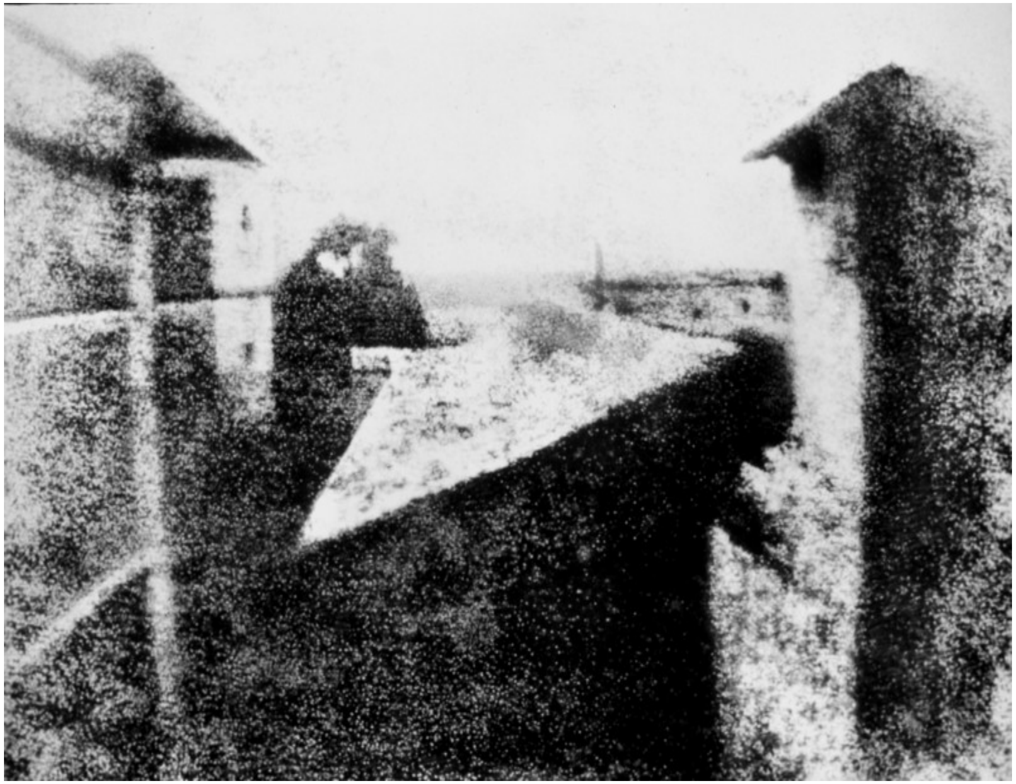
$$X = \lambda + \sqrt{\lambda}X$$

- Signal to Noise Ratio (SNR)

  Let X be a random variable then

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

- Nicephore Niepce, the first photography (1827)



exposure time : 8 hours

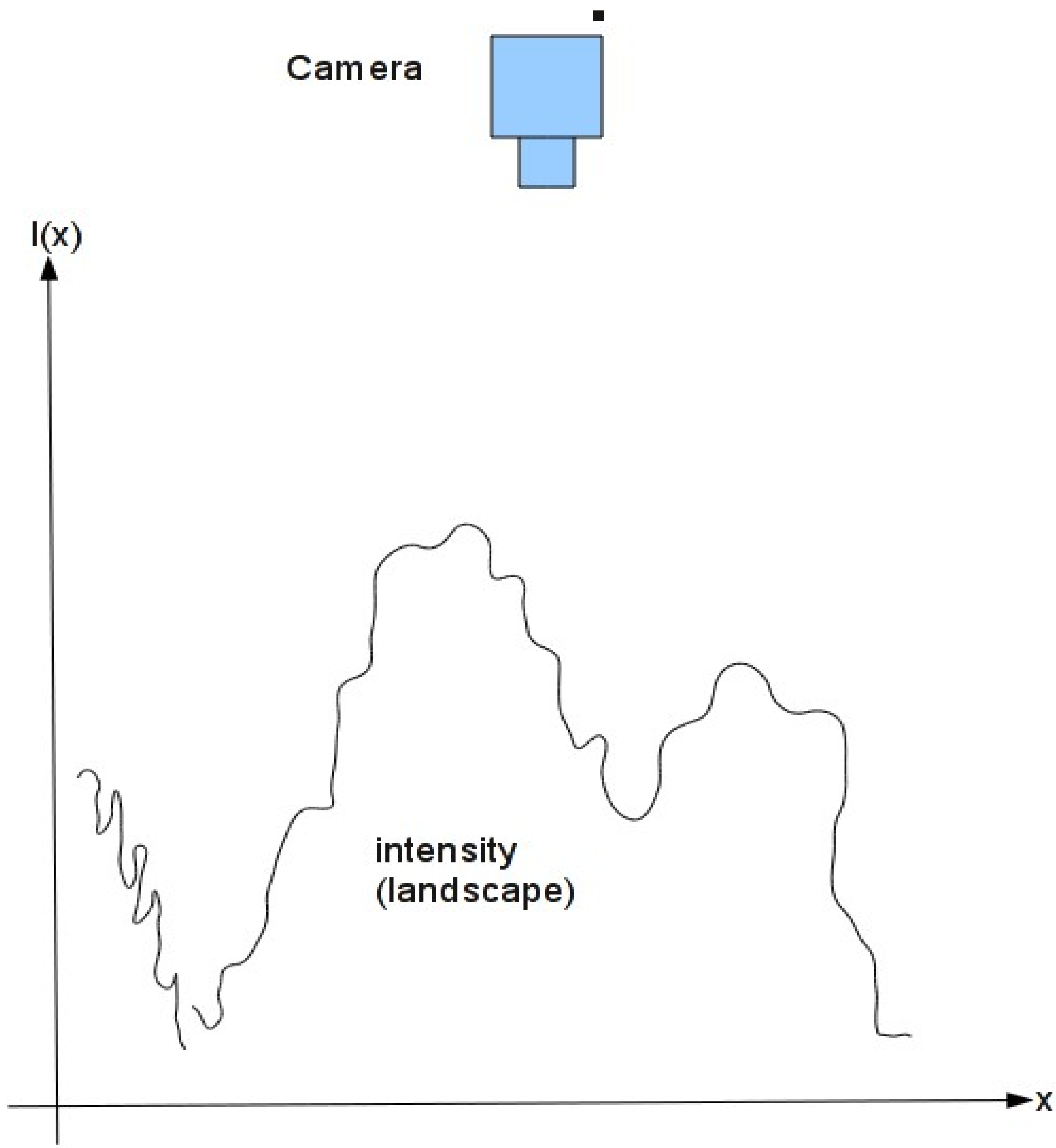
- Acquisition model (1D setting)

  Real (non observable) landscape : l

  (deterministic intensity field)

$$l(t, x),\ \forall (t, x) \in \mathbb{R}^+ \times \mathbb{R}$$

which represents the photon emission at time t and position x


Camera
I(x)
intensity
(landscape)
x

- Camera : optical system + pixel sensor

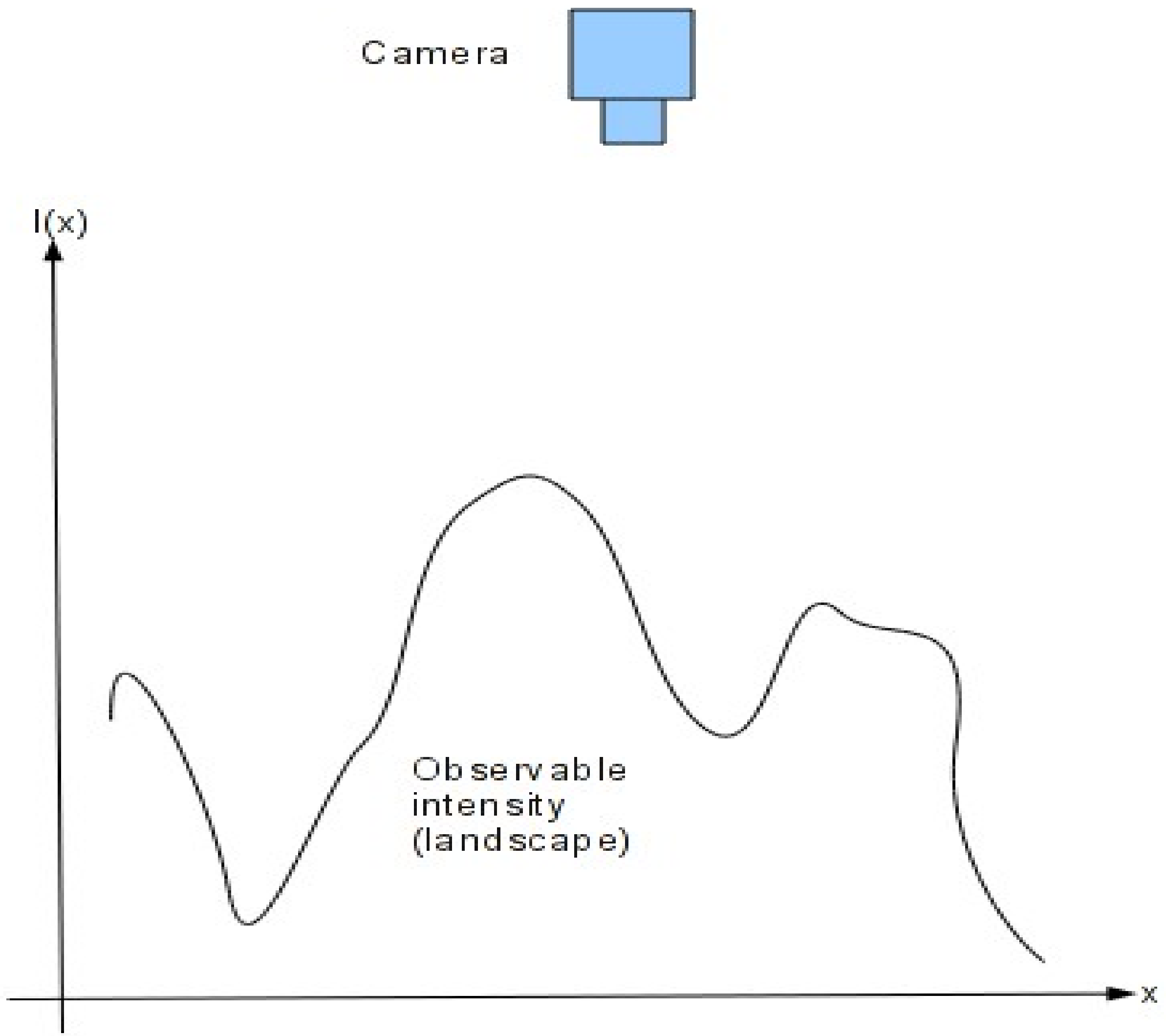
$point\ spread\ function\ g$ photon integrator

(band-limited)

We call ”ideal landscape” u the deterministic function

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

$$(f * g)(x) = \int_{-\infty}^{+\infty} f(y)g(x-y)dy$$



From now on l is seen through the optical system and already contains g

$$l := l * g$$

- Assumptions u

$$u \in L^1 \cap L^2$$

Band limited, thanks to the $point\ spread\ function\ g$

$$\hat{u}(\xi) \text{ supported on } [-\pi, \pi]$$

$$\hat{u}(\xi) = \int_{-\infty}^{\infty} u(x) e^{-ix\xi} dx$$

- A one pixel photography is :

  an evaluation of the Poisson process $\mathbf{P}_l$ associated to the intensity field l at $[t_1, t_2] \times [x - \frac{1}{2}, x + \frac{1}{2}]$

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

Exposure time

Normalized pixel sensor centered at x

- A photography is the observation of

$$obs(n) \sim \mathbf{P}_l\left([0,\Delta t]\times[n-\frac{1}{2},n+\frac{1}{2}]\right)$$
$$\sim \mathcal{P}\left(\int_0^{\Delta t}\int_{n-\frac{1}{2}}^{n+\frac{1}{2}} l(t,y)dydt\right) \quad \text{for } n\in\mathbb{Z}$$

$$\mathbb{P}\left(obs(n)=k\right)=\frac{\left(\int_0^{\Delta t}\int_{n-\frac{1}{2}}^{n+\frac{1}{2}} l(t,y)dydt\right)^k e^{-\left(\int_0^{\Delta t}\int_{n-\frac{1}{2}}^{n+\frac{1}{2}} l(t,y)dydt\right)}}{k!}.$$

- The pixel sensors are disjoint

$obs(n)$ are in independent Poisson random variables

$$\mathbb{P}\left(\{obs(n)=k\}\cap\{obs(m)=l\}\right)=\mathbb{P}\left(\{obs(n)=k\}\right)\mathbb{P}\left(\{obs(m)=l\}\right)\ \ (n\neq m).$$

- The continuous observed landscape $e(x)$ is defined as the Shannon-Whittaker interpolated of the observed values $obs(n)$ known $\text{for } n \in \mathbb{Z}$

$$e(x) = \sum_{k \in \mathbb{Z}} obs(k) sinc(x - k)$$

Thus $\hat{e}(\xi) = \sum_{k \in \mathbb{Z}} e(k) e^{-ik\xi} \mathbb{1}_{[-\pi,\pi]}(\xi)$

and $e(n) = obs(n) \; \forall n \in \mathbb{Z}$.

$$sinc(x) = \frac{sin(\pi x)}{\pi x}$$

- u being band-limited such that

$$\hat{u}(\xi) \text{ supported on } [-\pi, \pi]$$

if $obs(n) = u(n)$ one can deduce

$$e(x) = u(x) \text{ from } e(n) = u(n) \text{ for } n \in \mathbb{Z}$$

Unfortunately this is only the case when the SNR is infinite requiring an infinite exposure time as we shall see.

- The fundamental principle of photography :

  Let u be an ideal landscape then the SNR at pixel n of $e(n) = obs(n)$

  $$\sqrt{N\Delta t u(n)}$$

  where $N\Delta t$ is the exposure time

$$\mathbb{E}(\mathbb{u}_{est}(n)) = u(n) \qquad SNR(X) := \frac{|\mathbb{E}X|}{\sqrt{var(X)}}$$

$$var(\mathbb{u}_{est}(n)) = \frac{N\Delta t u(n)}{(N\Delta t)^2} = \frac{u(n)}{N\Delta t}$$

- Proof :

$$obs(n) \sim \mathbf{P}_l([0, N\Delta t] \times [n - \frac{1}{2}, n + \frac{1}{2}]) \sim \mathcal{P}\left(N\Delta t u(n)\right)$$

Let $\mathbb{u}_{est}(n) \sim \frac{\mathcal{P}\left(N\Delta t u(n)\right)}{N\Delta t}$ be the estimated landscape

$\mathbb{u}_{est}(n)$ is an unbiased estimator : $\mathbb{E}(\mathbb{u}_{est}(n)) = u(n)$

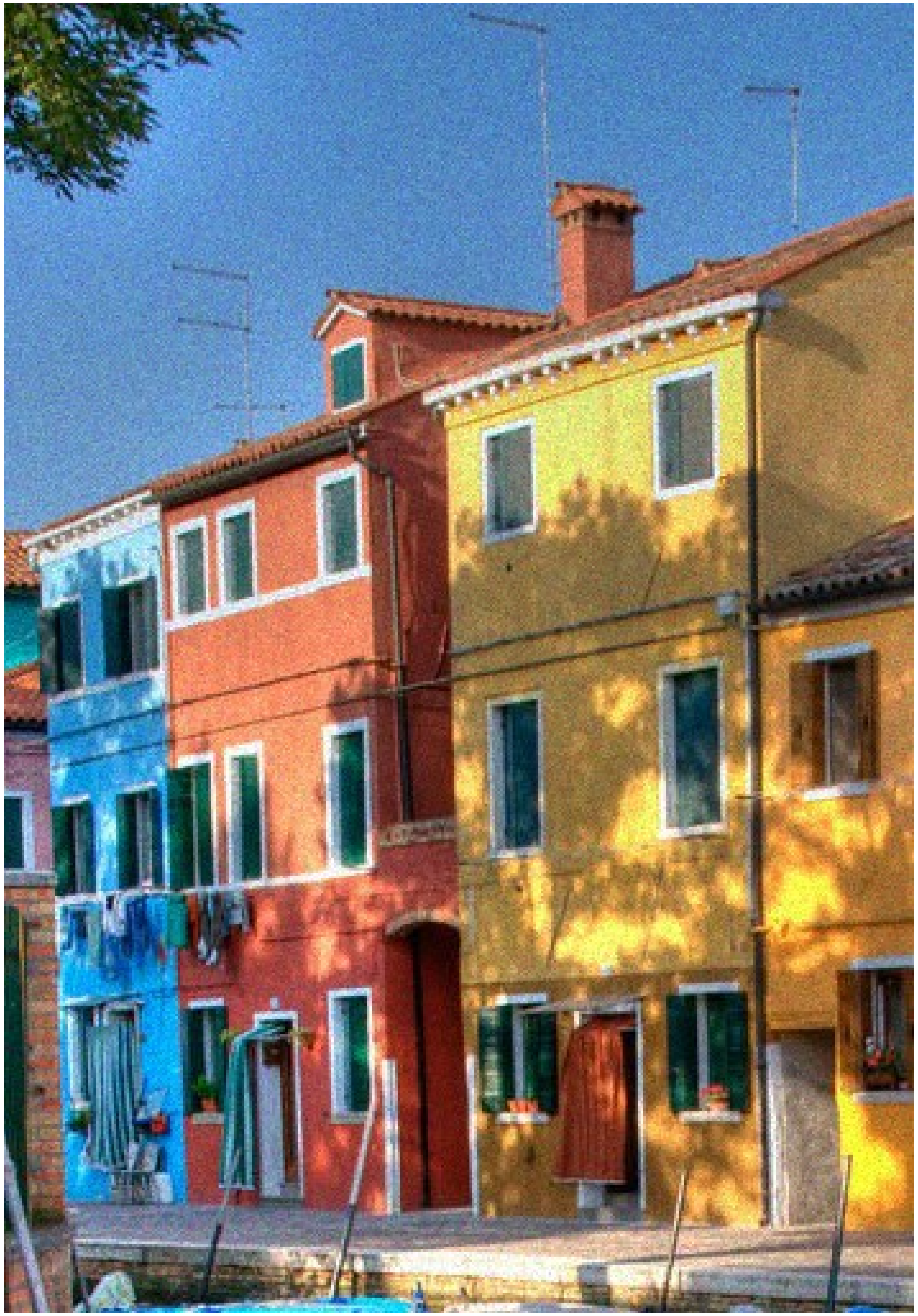
$$var(\mathbb{u}_{est}(n)) = \frac{N\Delta t u(n)}{(N\Delta t)^2} = \frac{u(n)}{N\Delta t}$$

$$SNR(\mathbb{u}_{est}(n)) = \sqrt{N\Delta t u(n)}$$

Remark : in passive systems the only way to increase the SNR is to increase $N\Delta t$

Exposure time= 1 , SNR=100, numbers of photons (average) : 10^4.

Exposure time= 10^{-1},SNR=33, numbers of photons (average) : 10^3.



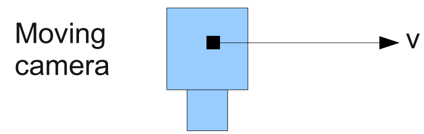
Exposure time= 10^{-2} SNR=10, numbers of photons (average) :10^2 .

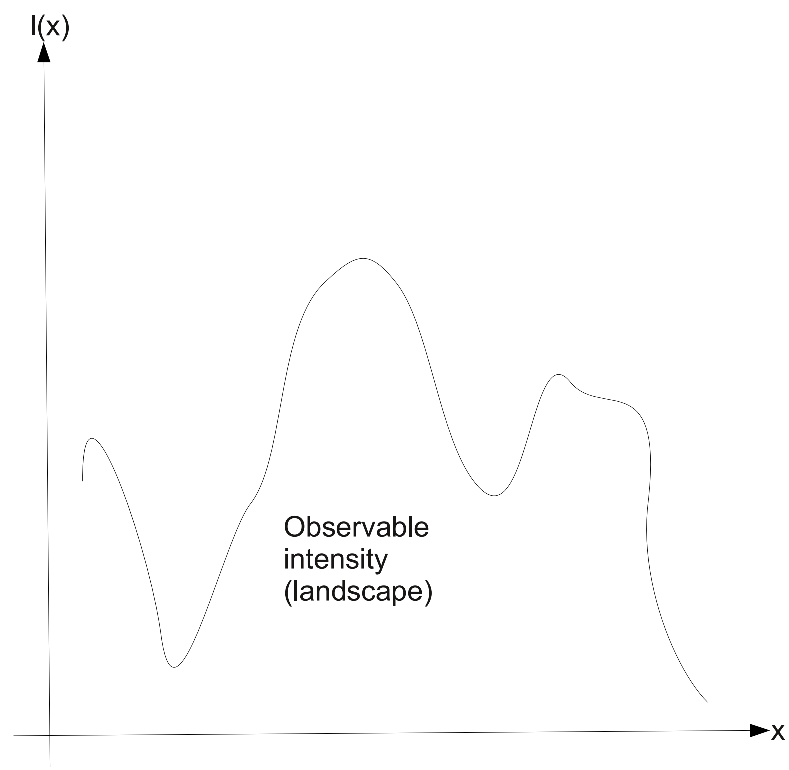
Exposure time= 10^{-3} SNR=3, numbers of photons (average) : 10^1.

Exposure time=10^{-4}, SNR=1, numbers of photons (average) : 1.

Exposure time= 10^{-5} SNR=0.3, numbers of photons (average) : 0.1 .

- Moving landscape





From now on u(x)=u(x-vt)

- Motion blur

  For a moving landscape at velocity v the observed value at pixel x is

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

  the image obtained by a convolution of the ideal landscape u and a window shaped kernel

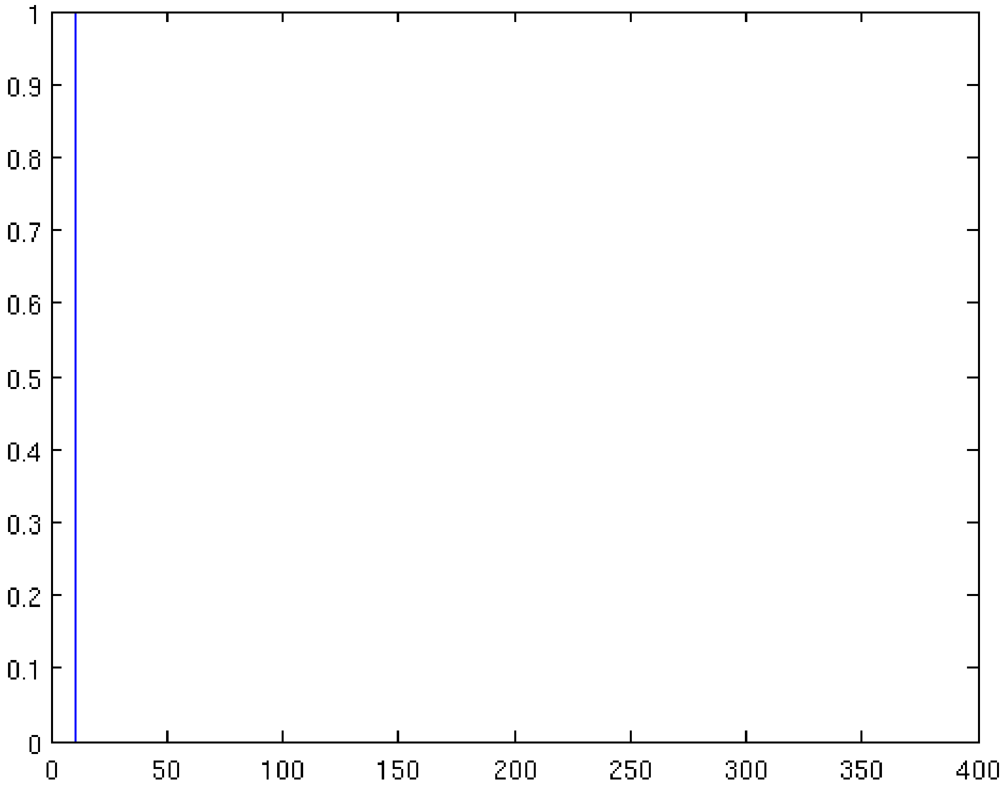
- Motion blurs are not invertible in general as

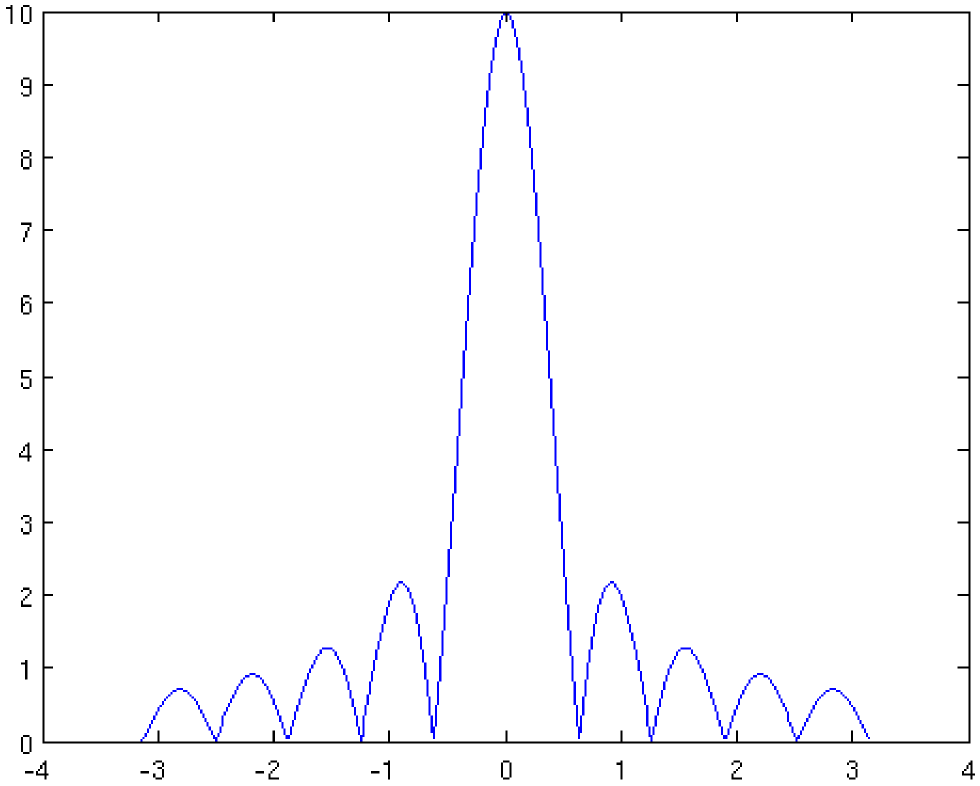
$$\mathcal{F}(\mathbb{1}_{[0,v\Delta t]})(\xi) = \int_0^{v\Delta t} e^{-ix\xi} dx = 2\frac{sin(\frac{\xi v\Delta t}{2})}{\xi} e^{-i\xi\frac{v\Delta t}{2}}$$

has zero(es) in the support of $\hat{u}(\xi)\,:\,[-\pi,\pi]$

if $|v|\Delta t \geq 2$

- Motion blur of 10 pixels (kernel)





The blur function : 10 pixels blur (left), the modulus of it's Fourier transform : only a few zeroes (right).

- Classic solution : the snapshot

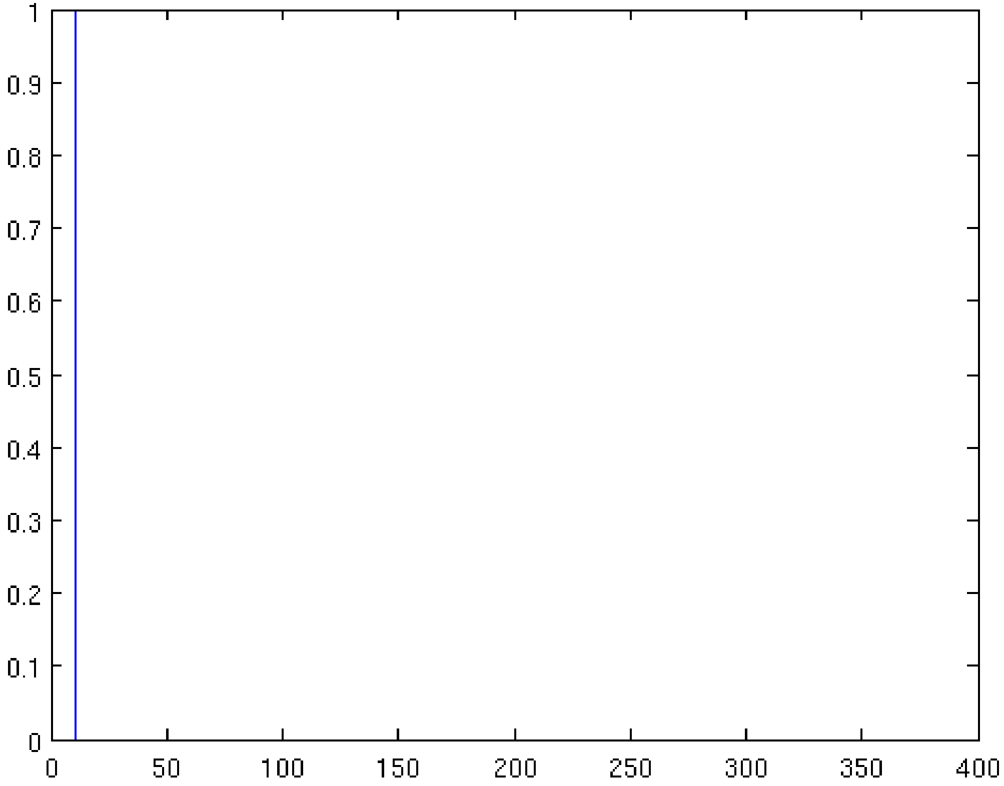
  Namely the use of an exposure time Δt such that $|v|\Delta t < 2$

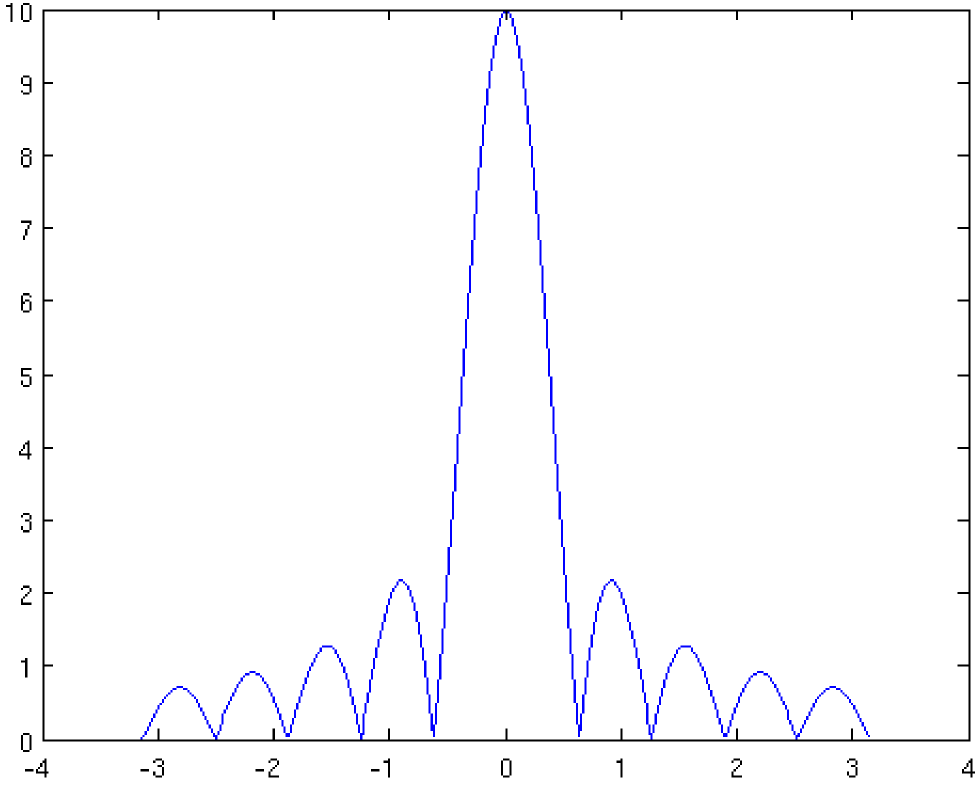
  guaranteeing no zero of

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

  in the support of $\hat{u}(\xi) : [-\pi, \pi]$

• Effect of a blur of 10 pixels, SNR=100.





The blur function : 10 pixels blur (left), the modulus of it's Fourier transform : only a few zeroes (right).

From left to right : the landscape, the observed, the restored (bestial).

Simulation algorithm of snapshot and deconvolution

1) Take a landscape

2) Convolve with the blur function to obtain the blurry landscape (intensities for step 3)

3) Simulate the observed : simulate Poisson r.v.

4) Deconvolution : Wiener filter with oracle

$$\hat{w}(\xi) = \frac{\hat{\alpha}(\xi)^*}{|\hat{\alpha}(\xi)|^2 + \frac{|\hat{\eta}(\xi)|^2}{|\hat{u}(\xi)|^2}} \text{ where } \eta = obs - u$$

$$\text{and } \alpha(t) = \mathbb{1}_{[0,v\Delta t]}(t)$$

The Wiener filter is the optimum of

$$\mathbb{E}|obs * w - u|^2 \text{ when } obs = u * \alpha + \eta$$

From left to right : the landscape, the observed, the restored (oracle Wiener filtering).

- Thus we cannot control the SNR anymore when $\Delta t < \dfrac{2}{|v|}$

  leading to a poor SNR particularly if |v| is big

- But...

  someone found a solution to use arbitrary Δt

Mitsubishi Electric Develops Deblurring Flutter Shutter Camera

by Karen M. Cheung DIGG.COM MY YAHOO! ADD DEL.ICIO.US

August 30, 2006 - Following this month's 33rd Annual Siggraph Conference in Boston, MA, a research team at Mitsubishi Electric is catching the attention of camera manufacturers for their photo motion deblurring technology, called a flutter shutter camera.

The flutter shutter camera is a modified camera that can capture moving objects at an exposure time of over 50 milliseconds, like high speed motion cameras. Using a coded exposure sequence, the new flutter shutter camera could recover text from a speeding car and sharpen images, according to the researchers.

Introduced in early August, three Mitsubishi Electric researchers presented the abstract, "Coded Exposure Photography: Motion Deblurring using Fluttered Shutter" at the largest computer and graphics conference, Siggraph. After one year of research development, Mitsubishi Electric Research Lab (MERL) senior researcher Ramesh Raskar, MERL visiting researcher Amit Agrawal, and Northwestern University computer science assistant professor Jack Tumblin launched the new prototype with the goal of deblurring photos.

The prototype is made with an 8 megapixel Canon PowerShot Pro1, although it could be applied to any camera. Instead of leaving the shutter open during one exposure duration, the camera's attached lens filter flutters the shutter multiple times during a single exposure, based on a carefully chosen binary sequence.

Raskar, who celebrates his sixth year at Mitsubishi this month, woke up one day with the idea, according to the researcher. Raskar said the fluttered shutter method was "so simple" and wondered if it could work.

Above: Mitsubishi Electric Research Lab (MERL) senior researcher Ramesh Raskar with a flutter shutter prototype

"We have UV filters. We have polarizing filters. What about time filters?" said senior research scientist Ramesh Raskar at Mitsubishi Electric. Just a few weeks ago, Raskar and the Mitsubishi team saw the dream actualized with its official introduction. This time filter is made up of an external ferro-electric shutter that flutters based on a rapid binary sequence.

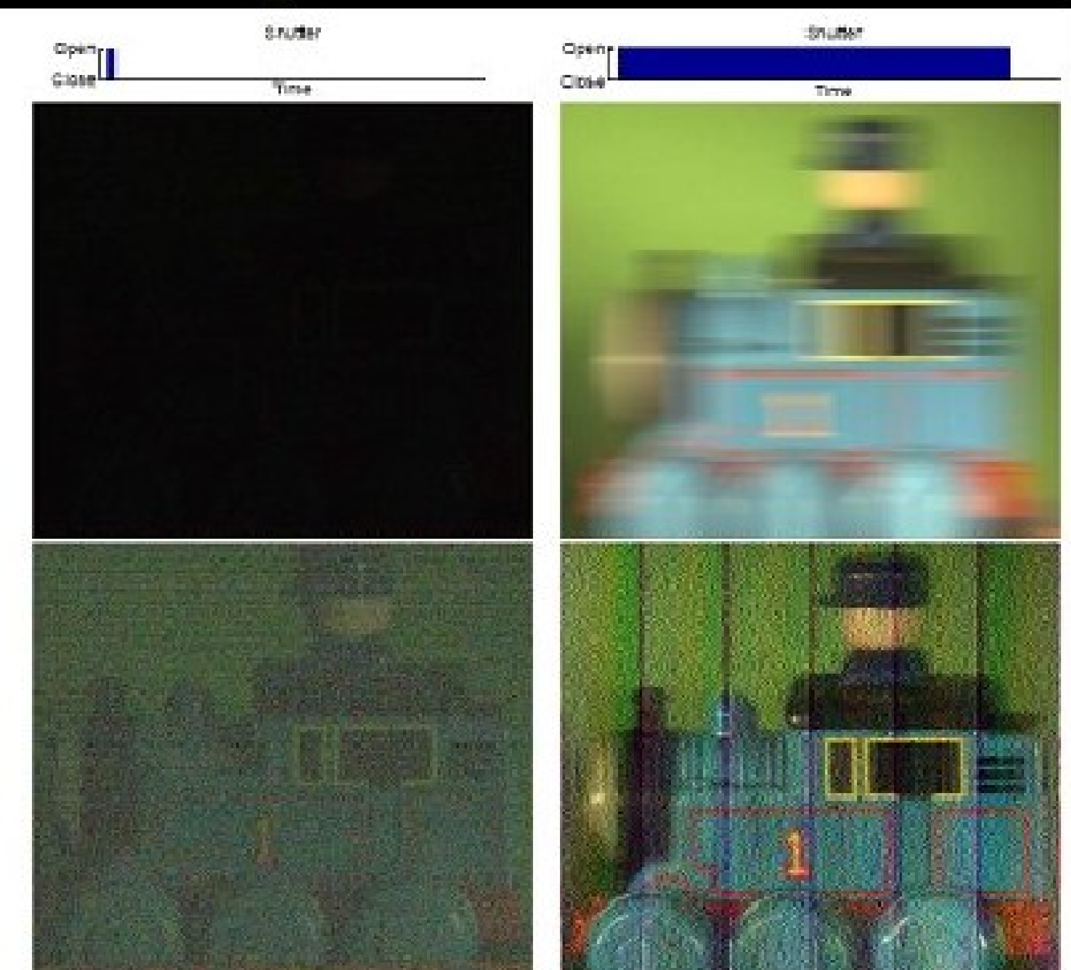
Traditional cameras typically have a single shuttered exposure in which moving subjects result in blurry images. On standard cameras, object motion can be described by convolution of a sharp image with a temporal box filter, according to researchers, thus destroying high frequency spatial details in the image. The coded exposure camera, on the other hand, uses a method called deconvolution, in which the convolution filter changes to a broadband filter and preserves spatial frequencies and image detail.

A post-capture linear system algorithm is then applied to recover image sharpness. The algorithm is "fundamentally different from other deblurring algorithms," stated authors of the abstract. The algorithm, Ax=b (a Matlab Code), is the simplest deconvolution algorithm possible, according to researchers. Unlike Photoshop or other deblurring methods, this deconvolution does not result in ringing or deconvolution artifacts, halo-like distortions on the image.

Coded Exposure Photography: Assisting Motion Deblurring using Fluttered Shutter

Raskar, Agrawal, Tumblin (Siggraph2006)

Short Exposure | **Traditional** | **Coded**



← Shutter →

← Captured Photos →

← Deblurred Results →

Image is dark and noisy

Result has Banding Artifacts and some spatial frequencies are lost

Decoded image is as good as image of a static scene

The camera can even capture images from a flying plane, according to Raskar. With large-scale aerial photography applications, such as Google Maps, the camera can use an aperture ten times smaller than currently used on the expensive aerial cameras. For planes mapping the entire earth, this would reduce time and camera costs significantly.

Some may wonder why not use a short shutter speed, such as those on action and sport modes for shooting fast-moving

Fluttered Shutter Camera

Raskar, Agrawal, Tumblin Siggraph2006

Ferroelectric shutter in front of the lens is turned opaque or transparent in a rapid binary sequence

- The numerical flutter shutter setup

  0) k=0

  1)Take the k-th image using exposure time Δt

  2) Multiply it by a weight in $\alpha_k \in \mathbb{R}$

  3) Add to the previous

  4) Go to 1) or stop after N iterations

- The k-th elementary image at pixel n is

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

- Thus the output at pixel n is

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

Definition

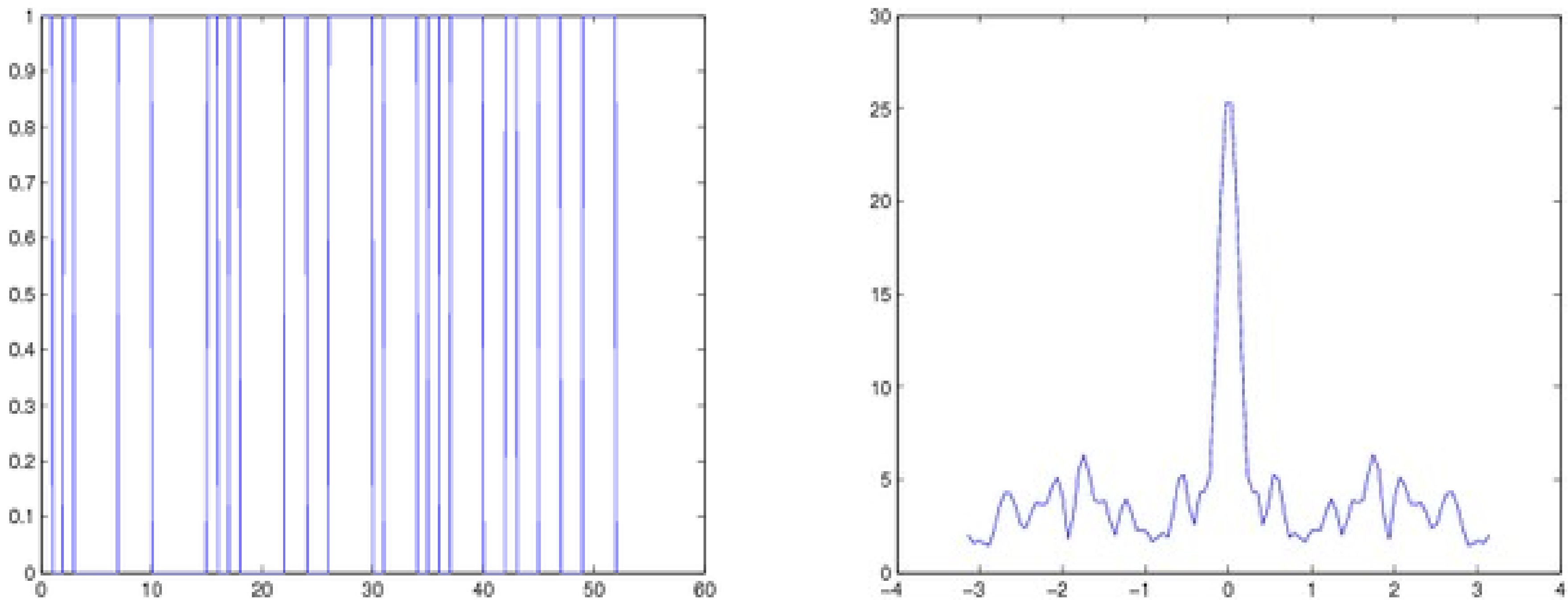
We call flutter shutter code the vector

$$(\alpha_k)_{k=0,\cdots,N-1}$$

We call flutter shutter function

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

- Example of flutter shutter function



The Raskar function and the modulus of its Fourier transform.

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

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

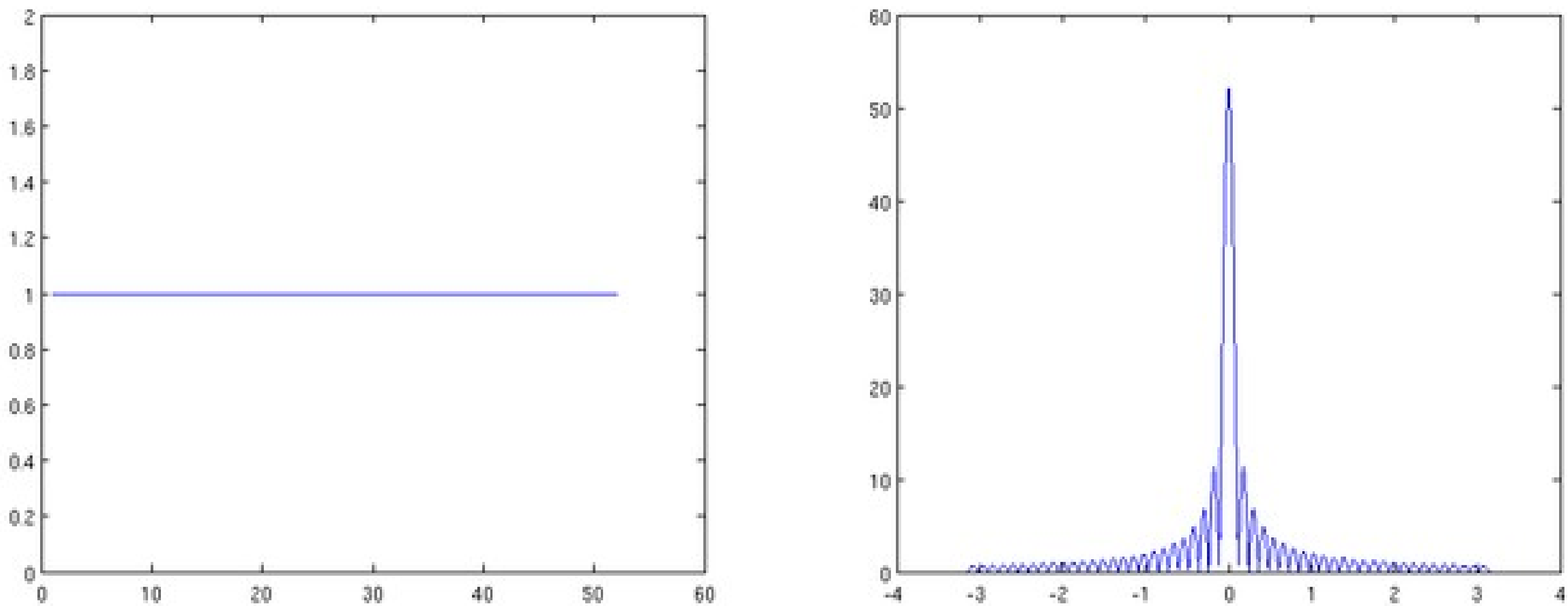
We call *numerical flutter shutter* its band limited interpolate

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

- Examples :

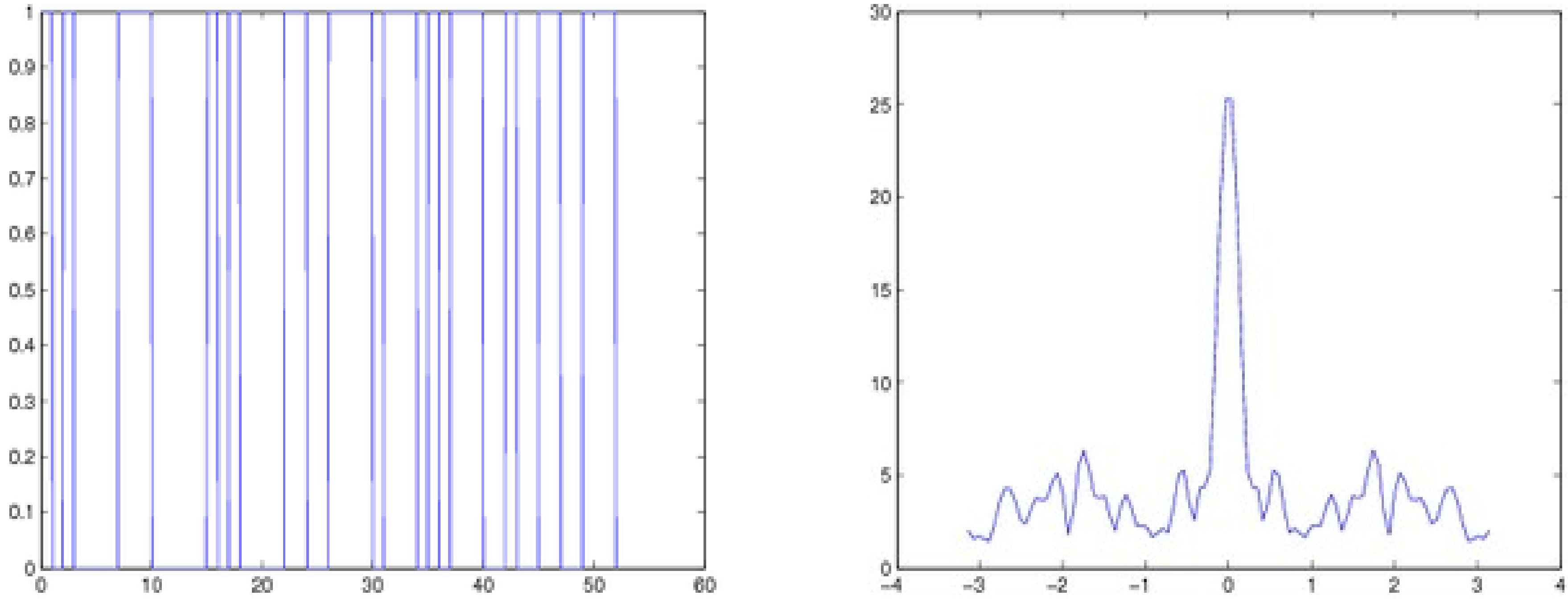
1. $\alpha_k = 1 \; \forall k \in \{0, ..., N-1\}$ (pure accumulation prone to motion blur)
2. $\alpha_k = 0$ or $1 \; \forall k \in \{1, ..., N-1\}$ with $\sum \alpha_0 = \frac{N}{2}$ (Raskar's *flutter shutter* has this generic form)
3. $\alpha_0 = 1$ and $\alpha_k = 0 \; \forall k \in \{1, ..., N-1\}$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

- Example 1 : accumulation



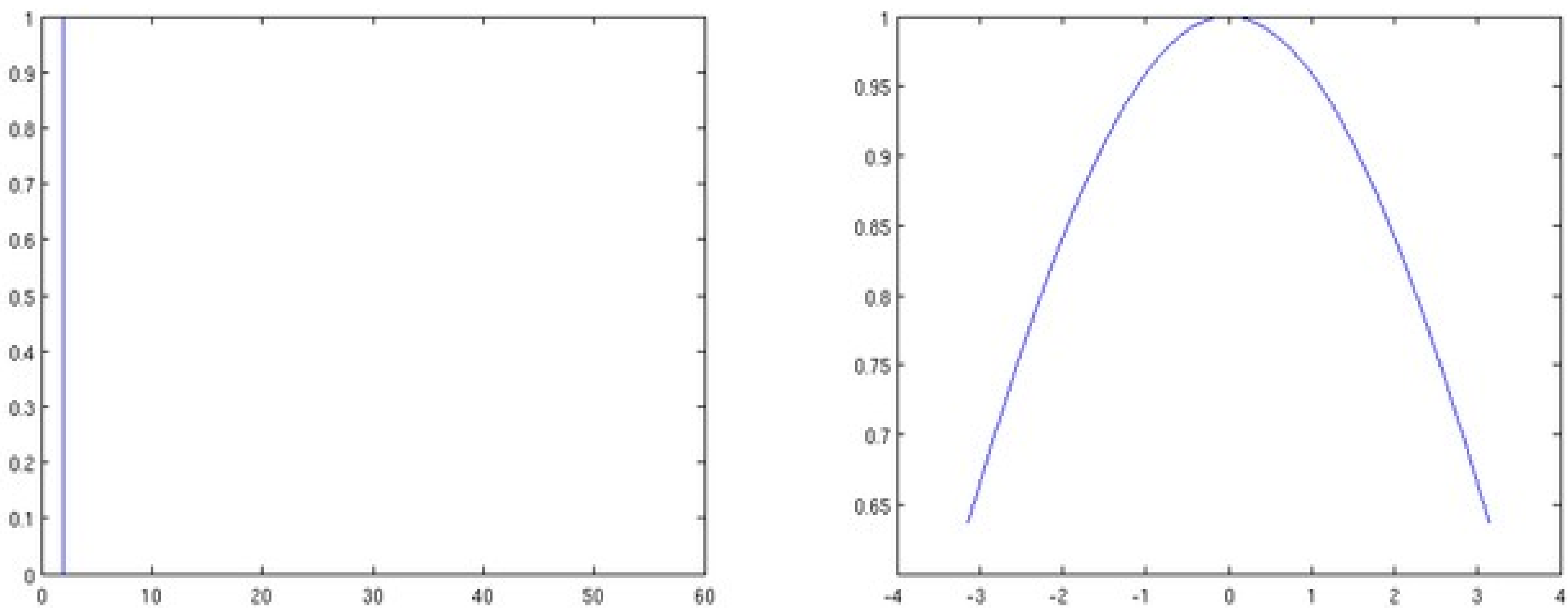
The accumulation function and the modulus of its Fourier transform.

- Example 2 : Raskar's code



The Raskar function and the modulus of its Fourier transform.

- Example 3 : the standard snapshot



The standard snapshot function and the modulus of its Fourier transform.

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

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

and

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

From the *numerical flutter shutter* samples definition

$$\begin{aligned}
\mathbb{E}\left(obs(x)\right) &= \sum_{k=0}^{N-1} \alpha_k \int_{k\Delta t}^{(k+1)\Delta t} u(x - vt)dt \\
&= \sum_{k=0}^{N-1} \alpha_k \int_{k}^{(k+1)} \Delta t u(x - v\Delta t s)ds \\
&= \int_{0}^{N} \Delta t \alpha(s) u(x - v\Delta t s)ds
\end{aligned}$$

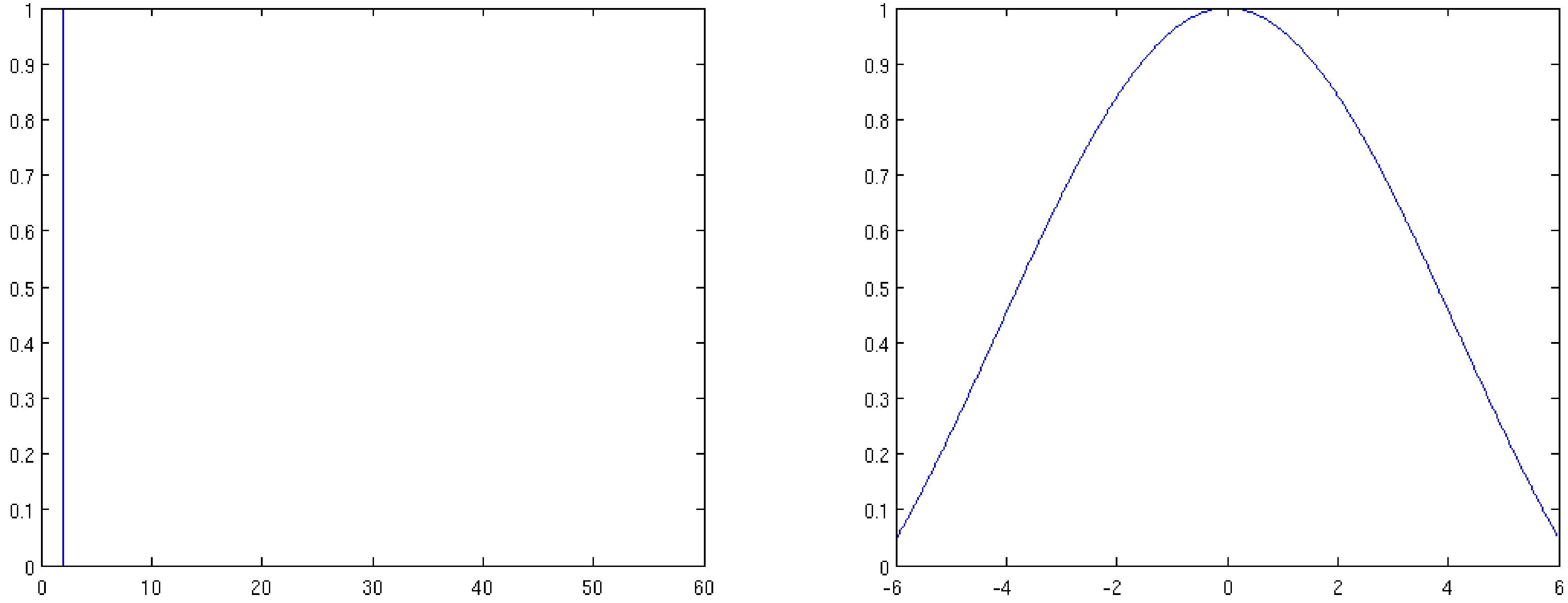
where $\alpha = \sum_{k=0}^{N-1} \alpha_k \mathbb{1}_{[k,k+1[}$. Thus,

$$\begin{aligned}
\mathbb{E}\left(obs(x)\right) &= \int_{0}^{Nv\Delta t} \frac{1}{v}\alpha(\frac{y}{v\Delta t})u(x-y)dy \\
&= \int_{-\infty}^{+\infty} \frac{1}{v}\alpha(\frac{y}{v\Delta t})u(x-y)dy \\
&= \left(\frac{1}{v}\alpha(\frac{\cdot}{v\Delta t}) * u\right)(x).
\end{aligned}$$

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

Remark that $obs(x)$ is not necessarily a Poisson random variable.

- Experiment : a 1.9 pixel blur



A 1.9 pixel blur snapshot function and it the modulus of its Fourier transform.

The observed (left) : snapshot blur of 1.9 pixel (left), SNR 50. The restored (middle), RMSE=9.8. The residual noise (right).

$$RMSE\,(u, \mathbb{u}_{est}) := \sqrt{\frac{\int_D |u(x) - \mathbb{u}_{est}(x)|^2 dx}{measure(D)}}$$

- Inverse filter

Step 1: the noiseless case when

$obs(n) = \left(\frac{1}{v}\alpha(\frac{\cdot}{v\Delta t}) * u\right)(n)$ (obtained for $n \in \mathbb{Z}$)

is band limited, thus can be interpolated to $obs(x)$, for $x \in \mathbb{R}$.

Then

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

By hypothesis : $\hat{u}(\xi) = 0$ for $|\xi| > \pi$.

Hence for the invertibility we must require $|\hat{\alpha}(\xi v\Delta t)| > 0$ for $\xi \in [-\pi, \pi]$

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

For an invertible flutter, consider the inverse filter γ defined by

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

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

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

We observe $e(n) = \mathbb{E}(obs(n))$ for $n \in \mathbb{Z}$ the deterministic part of $obs(n)$.
We wish to compute $\hat{e}(\xi)$ from $(e(n))_{n \in \mathbb{Z}}$.
$e(x)$ is band limited, we interpolate it using the sinc-interpolation:

$$e(x) = \sum_{n \in \mathbb{Z}} e(n) sinc(x - n), \text{ thus } \hat{e}(\xi) = \sum_{n \in \mathbb{Z}} e(n) e^{-in\xi} \mathbb{1}_{[-\pi,\pi]}(\xi). \tag{2}$$

So the ideal deconvolved landscape $d(x)$ obtained by combining (1) and (2) is

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

Step 2: the noisy case

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

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

- Now we have a model for the observed
- We wish to compute the SNR
- And compare the SNR for different strategies (flutter shutter functions)

Definition (Signal to noise ratios)
If $\hat{\mathbb{u}}_{est}(\xi)$ is an estimation of $\hat{u}(\xi)$ based on a noisy observation We call "spectral SNR" of $\mathbb{u}_{est}$ the frequency dependent ratio defined by

$$SNR^{spectral}(\hat{\mathbb{u}}_{est}(\xi)) := \frac{|\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)|}{\sqrt{(var(\hat{\mathbb{u}}_{est}(\xi))}} \qquad \text{for } \xi \in [-\pi, \pi]; \tag{3}$$

$$SNR^{spectral-averaged}(\hat{\mathbb{u}}_{est}) := \frac{\frac{1}{2\pi}\int |\mathbb{E}\hat{\mathbb{u}}_{est}(\xi)| \mathbb{1}_{[-\pi,\pi]}(\xi) d\xi}{\sqrt{\frac{1}{2\pi}\int var(\hat{\mathbb{u}}_{est}(\xi)) \mathbb{1}_{[-\pi,\pi]}(\xi) d\xi}}. \tag{4}$$

Theorem *The numerical flutter shutter has a spectral SNR equal to*

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

Step0. (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 $\sum_{n\in\mathbb{Z}} f(n) = \hat{f}(0) = ||f||_{L^1}$.

Step1. (Variance of $F(\mathbb{u}_{est,num})(\xi)$)

$$
\begin{aligned}
var(\mathcal{F}(\mathbb{u}_{est,num})(\xi)) &= \frac{var\left(\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} = \frac{\sum_{n\in\mathbb{Z}} var(obs(n))|e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)|^2}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} \\
&= \frac{\sum_{n\in\mathbb{Z}} \left(\frac{1}{v}\alpha^2(\frac{\cdot}{v\Delta t}) * u\right)(n)\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} \\
&= \frac{||\frac{1}{v}\alpha^2(\frac{\cdot}{v\Delta t}) * u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} \qquad \text{(by Poisson summation formula)} \\
&\qquad\qquad (5) \\
&= \frac{\frac{1}{v}||\alpha^2(\frac{\cdot}{v\Delta t})||_{L^1}||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} = \frac{\frac{1}{v}||\alpha(\frac{\cdot}{v\Delta t})||_{L^2}^2||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} \\
&= \frac{\frac{v\Delta t}{v}||\alpha||_{L^2}^2||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t^2|\hat{\alpha}(\xi v\Delta t)|^2} = \frac{||\alpha||_{L^2}^2||u||_{L^1}\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t|\hat{\alpha}(\xi v\Delta t)|^2}.
\end{aligned}
$$

Step2. (Expected value of $F(\mathbb{u}_{est,num})(\xi)$)

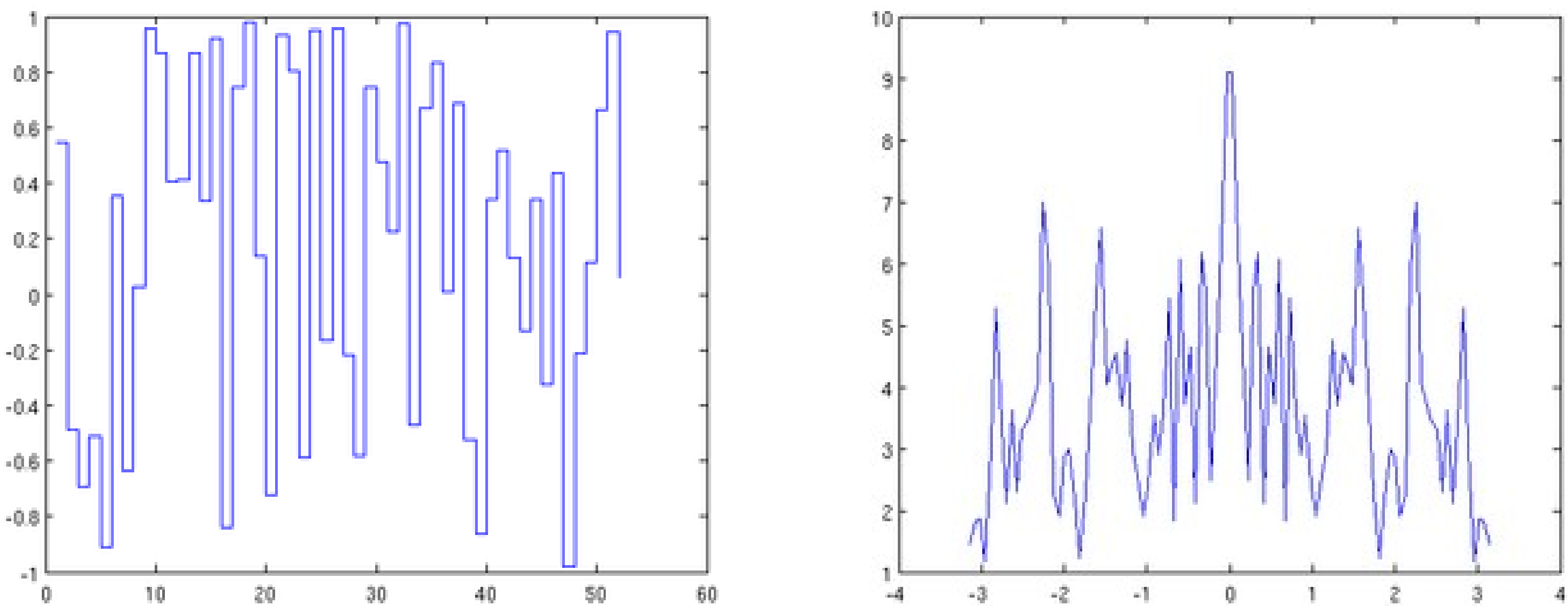
$$\mathbb{E}(\mathcal{F}(\mathbb{u}_{est,num}(\xi))) = \frac{\left(\mathbb{E}\sum_{n\in\mathbb{Z}} obs(n)e^{-in\xi}\mathbb{1}_{[-\pi,\pi]}(\xi)\right)}{\Delta t\hat{\alpha}(\xi v\Delta t)} = \frac{\sum_{m\in\mathbb{Z}}\mathcal{F}\left(\frac{1}{v}\alpha(\frac{\cdot}{v\Delta t}) * u\right)(\xi+2\pi m)\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta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

$$= \frac{\mathcal{F}\left(\frac{1}{v}\alpha(\frac{\cdot}{v\Delta t}) * u\right)(\xi)\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t\hat{\alpha}(\xi v\Delta t)} = \frac{\frac{v\Delta t}{v}\hat{\alpha}(\xi v\Delta t)\hat{u}(\xi)\mathbb{1}_{[-\pi,\pi](\xi)}}{\Delta t\hat{\alpha}(\xi v\Delta t)} = \hat{u}(\xi)\mathbb{1}_{[-\pi,\pi]}(\xi).$$

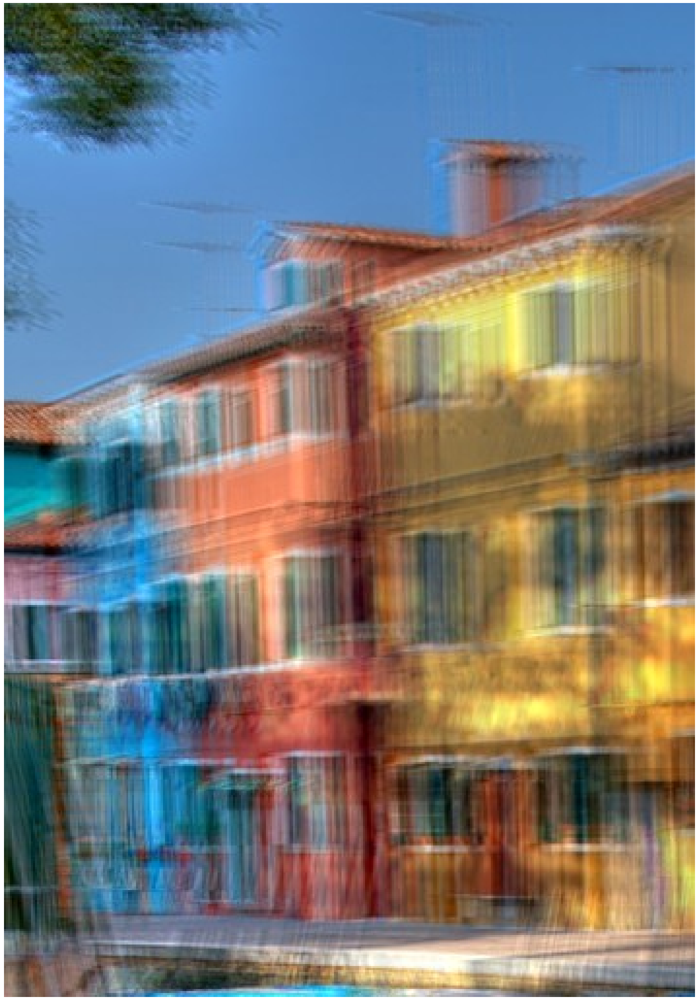
Thus the definition of the spectral SNR (3) entails

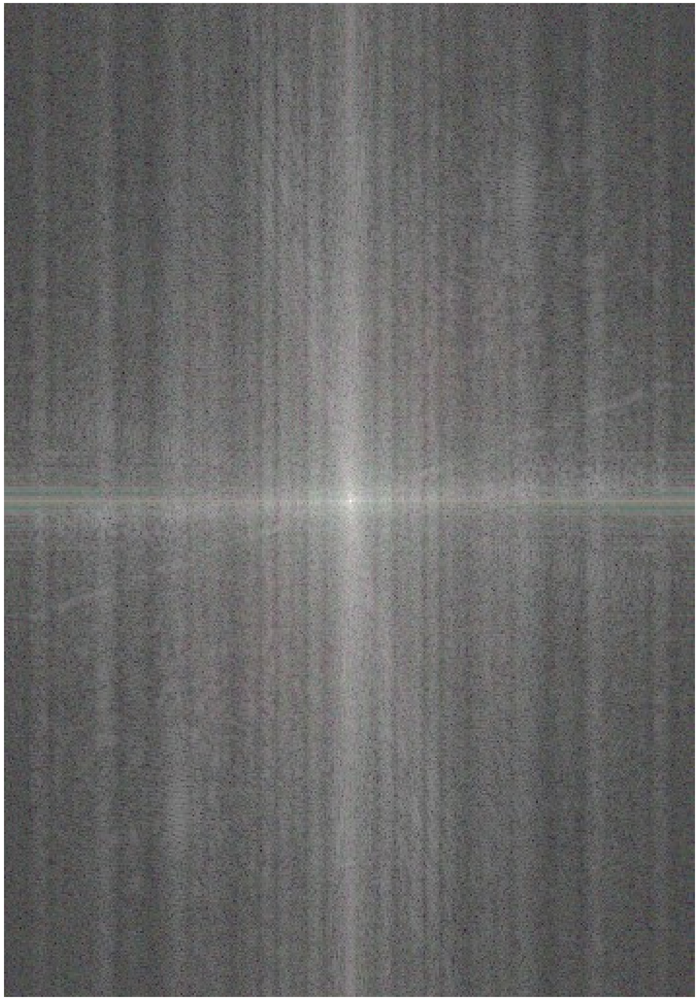
$$SNR^{spectral}(\mathbb{u}_{est,num}(\xi)) = \mathbb{1}_{[-\pi,\pi]}(\xi)\frac{\sqrt{\Delta}t|\hat{u}(\xi)||\hat{\alpha}(\xi v\Delta t)|}{\sqrt{||u||_{L^1}}||\alpha|||_{L^2}}.$$

- Experiment : a random code

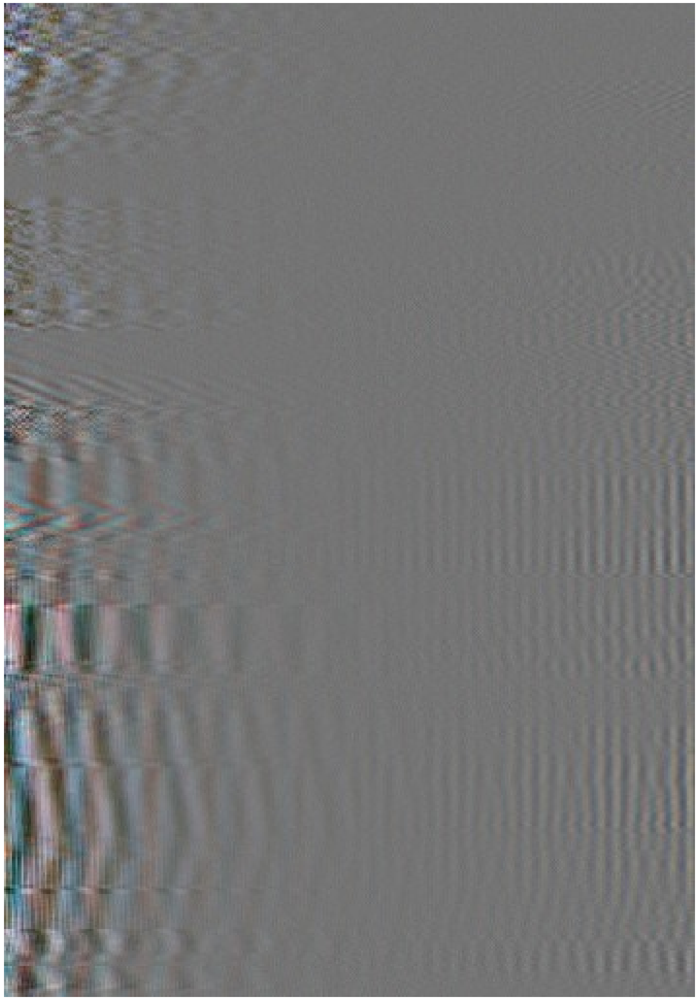


The rand-code function and it the modulus of its Fourier transform.





The observed : rand-code blur of 52 pixels (left). The log-modulus of its DFT.



The restored (left), the residual noise (right), RMSE=2.19.

- White noise deconvolution (Raskar's code)

- White noise deconvolution (rand code)

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

Proof. Among all gain control functions $\alpha(t)$ one giving the best $SNR^{spectral-averaged}$ (4) is given by minimizing the averaged variance of $\hat{\mathbb{u}}_{est}$

$$F(\alpha) = \frac{||\alpha||_2^2}{2\pi\Delta t} \int_{-\pi}^{\pi} \frac{d\xi}{|\hat{\alpha}(\xi v \Delta t)|^2} \text{ (dropping the constant in } u)$$

$$= \frac{||\alpha||_2^2}{\Delta t} \int_{-\pi v\Delta t}^{\pi v \Delta t} \frac{1}{|\hat{\alpha}(\xi)|^2} \frac{d\xi}{2\pi v \Delta t} \geq \frac{||\alpha||_2^2}{\Delta t} \frac{1}{\int_{-\pi v\Delta t}^{\pi v\Delta t} |\hat{\alpha}(\xi)|^2 \frac{d\xi}{2\pi v\Delta t}}.$$

Crucial point : Jensen formula. Since $x > 0 \mapsto \frac{1}{x}$ is strictly convex the equality occurs when $|\hat{\alpha}(\xi)|^2 = \mathbb{1}_{[-\pi v\Delta t, -\pi v \Delta t]}(\xi)$ up to a constant renormalization. This leads to $\alpha^*(t) = sinc(tv\Delta t)$ (up to a normalization constant). $\square$

Corollary (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 (3) 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 (4) 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. (Parseval formula : $||f||^2_{L^2} = \frac{1}{2\pi}||\mathcal{F}(f)||^2_{L^2}$)
From the ideal flutter shutter function and using the Parseval formula we deduce $||\alpha^*||^2_{L^2} = v\Delta t$.
Then from numerical flutter shutter SNR we deduce that $SNR^{spectral}(\xi) = \frac{\mathbb{1}_{[-\pi,\pi]}(\xi)}{\sqrt{v}} \frac{|\hat{u}(\xi)|}{\sqrt{||u||_{L^1}}}$. Moreover we have

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

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

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

Corollary *(Snapshot SNR)*
For a standard snapshot using a exposure time of Δt the spectral SNR (3) is

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

Proof. Direct consequence of the numerical flutter shutter SNR and

$$\mathcal{F}(\mathbb{1}_{[0,v\Delta t]})(\xi) = \int_0^{v\Delta t} e^{-ix\xi}dx = 2\frac{sin(\frac{\xi v\Delta t}{2})}{\xi}e^{-i\xi\frac{v\Delta t}{2}}. \quad (11)$$

$\square$

Theorem (Best exposure time for landscape recovery)
Consider a landscape $u(x-vt)$ moving at velocity v. Then for a snapshot the $SNR^{spectral-averaged}$ (4) is maximized when $v\Delta t^* \approx 1.0909$" and is equal to

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

Proof. Similarly to the ideal flutter shutter function, the energy to be minimized in order to guarantee the best $SNR^{spectral-averaged}$ *after* deconvolution is

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

Then

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

Which vanishes when $b^* = v\Delta t^* \approx 1.0909$.

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

Corollary (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

$$R = \frac{SNR^{spectral-average}(flutter, ideal)}{SNR^{spectral-average}(snapshot)} \approx 1.1715.$$

Proof. This is a direct consequence of the best snapshot SNR and ideal numerical flutter shutter SNR. $\square$

- Numerical simulation, examples

Step1.

> Compute $\tilde{u}(m,n)$ the 2D-DFT of u : For $m = -\frac{M}{2}, ..., \frac{M}{2} - 1$ and $n = -\frac{N}{2}, ..., \frac{N}{2} - 1$:
>
> $\tilde{u}(m,n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} u(k,l) \omega_M^{-km} \omega_N^{-nl}$ where $\omega_N = \exp\left(\frac{2i\pi}{N}\right)$.
>
> Compute the motion kernel a
>
> > For each $m = -\frac{M}{2}, ..., \frac{M}{2} - 1$
> >
> > For each $n = -\frac{N}{2}, ..., \frac{N}{2} - 1$
> >
> > $$a(m,n) = \frac{sin\left(\frac{\pi v n}{N}\right)}{\frac{\pi v n}{N}} \sum_{k=0}^{L-1} a_k e^{-i\left(\frac{2\pi v n}{N}\right)(k+0.5)}$$
>
> Compute the product of the 2D-DFT of u and of a(m,n)
>
> Compute the inverse DFT of the previous, store it in e(m,n)

Here e(m,n) contains the ideal noiseless observed up to the periodization effect.

> Crop the result to avoid periodization effect

Step1b.

> Compute the 2D-DFT of u
>
> Compute the "noise" kernel b
>
> > For each $m = -\frac{M}{2}, ..., \frac{M}{2} - 1$
> >
> > For each $n = -\frac{N}{2}, ..., \frac{N}{2} - 1$
> >
> > $$a(m,n) = \frac{sin\left(\frac{\pi v n}{N}\right)}{\frac{\pi v n}{N}} \sum_{k=0}^{L-1} a_k^2 e^{-i\left(\frac{2\pi v n}{N}\right)(k+0.5)}$$
>
> Compute the product of the 2D-DFT of u and of b(m,n)
>
> Compute the inverse DFT of the previous, store it in f(m,n)

Here f(n,m) contains the variance of the observed value o(m,n).

Step2.

> For each (m,n) simulate the Poisson random variable with intensity f(m,n) and the desired SNR using remark2, store it in n(m,n)
> For each (m,n) the observed o(m,n)=e(m,n)+n(m,n)-f(m,n)

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

Step3.

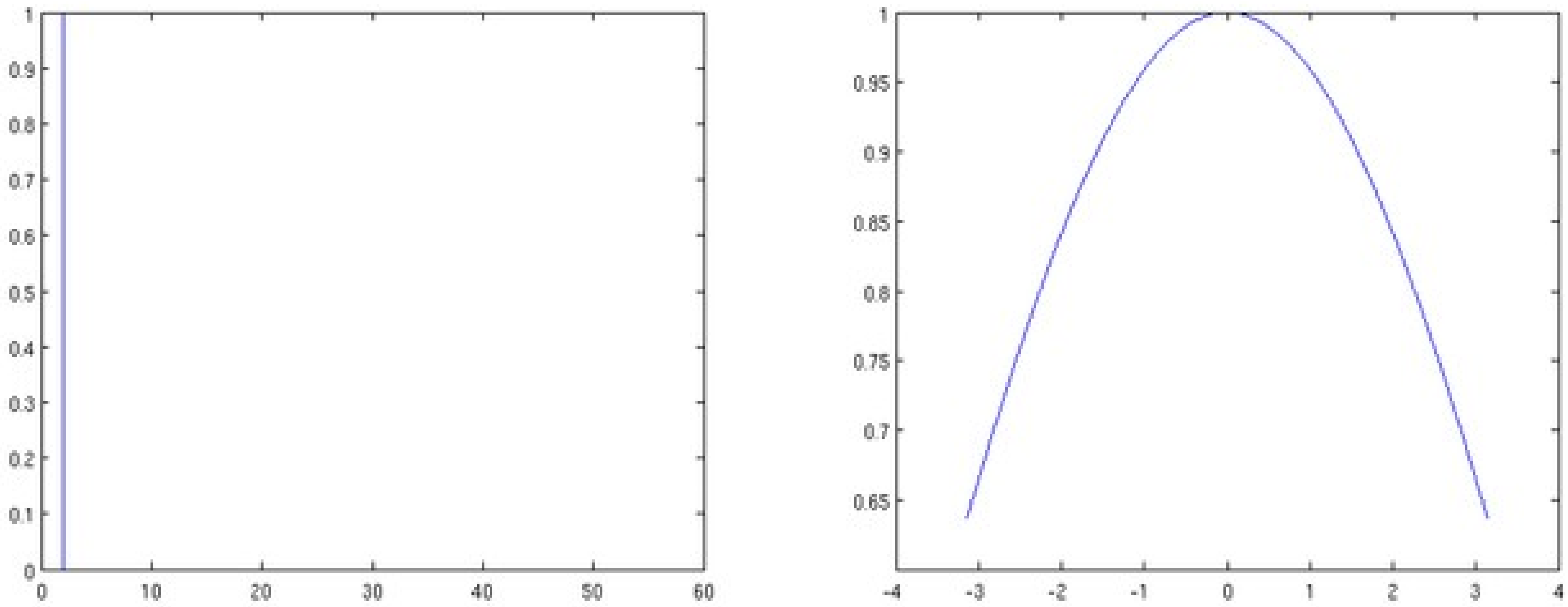
> Use classic mirror symmetry among the columns obtain u_s(m,n)
>
> Compute the 2D-DTF of u_s
>
> Compute the motion kernel like in Step2.
>
> Divide the 2D-DTF of u_s by the motion kernel
>
> Compute the inverse DFT of the previous
>
> Crop to remove the mirror symmetry

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

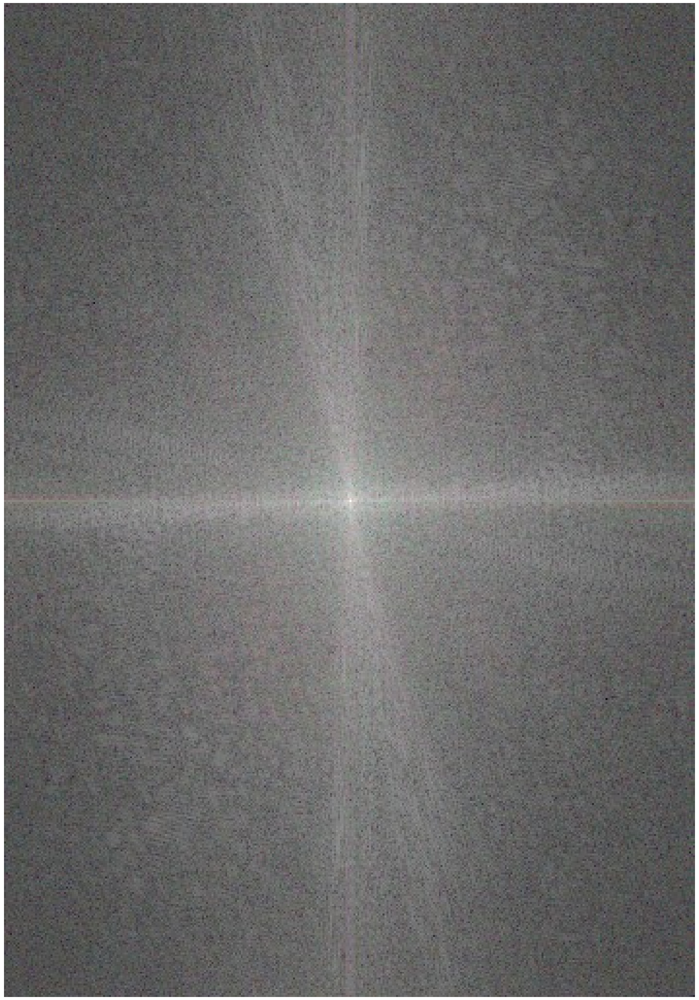
Step4.

> Compute the RMSE and RMSE_CI after cropping to avoid border effects.

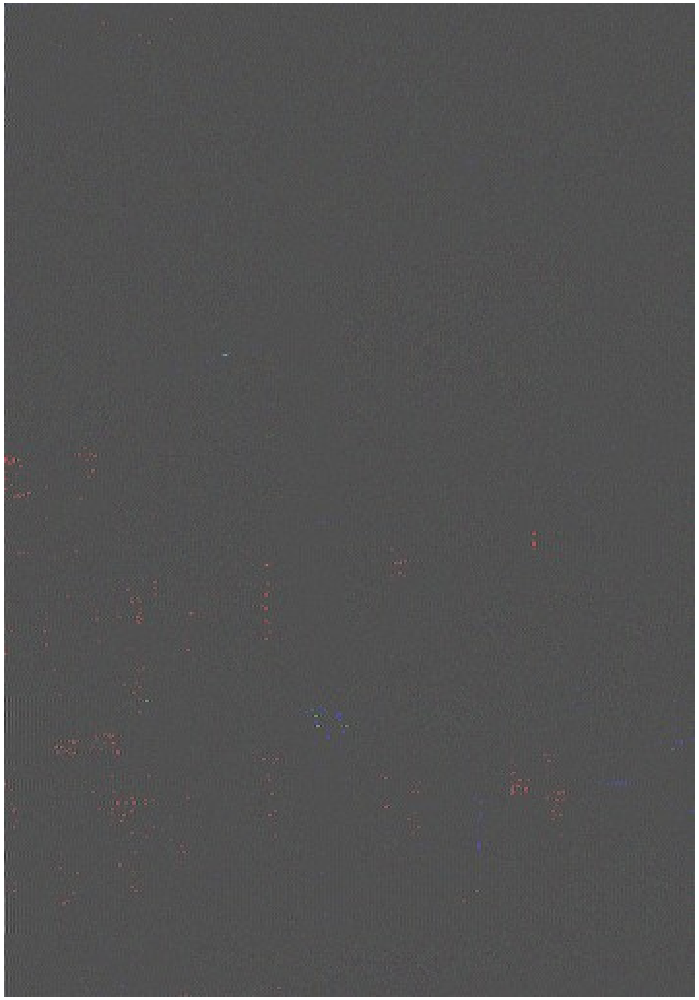
- Example 1 : the standard snapshot



The standard snapshot function and the modulus of its Fourier transform.

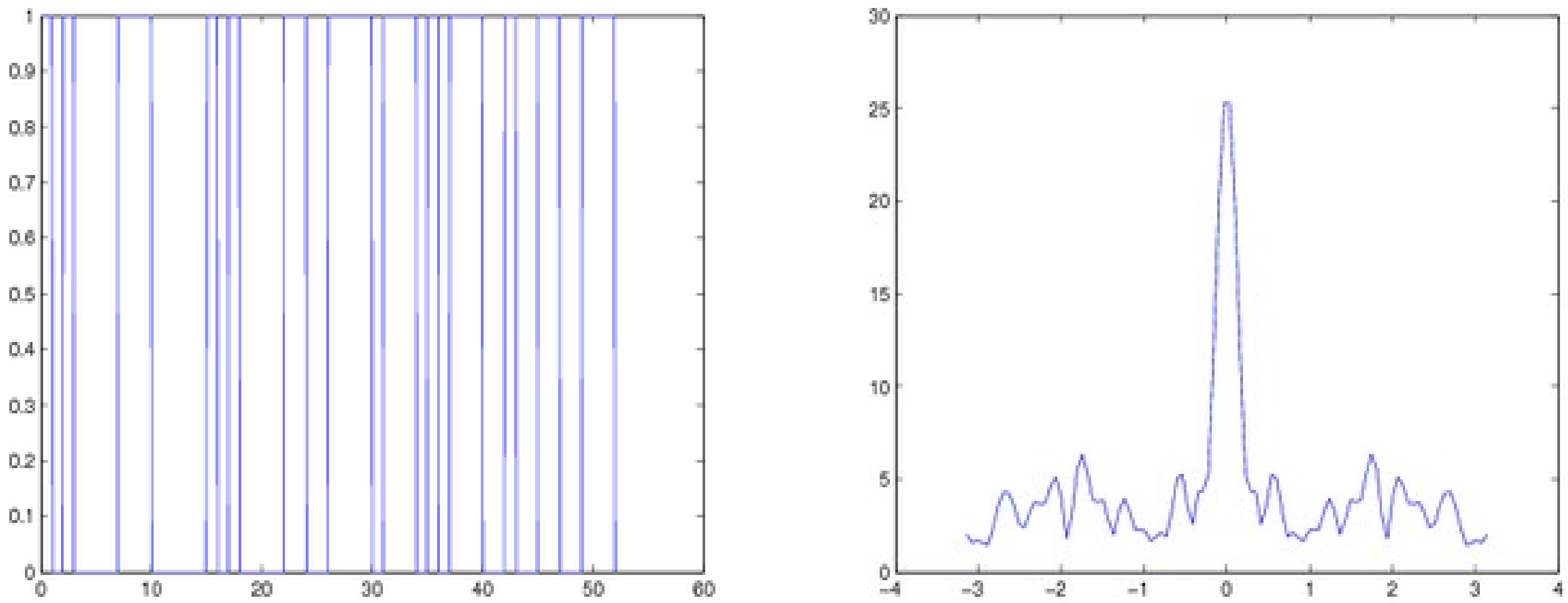


The observed : snapshot, blur of 1 pixel (left). The log-modulus of its DFT.

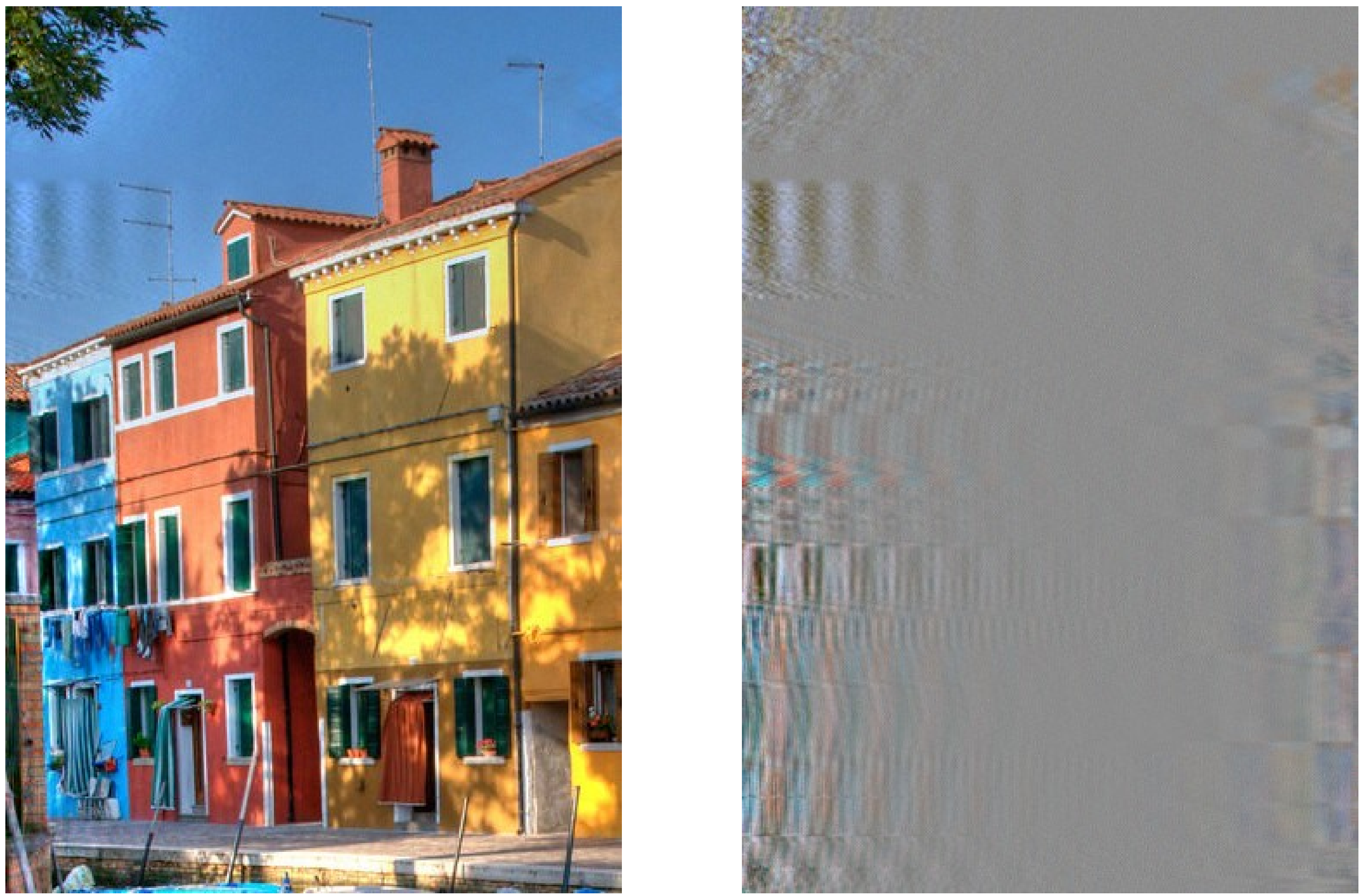
The restored (left), the residual noise (right), RMSE=1.58.

- Example 2 : Raskar's code



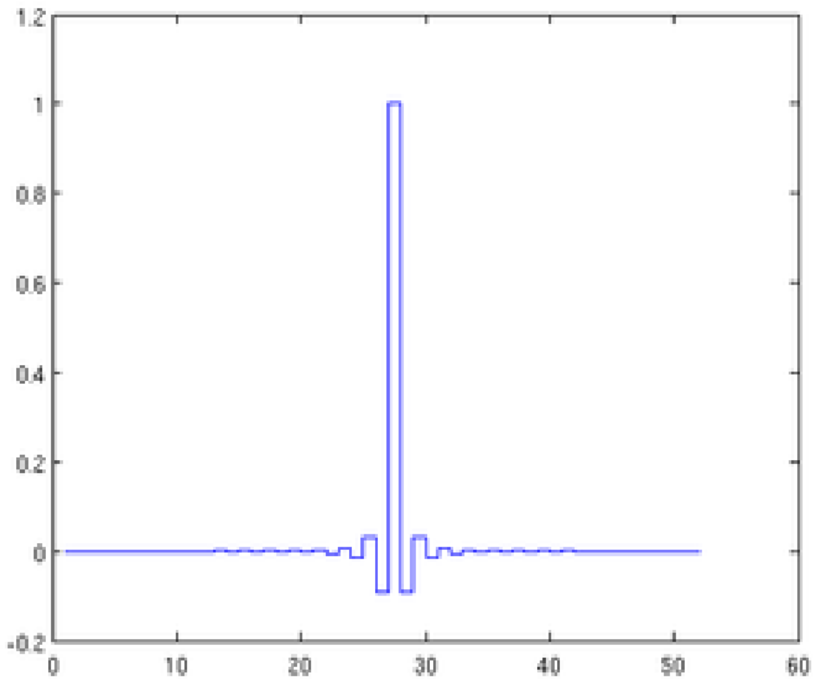
The Raskar function and the modulus of its Fourier transform.

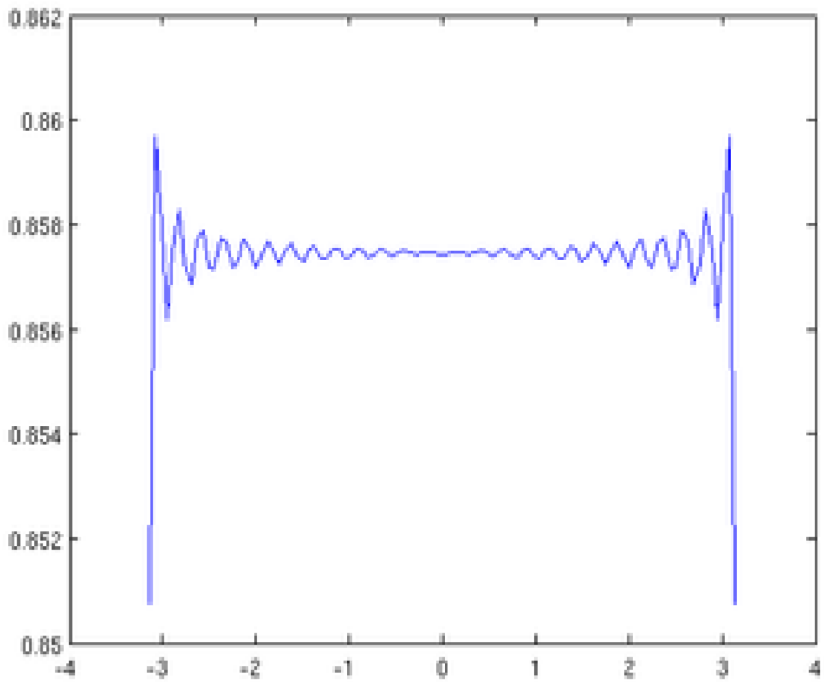
The observed : Raskar's code, blur of 52 pixels (left). The log-modulus of its DFT.



The restored (left), the residual noise (right), RMSE=3.37.
(RMSE of rand code : 2.19)

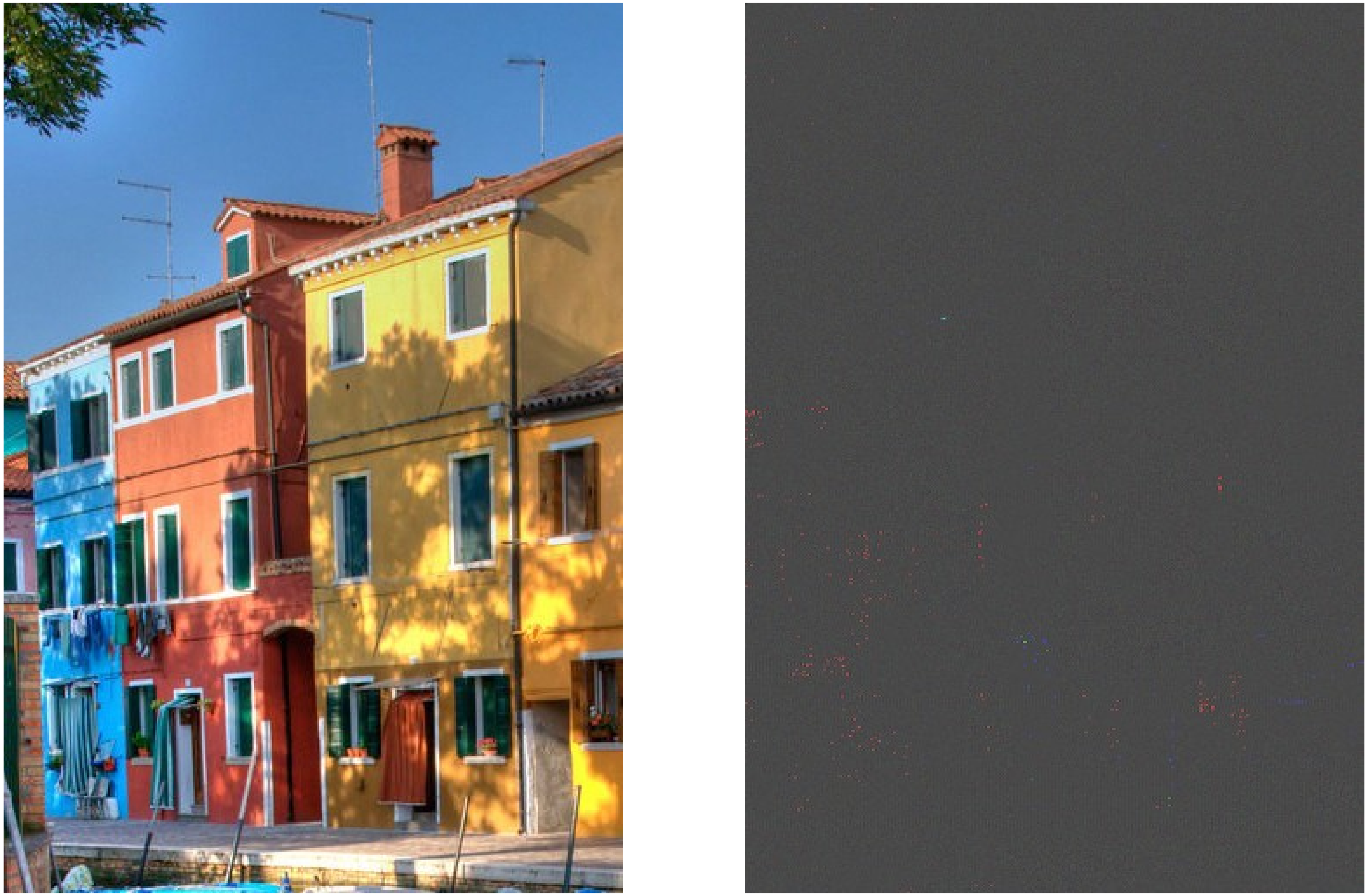
- Example 3 : the sinc code





The sinc-code function and the modulus of its Fourier transform.

The observed : snapshot, blur of 52 pixels (left). The log-modulus of its DFT.



The restored (left), the residual noise (right), RMSE=1.56.

- Conclusion

  The Flutter Shutter paradox :

  even for an infinite exposure time the SNR remains finite, contrarily to the classic steady photography

  The optimal flutter function is a sinc, and slightly increases the SNR compared to the best snapshot

- Thanks!

- Try it : (IPOL)

  https://edit.ipol.im/edit/algo/mrt_flutter_shutter/

  Detailed description of the algorithm

  Standard C++ code, commented

  Running demo