Models of flutter shutter cameras

Yohann Tendero
yohann.tendero@ipsa.fr

Join work with: Jean-Michel Morel
Stanley Osher
Bernard Rougé

June, 21th 2021. Peyresq summer school.

Syllabus

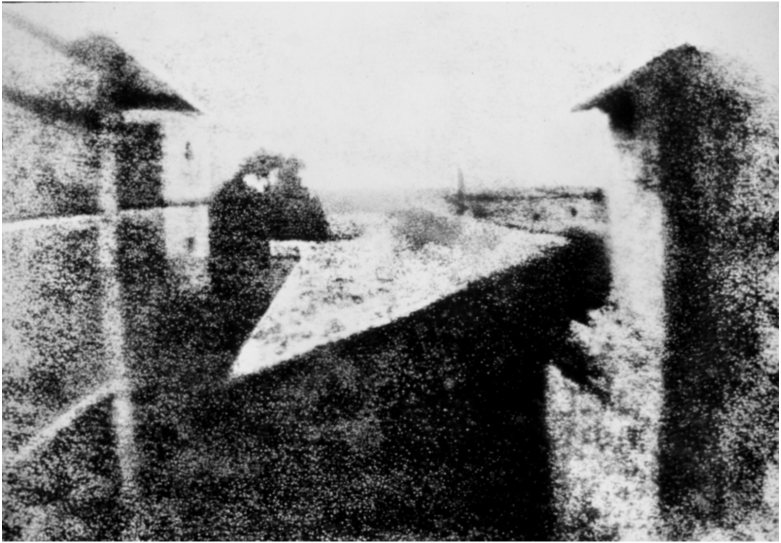
- (YT) Modèles continus et discrets pour le "flutter shutter"
- (Pauline Trouvé) Modèles en co-conception et démarches d'optimisation associées
- (YT) Exemples d'optimisation - choix du modèles discret/discret vs continu/discret)
- (Pauline Trouvé) Deep Codesign

Forewords

Computational photography

- Aims at optimizing the overall imaging chain:
  from the acquisition device to the final image produced by the algorithms.
- In other words, we increase the "final image quality" using new algorithms designed jointly with the cameras.
- Some examples of applications :
  - Focus stacking. Camera combines several images at different focus to produce an "always in focus picture"
  - High Dynamic range. Camera combines several images to avoid over/underexposure.
  - Face detection. Used to detect "regions of interest" and tune the focus + apply contrast change to produce the picture.
  - Flutter shutter. Modify the camera shutter to avoid motion blurs.

Nicephore Niepce, the first photography (1827). Exposure time: 8 hours.



Overview

- Image quality and exposure time. Acquisition models.
- Signal to noise ratio, fundamental thm. of photography
- Exposure time and motion blur
- Flutter shutter camera (principles and models)

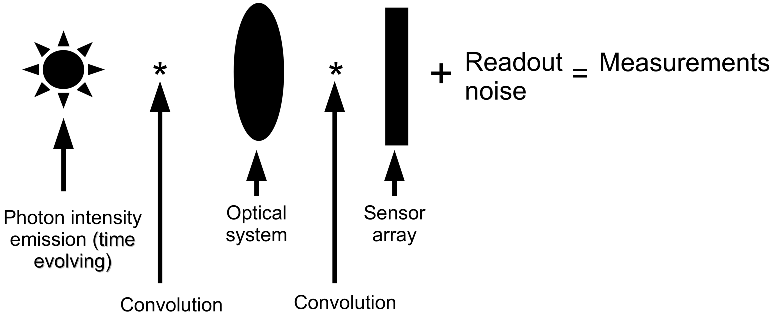
Passive acquisition systems: image quality and exposure time

Passive acquisition systems: image quality and exposure time

Passive acquisition systems: image quality and exposure time

Passive acquisition systems: image quality and exposure time

From the light source to the pixels



Sensor array : photon counter.

Poisson random variable

Poisson random variable (r.v.): $X \sim \mathcal{P}(\lambda)$, $\lambda > 0$ is called intensity

The realizations of a Poisson r.v. are supported on $\mathbb{N}$

For any $k \in \mathbb{N}$ we have $\mathbb{P}(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}$

We have $\mathbb{E}(X) = \lambda$ and $\mathrm{var}(X) = \lambda$

Simulating Poisson random variables

Simulation of a Poisson random variable X with intensity λ (D. Knuth):

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

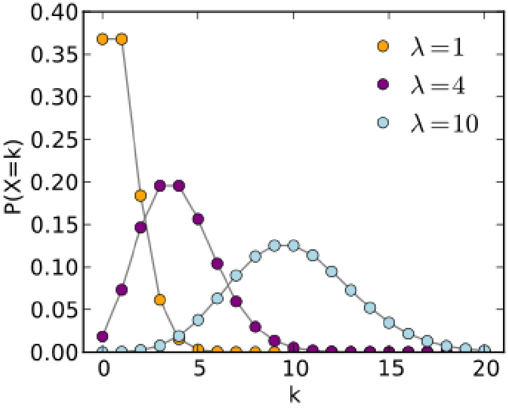
> Let $g = exp(-\lambda)$; em=-1; t =1; boolean rejected=true;
>
> While (rejected) DO
>
> > em=em+1;
> >
> > t=t.rand; (where rand is an uniform on [0,1] random generator)
> >
> > If $(t > g)$ then : X=em; rejected=true; endif;
>
> endwhile;

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

(The above algorithm returns X).

Behavior of Poisson r.v. for large intensities

Convergence to Gaussian distribution for large intensities



(This explains the "If $\lambda \leqslant 50$" in the algorithm).

Usefulness of using the Gaussian approximation for large λ

When simulating random variables :

- we always assume that we can obtain independent realization of $\mathcal{U}[0,1]$
- this is, obviously, no really the case: the realizations are deterministic.
  They satisfy "nice" statistical properties so that "they look as random as possible"
- However, every random generator is periodic!
- and, obviously, the realizations doesn't look like independent realizations
- So it is always worth estimating how many calls to the pseudo-random generator will be needed.

Recall: simulating Gaussian random variables

Simulation of a Gaussian random variable (polar Box-Muller)

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

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

(As usual, we tacitly assume *rand* $\sim \mathcal{U}[0,1]$).

Also, using a sin we could get two independent r.v. with one call to *rand*

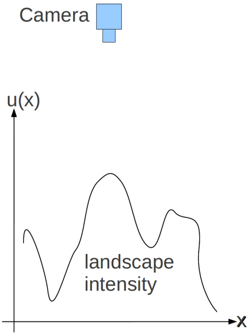
Simplest observation model

Often, one just assumes that the observed image obs is given by

$$\mathrm{obs}(n) = g \star I(n) + \eta(n)$$

where I denotes the scene, g is a PSF that models the camera optical system and $\eta(n)$ is an (additive) Gaussian white noise. Yet, one observes (see eg. Colom et al., Analysis and Extension of the Percentile Method, Estimating a Noise Curve from a Single Image) that the noise variance depends on the local signal value. A flutter shutter camera modifies the values observed by the camera and another model can therefore be used.

Photon acquisition model



- Δt length of a time interval
- Ideal observable landscape:
  $\tilde{u} = \underbrace{\mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]}}_{\text{Pixel sensor}} * \underbrace{g}_{\text{Optical system (PSF)}} * \underbrace{l}_{\text{Photon emission}}.$

Ideal case: a "Δt snapshot" at a pixel at position n is a Poisson random variable

$$\mathbf{P}_{g*l}([0,\Delta t] \times [n-\frac{1}{2}, n+\frac{1}{2}]) \sim \mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n)dt\right).$$

$X \sim \mathcal{P}(\lambda)$, $\mathbb{P}(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$.

Measuring image acquisition quality

Signal to Noise Ratio (SNR):
If X is an L^2 random variable the SNR is

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

Interpretation: Loosely speaking the SNR measures the relative fluctuation of a r.v. around its mean.

Fundamental Theorem of Photography

1) The value observed $\mathrm{obs}(n)$ at pixel n can be any realization of

$$\mathrm{obs}(n) \sim \mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n)dt\right) = \mathcal{P}\left(\Delta t \tilde{u}(n)dt\right).$$

Fundamental Theorem of Photography

1) The value observed $\mathrm{obs}(n)$ at pixel n can be any realization of

$$\mathrm{obs}(n) \sim \mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n)dt\right) = \mathcal{P}\left(\Delta t \tilde{u}(n)dt\right).$$

2) This pixel value is rescaled to be independent of the exposure time: $\frac{\mathrm{obs}(n)}{\Delta t}$.

3) The quantity stored (the digital image) is therefore $\frac{\mathrm{obs}(n)}{\Delta t}$

Theorem

The SNR of pixel n is $\sqrt{\Delta t \tilde{u}(n)}$, where $\tilde{u}(n)$ is the light intensity (photons per s) received at position n.

Fundamental Theorem of Photography

1) The value observed $\mathrm{obs}(n)$ at pixel n can be any realization of

$$\mathrm{obs}(n) \sim \mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n)dt\right) = \mathcal{P}\left(\Delta t \tilde{u}(n)dt\right).$$

2) This pixel value is rescaled to be independent of the exposure time: $\frac{\mathrm{obs}(n)}{\Delta t}$.

3) The quantity stored (the digital image) is therefore $\frac{\mathrm{obs}(n)}{\Delta t}$

Theorem

The SNR of pixel n is $\sqrt{\Delta t \tilde{u}(n)}$, where $\tilde{u}(n)$ is the light intensity (photons per s) received at position n.

$\mathbb{E}\left(\frac{\mathrm{obs}(n)}{\Delta t}\right) = \mathbb{E}\frac{\mathcal{P}(\Delta t \tilde{u}(n)dt)}{\Delta t} = \tilde{u}(n)$: this is what we would observe if there were no noise.

$\mathrm{var}\left(\frac{\mathrm{obs}(n)}{\Delta t}\right) = \frac{\tilde{u}(n)}{\Delta t}$. Hence, the SNR at pixel n is $\sqrt{\Delta t u(n)}$.

Direct consequences of Fundamental Theorem of Photography

Theorem

The SNR of pixel n is $\sqrt{\Delta t \tilde{u}(n)}$, where $\tilde{u}(n)$ is the light intensity (photons per s) received at position n.

We deduce that to increase the image quality (the SNR) we can

Direct consequences of Fundamental Theorem of Photography

Theorem

The SNR of pixel n is $\sqrt{\Delta t \tilde{u}(n)}$, where $\tilde{u}(n)$ is the light intensity (photons per s) received at position n.

We deduce that to increase the image quality (the SNR) we can

- Increase the exposure time Δt
- Increase the photon emission $\tilde{u}$, i.e.,

Direct consequences of Fundamental Theorem of Photography

Theorem

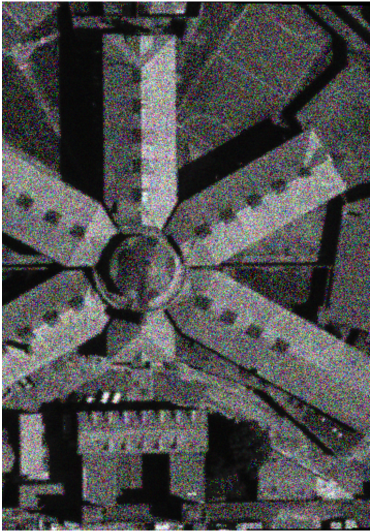
The SNR of pixel n is $\sqrt{\Delta t \tilde{u}(n)}$, where $\tilde{u}(n)$ is the light intensity (photons per s) received at position n.

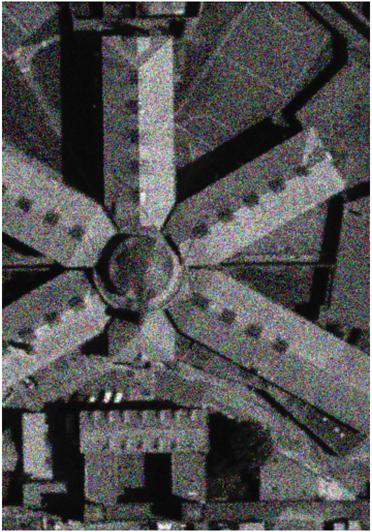
We deduce that to increase the image quality (the SNR) we can

- Increase the exposure time Δt
- Increase the photon emission $\tilde{u}$, i.e., use a flash.

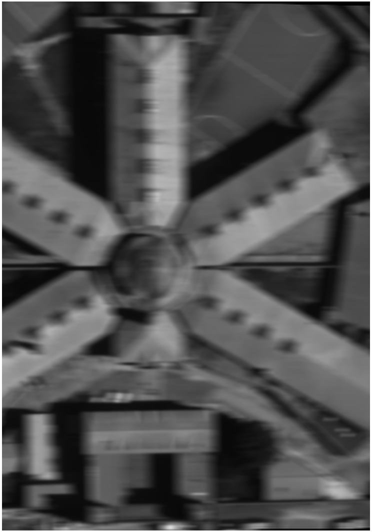
(However, a flash can only be used indoors).

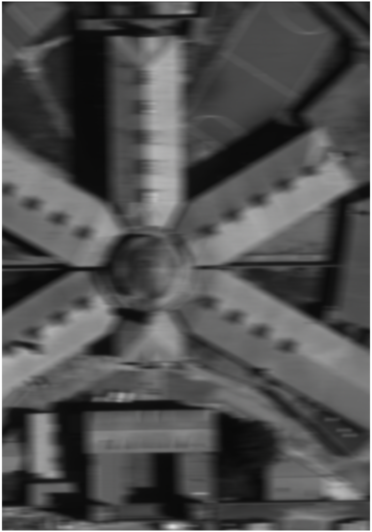
Dynamic scenes: exposure time vs motion blur





Dynamic scenes: exposure time vs motion blur





Recall: Fourier transform & convolution

For $f, g \in L^1(\mathbb{R})$ or $L^2(\mathbb{R})$ we have $f * g$ convolution of two functions

$$(f * g)(x) = \int_{\mathbb{R}} f(y)g(x-y)dy$$

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

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

and $\mathcal{F}^{-1}(\mathcal{F}(f))(x) := \widetilde{\mathcal{F}(f)}(x) = f(x) = \frac{1}{2\pi}\int_{\mathbb{R}} \mathcal{F}(f)(\xi)e^{ix\xi}d\xi$.
Moreover $\mathcal{F}(f * g)(\xi) = \mathcal{F}(f)(\xi)\mathcal{F}(g)(\xi)$ and (Plancherel)

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

Recall: band-limited functions and cardinal sine

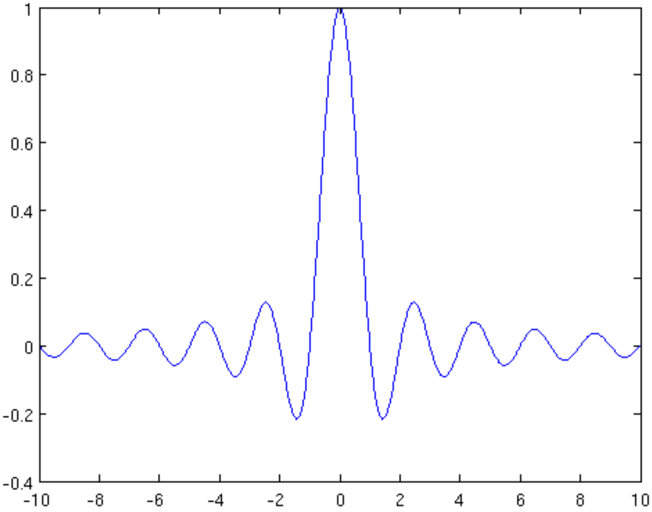
- Band-limited: $\hat{u}$ is band limited if $\hat{u}(\xi)$ is supported on $[-\pi, \pi]$
- Cardinal sine function $\mathrm{sinc}(x) := \frac{\sin(\pi x)}{\pi x}$

$$\mathcal{F}(\mathrm{sinc})(\xi) = \mathbb{1}_{[-\pi,\pi]}(\xi)$$

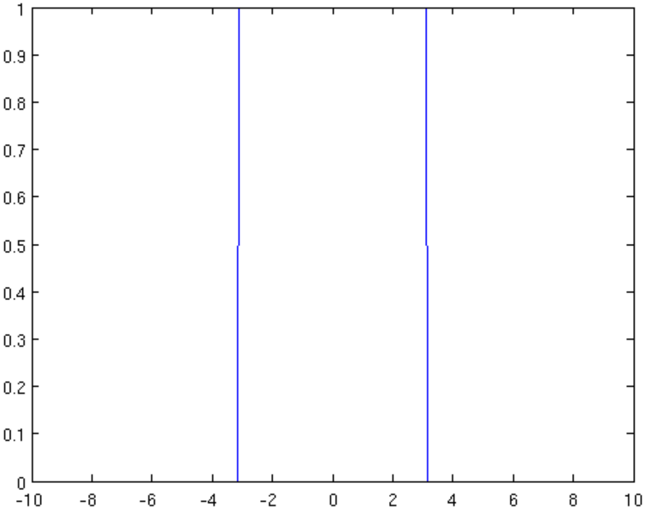
- The sinc function acts as a Dirac for band limited functions:

$$u * \mathrm{sinc}(x) = u(x) \quad \forall x \in \mathbb{R}.$$

Graph of cardinal sine



Fourier transform of cardinal sine

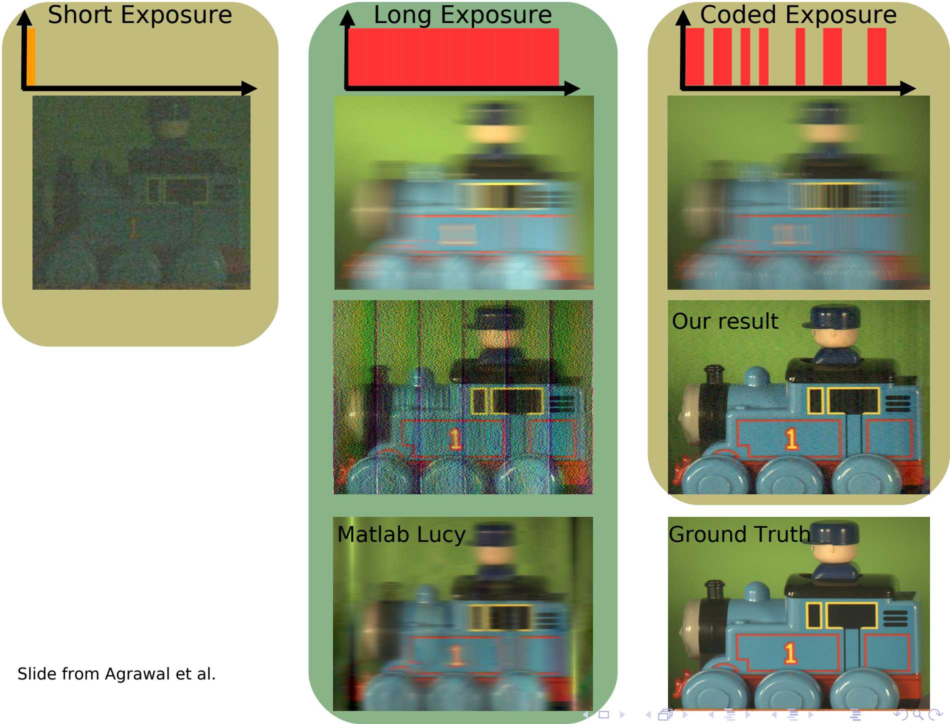


Photographing dynamic scenes could only be done using short exposure times, until...

Photographing dynamic scenes could only be done using short exposure times, until...

The Coded Exposure

Agrawal et al. "Resolving Objects at Higher Resolution from a Single Motion-Blurred Image", CVPR 2007.



Slide from Agrawal et al.

Counterintuitive solution

To reduce motion blur, increase it!

Move camera as picture is taken.

-Kernel is known (no motion estimation needed).

-Kernel identical over the image.

Levin *et al.*, *"Motion-invariant Photography", SIGGRAPH, 2008.*

Acquisition model: discrete one

Many Authors (eg Agrawal et al., Levin et al.) build the following model:

- The observed data is given by a finite sensor array
- The camera diaphragm (or shutter) opens/closes on sub-intervals of the time-aperture following a finite binary sequence called "code"

Acquisition model: discrete one

Many Authors (eg Agrawal et al., Levin et al.) build the following model:

- The observed data is given by a finite sensor array
- The camera diaphragm (or shutter) opens/closes on sub-intervals of the time-aperture following a finite binary sequence called "code"
- Do drawing.
- Hence,in the ideal noiseless case the observed values should be given by the discrete convolution

$$u \star \alpha$$

where u denotes the discrete version of the observed landscape, α is the "code" and $\star$ a discrete convolution

Acquisition model: discrete one

Many Authors (eg Agrawal et al., Levin et al.) build the following model:

- The observed data is given by a finite sensor array
- The camera diaphragm (or shutter) opens/closes on sub-intervals of the time-aperture following a finite binary sequence called "code"
- Do drawing.
- Hence,in the ideal noiseless case the observed values should be given by the discrete convolution

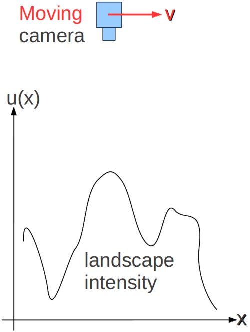
  $$u \star \alpha$$

  where u denotes the discrete version of the observed landscape, α is the "code" and $\star$ a discrete convolution
- and the noisy observed value is

  $$u \star \alpha + \eta$$

  where η is (in most cases) a Gaussian noise

Image model: continuous one



- Δt length of a time interval
- v relative velocity (unit: pixels per second)
- $\tilde{u} = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * l$ ideal observable landscape.

Ideal case: "Δt snapshot" at a pixel at position n is a Poisson random variable

$$\mathbf{P}_{g*l}([0,\Delta t] \times [n-\frac{1}{2}, n+\frac{1}{2}]) \sim \mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n - vt)dt\right).$$

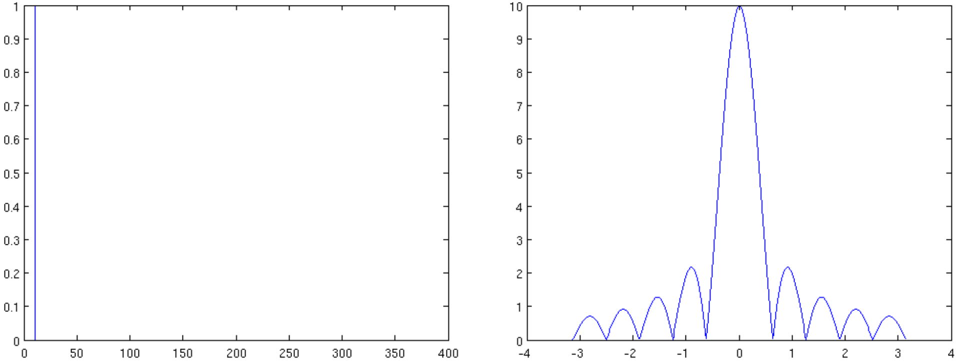
$X \sim \mathcal{P}(\lambda)$, $\mathbb{P}(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$.

Imaging dynamic scenes : standard motion blur kernel

- Observed value at pixel n : $\mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n - {\color{red}v}t)dt\right)$.
- We have $\int_0^{\Delta t} \tilde{u}(n - vt)dt = \frac{1}{|v|}(\tilde{u} * \mathbb{1}_{[0,v\Delta t]})(n)$
- and $\mathcal{F}\left(\mathbb{1}_{[0,v\Delta t]}\right) = \int_0^{v\Delta t} e^{-ix\xi}dx = 2\frac{\sin(\frac{\xi v\Delta t}{2})}{\xi}e^{-i\xi\frac{v\Delta t}{2}}$.
- Hence, $\mathcal{F}\left(\mathbb{1}_{[0,v\Delta t]}\right)$ has zeros on $[-\pi, \pi]$ the support of $\mathcal{F}(\tilde{u})$ if $|v|\Delta t \geqslant 2$.

Bottom line:
<u>the standard motion blur kernel is not invertible in general</u>.

- 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).

Imaging dynamic scenes : need to limit the exposure time or use a flutter shutter

- Observed value at pixel n :
$\mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n - vt)dt\right) = \mathcal{P}\left(\frac{1}{|v|}(\tilde{u} * \mathbb{1}_{[0,v\Delta t]})(n)\right)$.
- Hence, $\mathcal{F}\left(\mathbb{1}_{[0,v\Delta t]}\right)$ has zeros on $[-\pi, \pi]$ the support of $\mathcal{F}(\tilde{u})$ if $|v|\Delta t \geqslant 2$.

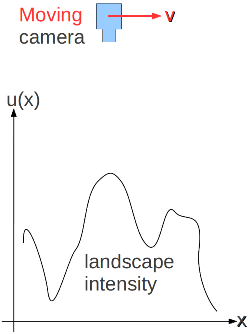
The non-negative quantity $|v|\Delta t$ is nothing but the length of the blur. (It's a distance).

To have an invertible kernel Δt must be such that $(0 <)\Delta t < \frac{2}{|v|}$.

This limits the image quality.

In other words, contrarily to the steady case: we can no more control the SNR using the exposure time.

Recall: Ideal image model (classic camera)



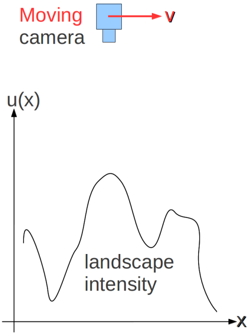
- Δt length of a time interval
- v relative velocity (unit: pixels per second)
- $\tilde{u} = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * l$ ideal observable landscape.

Ideal case: "Δt snapshot" at a pixel at position n is a Poisson random variable

$$\mathbf{P}_{g*l}([0,\Delta t] \times [n - \frac{1}{2}, n + \frac{1}{2}]) \sim \mathcal{P}\left(\int_0^{\Delta t} \tilde{u}(n - vt)dt\right).$$

$X \sim \mathcal{P}(\lambda)$, $\mathbb{P}(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$.

Ideal image model (flutter camera)



- Δt length of a time interval
- v relative velocity (unit: pixels per second)
- $\tilde{u} = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]} * g * I$ ideal observable landscape.

Ideal case: "Δt snapshot" at a pixel at position n is a Poisson random variable

$$\mathbf{P}_{g*I}([0,\Delta t] \times [n-\frac{1}{2}, n+\frac{1}{2}]) \sim \mathcal{P}\left(\int_0^{\Delta t} \alpha(t)\tilde{u}(n - vt)dt\right).$$

$X \sim \mathcal{P}(\lambda)$, $\mathbb{P}(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$.

Formalization : continuous model

Definition

Each observation at a pixel centered at x is corrupted by an additive noise $\eta(x)$ called readout noise. We assume that $\mathbb{E}(\eta(x)) = 0$, that $\mathrm{var}(\eta(x)) = \sigma_r < +\infty$ and that the obscurity noise has a variance $\sigma_o^2 < +\infty$. We have

$$\mathrm{obs}(x) \quad \sim \quad \mathcal{P}\left(\int_{t_1}^{t_2} \left(\tilde{u}(x - vt) + \sigma_o^2\right) dt\right) + \eta(x).$$

We assume that

$$\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} |\tilde{u}(x)| dx := \mu \in \mathbb{R}$$

exists and is finite, that $u := (\tilde{u} - \mu) \in L^1(\mathbb{R})$ and $[-\pi, \pi]$ band limited.

Formalization in the general case

Definition

Each observation at a pixel centered at x is corrupted by an additive noise $\eta(x)$ called readout noise. We assume that $\mathbb{E}(\eta(x)) = 0$, that $\mathrm{var}(\eta(x)) = \sigma_r < +\infty$ and that the obscurity noise has a variance $\sigma_o^2 < +\infty$. We have

$$\mathrm{obs}(x) \quad \sim \quad \mathcal{P}\left(\int_{t_1}^{t_2} \left(\tilde{u}(x - vt) + \sigma_o^2\right) dt\right) + \eta(x).$$

We assume that

$$\lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} |\tilde{u}(x)| dx := \mu \in \mathbb{R}$$

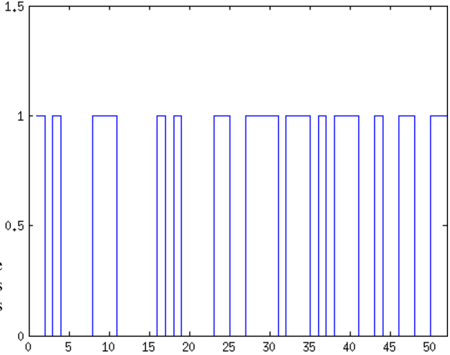
exists and is finite, that $u := (\tilde{u} - \mu) \in L^1(\mathbb{R})$ and $[-\pi, \pi]$ band limited.

Exposure function

3×10^6 candidate codes. We chose a code that (i) maximizes the minimum of the magnitude of the DFT values and (ii) minimizes the variance of the DFT values. The near-optimal code we found is

1010000111000001010000110011110111010111001001100111.

The Agrawal *et al.* code.

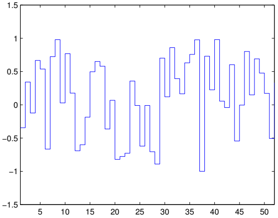


The binary exposure function for the Agrawal et al. code.

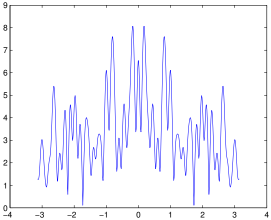
$$\text{Code: } (\alpha_0, \ldots, \alpha_{L-1}) \in \mathbb{R}^L \Leftrightarrow \text{Exposure function } \alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t, (k+1)\Delta t[}(t).$$

Numerical coded exposure setup

1. The camera takes a burst of L images using an exposure time Δt;
2. The k-th elementary image is assigned a numerical weight $\alpha_k \in \mathbb{R}$;
3. All images are added together to get *one* observed image.



A *signed* exposure function.



The modulus of its Fourier transform.

Numerical and analog flutter shutter

Code: $(\alpha_0, ..., \alpha_{L-1}) \in \mathbb{R}^L \Leftrightarrow$ Exposure function: $\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t[}(t)$.

Definition

- *Numerical* samples:
  $\mathrm{obs}(n) \sim \sum_{k=0}^{L-1} \alpha_k \left(\mathcal{P} \left(\int_{k\Delta t}^{(k+1)\Delta t} \tilde{u}(n - vt) + \sigma_o^2 dt \right) + \eta(k) \right)$.
- *Analog* samples ($\alpha(t) \in [0,1]$):
  $\mathrm{obs}(n) \sim \mathcal{P} \left[\left(\frac{1}{|v|} \alpha(\frac{\cdot}{v}) * \tilde{u} \right)(n) + \|\alpha\|_{L^1(\mathbb{R})} \sigma_o^2 \right] + \eta(n)$

Assumption: $\eta(n)$ i.i.d. and independent form the Poisson r.v.

Recalls:

- Velocity v : unit in pixel(s) per Δt.
- $\sigma_o^2 < +\infty$: variance of the obscurity noise.
- $\mathbb{E}(\eta(n)) = 0$, $\mathrm{var}(\eta(n)) = \sigma_r^2 < +\infty$.

Conclusion

There are two models for the flutter shutter (and two variants of flutter shutter: analog and numerical)

- (Discrete) $\mathrm{obs}(n) \sim (u \star \alpha + \eta)(n)$
- (Continuous)
  - *Numerical* samples: $\mathrm{obs}(n) \sim \sum_{k=0}^{L-1} \alpha_k \left(\mathcal{P} \left(\int_{k\Delta t}^{(k+1)\Delta t} \tilde{u}(n - vt) + \sigma_o^2 dt \right) + \eta(k) \right).$
  - *Analog* samples ($\alpha(t) \in [0,1]$): $\mathrm{obs}(n) \sim \mathcal{P}\left[\left(\frac{1}{|v|}\alpha(\frac{\cdot}{v}) * \tilde{u} \right)(n) + \|\alpha\|_{L^1(\mathbb{R})}\sigma_o^2 \right] + \eta(n)$

Code: $(\alpha_0, ..., \alpha_{L-1}) \in \mathbb{R}^L \Leftrightarrow$ Exposure function: $\alpha(t) = \sum_{k=0}^{L-1} \alpha_k \mathbb{1}_{[k\Delta t,(k+1)\Delta t}$

Short bibliography

- Agrawal et al., Resolving Objects at Higher Resolution from a Single Motion-blurred Image.
- Levin et al., Motion invariant photography
- Cossairt et al., When does computational imaging improve performance?
- T. et al., On a mathematical theory of coded exposure
- T. et al., The flutter shutter paradox