Gaussian Models for Images

Agnès Desolneux

CNRS and Ecole Normale Supérieure Paris-Saclay

Cours Master MVA

PART I : Gaussian texture images

Framework

- We work with discrete digital images $u \in \mathbb{R}^{M \times N}$ indexed on the set $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$.
- Each image is extended by periodicity :

$$u(k,l) = u(k \mod M, l \mod N) \quad \text{for all } (k,l) \in \mathbb{Z}^2.$$

- Consequence : Translation of an image :

Discrete Fourier transform of digital images

- Image domain : $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$
- Fourier domain $\hat{\Omega}$: the frequency 0 is placed at the center :

$$\hat{\Omega} = \left\{-\frac{M}{2}, \dots, \frac{M}{2} - 1\right\} \times \left\{-\frac{N}{2}, \dots, \frac{N}{2} - 1\right\}.$$

Definition :

- The **discrete Fourier transform (DFT)** of u is the **complex-valued** image $\hat{u}$ defined by :

$$\forall (s,t) \in \hat{\Omega}, \quad \hat{u}(s,t) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} u(k,l) e^{-\frac{2iks\pi}{M}} e^{-\frac{2ilt\pi}{N}}.$$

Discrete Fourier transform of digital images

- Image domain : $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$
- Fourier domain $\hat{\Omega}$: the frequency 0 is placed at the center :

$$\hat{\Omega} = \left\{ -\frac{M}{2}, \dots, \frac{M}{2} - 1 \right\} \times \left\{ -\frac{N}{2}, \dots, \frac{N}{2} - 1 \right\}.$$

Definition :

- The **discrete Fourier transform (DFT)** of u is the **complex-valued** image $\hat{u}$ defined by :

$$\forall (s,t) \in \hat{\Omega}, \quad \hat{u}(s,t) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} u(k,l) e^{-\frac{2iks\pi}{M}} e^{-\frac{2ilt\pi}{N}}.$$

- $|\hat{u}|$: **Fourier modulus (amplitude)** of u.
- $\arg(\hat{u})$: **Fourier phase** of u.

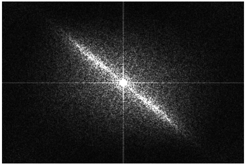
Discrete Fourier transform of digital images

Symmetry property :

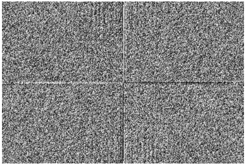
- $|\hat{u}|$: the **Fourier modulus** is even.
- $\arg(\hat{u})$: the **Fourier phase** is odd.

Visualization of the DFT :

Image u



Modulus $|\hat{u}|$



Phase $\arg(\hat{u})$

Modulus and phase of a digital image

Exchanging the modulus and the phase of two images : (ref : Oppenheim and Lim « the importance of phase in signals », 1981)

Practical exercise 1 :

Take two grey-level images of the same size. Compute their Fourier Transform, switch their phases, and take the inverse Fourier transforms. Visualize the two new images : what do you observe ?

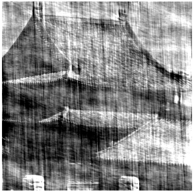
Modulus and phase of a digital image

Exchanging the modulus and the phase of two images : (ref : Oppenheim and Lim « the importance of phase in signals », 1981)

Image 1

Image 2

Modulus of 1
& phase of 2



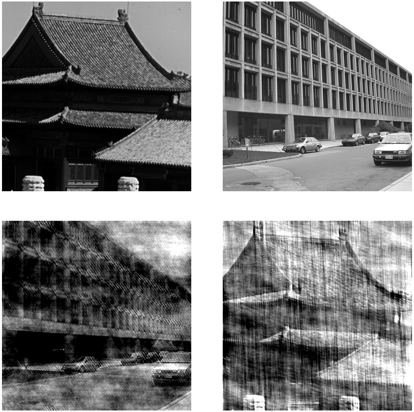
Modulus of 2
& phase of 1

Modulus and phase of a digital image

Exchanging the modulus and the phase of two images : (ref : Oppenheim and Lim « the importance of phase in signals », 1981)

Image 1

Image 2



Modulus of 1 $\&$ phase of 2

Modulus of 2 $\&$ phase of 1

- Geometric contours are mostly contained in the phase.

Modulus and phase of a digital image

Exchanging the modulus and the phase of two images :

Image 3

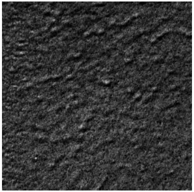
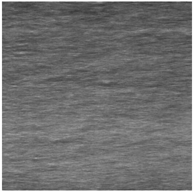


Image 4

Modulus of 3
& phase of 4



Modulus of 4
& phase of 3

- Textures are mostly contained in the modulus.

Modulus and phase of a digital image

Exchanging the modulus and the phase of two images :

Image 1

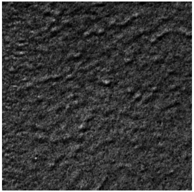
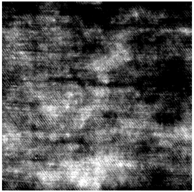
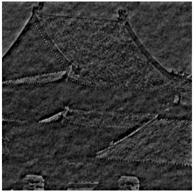


Image 4

Modulus of 1
& phase of 4





Modulus of 4
& phase of 1

- Geometric contours are mostly contained in the phase.
- Textures are mostly contained in the modulus.

Random phase textures

- We call *random phase texture* any image that is perceptually invariant to phase randomization.
- Phase randomization = replace the Fourier phase by a random phase.
- Definition : A random field $\theta : \hat{\Omega} \to \mathbb{R}$ is a ***random phase*** if
  1. Symmetry : θ is odd :
  $$\forall (s,t) \in \hat{\Omega}, \theta(-s,-t) = -\theta(s,t).$$
  2. Distribution : Each component $\theta(s,t)$ is
     - uniform over the interval $]-\pi,\pi]$ if $(s,t) \notin \{(0,0), \left(\frac{M}{2},0\right), \left(0,\frac{N}{2}\right), \left(\frac{M}{2},\frac{N}{2}\right)\}$,
     - uniform over the set $\{0,\pi\}$ otherwise.
  3. Independence : For each subset $\mathcal{S} \subset \hat{\Omega}$ that does not contain distinct symmetric points, the r.v. $\{\theta(s,t) | (s,t) \in \mathcal{S}\}$ are independent.

Random phase textures

- We call *random phase texture* any image that is perceptually invariant to phase randomization.
- Phase randomization = replace the Fourier phase by a random phase.
- Definition : A random field $\theta : \hat{\Omega} \to \mathbb{R}$ is a ***random phase*** if
  1. Symmetry : θ is odd :
  $$\forall (s,t) \in \hat{\Omega}, \theta(-s,-t) = -\theta(s,t).$$
  2. Distribution : Each component $\theta(s,t)$ is
     - uniform over the interval $]-\pi,\pi]$ if $(s,t) \notin \{(0,0), (\frac{M}{2},0), (0,\frac{N}{2}), (\frac{M}{2},\frac{N}{2})\}$,
     - uniform over the set $\{0,\pi\}$ otherwise.
  3. Independence : For each subset $\mathcal{S} \subset \hat{\Omega}$ that does not contain distinct symmetric points, the r.v. $\{\theta(s,t) | (s,t) \in \mathcal{S}\}$ are independent.

- **Theoretical exercise 2 :** Prove that the Fourier phase of a Gaussian white noise W is a random phase, and that the Fourier amplitudes follow a Rayleigh distribution.

Random phase textures

- We call *random phase texture* any image that is perceptually invariant to phase randomization.
- Phase randomization = replace the Fourier phase by a random phase.
- Definition : A random field $\theta : \hat{\Omega} \to \mathbb{R}$ is a ***random phase*** if
  1. Symmetry : θ is odd :
  $$\forall (s,t) \in \hat{\Omega}, \theta(-s,-t) = -\theta(s,t).$$
  2. Distribution : Each component $\theta(s,t)$ is
     - uniform over the interval $]-\pi,\pi]$ if $(s,t) \notin \left\{(0,0), \left(\frac{M}{2},0\right), \left(0,\frac{N}{2}\right), \left(\frac{M}{2},\frac{N}{2}\right)\right\}$,
     - uniform over the set $\{0,\pi\}$ otherwise.
  3. Independence : For each subset $\mathcal{S} \subset \hat{\Omega}$ that does not contain distinct symmetric points, the r.v. $\{\theta(s,t) | (s,t) \in \mathcal{S}\}$ are independent.

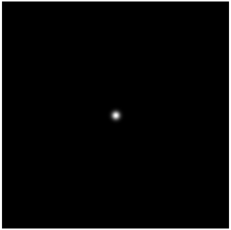
- **Theoretical exercise 2 :** Prove that the Fourier phase of a Gaussian white noise W is a random phase, and that the Fourier amplitudes follow a Rayleigh distribution.

- (Lazy) simulation : In Matlab, `theta = angle(fft2(randn(M,N)))`.
- *Random phase textures* constitute a “limited” subclass of the set of textures.

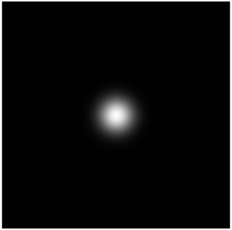
Random Phase Noise (RPN)

- Texture synthesis algorithm : ***random phase noise* (RPN)** : (Van Wijk, 1991)

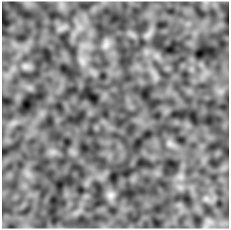
1. Compute the DFT $\hat{h}$ of the input h.
2. Compute a random phase θ.
3. Set $\hat{Z} = |\hat{h}|e^{i\theta}$ (or $\hat{Z} = \hat{h}e^{i\theta}$).
4. Return Z the inverse DFT of $\hat{Z}$.



Original image h



Modulus $\left|\hat{h}\right|$



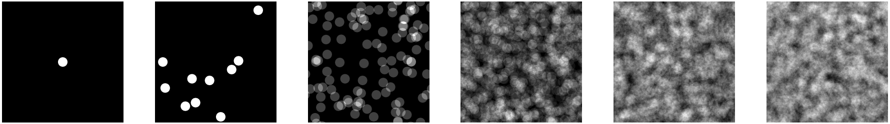
RPN associated with h

Discrete spot noise (Van Wijk, 1991)

- Let h be a discrete image called *spot*.
- Let (X_k) be a sequence of random translation vectors which are i.d.d. and uniformly distributed over Ω.
- The **discrete spot noise of order n associated with h** is the random image

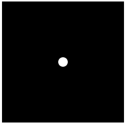
$$f_n(x) = \sum_{k=1}^{n} h(x - X_k).$$

(translations with periodic boundary conditions)

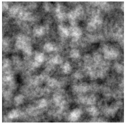


Spot h | $n = 10$ | $n = 10^2$ | $n = 10^3$ | $n = 10^4$ | $n = 10^5$

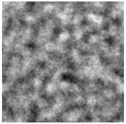
Limit of the DSN model ?



Spot h



$n = 10^4$



$n = 10^5$

?

$n = +\infty$

- For texture synthesis we are more particularly interested in the limit of the *DSN* : the ***asymptotic discrete spot noise*** (***ADSN***).
- The *DSN* of order n, $f_n(x) = \sum_k h(x - X_k)$, is the sum of the n i.i.d. random images $h(\cdot - X_k)$.
- **Central limit theorem for random vectors :**
  The sequence of random images $\left(\dfrac{f_n - n\mathbb{E}(h(\cdot - X_1))}{\sqrt{n}}\right)_{n\in\mathbb{N}^*}$ converges in distribution towards the **Gaussian random vector** $Y = (Y(x))_{x\in\Omega}$ with zero mean and covariance $\mathrm{Cov}(h(\cdot - X_1))$.

Asymptotic discrete spot noise (*ADSN*)

Expectation of the random translations :

$$
\begin{aligned}
\mathbb{E}(h(x - X_1)) &= \sum_{y \in \Omega} h(x-y)\mathbb{P}(X_1 = y) \\
&= \sum_{y \in \Omega} h(x-y)\frac{1}{MN} \\
&= \frac{1}{MN} \sum_{z \in \Omega} h(z) \\
&= \text{mean of } h.
\end{aligned}
$$

- $\mathbb{E}(h(x - X_1)) = m$, where m is the mean of h.

Asymptotic discrete spot noise (*ADSN*)

Covariance of the random translations : Let $x, y \in \Omega$,

$$\begin{aligned}
\mathrm{Cov}(h(x-X_1), h(y-X_1)) &= \mathbb{E}((h(x-X_1)-m)(h(y-X_1)-m)) \\
&= \sum_{z\in\Omega}(h(x-z)-m)(h(y-z)-m)\mathbb{P}(X_1=z) \\
&= \frac{1}{MN}\sum_{z\in\Omega}(h(x-z)-m)(h(y-z)-m) \\
&= C_h(x,y).
\end{aligned}$$

- $\mathrm{Cov}(h(x-X_1), h(y-X_1)) = C_h(x,y)$ where C_h is the **autocorrelation** of h :

$$C_h(x,y) = \frac{1}{MN}\sum_{z\in\Omega}\left(h(x-z)-m\right)\left(h(y-z)-m\right), \quad (x,y)\in\Omega.$$

Asymptotic discrete spot noise (*ADSN*)

- For texture synthesis we are more particularly interested in the limit of the *DSN* : the ***asymptotic discrete spot noise* (*ADSN*)**.

Expectation and covariance of the random translations :

- $\mathbb{E}(h(x - X_1)) = m$, where m is the mean of h.
- $\mathrm{Cov}(h(x - X_1), h(y - X_1)) = C_h(x, y)$ where C_h is the autocorrelation of h :

$$C_h(x, y) = \frac{1}{MN} \sum_{z \in \Omega} (h(x - z) - m)\,(h(y - z) - m) = C_h(0, y - x).$$

Definition of *ADSN* :

- The *ADSN* associated with h is the Gaussian vector $\mathcal{N}(0, C_h)$.

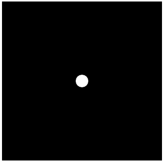
Simulation of the ADSN

Definition of *ADSN* : the *ADSN* associated with h is the Gaussian vector $\mathcal{N}(0, C_h)$.

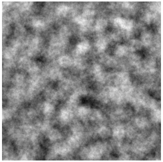
Convolution product : $(f * g)(x) = \sum_{y \in \Omega} f(x - y) g(y), \ x \in \Omega.$

Simulation of the *ADSN* :

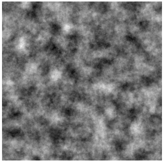
- Let $h \in \mathbb{R}^{M \times N}$ be a an image, m be the mean of h and W be a Gaussian white noise image.
- The random image $\frac{1}{\sqrt{MN}}(h - m) * W$ is the *ADSN* associated with h.



Spot h



DSN, $n = 10^5$



ADSN

ADSN Simulation

Theoretical exercise 3 : prove that $Y = \frac{1}{\sqrt{MN}}(h - m) * W$ is $\mathcal{N}(0, C_h)$ distributed.

ADSN Simulation

Theoretical exercise 3 : prove that $Y = \frac{1}{\sqrt{MN}}(h-m) * W$ is $\mathcal{N}(0, C_h)$ distributed.

- Y is obtained from W by applying a linear mapping. Since W is a Gaussian vector, Y is also a Gaussian vector.
- One just needs to show that $\mathbb{E}(Y(x)) = 0$ and $\mathrm{Cov}(Y(x), Y(y)) = C_h(x, y)$.

ADSN Simulation

Theoretical exercise 3 : prove that $Y = \frac{1}{\sqrt{MN}}(h-m) * W$ is $\mathcal{N}(0, C_h)$ distributed.

- Y is obtained from W by applying a linear mapping. Since W is a Gaussian vector, Y is also a Gaussian vector.
- One just needs to show that $\mathbb{E}(Y(x)) = 0$ and $\mathrm{Cov}(Y(x), Y(y)) = C_h(x,y)$.
- By linearity, $\mathbb{E}(Y(x)) = \frac{1}{\sqrt{MN}}(h-m) * \mathbb{E}(W)(x) = 0$.

ADSN Simulation

Theoretical exercise 3 : prove that $Y = \frac{1}{\sqrt{MN}}(h-m) * W$ is $\mathcal{N}(0, C_h)$ distributed.

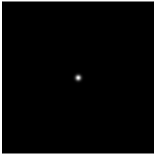
- Y is obtained from W by applying a linear mapping. Since W is a Gaussian vector, Y is also a Gaussian vector.
- One just needs to show that $\mathbb{E}(Y(x)) = 0$ and $\mathrm{Cov}(Y(x), Y(y)) = C_h(x, y)$.
- By linearity, $\mathbb{E}(Y(x)) = \frac{1}{\sqrt{MN}}(h-m) * \mathbb{E}(W)(x) = 0.$
- Let $x, y \in \Omega$,

$$
\begin{aligned}
\mathrm{Cov}(Y(x), Y(y)) &= \mathbb{E}(Y(x)Y(y)) \\
&= \frac{1}{MN}\mathbb{E}\left(\sum_{s\in\Omega}(h(s-x)-m)W(s)\sum_{t\in\Omega}(h(t-y)-m)W(t)\right) \\
&= \frac{1}{MN}\sum_{s,t\in\Omega}(h(s-x)-m)(h(t-y)-m)\quad\underbrace{\mathbb{E}(W(s)W(t))}_{=1 \text{ if } s=t \text{ and } 0 \text{ otherwise}} \\
&= \frac{1}{MN}\sum_{s\in\Omega}(h(s-x)-m)(h(s-y)-m) = C_h(x, y).
\end{aligned}
$$

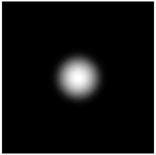
Differences between *RPN* and *ADSN*

(ref : Galerne, Gousseau, Morel, 2011)

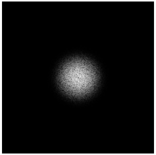
- *RPN* and *ADSN* both have a random phase.
- The Fourier modulus of *RPN* is equal to $|\hat{h}|$.
- The Fourier modulus of *ADSN* is the pointwise multiplication between $|\hat{h}|$ and a Rayleigh noise.



Spot h

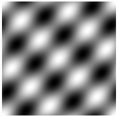


RPN Modulus

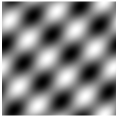


ADSN Modulus

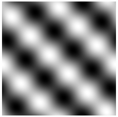
- ***RPN* and *ADSN* are two different processes.**



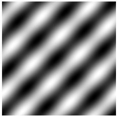
Spot h



RPN



An *ADSN* realization

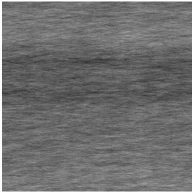


Another *ADSN* realization

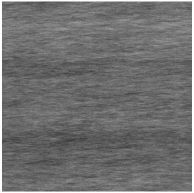
RPN and *ADSN* associated to texture images

- We add the original mean to *RPN* and *ADSN* realizations.
- Some textures are relatively well reproduced by *RPN* and *ADSN*.

Original image



RPN



ADSN

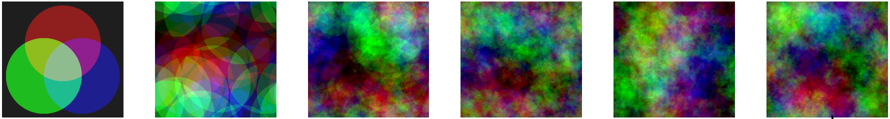
- ... But several developments are necessary to derive texture synthesis algorithms from sample.

Extension to color images

- We use the RGB color representation for color images.
- **Color *ADSN* :** The definition of Discrete Spot Noise extends to color images $h = (h_r, h_g, h_b)$.
- The color *ADSN* Y is the limit Gaussian process obtained in letting the number of spots tend to $+\infty$. It is simulated by :

$$Y = \frac{1}{\sqrt{MN}} \begin{pmatrix} (h_r - m_r\mathbf{1}) * W \\ (h_g - m_g\mathbf{1}) * W \\ (h_b - m_b\mathbf{1}) * W \end{pmatrix}, \quad W \text{ a Gaussian white noise.}$$

- One convolves each color channel with the **same** Gaussian white noise W.



Spot h | $n = 10$ | $n = 10^2$ | $n = 10^3$ | $n = 10^4$ | color *ADSN*

- Phase of color *ADSN* : The same random phase is added to the Fourier transform of each color channel.

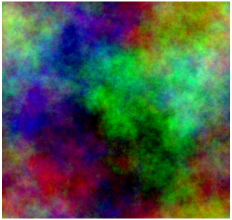
Extension to color images

- **Color *RPN* :** By analogy, the *RPN* associated with a color image $h = (h_r, h_g, h_b)$ is the color image obtained by **adding the same random phase** to the Fourier transform of each color channel.

Original image h

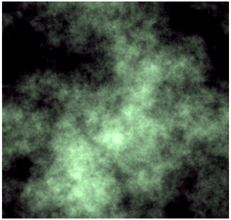
$$\hat{h} = \begin{pmatrix} |\hat{h}_R| e^{i\varphi_R} \\ |\hat{h}_G| e^{i\varphi_G} \\ |\hat{h}_B| e^{i\varphi_B} \end{pmatrix}$$

Color *RPN*



$$\hat{Z} = \begin{pmatrix} |\hat{h}_R| e^{i(\varphi_R + \theta)} \\ |\hat{h}_G| e^{i(\varphi_G + \theta)} \\ |\hat{h}_B| e^{i(\varphi_B + \theta)} \end{pmatrix}$$

"Wrong *RPN*" : each channel has the same random phase

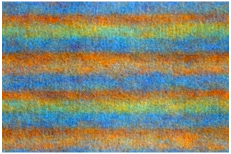


$$\hat{Z}_W = \begin{pmatrix} |\hat{h}_R| e^{i\theta} \\ |\hat{h}_G| e^{i\theta} \\ |\hat{h}_B| e^{i\theta} \end{pmatrix}$$

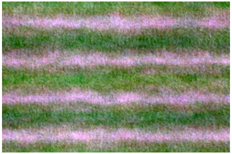
Extension to color images

- Another example with a real-world texture.

Original image h



Color *RPN*



"Wrong *RPN*"

- Preserving the original phase displacement between the color channels is essential for color consistency.
- ...however for most monochromatic textures, there is no huge difference.

Original image h

Color *RPN*

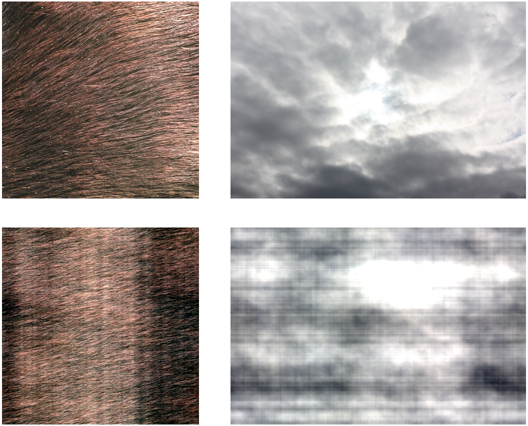
"Wrong *RPN*"

Avoiding artifacts due to non periodicity

- Both *ADSN* and *RPN* algorithms are based on the fast Fourier transform (FFT).
  $\Longrightarrow$ implicit hypothesis of periodicity
- Using non periodic samples yields important artifacts.

Spot h

ADSN



Avoiding artifacts due to non periodicity

- A good solution : Force the periodicity of the input sample.
- The original image h is replaced by its **periodic component** $p = \mathrm{per}(h)$, see L. Moisan's course.
- Definition of the periodic component p of h : p unique solution of

$$\begin{cases} \Delta p = \Delta_i h \\ \mathrm{mean}(p) = \mathrm{mean}(h) \end{cases}$$

where, noting N_x the neighborhood of $x \in \Omega$ for 4-connexity :

$$\Delta f(x) = 4f(x) - \sum_{y \in N_x} f(y) \quad \text{and} \quad \Delta_i f(x) = |N_x \cap \Omega| f(x) - \sum_{y \in N_x \cap \Omega} f(y).$$

These two Laplacians only differ at the border :
  - Δ : discrete Laplacian with periodic conditions
  - Δ_i : discrete Laplacian without periodic conditions (index i for interior)

- p is "visually close" to h (same Laplacian).
- p is fastly computed using the FFT...

FFT-based Poisson Solver

Periodic Poisson problem : Find the image p such that

$$\begin{cases} \Delta p = \Delta_i h \\ \mathrm{mean}(p) = \mathrm{mean}(h) \end{cases}$$

In the Fourier domain, this system becomes :

$$\begin{cases} \left(4 - 2\cos\left(\frac{2s\pi}{M}\right) - 2\cos\left(\frac{2t\pi}{N}\right)\right)\hat{p}(s,t) = \widehat{\Delta_i h}(s,t), \; (s,t) \in \Omega \setminus \{(0,0)\}, \\ \hat{p}(0,0) = \mathrm{mean}(h). \end{cases}$$

Algorithm to compute the periodic component :

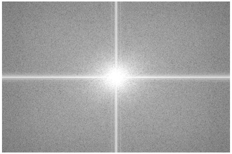
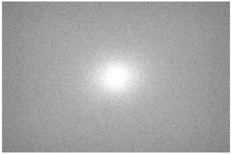
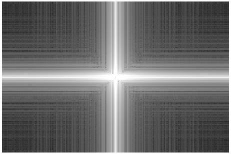
1. Compute $\Delta_i h$ the discrete Laplacian of h.
2. Compute $m = \mathrm{mean}(h)$.
3. Compute $\widehat{\Delta_i h}$ the DFT of $\Delta_i h$ using the forward FFT.
4. Compute the DFT $\hat{p}$ of p defined by

$$\begin{cases} \hat{p}(s,t) = \frac{\widehat{\Delta_i h}((s,t))}{-4+2\cos\left(\frac{2s\pi}{M}\right)+2\cos\left(\frac{2t\pi}{N}\right)} & \text{for } (s,t) \neq (0,0) \\ \hat{p}(0,0) = m \end{cases}$$

5. Compute p using the backward FFT (if necessary).

Periodic component : effects on the Fourier modulus

- p is “visually close” to h (same Laplacian).

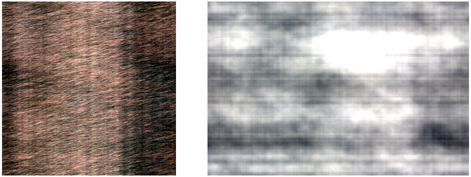
	Image h	Periodic component $p = \mathrm{per}(h)$	Smooth component $s = h - p \ (+m)$
Images			
Fourier modulus			

- The application $\mathrm{per} : h \mapsto p$ filters out the “cross structure” of the spectrum.

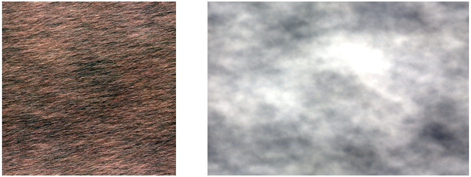
Avoiding artifacts due to non periodicity

Spot h

ADSN(h)



ADSN(p)



Synthesizing textures having arbitrary large size

Ad hoc solution : To synthesize a texture larger than the original spot h, one computes an “equivalent spot” $\tilde{h}$:

- Copy $p = \mathrm{per}(h)$ in the center of a constant image equal to the mean of h.
- Normalize the variance.
- Attenuate the transition at the inner border.

Spot h

Equivalent spot $\tilde{h}$

RPN(h)

RPN$\left(\tilde{h}\right)$

- Not really rigorous... The envelope changes the covariance.

Properties of the resulting algorithms

- Algorithms *RPN* and *ADSN* are both fast, with the complexity of the FFT $[\mathcal{O}(MN\log(MN))]$.
- Visual stability : All the realizations obtained from the same input image are visually similar.

Spot h

RPN 1

RPN 2

RPN 3

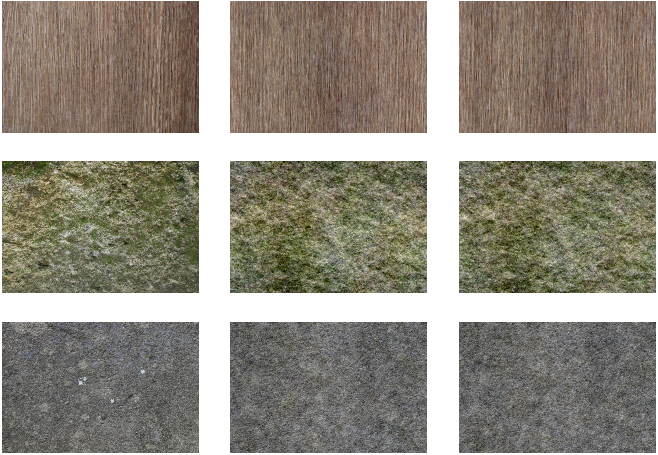
Practical exercise 4 :

Try the *RPN* algorithm online at
`http://http://www.ipol.im/pub/art/2011/ggm_rpn/`.
What do you observe ? Does it always « work well » ?

Numerical results : similarity of the textures

- In order to compare both algorithms, the same random phase is used for *ADSN* and *RPN*.

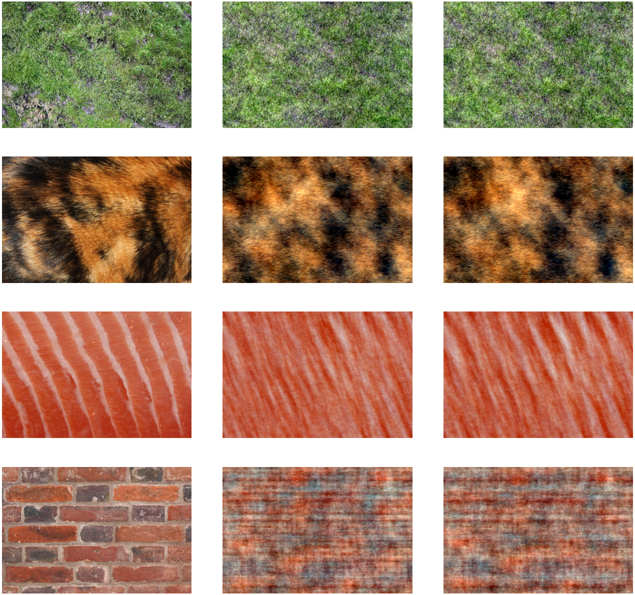
Image h	*ADSN*	*RPN*



- Both algorithms produce visually similar textures.

Numerical results : non random phase textures

Image h | *ADSN* | *RPN*



Texton associated with a Gaussian texture

Theoretical and practical exercise 5 :

1. Let g and h be two zero-mean spot functions. Show that they define the same *ADSN* model if and only if their Fourier amplitudes are equal.
2. Let h be a zero-mean spot function. We define the *texton* of h as the image $T(h) : \Omega \to \mathbb{R}$ that satisfies $\widehat{T(h)} = |\hat{h}|$. Prove that $T(h)$ is solution of the two following optimization problems : find

$$v : \Omega \to \mathbb{R} \text{ that maximizes } v(0) \text{ under the constraint } |\widehat{v}| = |\widehat{h}|.$$

$$v : \Omega \to \mathbb{R} \text{ that minimizes } \parallel \nabla \widehat{v} \parallel_p^p \text{ under the constraint } |\widehat{v}| = |\widehat{h}|.$$

3. Show that the texton has also the following properties :
   3.1 $T(h)$ is a real and symmetric image.
   3.2 $T(T(h)) = T(h)$.
   3.3 The operator T is 1-Lipschitz for the L^2 norm : if u and v are two images defined on Ω, then $\parallel T(u) - T(v) \parallel_2 \leq \parallel u - v \parallel_2$.
   3.4 The texton is translation invariant : $T(\tau_y(u)) = T(u)$ for all $y \in \Omega$.
4. Practice : take a grey-level texture image, compute its texton, and try to visualize it.

Texton associated with a Gaussian texture

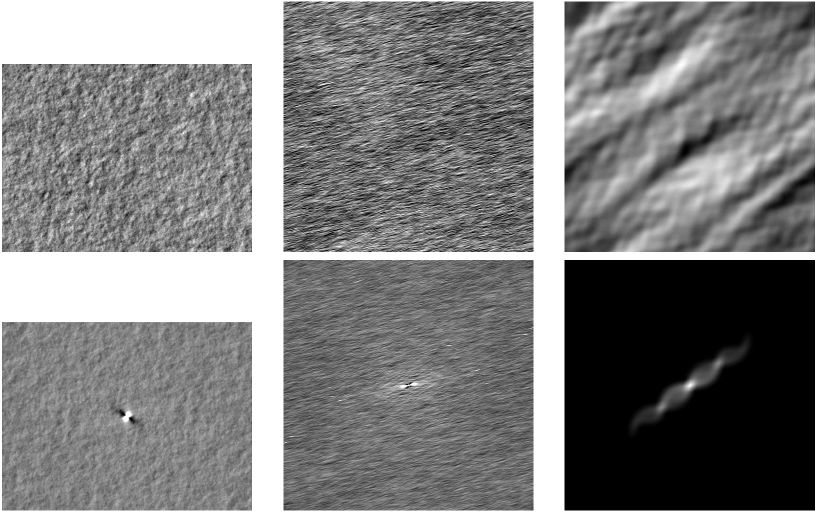
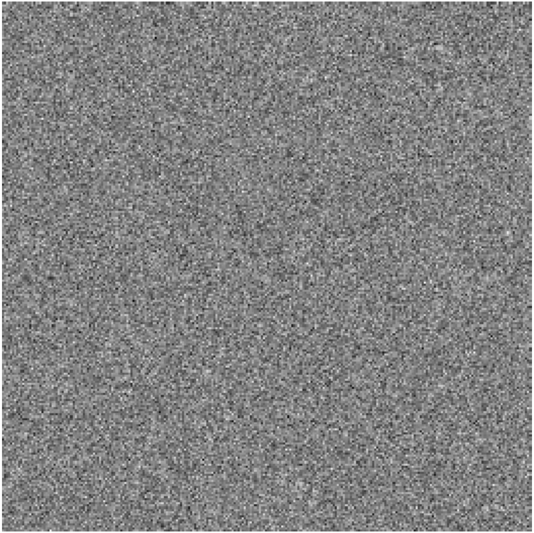


FIGURE: First line : texture images. Second line : texton images.

PART II : detections

Detecting geometric structures in images



What do you see ?

Helmholtz Principle (Non-accidentalness principle)

Two ways to state the non-accidentalness principle :

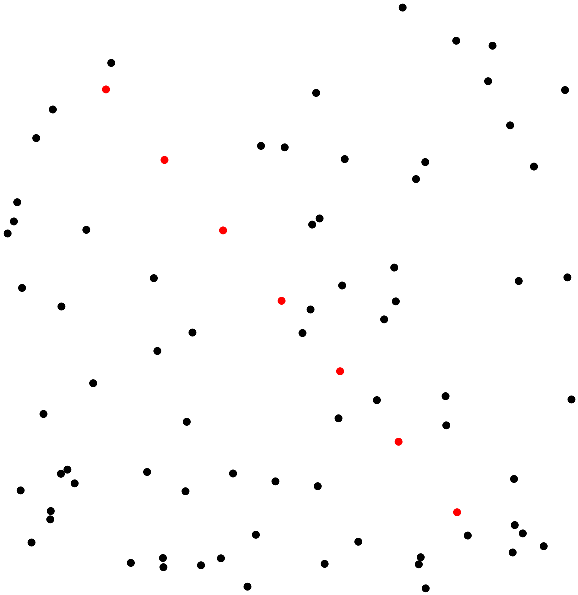
1. First way is common sense : *“we don’t see anything in a noise image”* (Attneave 1954)
2. Stronger statement : *“we perceive what has a low probability of arriving by accident”*, in other words *“if a large deviation from randomness occurs, then a structure is perceived”* (Witkin and Tenenbaum 1983, Lowe 1985)

F. Attneave, Some informational aspects of visual perception. *Psych. Rev.*, 1954.
A.P. Witkin and J. Tenenbaum. On the role of structure in vision, In *Human and Machine Vision*, 1983.

D. Lowe. *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, 1985.

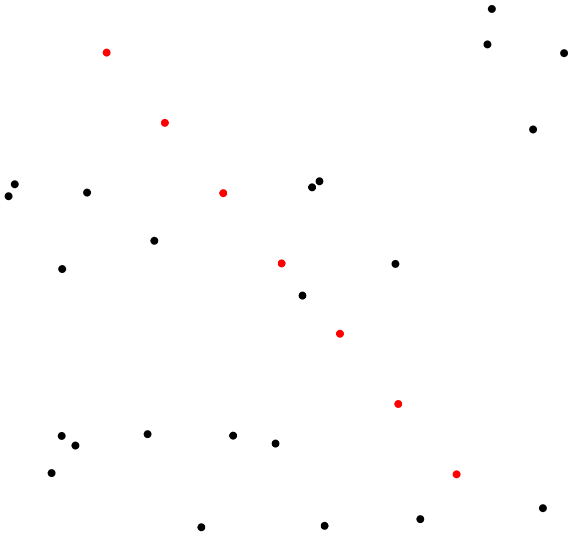
Illustrating example

Illustrating example



Illustrating example

Illustrating example



A contrario framework

General framework : just need an *a contrario* model of what the image is not (pure noise)

- Observe E a geometric event in an image

A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis : A Probabilistic Approach*, Springer-Verlag, 2008.

A contrario framework

General framework : just need an *a contrario* model of what the image is not (pure noise)

- Observe E a geometric event in an image
- Compute $\mathrm{NFA}(E)$ that is the expected number of occurrences of E in an image following the a contrario model.

A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis : A Probabilistic Approach*, Springer-Verlag, 2008.

A contrario framework

General framework : just need an *a contrario* model of what the image is not (pure noise)

- Observe E a geometric event in an image
- Compute $\mathrm{NFA}(E)$ that is the expected number of occurrences of E in an image following the a contrario model.
- **Definition :** Let $\epsilon > 0$. If $\mathrm{NFA}(E) < \epsilon$ then E is called *an ϵ-meaningful event*.

A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis : A Probabilistic Approach*, Springer-Verlag, 2008.

Detection of straight segments in an image

LSD (Line Segment Detector) algorithm of Grompone et. al. (2010) :

Let $\Omega = \{1, \ldots, M\} \times \{1, \ldots, N\}$ be a discrete domain and let $u_0 : \Omega \to \mathbb{R}$ be an image. It orientation field is $\theta_0 : \Omega \to S^1 = [0, 2\pi)$ given by

$$\forall x \in \Omega, \quad \theta_0(x) = \frac{\pi}{2} + \mathrm{Arg}\, \frac{\nabla u_0(x)}{\|\nabla u_0(x)\|},$$

where

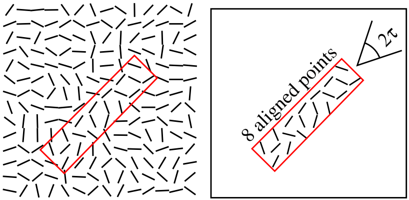
X1	X2
X3	X4

$$\nabla u_0 = \tfrac{1}{2} \begin{pmatrix} X_2 - X_1 + X_4 - X_3 \\ X_3 - X_1 + X_4 - X_2 \end{pmatrix}.$$

R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, A Fast Line Segment Detector with a False Detection Control, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

Let $r \subset \Omega$ be a rectangle with principal orientation $\varphi(r)$. Define the number of aligned pixels it contains by :

$$k(r;\theta_0) := \sum_{x \in r} \mathbb{1}_{|\theta_0(x) - \varphi(r)| \leq p\pi}.$$



R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, LSD : a Line Segment Detector, *Image Processing On Line (IPOL)*, 2012.

- Define a « pure noise model », that is a law P on orientation fields Θ, given here by : the $\Theta(x)$ are independent identically distributed uniformly on $[0, 2\pi)$.

- Define a « pure noise model », that is a law P on orientation fields Θ, given here by : the $\Theta(x)$ are independent identically distributed uniformly on $[0, 2\pi)$.

- Define the number of false alarms of the rectangle r in θ_0, under the a contrario model P by

$$\mathrm{NFA}_P(r;\theta_0) = N_{tests} \times \mathbb{P}_P[k(r;\Theta) \geq k(r;\theta_0)],$$

where N_{tests} is the number of tests, that is the number of rectangles in a $M \times N$ image ($\simeq (MN)^{5/2}$).

- In this definition, Θ is a random orientation field following the law P and $k(r;\Theta) = \sum_{x\in r} 1\!\!1_{|\Theta(x)-\varphi(r)|\leq p\pi}$ is then a random variable following a binomial distribution of parameters $n(r) = \#r$ and p.

- Define a « pure noise model », that is a law P on orientation fields Θ, given here by : the $\Theta(x)$ are independent identically distributed uniformly on $[0, 2\pi)$.

- Define the number of false alarms of the rectangle r in θ_0, under the a contrario model P by

$$\mathrm{NFA}_P(r; \theta_0) = N_{tests} \times \mathbb{P}_P[k(r; \Theta) \geq k(r; \theta_0)],$$

where N_{tests} is the number of tests, that is the number of rectangles in a $M \times N$ image ($\simeq (MN)^{5/2}$).

- In this definition, Θ is a random orientation field following the law P and $k(r; \Theta) = \sum_{x \in r} \mathbb{1}_{|\Theta(x) - \varphi(r)| \leq p\pi}$ is then a random variable following a binomial distribution of parameters $n(r) = \#r$ and p.

- When $\mathrm{NFA}_P(r; \theta_0) < \epsilon$, we say that the rectangle r is ϵ-meaningful.

Main property of the NFA

Theoretical exercise 6 : prove the following proposition :

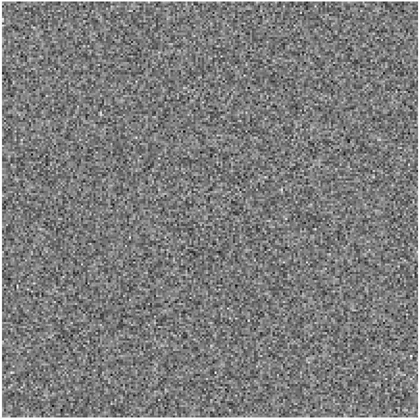
Proposition

When Θ is a random orientation field following the law P, then the $\mathrm{NFA}_P(r_i; \Theta)$, $1 \leq i \leq N_{tests}$, *become random variables and*

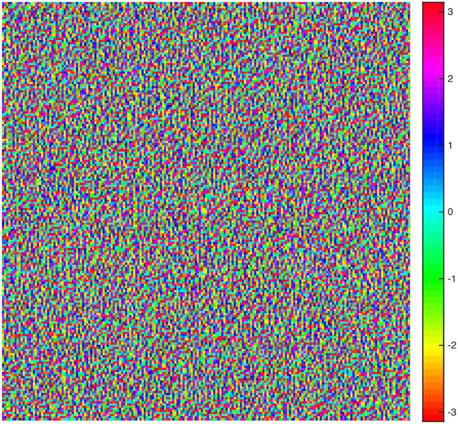
$$\mathbb{E}_P \left(\sum_{i=1}^{N_{test}} 1\!\!\mathrm{I}_{\mathrm{NFA}_P(r_i;\Theta)<\epsilon} \right) < \epsilon.$$

In other words, the expected number (under law P) of ϵ-meaningful rectangles is less than ϵ.

Example 0

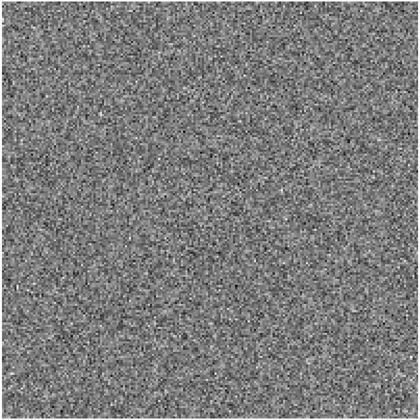


Original image u_0



Orientation field θ_0

Example 0

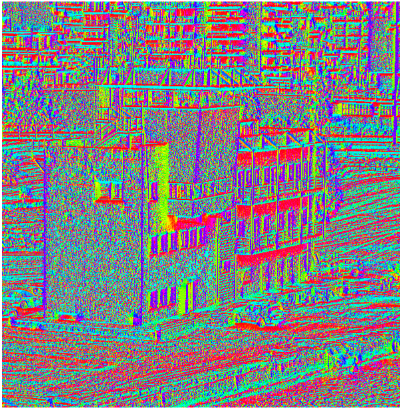


Original image u_0

LSD result

Example 1

Original image u_0



Orientation field θ_0

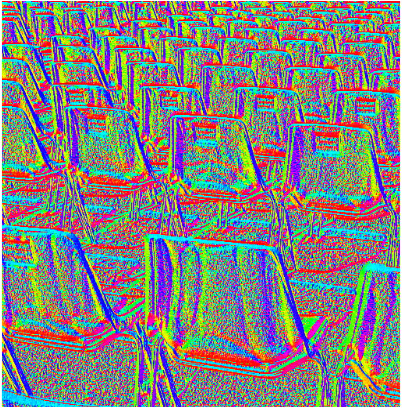
Example 1

Original image u_0

LSD result

Example 2

Original image u_0



Orientation field θ_0

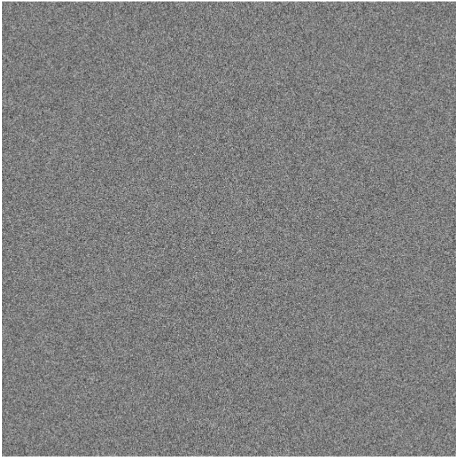
Example 2

Original image u_0

LSD result

Other Gaussian noise images

White noise image : the a contrario background noise model in the LSD algorithm

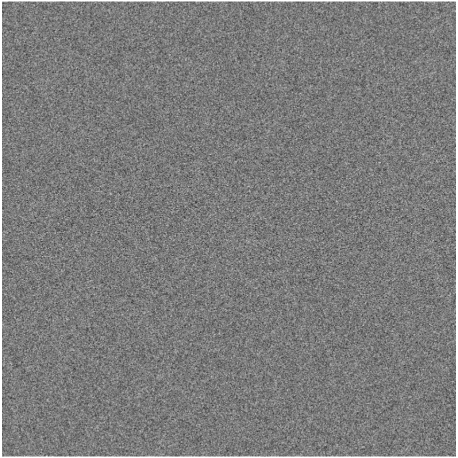


Other Gaussian noise images

Power-law Gaussian noise images, given by

$$\mathbb{E}(|\widehat{U}(\xi)|^2) = |\xi|^{-2\beta},$$

for β from 0 to 2.5 (every 0.5).

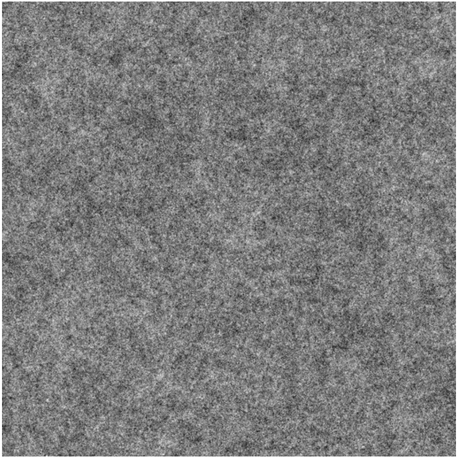


Other Gaussian noise images

Power-law Gaussian noise images, given by

$$\mathbb{E}(|\widehat{U}(\xi)|^2) = |\xi|^{-2\beta},$$

for β from 0 to 2.5 (every 0.5).

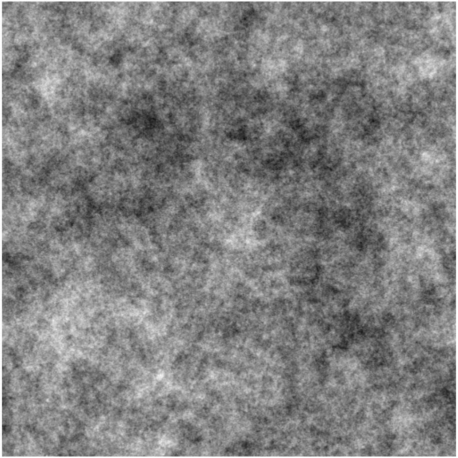


Other Gaussian noise images

Power-law Gaussian noise images, given by

$$\mathbb{E}(|\widehat{U}(\xi)|^2) = |\xi|^{-2\beta},$$

for β from 0 to 2.5 (every 0.5).

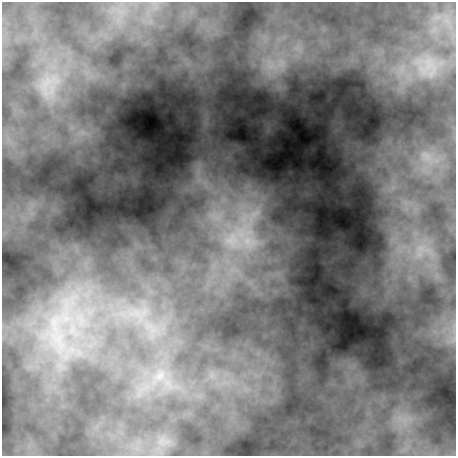


Other Gaussian noise images

Power-law Gaussian noise images, given by

$$\mathbb{E}(|\widehat{U}(\xi)|^2) = |\xi|^{-2\beta},$$

for β from 0 to 2.5 (every 0.5).

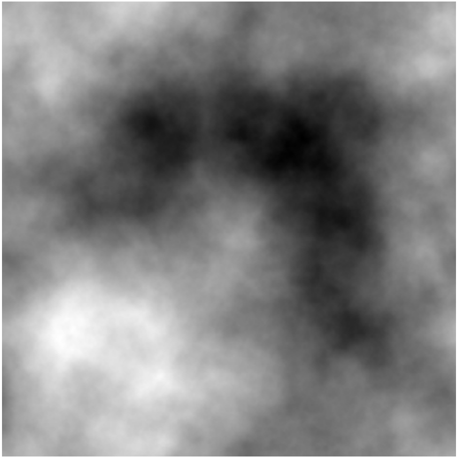


Other Gaussian noise images

Power-law Gaussian noise images, given by

$$\mathbb{E}(|\widehat{U}(\xi)|^2) = |\xi|^{-2\beta},$$

for β from 0 to 2.5 (every 0.5).

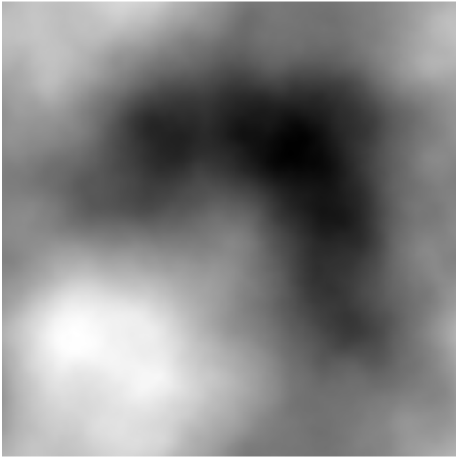


Other Gaussian noise images

Power-law Gaussian noise images, given by

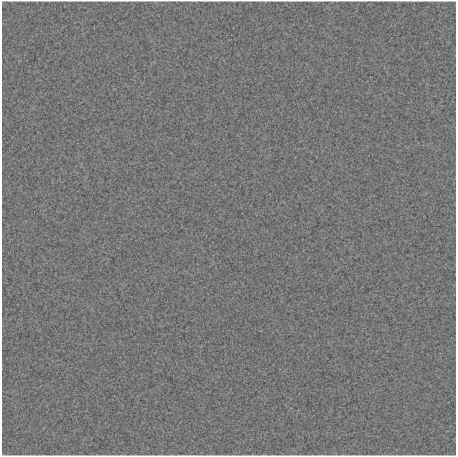
$$\mathbb{E}(|\widehat{U}(\xi)|^2) = |\xi|^{-2\beta},$$

for β from 0 to 2.5 (every 0.5).



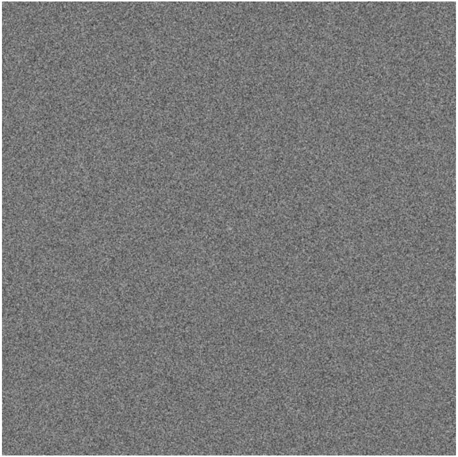
Visual perception

What do you see in a white noise image ?



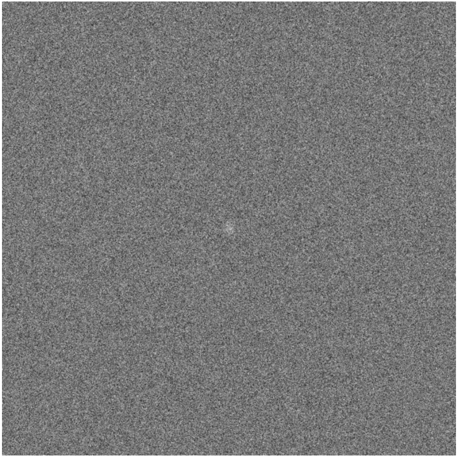
Visual perception

What do you see in a white noise image ?



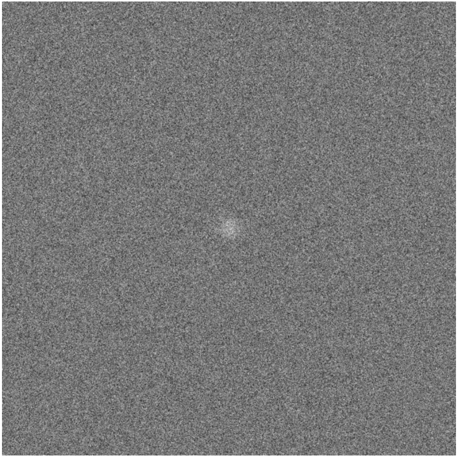
Visual perception

What do you see in a white noise image ?



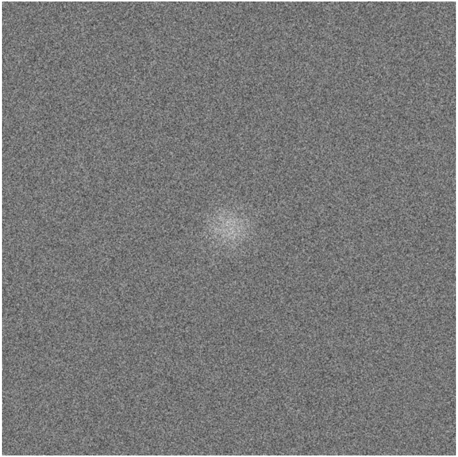
Visual perception

What do you see in a white noise image ?



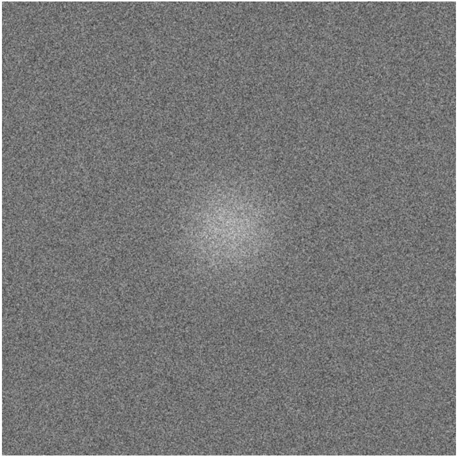
Visual perception

What do you see in a white noise image ?



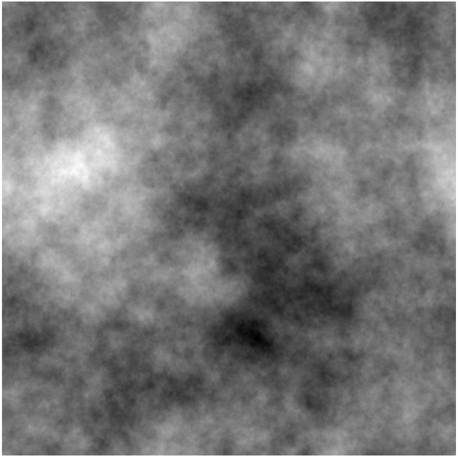
Visual perception

What do you see in a white noise image ?



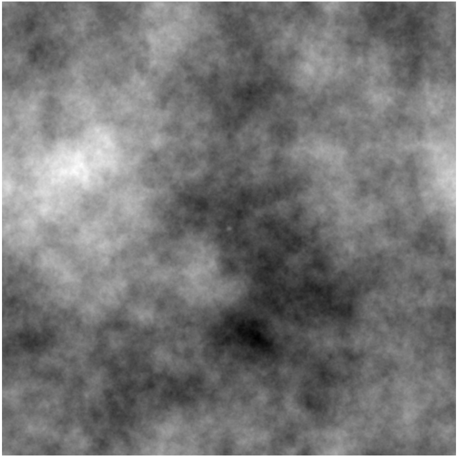
Visual perception

And in a power-law Gaussian image ?



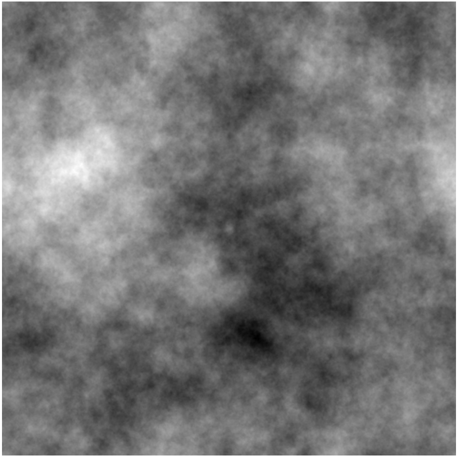
Visual perception

And in a power-law Gaussian image ?



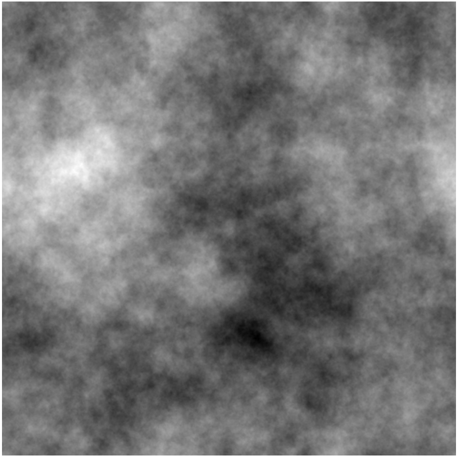
Visual perception

And in a power-law Gaussian image ?



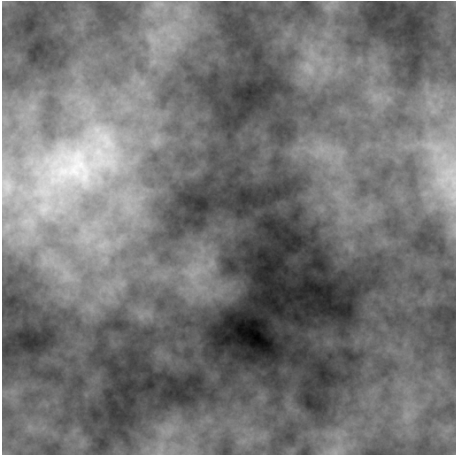
Visual perception

And in a power-law Gaussian image ?



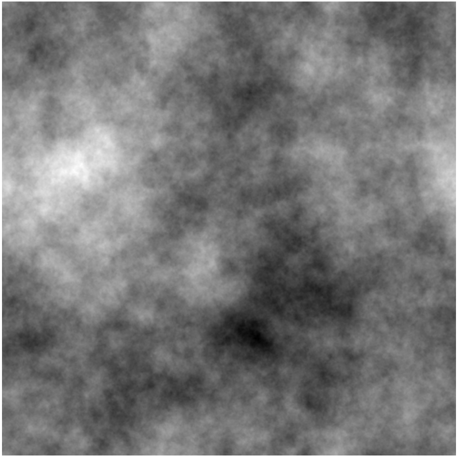
Visual perception

And in a power-law Gaussian image ?



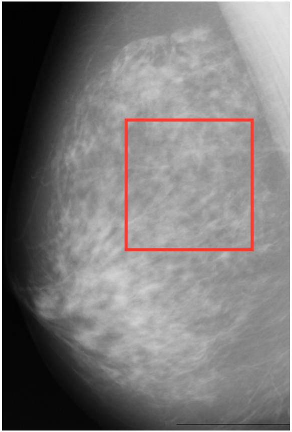
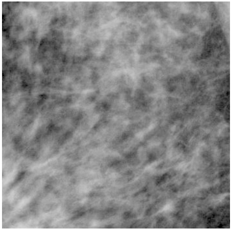
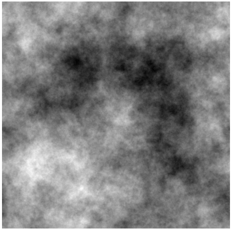
Visual perception

And in a power-law Gaussian image ?



Application : medical imaging

Questions of detectability of lesions (masses) in mammogram images.

Refs : Burgess / Grosjean-Moisan

Last exercise

Practical exercise 7 :

Sample power-law Gaussian images, and check by yourself the perceptual phenomenon of the previous slides. Try different values of β.

Conclusion of Part II

- A contrario approach : a computational approach to detect geometric structures in images.
- Many other applications : shape recognition, image matching, clustering, stereovision, denoising (grain filter), etc.
- Can we turn it into a generative approach ? -> Yes, see exponential distributions.