Les processus ponctuels déterminantaux à l'intersection de la géométrie stochastique et du traitement d'image

Agnès Desolneux

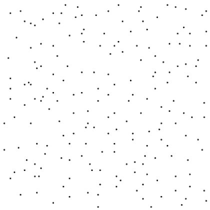
CNRS et Centre Borelli (ENS Paris-Saclay)

Journées communes GeoSto et MIA,
Rouen, 23 septembre 2022

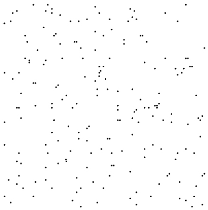
Travail en collaboration avec **Claire Launay** (Albert Einstein College of Medicine, New-York) et **Bruno Galerne** (Université d'Orléans).

Determinantal Point Processes

Determinantal Point Processes (DPP) provide a family of models of random configurations that favor **diversity** or **repulsion** between points :



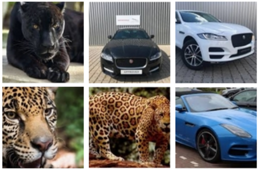
(a) Realization of a DPP



(b) Realization of a Bernoulli process

- **On continuous domains :** Introduced by Macchi (1975) for modeling fermions, regain of interest in spatial statistics (Lavancier, Møller, Rubak, 2015).

Determinantal Point Processes



I went to this place two weeks ago with my aunt and my cousins. It was a lovely sunny afternoon. We had a chocolate cake and drank an apricot juice. The employees were charming and really helpful. We stayed there the whole afternoon, laughing, playing and enjoying the nice weather. Thanks again ! I definitely recommend it !

- **On discrete domains :** Various applications in machine learning based on selection of diverse subsets :
  - Recommendation systems (Wilhelm et al., 2018).
  - Text summarization (Kulesza, Taskar, 2012 ; Dupuy, Bach, 2017).
  - Feature selection (Belhadji, Bardenet, Chainais, 2018).
  - . . .
- Advantages of (discrete) DPPs (compared to Gibbs processes) :
  - Similarity between points encoded in a **matrix** K **called kernel**
  - Moments and marginal probabilities have closed form formulas
  - Exact simulation algorithm

Discrete determinantal point processes

In this talk we work on a discrete set made of N elements that we identify with $\mathcal{Y} = \{1, \dots, N\}$.

Definition

Let K be a Hermitian matrix of size $N \times N$ such that

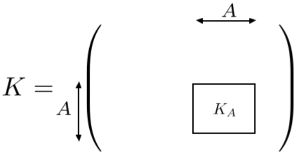
$$0 \preceq K \preceq I.$$

The random subset $Y \subset \mathcal{Y}$ defined by the inclusion probabilities

$$\forall A \subset \mathcal{Y}, \quad \mathbb{P}(A \subset Y) = \det(K_A)$$

is determinantal point process of kernel K.

One writes $Y \sim \mathrm{DPP}(K)$.



Properties of DPP

- The diagonal coefficients K_{ii} define the inclusion probability of each element i :

$$\mathbb{P}(i \in Y) = K_{ii}.$$

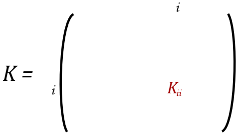
$$K = \begin{pmatrix} & & i & \\ & & & \\ i & & K_{ii} & \\ & & & \end{pmatrix}$$

Let $\{\lambda_1, \ldots, \lambda_N\} \in \mathbb{R}$ be the eigenvalues of K.

Properties of DPP

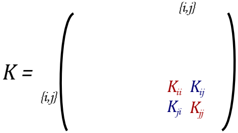
- The diagonal coefficients K_{ii} define the inclusion probability of each element i :

$$\mathbb{P}(i \in Y) = K_{ii}.$$



- The off-diagonal coefficients K_{ij} gives the repulsion between the points i and j :

$$\mathbb{P}(\{i,j\} \subset Y) = \mathbb{P}(i \in Y)\mathbb{P}(j \in Y) - |K_{ij}|^2.$$



- A DPP is repulsive since $\mathbb{P}(\{i,j\} \subset Y)$ is always smaller than in the case of independent point selection (Bernoulli process).
- By construction, DPPs are simple random sets.

Let $\{\lambda_1, \ldots, \lambda_N\} \in \mathbb{R}$ be the eigenvalues of K.

Properties of DPP

Cardinality : it satisfies $|Y| \sim \sum_{i \in \mathcal{Y}} \mathcal{B}\text{er}(\lambda_i)$

(sum of independent Bernoulli random variables of parameter λ_i). Hence

$$\mathbb{E}(|Y|) = \sum_{i \in \mathcal{Y}} \lambda_i = \mathrm{Tr}(K) = \sum_{i \in \mathcal{Y}} K_{ii}$$

$$\mathrm{Var}(|Y|) = \sum_{i \in \mathcal{Y}} \lambda_i (1 - \lambda_i)$$

$$K = \begin{pmatrix} K_{11} & & & \\ & \ddots & & \\ & & K_{ii} & \\ & & & \ddots \\ & & & & K_{NN} \end{pmatrix}$$

Properties of DPP

Cardinality : it satisfies $|Y| \sim \sum_{i \in \mathcal{Y}} \mathcal{B}\text{er}(\lambda_i)$

(sum of independent Bernoulli random variables of parameter λ_i). Hence

$$\mathbb{E}(|Y|) = \sum_{i \in \mathcal{Y}} \lambda_i = \mathrm{Tr}(K) = \sum_{i \in \mathcal{Y}} K_{ii}$$

$$\mathrm{Var}(|Y|) = \sum_{i \in \mathcal{Y}} \lambda_i (1 - \lambda_i)$$

$$K = \begin{pmatrix} K_{11} & & & \\ & \ddots & & \\ & & K_{ii} & \\ & & & \ddots \; K_{NN} \end{pmatrix}$$

Two examples of DPP :

- Bernoulli Point Process :
  Y_i are independent following some Bernoulli distribution with parameter p_i. This is a DPP for the diagonal kernel $K = \mathrm{diag}(p_1, \ldots, p_N)$.
- Projection DPP :

$$\forall i \in \mathcal{Y}, \quad \lambda_i = 0 \text{ or } 1.$$

  Notice that for projection DPP the cardinality $|Y|$ is fixed : $|Y| = \sum_i \lambda_i$.

Properties of DPP

Cardinality : it satisfies $|Y| \sim \sum_{i\in\mathcal{Y}} \mathcal{B}\text{er}(\lambda_i)$

(sum of independent Bernoulli random variables of parameter λ_i). Hence

$$K = \begin{pmatrix} K_{11} & & & & \\ & \ddots & & & \\ & & K_{ii} & & \\ & & & \ddots & \\ & & & & K_{NN} \end{pmatrix}$$

$$\mathbb{E}(|Y|) = \sum_{i\in\mathcal{Y}} \lambda_i = \mathrm{Tr}(K) = \sum_{i\in\mathcal{Y}} K_{ii}$$

$$\mathrm{Var}(|Y|) = \sum_{i\in\mathcal{Y}} \lambda_i(1-\lambda_i)$$

Two examples of DPP :

- Bernoulli Point Process :
  Y_i are independent following some Bernoulli distribution with parameter p_i. This is a DPP for the diagonal kernel $K = \mathrm{diag}(p_1, \dots, p_N)$.
- Projection DPP :
  $$\forall i \in \mathcal{Y}, \quad \lambda_i = 0 \text{ or } 1.$$
  Notice that for projection DPP the cardinality $|Y|$ is fixed : $|Y| = \sum_i \lambda_i$.

Exact sampling algorithm using the spectral decomposition of K (Hough-Krishnapur-Peres-Virág)

Motivation

Take advantage of the repulsive nature of DPP to :

- Sample subsets of well-spread pixels in image domain and use them for texture modeling based on shot noise.
- Subsample the set of patches of an image to efficiently summarize the diversity of the patches.

Outline

I. Determinantal point processes on pixels

II. Shot noise models driven by Determinantal Pixel Processes

III. Identifiability and Inference for Determinantal Pixel Processes

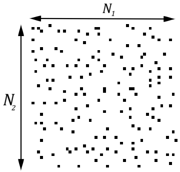
IV. Subsampling image patches using Determinantal Point Processes

Determinantal pixel processes (DPixP)

Framework for images :
Image domain : a discrete grid Ω of size $N_1 \times N_2$, then $N = N_1 N_2$ is the total number of pixels.

We consider a DPP Y defined on Ω, with kernel K, a matrix of size $N \times N$.

Hypothesis : Y *is* ***stationary*** *(with periodic boundary conditions)*



A sample

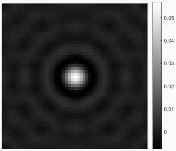
Determinantal pixel processes (DPixP)

Framework for images :
Image domain : a discrete grid Ω of size $N_1 \times N_2$, then $N = N_1 N_2$ is the total number of pixels.

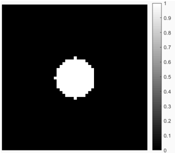
We consider a DPP Y defined on Ω, with kernel K, a matrix of size $N \times N$.

Hypothesis : Y *is* ***stationary*** *(with periodic boundary conditions)*

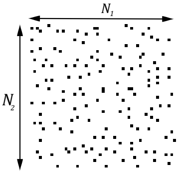
- K is a block-circulant matrix with circulant blocks : There exists a function $C : \Omega \to \mathbb{C}$ s.t.
$$\forall x, y \in \Omega, \quad K_{xy} = C(x - y).$$
- K is diagonalized in the 2D Discrete Fourier transform and the eigenvalues of K are the Fourier coefficients of C.



Kernel function C



Fourier coefficients $\widehat{C}$



A sample

The 2D discrete Fourier transform

Let $f : \Omega \to \mathbb{C}$ be a function defined on $\Omega = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\}$. Its discrete Fourier transform $\widehat{f}$ is the function defined on Ω by

$$\forall \xi \in \Omega, \ \ \widehat{f}(\xi) = \sum_{x \in \Omega} f(x) e^{-2i\pi \langle x, \xi \rangle},$$

where for $x = (x_1, x_2) \in \Omega$ and $\xi = (\xi_1, \xi_2) \in \Omega$, we denote the scalar product

$$\langle x, \xi \rangle = \frac{x_1 \xi_1}{N_1} + \frac{x_2 \xi_2}{N_2}.$$

The 2D discrete Fourier transform

Let $f : \Omega \to \mathbb{C}$ be a function defined on $\Omega = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\}$. Its discrete Fourier transform $\widehat{f}$ is the function defined on Ω by

$$\forall \xi \in \Omega, \;\; \widehat{f}(\xi) = \sum_{x \in \Omega} f(x) e^{-2i\pi \langle x, \xi \rangle},$$

where for $x = (x_1, x_2) \in \Omega$ and $\xi = (\xi_1, \xi_2) \in \Omega$, we denote the scalar product

$$\langle x, \xi \rangle = \frac{x_1 \xi_1}{N_1} + \frac{x_2 \xi_2}{N_2}.$$

1. **Inversion :** we can recover f from $\widehat{f}$, by the inverse discrete Fourier transform
$$\forall x \in \Omega, \;\; f(x) = \frac{1}{|\Omega|} \sum_{\xi \in \Omega} \widehat{f}(\xi) e^{2i\pi \langle x, \xi \rangle}.$$
2. **Parseval Theorem :**
$$\|f\|_2^2 = \sum_{x \in \Omega} |f(x)|^2 = \frac{1}{|\Omega|} \sum_{\xi \in \Omega} |\widehat{f}(\xi)|^2 = \frac{1}{|\Omega|} \|\widehat{f}\|_2^2.$$
3. **Convolution/Product :** The (periodic) convolution being defined by
$$\forall x \in \Omega, \; f \star g(x) = \sum_{y \in \Omega} f(y) g(x - y), \;\text{ then } \forall \xi \in \Omega, \; \widehat{f \star g}(\xi) = \widehat{f}(\xi) \widehat{g}(\xi).$$

Determinantal pixel processes (DPixP)

Definition

Let $C : \Omega \to \mathbb{C}$ be a function defined on Ω such that

$$\forall \xi \in \Omega, \quad \widehat{C}(\xi) \text{ is real and } 0 \leqslant \widehat{C}(\xi) \leqslant 1.$$

Such a function will be called an admissible kernel. A random set $X \subset \Omega$ is called a determinantal pixel process (DPixP) with kernel C, if

$$\forall A \subset \Omega, \quad \mathbb{P}(A \subset X) = \det(K_A),$$

with K_A the matrix of size $|A| \times |A|$ s.t. $K_A = (C(x-y))_{x,y \in A}$.

Properties of DPixP

Cardinality : $|X| \sim \sum_{\xi\in\Omega} \mathcal{B}\text{er}(\widehat{C}(\xi))$ and in particular

$$\mathbb{E}(|X|) = \sum_{\xi\in\Omega} \widehat{C}(\xi) = |\Omega| C(0) \quad \text{and} \quad \mathrm{Var}(|X|) = \sum_{\xi\in\Omega} \widehat{C}(\xi)(1-\widehat{C}(\xi))$$

Properties of DPixP

Cardinality : $|X| \sim \sum_{\xi \in \Omega} \mathcal{B}\text{er}(\widehat{C}(\xi))$ and in particular

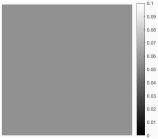
$$\mathbb{E}(|X|) = \sum_{\xi \in \Omega} \widehat{C}(\xi) = |\Omega| C(0) \quad \text{and} \quad \mathrm{Var}(|X|) = \sum_{\xi \in \Omega} \widehat{C}(\xi)(1 - \widehat{C}(\xi))$$

Two examples :

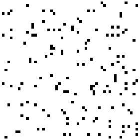
1. Bernoulli Process :

$$C(0) = p \;\text{ and }\; C(x) = 0,\ \forall x \in \Omega \setminus \{0\}$$

$$\Leftrightarrow \quad \forall \xi \in \Omega, \widehat{C}(\xi) = p.$$



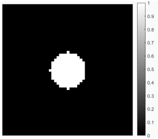
$\widehat{C}$



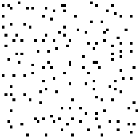
Realization

2. Projection DPixP :

$$\forall \xi \in \Omega, \quad \widehat{C}(\xi)(1 - \widehat{C}(\xi)) = 0.$$



$\widehat{C}$



Realization

Properties of DPixP

Remark : Bernoulli point processes have the property of being the processes such that $\mathrm{Var}(|X|)$ is maximal among all DPixP with same $\mathbb{E}(|X|)$.

Indeed, let $p \in [0,1]$ and let C be any admissible kernel such that $\mathbb{E}(|X|) = \sum_{\xi\in\Omega} \widehat{C}(\xi) = p|\Omega|$. Then, by Cauchy-Schwarz inequality,

$$\begin{aligned}
\mathrm{Var}(|X|) &= \sum_{\xi\in\Omega} \widehat{C}(\xi) - \sum_{\xi\in\Omega} \widehat{C}(\xi)^2 = p|\Omega| - \sum_{\xi\in\Omega} \widehat{C}(\xi)^2 \\
&\leqslant p|\Omega| - \frac{1}{|\Omega|}\left(\sum_{\xi\in\Omega} \widehat{C}(\xi)\right)^2 = p(1-p)|\Omega|.
\end{aligned}$$

And the equality holds when all $\widehat{C}(\xi)$ are equal to p, i.e. $C = p\delta_0$.

Sequential simulation of a DPixP

Let us denote, for $\xi \in \Omega$, the function φ_ξ defined on Ω by

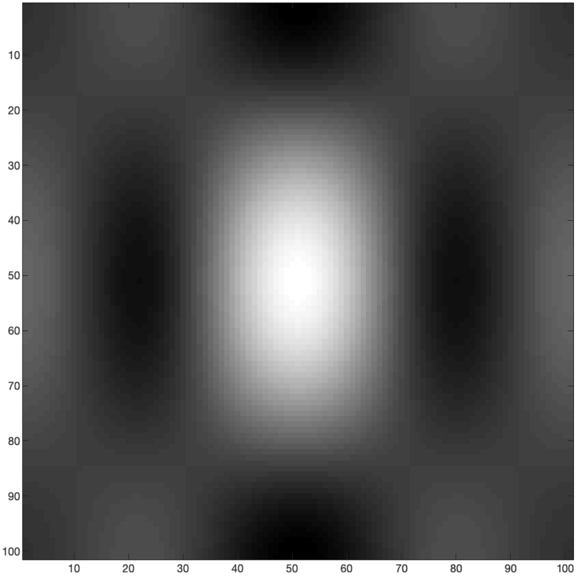
$$\forall x \in \Omega, \quad \varphi_\xi(x) = \frac{1}{\sqrt{MN}} e^{2i\pi\langle x,\xi\rangle}.$$

Then $\{\varphi_\xi\}_{\xi\in\Omega}$ is an orthonormal basis of $L^2(\Omega;\mathbb{C})$.

Algorithm : Sequential simulation of a DPixP

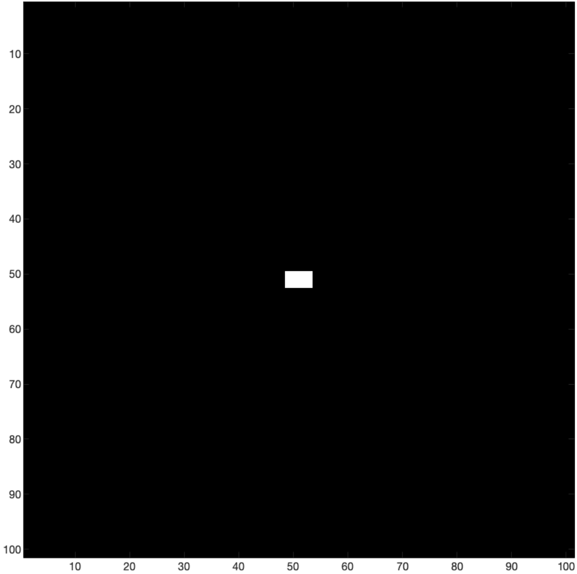
- Sample a random field $U = (U_\xi)_{\xi\in\Omega}$ where the U_ξ are i.i.d. uniform on $[0,1]$.
- Define the “active frequencies” $\{\xi_1,\dots,\xi_n\} = \{\xi \in \Omega; U(\xi) \leqslant \widehat{C}(\xi)\}$, and denote,
  $$\forall x \in \Omega, \quad v(x) = (\varphi_{\xi_1}(x),\dots,\varphi_{\xi_n}(x)) \in \mathbb{C}^n.$$
- For $k = 1$ to n do :
  - Sample X_1 uniform on Ω, and define $e_1 = v(X_1)/\|v(X_1)\|$.
  - For $k = 2$ to n, sample X_k from the probability density p_k on Ω, defined by
    $$\forall x \in \Omega, \quad p_k(x) = \frac{1}{n-k+1}\left(\frac{n}{MN} - \sum_{j=1}^{k-1} |e_j^* v(x)|^2\right)$$
  - Define $e_k = w_k/\|w_k\|$ where $w_k = v(X_k) - \sum_{j=1}^{k-1} e_j^* v(X_k) e_j$.
- Return $X = (X_1,\dots,X_n)$.

Sequential simulation of a DPixP : example



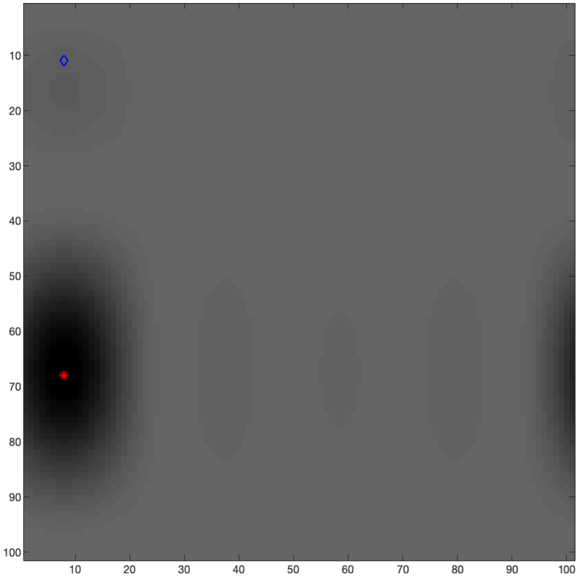
Kernel C

Sequential simulation of a DPixP : example



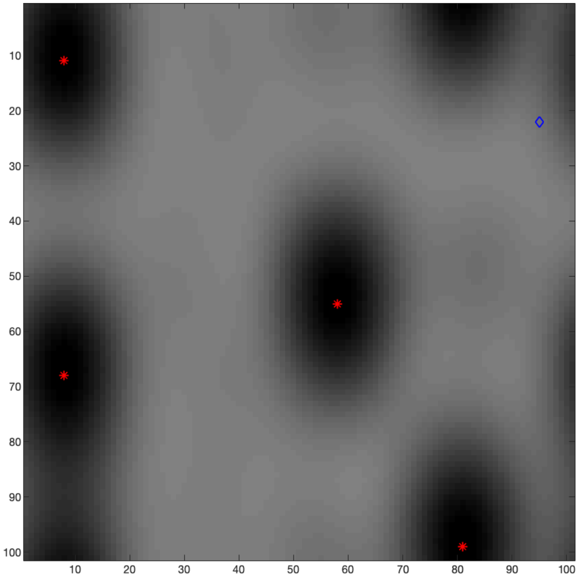
In the frequency domain : $\widehat{C}$

Sequential simulation of a DPixP : example



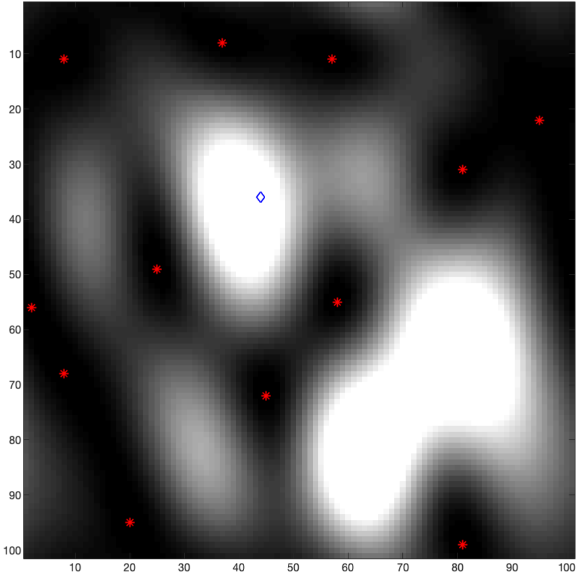
Sequential sampling at step 2

Sequential simulation of a DPixP : example



Sequential sampling at step 5

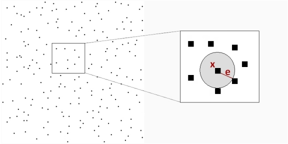
Sequential simulation of a DPixP : example



Sequential sampling at step 13

DPixP and hard-core repulsion

Can we impose a minimal distance between points of a DPixP ? What are the consequences on the kernel C ?



Proposition

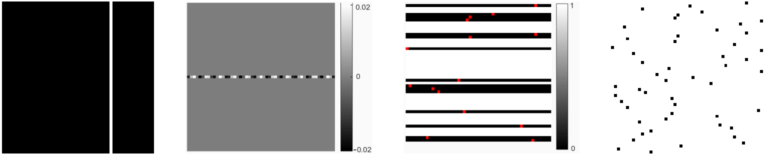
Let us consider $X \sim \mathrm{DPixP}(C)$ on Ω and $e \in \Omega$. Then the following propositions are equivalent :

1. For all $x \in \Omega$, the probability that x and $x + e$ belong simultaneously to X is zero.
2. For all $x \in \Omega$, the probability that x and $x + \lambda e$ belong simultaneously to X is zero for $\lambda \in \mathbb{Q}$ such that $\lambda e \in \Omega$.
3. There exists $\theta \in \mathbb{R}$ such that the only frequencies $\xi \in \Omega$ such that $\widehat{C}(\xi)$ is nonzero are located on the discrete line defined by $\langle e, \xi \rangle = \theta$.
4. X contains almost surely at most one point on every discrete line of direction e.

This is called directional repulsion.

DPixP and hard-core repulsion

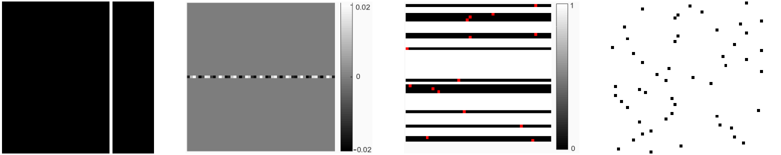
Example : Horizontal repulsion



$\widehat{C}$ | Real part of C | Density during sampling | Realization

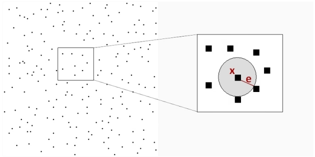
DPixP and hard-core repulsion

Example : Horizontal repulsion



$\widehat{C}$ Real part of C Density during sampling Realization

Conclusion on hard-core repulsion : The only DPixP imposing a minimum distance between the points is the degenerate DPixP made of a single pixel.

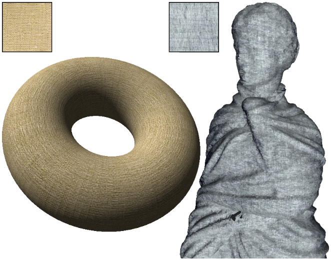


Shot noise and texture modeling

The **spot noise** was introduced by J. van Wijk (*Computer Graphics*, 1991) for texture synthesis. Using a Poisson points process $\{x_i\} \subset \mathbb{R}^2$, it has the form

$$\forall x \in \mathbb{R}^2, \quad S(x) = \sum_i \beta_i g(x - x_i).$$

Lagae et al. “Procedural noise using sparse Gabor convolution”, SIGGRAPH 2009



Galerne, Leclaire, Moisan, “Texton noise”, CGF 2017, based on Gaussian limit of Poisson shot noise.

Shot noise driven by a DPixP

Definition : Shot noise driven by a DPixP

Let C be an admissible kernel, and let g be a function defined on Ω. Then, the shot noise random field S driven by the DPixP of kernel C and the spot g is defined by

$$\forall x \in \Omega, \quad S(x) = \sum_{x_i \in X} g(x - x_i),$$

where $X = \{x_i\}$ is a DPixP of kernel C.

Shot noise driven by a DPixP

Definition : Shot noise driven by a DPixP

Let C be an admissible kernel, and let g be a function defined on Ω. Then, the shot noise random field S driven by the DPixP of kernel C and the spot g is defined by

$$\forall x \in \Omega, \quad S(x) = \sum_{x_i \in X} g(x - x_i),$$

where $X = \{x_i\}$ is a DPixP of kernel C.

To compute the moments (mean, variance, kurtosis, etc.) of S, we first need to have a “Mecke-Campbell-Slivnyak” type formula in the DPixP framework.

Proposition : Moments formula

Let X be a DPixP of kernel C, let $k \geqslant 1$ be an integer, and let f be a function defined on Ω^k. Then

$$\mathbb{E}\left[\sum_{x_{i_1},\dots,x_{i_k} \in X}^{\neq} f(x_{i_1},\dots,x_{i_k})\right] = \sum_{y_1,\dots,y_k \in \Omega} f(y_1,\dots,y_k)\det(C(y_i - y_j)_{1\leqslant i,j\leqslant k})$$

Shot noise driven by a DPixP : Moments

1. Mean value :
$$\mathbb{E}(S(0)) = C(0) \sum_{y \in \Omega} g(y) = C(0)\widehat{g}(0).$$
2. Covariance : (assume $\widehat{g}(0) = 0$)
$$\forall x \in \Omega, \quad \Gamma_S(x) := \mathrm{Cov}(S(0), S(x)) = C(0) g \star g_-(x) - (g \star g_- \star |C|^2)(x),$$
where $g_-(x) := g(-x)$. And therefore
$$\mathrm{Var}(S(0)) = C(0) \sum_{y \in \Omega} g(y)^2 - (g \star g_- \star |C|^2)(0)$$
and $\widehat{\Gamma_S}(\xi) = |\widehat{g}(\xi)|^2 (C(0) - \widehat{|C|^2}(\xi)).$

The variance depends on the spot g and the DPP kernel C in a non trivial way.

Shot noise driven by a DPixP

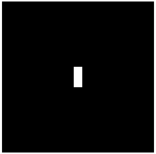
$$\mathrm{Var}(S(0)) = C(0)\sum_{y\in\Omega} g(y)^2 - (g \star g_- \star |C|^2)(0)$$
$$= \frac{n}{|\Omega|^2}\sum_{\xi\in\Omega} |\widehat{g}(\xi)|^2 - \frac{1}{|\Omega|^2}\sum_{\xi,\xi'\in\Omega} |\widehat{g}(\xi-\xi')|^2 \widehat{C}(\xi)\widehat{C}(\xi').$$

Proposition : Shot noise with extreme variance

Consider a spot function $g : \Omega \to \mathbb{R}^+$ and $n \in \mathbb{N}$ an expected cardinality for the DPixP.
Maximal variance : The DPixP with expected cardinality n associated with the *spot* g reaching maximal variance is the **Bernoulli process**.
Minimal variance : The DPixP with expected cardinality n associated with the *spot* g reaching minimal variance is the **projection DPixP** of n points, such that the n frequencies $\{\xi_1, ..., \xi_n\}$ associated with the non-zero Fourier coefficients are localized to maximize $\sum_{\xi,\xi'\in\{\xi_1,\dots,\xi_n\}} |\widehat{g}(\xi-\xi')|^2$.

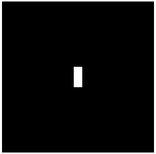
To approximate the maximization of the quadratic functional we use a simple greedy algorithm.

Shot noise driven by a DPixP

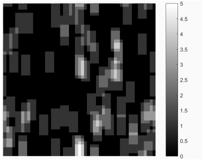


Spot g

Shot noise driven by a DPixP

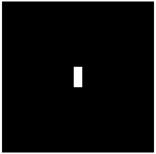


Spot g

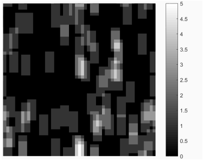


Shot noise with maximal variance (BPP)

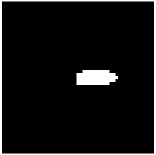
Shot noise driven by a DPixP



Spot g

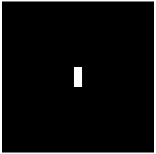


Shot noise with maximal variance (BPP)

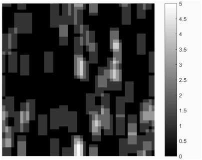


Fourier Coefficients from greedy algorithm

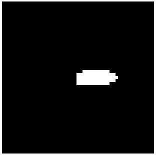
Shot noise driven by a DPixP



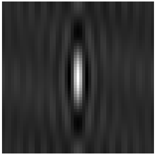
Spot g



Shot noise with maximal variance (BPP)

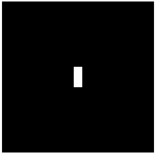


Fourier Coefficients from greedy algorithm

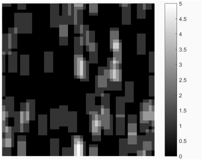


Kernel C

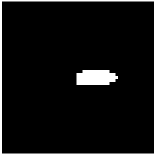
Shot noise driven by a DPixP



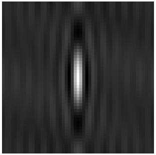
Spot g



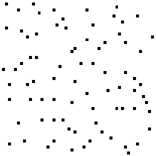
Shot noise with maximal variance (BPP)



Fourier Coefficients from greedy algorithm

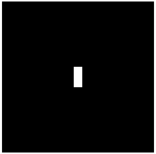


Kernel C

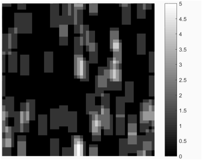


A realization of $\mathrm{DPixP}(C)$

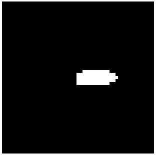
Shot noise driven by a DPixP



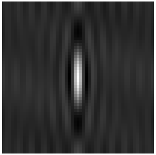
Spot g



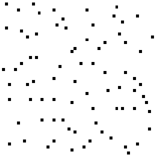
Shot noise with maximal variance (BPP)



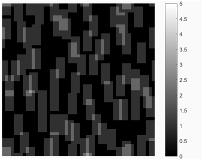
Fourier Coefficients from greedy algorithm



Kernel C

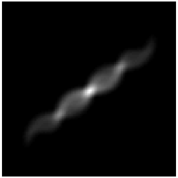


A realization of $\mathrm{DPixP}(C)$

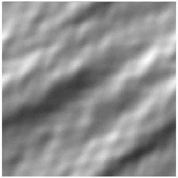


Shot noise with minimal variance

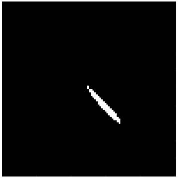
Shot noise driven by a DPixP



Spot g

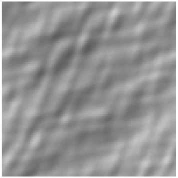


Shot noise with maximal variance (BPP)



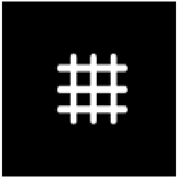
Fourier Coefficients from greedy algorithm

Kernel C de ce DPixP

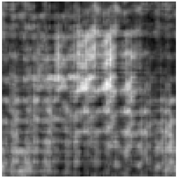


Shot noise with minimal variance

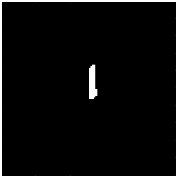
Shot noise driven by a DPixP



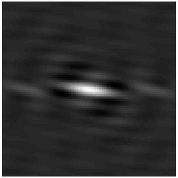
Spot g



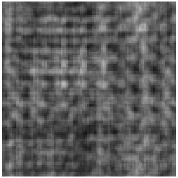
Shot noise with maximal variance (BPP)



Fourier Coefficients from greedy algorithm



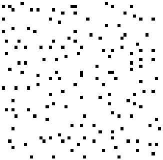
Kernel C de ce DPixP



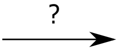
Shot noise with minimal variance

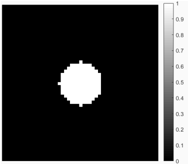
Inference for DPixP

Inference : We look for a kernel C that would correspond to one (or several) realizations of a subset of pixels.



A given realization

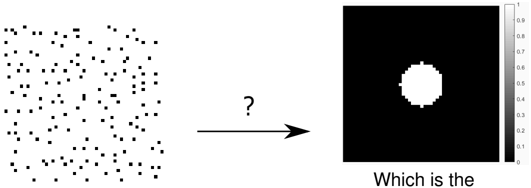




Which is the corresponding DPixP(C) ?

Inference for DPixP

Inference : We look for a kernel C that would correspond to one (or several) realizations of a subset of pixels.



A given realization

Which is the corresponding $\mathrm{DPixP}(C)$?

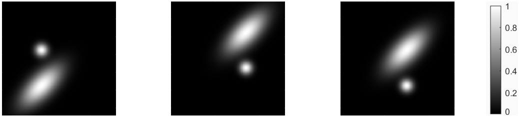
Identifiability of the problem :
What is the equivalence class of a given kernel C ?

Inference for DPixP - Identifiability

Proposition

Let C_1, C_2 be two kernels defined on Ω, satisfying some *reasonable hypotheses*[1].
Then, $\mathrm{DPixP}(C_1) = \mathrm{DPixP}(C_2)$ if and only if the Fourier coefficients of C_2 are **translated and/or symmetric with respect to** $(0, 0)$ from the Fourier coefficients of C_1

Three DPixP kernels belonging the same equivalence class : they parameterize the same DPixP



$\widehat{C_1}$ $\widehat{C_2}$ $\widehat{C_3}$

. [1] Hartfiel, D. J., and Loewy, R. On matrices having equal corresponding principal minors. (Apr. 1984).

Inference for DPixP

- **Input** : J realizations, $Y_1, \dots, Y_J$, from the same DPiXP with unknown C kernel.
- **Empirical estimator of the cardinality** $n = \frac{1}{J}(|Y_1| + \cdots + |Y_J|)$
- Let us consider the conditional distribution

$$p_C(x) = \begin{cases} \mathbb{P}(x \in X |\ 0 \in X) = C(0) - \dfrac{|C(x)|^2}{C(0)} & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases}$$

- Using **stationarity** an empirical estimator of p_C is

$$\theta_J(x) = \begin{cases} \dfrac{1}{nJ} \displaystyle\sum_{i=1}^{J} \sum_{y \in \Omega} 1_{Y_i}(y) 1_{Y_i}(y+x) & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases}.$$

Inference for DPixP

- **Input** : J realizations, $Y_1, \dots, Y_J$, from the same DPiXP with unknown C kernel.
- **Empirical estimator of the cardinality** $n = \frac{1}{J}(|Y_1| + \cdots + |Y_J|)$
- Let us consider the conditional distribution

$$p_C(x) = \begin{cases} \mathbb{P}(x \in X|\ 0 \in X) = C(0) - \dfrac{|C(x)|^2}{C(0)} & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases}$$

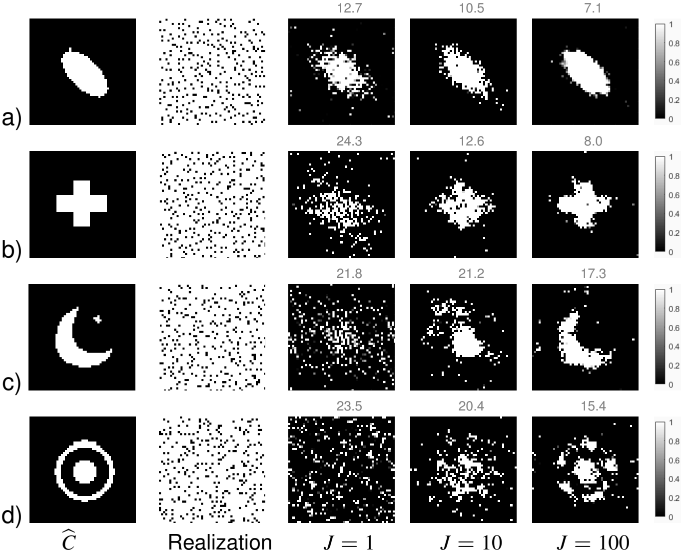
- Using **stationarity** an empirical estimator of p_C is

$$\theta_J(x) = \begin{cases} \dfrac{1}{nJ} \displaystyle\sum_{i=1}^{J} \sum_{y \in \Omega} 1_{Y_i}(y) 1_{Y_i}(y+x) & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases}.$$

- We propose to solve $\min_C \|p_C - \theta_J\|_2^2$ under the set of admissible kernels with expected cardinality n using projected gradient descent.
- Convex constraint but highly non convex functional, a careful initialization is important (heuristic).

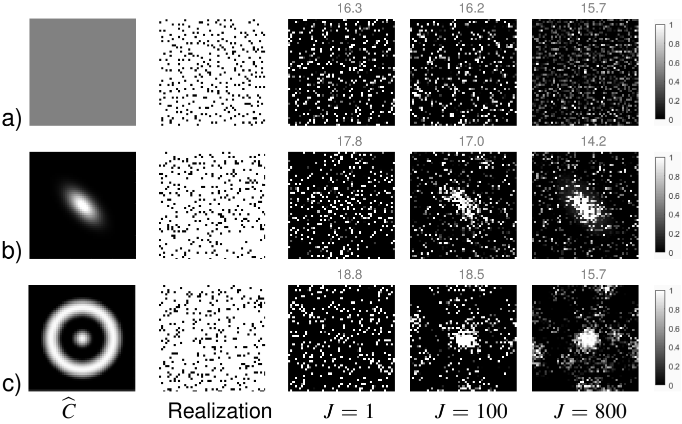
Inference for DPixP

Inference of the Fourier coefficients from 1, 10 and 100 realizations. (ℓ^2 distance)



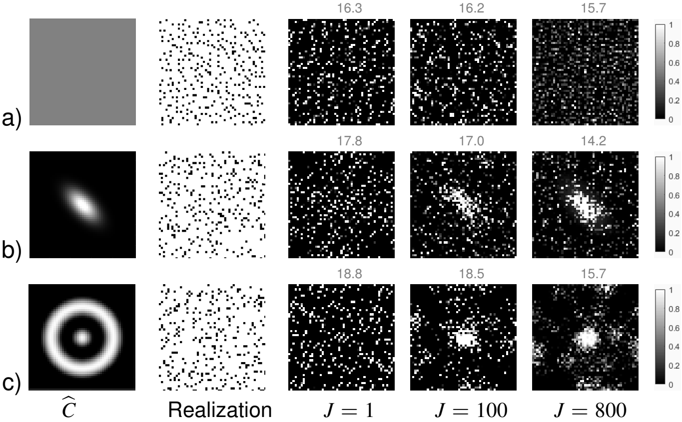
Inference for DPixP

Inference of the Fourier coefficients from 1, 10 and 100 realizations. (ℓ^2 distance)



Inference for DPixP

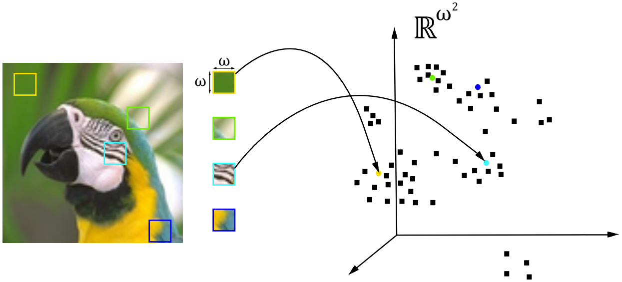
Inference of the Fourier coefficients from 1, 10 and 100 realizations. (ℓ^2 distance)



Conclusion : Satisfying results for projection DPixP, using a fast estimation algorithm.

Subsampling image patches using DPP

DPPs are widely used in statistics and in machine learning for selecting diverse subsets of points : k-means initialization, text summary (Kulesza-Taskar, Dupuy-Bach ..,), feature selections (Belhadji-Bardenet-Chainais), etc.



Patches of an image are seen as points in patch space [1].

Question : What is the best kernel K to subsample image patches ?

1. Houdard, A., Some advances in patch-based image denoising, Thèse de doctorat, 2018.

Discrete DPPs and L-ensembles

- Back to the general discrete setting with $\mathcal{Y} = \{1, \dots, N\}$ and a matrix K to determine $Y \sim \mathrm{DPP}(K)$.
- K is Hermitian and has its eigenvalues in the interval $[0, 1]$.
- If 1 is not an eigenvalue of K, one sets $L = K(I - K)^{-1}$ and one has the marginal probability

$$\forall A \subset \mathcal{Y}, \quad \mathbb{P}(Y = A) = \frac{\det(L_A)}{\det(I + L)}.$$

- Conversely, given any Hermitian matrix $L \succeq 0$ defines a DPP by setting $K = L(L + I)^{-1}$ the spectrum of which is within $[0, 1)$. This is called an L-ensemble.
- An L-ensemble kernel L is easier to manipulate for parametric modeling (e.g. rescale by multiplying by any constant etc.). K and L share the same eigenvectors.

Subsampling image patches using DPP

We define on the set of patches $\mathcal{P} = \{p_i, 1 \leqslant i \leqslant N\}$ an admissible matrix K or an L-ensemble kernel L to define $K = L(L+I)^{-1}$.
We consider several examples of kernels :

- Gaussian kernel based on the intensity of the patches :

$$L_{ij} = \exp\left(-\frac{\|p_i - p_j\|_2^2}{s^2}\right)$$

  The parameter s is fixed as the median of the distances of intensities between the patches.

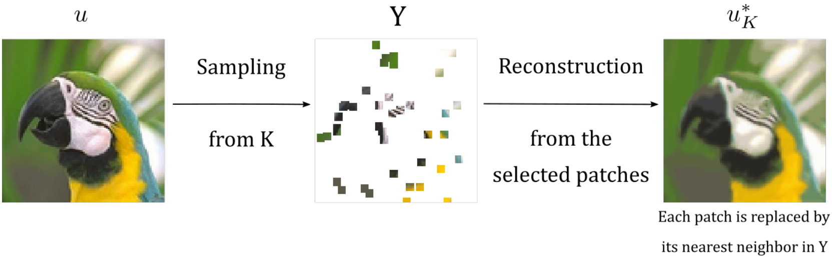
- Gaussian kernel based on the k first PCA components of patches :

$$L_{ij} = \exp\left(-\frac{\|PCA_i - PCA_j\|_2^2}{s^2}\right)$$

- Kernel based on a quality/diversity decomposition, where $q_i \in \mathbb{R}^+$, $\phi_i \in \mathbb{R}^D$, s.t. $\|\phi_i\|_2 = 1$, $L_{ij} = q_i \phi_i^T \phi_j q_j$
- Projection kernel K obtained in maximizing a reconstruction evaluation

$$\mathbb{E}\left(\sum_{p_i \in \mathcal{P}} \sum_{Q \in \mathcal{Q}} \mathbf{1}_{\|p_i - Q\|_2 \leqslant \alpha}\right), \text{ where } \mathcal{Q} \sim \mathrm{DPP}(K).$$

Subsampling image patches using DPP

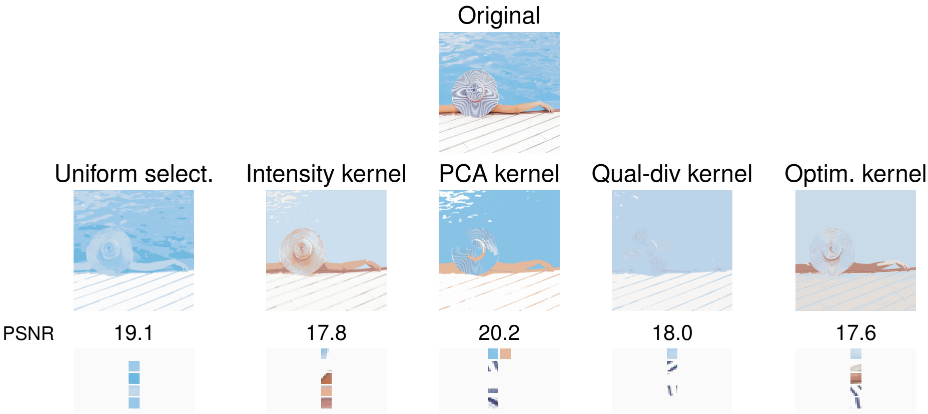


Reconstruction of an image from patches sampled by DPP :

Each patch in the image is replaced by its closest representative in the subset $Y \sim \mathrm{DPP}(K)$ (nearest neighbor for the ℓ_2-distance).

Comparison of the different kernels for patch subsampling

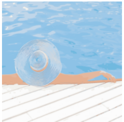
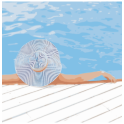
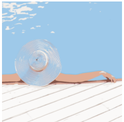
Expected cardinality of the DPP : 5 patches.
Each patch in the image is replaced by its closest representative in the subset $Y \sim \mathrm{DPP}(K)$ (nearest neighbor for the ℓ_2-distance).



Comparison of the different kernels for patch subsampling

Expected cardinality of the DPP : 25 patches.
Each patch in the image is replaced by its closest representative in the subset $Y \sim \mathrm{DPP}(K)$ (nearest neighbor for the ℓ_2-distance).

Original

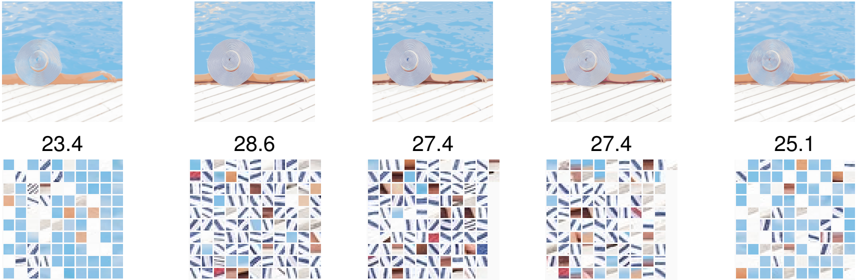
	Uniform select.	Intensity kernel	PCA kernel	Qual-div kernel	Optim. kernel
					
PSNR	21.3	24.3	24.4	22.6	22.5
					

Comparison of the different kernels for patch subsampling

Expected cardinality of the DPP : 100 patches.
Each patch in the image is replaced by its closest representative in the subset $Y \sim \mathrm{DPP}(K)$ (nearest neighbor for the ℓ_2-distance).

Original

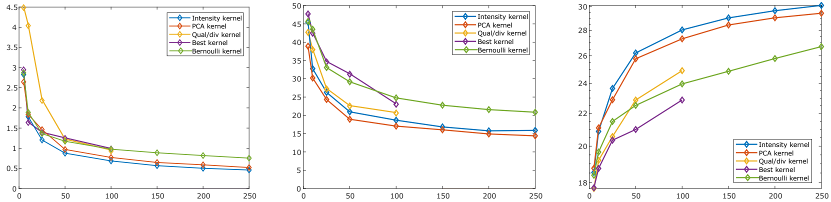
	Uniform select.	Intensity kernel	PCA kernel	Qual-div kernel	Optim. kernel
PSNR	23.4	28.6	27.4	27.4	25.1



Comparison of the different kernels for patch subsampling

Reconstruction errors for the previous image VS. expected cardinality

- $\{p_i, 1 \leqslant i \leqslant N\}$, patches of the image
- $\mathcal{Q} \sim \mathrm{DPP}(K)$, subset of patches sampled using the given DPP



(a) $E_1 = \frac{1}{N} \sum_{i=1}^{N} d(p_i, \mathcal{Q})^2$ (b) $E_2 = \max_{i \in \{1, \dots, N\}} d(p_i, \mathcal{Q})^2$ (c) PSNR

Conclusion :

- Uniform sampling lags always behind.
- Qual/div and optimized kernels are not competitive and limited in cardinal by construction.
- Intensity and PCA kernels are the best choice for every measurements.

Conclusion and perspectives

- (Fast) sampling algorithms for DPPs ?
- Many questions for texture modeling : from an image, estimate the spot function and the kernel of the DPP ?
- Selecting the $\ll$ best $\gg$ kernel for representing the patches of an image depending on the final task (compression, denoising, texture synthesis, etc.).
- Geometry of the shot noise driven by a DPP ?

Conclusion and perspectives

- (Fast) sampling algorithms for DPPs ?
- Many questions for texture modeling : from an image, estimate the spot function and the kernel of the DPP ?
- Selecting the $\ll$ best $\gg$ kernel for representing the patches of an image depending on the final task (compression, denoising, texture synthesis, etc.).
- Geometry of the shot noise driven by a DPP ?

MERCI !

References

- *Determinantal Point Processes for Image Processing*, C. Launay, B. Galerne, A. Desolneux, SIAM Journal on Imaging Sciences, 14(1), March 2021.
- *Exact Sampling of Determinantal Point Processes without Eigendecomposition*, C. Launay, B. Galerne, A. Desolneux, Journal of Applied Probability, vol. 57, no. 4, Décembre 2020.
- *Determinantal Patch Processes for Texture Synthesis*, C. Launay, A. Leclaire. Communication pour le GRETSI 2019.
- *Étude de la Répulsion des Processus Pixelliques Déterminantaux*, A. Desolneux, B. Galerne, C. Launay. Communication pour le GRETSI 2017.

- Papers and some associated codes are available online [2].

2. `https://claunay.github.io/`

Spectral sampling algorithm

Exact sampling algorithm using spectral decomposition of K (Hough-Krishnapur-Peres-Virág)

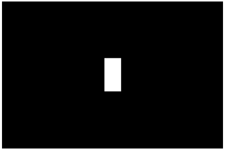
- Eigendecomposition (λ_j, v^j) of the matrix K.
- Select active frequencies : Sample a Bernoulli process $\mathbf{X} \in \{0,1\}^N$ with parameter $(\lambda_j)_j$.
  Denote n the number of active frequencies, $\{\mathbf{X} = 1\} = \{j_1, \dots, j_n\}$.
  and the matrix $V = \begin{pmatrix} v^{j_1} & v^{j_2} & \cdots & v^{j_n} \end{pmatrix} \in \mathbb{R}^{N \times n}$ with $V_k \in \mathbb{R}^n$ the k-th row of V, for $k \in \mathcal{Y}$.
- Output the sequence $Y = \{y_1, y_2, \dots, y_n\}$ sequentionaly sampled as follows :
  For $l = 1$ to n :
  - Draw a point $y_l \in \mathcal{Y}$ from the probability distribution

$$p_k^l = \frac{1}{n-l+1}\left(\|V_k\|^2 - \sum_{m=1}^{l-1} |\langle V_k, e_m\rangle|^2\right), \forall k \in \mathcal{Y}.$$

  - If $l < n$, define $e_l = \frac{w_l}{\|w_l\|} \in \mathbb{R}^n$ where $w_l = V_{y_l} - \sum_{m=1}^{l-1} \langle V_{y_l}, e_m\rangle e_m$.

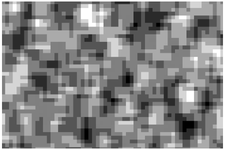
Shot noise driven by a DPixP : Limit theorems

- **Law of large numbers** and **central limit theorem** exist for shot noise based on DPixP.
- One needs to use increasing-domain asymptotics : Expand the DPP to $\mathbb{Z}^2$ and let the support of the kernel grow [3] : $S_M(y) = \frac{1}{M^2} \sum_{x \in X} g\left(y - \frac{x}{M}\right)$.

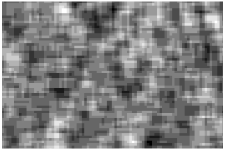


(a) *Spot*

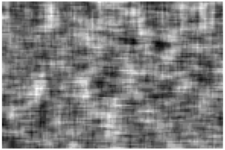
(b) S_M, $M = 1$



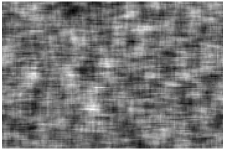
(c) S_M, $M = 2$



(d) S_M, $M = 3$



(e) S_M, $M = 6$



(f) $\mathcal{N}(0, \Sigma(C))$

3. Shirai, Takahashi, 2003. Soshnikov, 2002.

Shot noise driven by a DPixP : Limit theorems

For limit theorems, one needs to use increasing-domain asymptotics :
Expand the DPP to $\mathbb{Z}^2$ and let the support of the kernel grow [4].

Proposition

Let g be a continuous function on $\mathbb{R}^2$ with compact support, $X \sim \mathrm{DPixP}(C)$ and S_M the shot noise : $S_M(y) = \frac{1}{M^2} \sum_{x \in X} g\left(y - \frac{x}{M}\right), \forall y \in \mathbb{Z}^2$. Then,

$$S_M(0) = \frac{1}{M^2} \sum_{x \in X} g\left(-\frac{x}{M}\right) \xrightarrow[M\to\infty]{} C(0) \int_{\mathbb{R}^2} g(x)dx, \text{ a.s and in } L^1. \quad (1)$$

If g has zero mean, $\forall x_1, ..., x_m \in \mathbb{Z}^2$,

$$\sqrt{M^2}\left(S_M(x_1), \cdots, S_M(x_m)\right) \xrightarrow[M\to\infty]{\mathcal{L}} \mathcal{N}(0, \Sigma(C)) \quad (2)$$

with, for all $k, l \in \{1, \cdots, m\}$,

$$\Sigma(C)(k,l) = \left(C(0) - \|C\|_2^2\right) R_g(x_l - x_k).$$

where R_g is the autocorrelation of g.

4. Shirai, Takahashi, 2003. Soshnikov, 2002.

Inference for DPixP - Identifiability

Proposition

Let C_1, C_2 be two kernels defined on Ω, satisfying some *reasonable hypotheses*[1] with associated matrices K_1 and K_2 s.t. K_1 is irreducible. If $N \geqslant 4$, we suppose also that, for all partition of $\mathcal{Y}$ in two subsets α, β, $|\alpha| \geqslant 2, |\beta| \geqslant 2$, rank $(K_1)_{\alpha \times \beta} \geqslant 2$.
Then, $\mathrm{DPixP}(C_1) = \mathrm{DPixP}(C_2)$ if and only if the Fourier coefficients of C_2 are **translated and/or symmetric with respect to** $(0,0)$ from the Fourier coefficients of C_1 that is

$$\mathrm{DPixP}(C_1) = \mathrm{DPixP}(C_2) \Longleftrightarrow \exists\, \tau \in \Omega \text{ s.t. either } \forall \xi \in \Omega,\ \widehat{C}_2(\xi) = \widehat{C}_1(\xi - \tau)$$
$$\text{ou } \forall \xi \in \Omega,\ \widehat{C}_2(\xi) = \widehat{C}_1(-\xi - \tau).$$

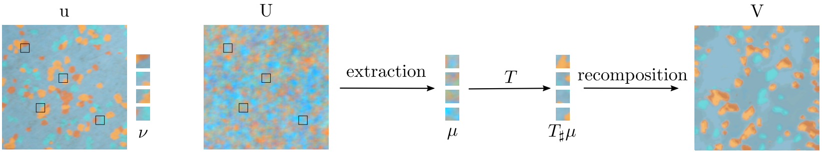
Two cases if K_1 do not satisfy the hypotheses :

- K_1 is irreducible but there exists a partition (α, β) s. t. the rank$(K_1)_{\alpha \times \beta} = 1$.
- K_1 is similar by permutation of a block diagonal matrix with similar blocks : This is a degenerate case e.g. with intermixed independent copies of the same DPP on a smaller grid.

Texture synthesis by example

Generate a texture image visually similar to an input texture image

- Strategy [5] :
  - Generate a Gaussian random field U with same mean and covariance as the input texture [6].
  - Define an optimal transport map T to correct the Gaussian patch distribution from the empirical patch distribution of the original texture.
  - Use T to correct the local features of the Gaussian image U.



5. Galerne, Leclaire, Rabin. A texture synthesis model based on semi-discrete optimal transport in patch space (2018).
6. Galerne., Gousseau, Morel, Random Phase Textures : Theory and Synthesis (2011).

Acceleration of a texture synthesis by example algorithm

- Synthesis time is highly dependent on the size of the patch distribution.
- Initial strategy : uniform selection of 1000 patches.
- **Contribution** [7] : Subsampling of the patch space using a DPP to better represent the patch set.

  Proposition : Select only 100 or 200 patches thanks to a DPP of kernel $K = L(L+I)^{-1}$ with

$$\forall i,j \in \{1,\dots,I\}, \quad L_{ij} = \exp\left(-\frac{\|p_i - p_j\|_2^2}{s^2}\right)$$

7. C. Launay, A. Leclaire., Determinantal Patch Processes for Texture Synthesis, In GRETSI 2019.

Acceleration of a texture synthesis by example algorithm

- Selection of a subset of patches with the DPP

$$\mathcal{Q} = \{q_j, 1 \leqslant j \leqslant J\} \sim \mathrm{DPP}(K).$$

- Estimation of the summarized patch distribution

$$\nu^* = \sum_{j=1}^{J} \nu_j^* \delta_{q_j}$$

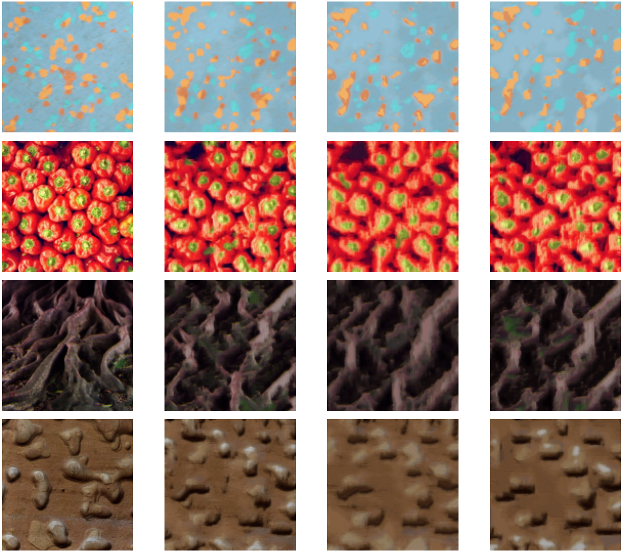
  with weights ν_j^* obtained by minimizing the Wasserstein distance between ν and the empirical distribution of all the patches.
- DPP simulation : Done only once during the estimation of the transport map T.

Acceleration : To synthesize an image of size 1024×1024 :

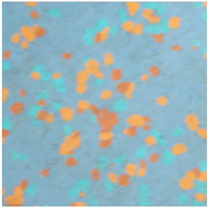
- Original algorithm : 1000 patches. Time : 1.7".
- Proposed DPP-based strategy :

Nb of patches	50	100	200
Time	0.19"	0.28"	0.47"

Acceleration of a texture synthesis by example algorithm

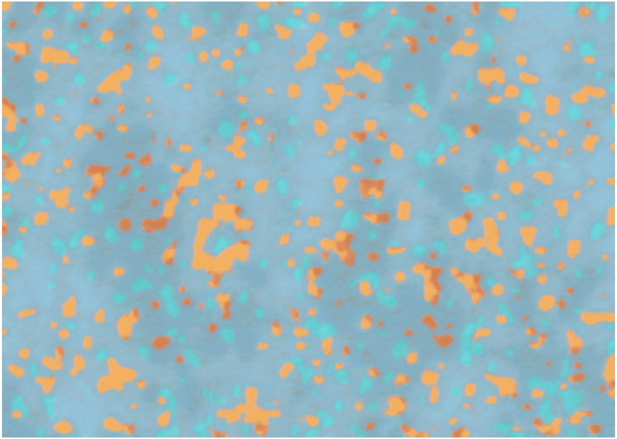


Comparaisons - 1000 patchs / 100 patchs sampled with DPP



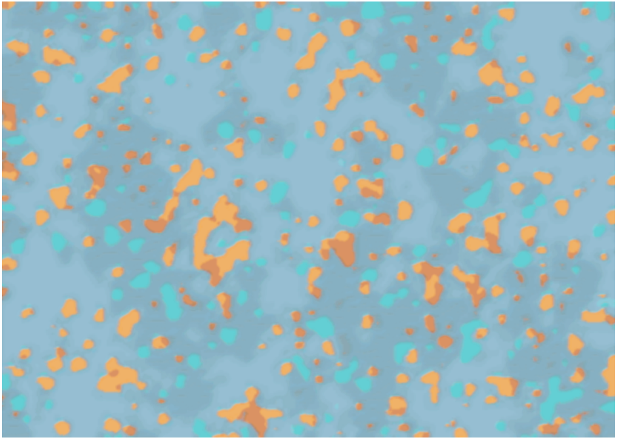
Original texture

Comparaisons - 1000 patchs / 100 patchs sampled with DPP



1000 patches sampled uniformly

Comparaisons - 1000 patchs / 100 patchs sampled with DPP



100 patches sampled with DPP

In general the visual quality is maintained, but one observe some detail loss for complex textures.