



Discrete determinantal point processes and their application to image processing

Claire Launay

► To cite this version:

Claire Launay. Discrete determinantal point processes and their application to image processing. Probability [math.PR]. Université Paris Cité, 2020. English. NNT : 2020UNIP7034 . tel-03189384

HAL Id: tel-03189384

<https://theses.hal.science/tel-03189384>

Submitted on 3 Apr 2021

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

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



Université de Paris

Laboratoire MAP5 (CNRS UMR 8145)

École doctorale 386 : Sciences Mathématiques de Paris Centre

THÈSE

présentée par

Claire Launay

pour obtenir le grade de

DOCTEURE D'UNIVERSITÉ DE PARIS

Spécialité : Mathématiques Appliquées

Processus ponctuels déterminantaux discrets et leur application au traitement des images

Soutenue le 22 juin 2020 devant un jury composé de

Pierre Chainais	Ecole Centrale de Lille	Rapporteur
Marianne Clausel	Université de Lorraine	Examinatrice
Agnès Desolneux	CNRS, ENS Paris Saclay	Directrice de thèse
Anne Estrade	Université de Paris	Présidente du jury
Bruno Galerne	Université d'Orléans	Directeur de thèse
Frédéric Lavancier	Université de Nantes	Rapporteur

RÉSUMÉ

Les processus ponctuels déterminantaux (Determinantal Point Processes ou DPP en anglais) sont des modèles probabilistes qui modélisent les corrélations négatives ou la répulsion à l'intérieur d'un ensemble d'éléments. Ils ont tendance à générer des sous-ensembles d'éléments diversifiés ou éloignés les uns des autres. Cette notion de similarité ou de proximité entre les points de l'ensemble est définie et conservée dans le noyau associé à chaque DPP. Cette thèse étudie ces modèles dans un cadre discret, définis dans un ensemble discret et fini d'éléments. Nous nous sommes intéressés à leur application à des questions de traitement d'images, lorsque l'ensemble de points de départ correspond aux pixels ou aux patchs d'une image. Les Chapitres 1 et 2 introduisent les processus ponctuels déterminantaux dans un cadre discret général, leurs propriétés principales et les algorithmes régulièrement utilisés pour les échantillonner, c'est-à-dire pour sélectionner un sous-ensemble de points distribué selon le DPP choisi. Dans ce cadre, le noyau d'un DPP est une matrice. L'algorithme le plus utilisé est un algorithme spectral qui repose sur le calcul des valeurs propres et des vecteurs propres du noyau du DPP. Dans le Chapitre 2, nous présentons un algorithme d'échantillonnage qui repose sur une procédure de *thinning* (ou amincissement) et sur une décomposition de Cholesky mais qui n'a pas besoin de la décomposition spectrale du noyau. Cet algorithme est exact et, sous certaines conditions, compétitif avec l'algorithme spectral. Le Chapitre 3 présente les DPP définis sur l'ensemble des pixels d'une image, appelés processus pixelliques déterminantaux (Determinantal Pixel Processes ou DPiXP en anglais). Ce nouveau cadre impose des hypothèses de périodicité et de stationnarité qui ont des conséquences sur le noyau du processus et sur les propriétés de répulsion générée par ce noyau. Nous étudions aussi ce modèle appliqué à la synthèse de textures gaussiennes, grâce à l'utilisation de modèles *shot noise*. Nous nous intéressons également à l'estimation du noyau de DPiXP à partir d'un ou plusieurs échantillons. Le Chapitre 4 explore les processus ponctuels déterminantaux définis sur l'ensemble des patchs d'une image, c'est-à-dire la famille des sous-images carrées d'une taille donnée dans une image. L'objectif est de sélectionner une proportion de ces patchs, suffisamment diversifiée pour être représentative de l'information contenue dans l'image. Une telle sélection peut permettre d'accélérer certains algorithmes de traitements d'images basés sur les patchs, voire d'améliorer la qualité d'algorithmes existants ayant besoin d'un sous-échantillonnage des patchs. Nous présentons une application de cette question à un algorithme de synthèse de textures.

Mots clés: Processus ponctuels déterminantaux, échantillonnage, pixels, modèles shot noise, inférence, textures, patchs.

ABSTRACT

Determinantal point processes (DPPs in short) are probabilistic models that capture negative correlations or repulsion within a set of elements. They tend to generate diverse or distant subsets of elements. This notion of similarity or proximity between elements is defined and stored in the kernel associated with each DPP. This thesis studies these models in a discrete framework, defined on a discrete and finite set of elements. We are interested in their application to image processing, when the initial set of points corresponds to the pixels or the patches of an image. Chapter 1 and 2 introduce determinantal point processes in a general discrete framework, their main properties and the algorithms usually used to sample them, i.e. used to select a subset of points distributed according to the chosen DPP. In this framework, the kernel of a DPP is a matrix. The main algorithm is a spectral algorithm based on the computation of the eigenvalues and the eigenvectors of the DPP kernel. In Chapter 2, we present a sampling algorithm based on a thinning procedure and a Cholesky decomposition but which does not require the spectral decomposition of the kernel. This algorithm is exact and, under certain conditions, competitive with the spectral algorithm. Chapter 3 studies DPPs defined over all the pixels of an image, called Determinantal Pixel Processes (DPixPs). This new framework imposes periodicity and stationarity assumptions that have consequences on the kernel of the process and on properties of the repulsion generated by this kernel. We study this model applied to Gaussian textures synthesis, using shot noise models. In this chapter, we are also interested in the estimation of the DPixP kernel from one or several samples. Chapter 4 explores DPPs defined on the set of patches of an image, that is the family of small square images contained in the image. The aim is to select a proportion of these patches, diverse enough to be representative of the information contained in the image. Such a selection can speed up certain patch-based image processing algorithms, or even improve the quality of existing algorithms that require patch subsampling. We present an application of this question to a texture synthesis algorithm.

Keywords: Determinantal point processes, sampling, pixels, shot noise models, inference, textures, patches.

Remerciements

Pendant ces derniers mois, j'ai régulièrement pensé à la façon dont je voulais remercier les gens qui m'ont accompagnée pendant ces quelques années. Souvent, je me souvenais d'anecdotes drôles, émouvantes ou marquantes, parfois je trouvais de jolies tournures ou une idée un peu originale... Et évidemment, au moment d'écrire mes remerciements, à la toute fin de ma thèse, je ne me souviens de rien. N'espérez donc rien de plus des lignes qui vont suivre qu'un grand merci un peu banal mais très sincère à mes collègues et à mes proches. Evidemment, je souhaite tout d'abord remercier profondément mes directeurs de thèse, Bruno Galerne et Agnès Desolneux. J'ai été chanceuse d'avoir pu travailler avec vous deux pendant ces quatre années. Un grand merci pour m'avoir accompagnée, pour avoir toujours su me rassurer et m'aiguiller et surtout pour être restés disponibles malgré les déménagements, les grèves et un confinement. Je n'avais jamais autant foi en la recherche et en notre travail qu'après nos rendez-vous. Agnès, ton expertise et ton recul m'ont sortie de l'impasse à de nombreuses reprises. Je te remercie particulièrement pour ta patience et ta capacité à me remotiver dans les moments de doute. Bruno, je garderai en souvenir ces heures passées face à ton tableau noir, les mains dans le cambouis, à se battre face aux DPPs. Ta passion et ton enthousiasme ont toujours su me redonner confiance.

Je tiens également à remercier très chaleureusement les membres de mon jury de thèse, Pierre Chainais, Marianne Clausel, Anne Estrade et Frédéric Lavancier. Tout particulièrement Pierre et Frédéric qui ont accepté de tenir le rôle, et la charge, de rapporteurs en ce printemps confiné. Vos questions et remarques pertinentes ont sans aucun doute amélioré mon manuscrit. Pierre, merci pour avoir suivi de si près mon parcours en tant que membre mon comité de suivi de thèse et surtout pour tes encouragements et ta bienveillance. Anne, tu as toi aussi fait partie de mon comité de suivi, merci encore pour ton aide en tant que nouvelle directrice du laboratoire et pour avoir rendu plus facile l'organisation de cette soutenance hybride.

Je l'ai souvent répété pendant ma thèse, j'ai eu le bonheur de passer mes trois années de doctorat et cette dernière année d'ATER au MAP5, à l'université Paris Descartes. L'ambiance y est toujours conviviale et accueillante, c'était un plaisir de venir y travailler au quotidien et j'ai le cœur gros de devoir quitter

ce laboratoire. Je souhaite sincèrement remercier Fabienne Comte, directrice du laboratoire jusqu'en début d'année 2020. Ton aide a été précieuse au début de ma thèse, je n'oublie pas que tu en as même été la directrice principale pendant quelques mois. Anne, tu as pris le relais de Fabienne avec un certain sens du timing, il faut l'avouer, ce même sens du rythme dont tu as fait preuve lors des 15 ans du MAP5. J'espère que le MAP&Muz Band a de beaux jours devant lui. Un grand merci également à Marie-Hélène, pour sa bonne humeur quotidienne et sa gestion experte du laboratoire, impeccablement secondée par Sandrine puis Julien. Je remercie encore Maureen, Christophe pour leur gentillesse et leur disponibilité, sans oublier Max, Arnaud, Azzedine et Isabelle pour leur aide. Ces quatre années de doctorat m'ont aussi permis de découvrir l'enseignement, dans lequel je me suis particulièrement épanouie. Cette expérience n'aurait pas été la même sans l'équipe pédagogique que j'ai côtoyée, merci à Annie, Florent, Marcela, Nathael et Georges. Marcela, merci encore pour tes nombreux encouragements. Enfin merci aux maîtres de conférence et professeurs qui font vivre le laboratoire. Certains d'entre vous ont été mes enseignants lorsque j'étais étudiante à Paris Descartes et c'était un plaisir de vous retrouver comme collègues. Je pense particulièrement à Julie Delon, George Koepfler et Lionel Moisan pour m'avoir donné envie de poursuivre dans cette voie et pour avoir joué un rôle précieux dans mon parcours, de la L2 à la thèse. Je tiens également à remercier l'équipe DPP de Lille qui m'a accueillie à plusieurs reprises et auprès de qui j'ai aussi beaucoup appris, en particulier Rémi Bardenet, Adrien Hardy, Mylène Mayda et Guillaume Gautier. Guillaume, merci pour tous tes retours toujours enrichissants, pour m'avoir fait visiter Lille et bien sûr pour ta boîte à outils DPPy. Arthur, toi c'est à Bordeaux que tu t'es installé et que tu m'as accueillie. Merci pour tes mails toujours rassurants et motivants (ils m'ont été très utiles à la fin de ma thèse) et pour nos discussions passionnantes et pour le travail que nous avons entamé ensemble.

Pendant ces quatre années au MAP5, j'y ai aussi rencontré des amis. Dans le bureau 725-C1, où j'ai mis les pieds en octobre 2016, j'ai eu l'impression d'arriver dans une équipe joyeuse et soudée. Noura et Alasdair vous terminiez vos contrats mais ces quelques mois ont suffi pour nous lier durablement. Merci à mes compagnons de route, Rémy (ta bienveillance, toujours à quelques portes), Anne-Sophie (ton rire et nos séances de sport me manquent), Antoine (surtout là pour un goûter presque mérité), Cambyse (et ton légendaire sens de la nuance lors de nos débats), Alexandre (et tes engagements parfois surprenants), Valentin (ces conférences avec toi étaient un plaisir), sans oublier Mario (et tes visites trop ponctuelles). Anton, Pierre-Louis et Rémi, vous avez su redonner un bel élan au bureau et je sais qu'avec vous, il est entre de bonnes mains (vertes). Pierre et Vincent, je vous dois énormément à tous les deux, et ma thèse également. Pierre, tu as toujours répondu présent, que ce soit pour

partager des pizza-burratas ou pour aller chercher des copies dans les orties. Vincent, merci pour ta patience et ta capacité à passer des heures à aider un copain en détresse : tous les doctorants du laboratoire te sont redevables. Et puis surtout, on a créé deux Burger Quiz ensemble, ce n'est pas rien ! Merci à tous les deux pour votre aide et vos conseils. Au 7e étage, ma route a également croisé celle d'autres thésards et jeunes docteurs : Alan (merci d'avoir pris ma place au conseil), Alessandro (j'attends tes conseils, et ta venue, à New York !), Alkéos (tant de débats passionnés en ta compagnie), Arthur (tu es notre roi à tous), Andrea et Christelle (nous n'avons pas fait suffisamment de karaoké ensemble), Fabien (ta gentillesse nous manque), Florian, Ismaël, Juliana (j'emporte avec moi ton bracelet), Julie, Léo, Marta, Matias, Maurizia, Ousmane, Safa, Sinda, Yen, Vivien et Warith (et ton enthousiasme à toute épreuve). Sans oublier les rapportés au grand cœur, Mélina, Jean-Marc, Anaïs et Newton, nos rendez-vous quasi-hebdomadaires me manqueront ! Je garderai un très bon souvenir de mes passages à Cachan-Paris Saclay, où j'ai côtoyé des doctorants passionnés et passionnants, Axel, Charles, Jérémy, Mariano, Marie, Pierre, Thibaud, Thibaud, Tina et Pashmina.

Et puis, tout au long de ma thèse, j'ai pu compter sur ma famille et mes amis toujours présents, même loin de Paris. Merci à tous, ceux qui pendant 4 ans, ont fait semblant de s'intéresser au sujet de ma thèse, jusqu'à essayer d'en apprendre le titre. Il y a d'abord les Parisiennes, enfin plus largement mes amies de BL. Après 10 ans, vous êtes toujours là, je suis fière de vous avoir pour amies. Chloé (ça va être long loin de toi), Pauline (c'est toi la prochaine !), Clélia, Alice, Xena, Juliette, Justine, Adèle, Hélène, Le Mao, Joséphine, Manon. Merci à toutes de m'avoir épaulée, supportée et réconfortée quand j'en avais besoin.

A mes amies d'HIDA, Appoline (le rythme des 3 semaines va devenir difficile à tenir), Marine (pour de nombreuses siestes avec toi), Anaïs, Héloïse, Clémence et Julie et leur rapporté.e, je suis si contente que nous soyons restées si proches malgré la distance. Des loups garous à la naissance d'Adèle, nous en avons parcouru du chemin ensemble. Merci Appo et Marine pour les presque relectures. C'est vous toutes qui avez fait qui je suis. Alice, Emeline, Julie et Sophie, j'ai hâte de voir où nos routes respectives nous mèneront. J'espère être présente pour célébrer chaque étape. Clément et Héloïse, un grand merci à vous pour tous vos encouragements. Notre précieuse amitié continue son chemin depuis l'enfance.

Une grande pensée à l'équipe des Gnolois, toujours là pour m'encourager et pour fêter ce qui peut l'être. Un merci tout particulier à Clem, Tom et Fantine pour toutes ces soirées de tarot où je n'ai pas pris et pour avoir supporté nos voix en chœur et en boucle ces derniers mois. Merci aussi aux copains rapportés de prépa, Romain, Pierre, Manon, Chloé, Matthias, Erwan, Matthias, Chloé et nos discussions politiques passionnantes qui me redonnent

foi en l'avenir.

Un grand merci à toute la famille Guilleux, qui s'agrandit d'année en année, pour m'avoir accueillie et pour avoir fait de Nantes un troisième foyer. Jacques et Claudine, Valentin et Camille, Simon, Amélia, Andréa et Ezra, j'espère que vous viendrez nous rendre visite très bientôt, d'un côté ou de l'autre de l'Atlantique. Sans oublier Zola et Ficelle, je sais qu'ils me soutiennent.

Enfin, je remercie ma famille (Launay-Gaudichet), qui m'a toujours encouragée quel que soit mon projet. Avec un rare enthousiasme, pendant 4 ans, vous avez été curieux du monde de la recherche en maths et m'avez posé des questions sur mon travail. Je ne remercierai jamais assez mes parents, Bernadette et Jean-Jacques, et ma sœur, Lucile, pour leur soutien inconditionnel et pour tous ces beaux moments partagés, à Angers ou en vacances, et pour tous ceux à venir. J'en profite pour embrasser Léo, Paul et Julien. Je suis une tata comblée et fière de notre famille.

Et puis, Alexis, tu sais déjà tout, et à quel point je te dois beaucoup. Jusqu'au bout de ma thèse, tu m'as portée et soutenue. La vie avec toi est douce et drôle et j'ai hâte de continuer notre aventure sur un autre continent.

Contents

Notations	11
1 Introduction	15
1.1 Discrete Point Processes	16
1.2 Determinantal Point Processes (DPPs)	21
1.3 Applications to Image Processing	26
1.4 Detailed Outline of the Manuscript	29
1.5 Contributions	35
2 Sampling Discrete DPPs	37
2.1 Introduction	37
2.2 Usual Sampling Method and Related Works	39
2.2.1 Spectral Algorithm	39
2.2.2 Other Sampling Strategies	41
2.3 Sequential Sampling Algorithm	44
2.3.1 Explicit General Marginal of a DPP	44
2.3.2 Sequential Sampling Algorithm of a DPP	46
2.4 Sequential Thinning Algorithm	47
2.4.1 General Framework of Sequential Thinning	47
2.4.2 Sequential Thinning Algorithm for DPPs	49
2.4.3 Computational Complexity	52
2.5 Experiments	53
2.5.1 DPP Models for Runtime Tests	53
2.5.2 Runtimes	54
2.6 Conclusion	60
3 Determinantal Point Processes on Pixels	63
3.1 Introduction	63
3.2 Determinantal Pixel Processes (DPixPs)	64
3.2.1 Notations and Definitions	65
3.2.2 Properties	67
3.2.3 Hard-core Repulsion	71
3.3 Shot Noise Models Based on DPixPs	73

3.3.1	Shot Noise Models and Micro-textures	73
3.3.2	Extreme Cases of Variance	76
3.3.3	Convergence to Gaussian Processes	78
3.4	Inference for DPixPs	82
3.4.1	Equivalence Classes of DPP and DPixP	83
3.4.2	Estimating a DPixP Kernel from One Realization	89
3.4.3	Estimating a DPixP Kernel From Several Realizations	93
3.5	Conclusion	97
4	Determinantal Point Processes on Patches	99
4.1	Introduction	99
4.2	Determinantal Patch Processes	101
4.2.1	DPP Kernels to Sample in the Space of Image Patches	101
4.2.2	Minimizing the Selection Error	104
4.2.3	Experiments	107
4.3	Application to a Method of Texture Synthesis	111
4.3.1	Texture Synthesis with Semi-Discrete Optimal Transport	112
4.3.2	DPP Subsampling of the Target Distribution	114
4.3.3	Results	118
4.4	Conclusion	122
5	Conclusion and Perspectives	127
5.1	Exact Determinantal Point Processes Sampling	127
5.2	Determinantal Pixel Processes	129
5.3	Determinantal Point Processes on Patches	131
A	Explicit General Marginal of a DPP	135
A.1	Möbius Inversion Formula	135
A.2	Cholesky Decomposition Update	136
A.2.1	Add a Line	136
A.2.2	Add a Bloc	136
B	Convergence of Shot Noise Models Based on DPixP	139
B.1	Ergodic Theory	139
B.2	Proof of Proposition 3.3.4 - Law of Large Numbers	141
B.3	Proof of Proposition 3.3.4 - Central Limit Theorem	144
C	Identifiability of a DPixP	151
C.1	Remark 3.4.1, Case 2	151
C.2	Remark 3.4.1, Case 3: K_1 is not irreducible	153

Notations

- \mathcal{Y} is the underlying space on which is defined the point processes.
- Y and X denote given point processes.
- ρ is the intensity of a point process. It is a function defined on \mathcal{Y} and if $x \in \mathcal{Y}$, $\rho(x) = \mathbb{P}(x \in Y)$. If the point process is homogeneous, ρ is a constant.
- $|\cdot|$ defined on the set of subsets of \mathcal{Y} is the cardinality of the subset: it counts the number of elements contained in the subset. $|\cdot|$ applied to a point of \mathcal{Y} or to a vector denotes its modulus.
- $\mathcal{M}_N(\mathbb{C})$ is the set of matrices of size $N \times N$, with complex coefficients.
- \overline{M} is the complex conjugate matrix of the matrix M .
- M^* is the conjugate transpose of the matrix M , $M^* = \overline{M}^t$.
- Similarly, \bar{v} is the complex conjugate vector of the vector v and v^* is the conjugate transpose of v .
- $M_{A \times B}$ denote for all subset A and B of \mathcal{Y} the matrix $(M(x, y))_{(x, y) \in A \times B}$ and $M_A = M_{A \times A}$.
- A^c is the complement of A in \mathcal{Y} if A is a subset of \mathcal{Y} .
- I^A is the matrix whose diagonal coefficients indexed by the elements of A are equal to 1 and whose other coefficients are zero.
- $\det(M)$ is the determinant of the square matrix M .
- $\text{Tr}(M)$ is the trace of the matrix M , that is the sum of its diagonal elements.
- $\text{rank}(M)$ is the rank of the matrix M .
- λ_{\max} is the maximum eigenvalue of a given matrix.

- $M \succeq 0$ means that the eigenvalues of M are bounded below by zero. On the contrary, $M \preceq I$ means that they are bounded above by one.
- K denotes for the (marginal) kernel of determinantal point processes, it is a positive semidefinite Hermitian matrix, whose eigenvalues are bounded above by one.
- L denotes a positive semi-definite matrix that can define a L -ensemble.
- $\langle ., . \rangle$ is the canonical scalar product on a Euclidean space, $\|.\|$ is the associated norm.
- $v_{1:k}$ denotes the vector (v_1, \dots, v_k) , for a given $k > 0$. In particular, $0_{1:k}$ is the null vector of size k .
- Ω is the image domain: a 2-dimensional discrete grid. If Ω is of size $N_1 \times N_2$, then we consider $\Omega = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\} \subset \mathbb{Z}^2$. Note that the functions defined on Ω can be extended to \mathbb{Z}^2 by periodicity.
- $u : \Omega \rightarrow \mathbb{R}^d$ is the image defined on Ω with d color channels.
- $\tau_y u$ is the translation of the image u by the vector y .
- $\widehat{\Omega}$ is the Fourier domain associated to Ω . For instance, if N_1 and N_2 are even, $\widehat{\Omega} = \{-\frac{N_1}{2}, \dots, \frac{N_1}{2} - 1\} \times \{-\frac{N_2}{2}, \dots, \frac{N_2}{2} - 1\}$.
- Ω^* denotes $\Omega \setminus \{0\}$, the image domain minus the origin.
- $\widehat{f} = \mathcal{F}(f)$ is the discrete Fourier transform of the function $f : \Omega \rightarrow \mathbb{C}$. $\mathcal{F}^{-1}(\widehat{f})$ is the inverse Fourier transform of \widehat{f} .
- f_- , given a function $f : \Omega \rightarrow \mathbb{C}$, is the function defined for $x \in \Omega$ by $f_-(x) = f(-x)$.
- $f * g$ denotes the convolution operation of the function f and g .
- $R_g : \Omega \rightarrow \mathbb{C}$ denotes the autocorrelation of the function g .
- S is the shot noise random field based on a point process X and a spot function g , both defined on Ω .
- $\text{Ber}(p)$ is a Bernoulli variable with parameter p .
- $\mathcal{N}(m, \Sigma)$ is the Gaussian distribution with mean m and covariance matrix Σ .
- \mathbb{T}^2 is the torus of dimension two.

- $\ell^2(\mathbb{Z}^2)$ is the set of functions f defined on \mathbb{Z}^2 such that

$$\|f\|_2^2 = \sum_{x \in \mathbb{Z}^2} \|f(x)\|^2 < \infty.$$

- $L^2(\mathbb{T}^2)$ is the set of functions f defined on \mathbb{T}^2 such that

$$\|f\|_2^2 = \int_{x \in \mathbb{T}^2} |f(x)|^2 dx < \infty.$$

- $\mathcal{D}_N \subset \mathcal{M}_N(\mathbb{C})$ is the set of diagonal matrices of size $N \times N$ such that its coefficients are of modulus one.
- $\widehat{\mathcal{C}}_n$ is the set of function \widehat{C} defined on $\widehat{\Omega}$ whose inverse Fourier transform is an admissible DPixP kernel function, that is

$$\{\widehat{C} \in \mathbb{R}^N \text{ such that } \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi) = n \text{ and } \forall \xi \in \widehat{\Omega}, 0 \leq \widehat{C}(\xi) \leq 1\}.$$

- $proj$ denotes the algorithm that projects a function defined on $\widehat{\Omega}$ on the set $\widehat{\mathcal{C}}_n$.
- $\mathcal{P} = \{P_i, i = 1, \dots, N\}$, the set of patches of size $(2\rho + 1) \times (2\rho + 1) \times d$ of the image u , given a $\rho \in \mathbb{N}$.
- \mathbf{P} is the matrix gathering all the patches of the image, of size $N \times d(2\rho + 1)^2$, with $N = N_1 \times N_2$.
- \bar{u} , given an image u , is the mean image $\frac{1}{|\Omega|} \sum u(x)$ and t_u is the normalized version of the image u : $t_u = \frac{1}{\sqrt{|\Omega|}}(u - \bar{u})\mathbf{1}_\Omega$.
- Ω^ℓ , given $\ell = 0, \dots, L - 1$ is the coarser image domain $\Omega \cap 2^\ell \mathbb{Z}^2$ and u^ℓ is the subsampled version of u on Ω^ℓ .
- $W_2^2(\mu, \nu)$ is the L2-Wasserstein distance between the probability distributions μ and ν such that

$$W_2^2(\mu, \nu) = \inf_{(\pi_{i,j})} \sum_{i,j} \pi_{i,j} \|y_i - x_j\|^2.$$

Chapter 1

Introduction

Contents

1.1	Discrete Point Processes	16
1.2	Determinantal Point Processes (DPPs)	21
1.3	Applications to Image Processing	26
1.4	Detailed Outline of the Manuscript	29
1.5	Contributions	35

In this thesis, we are interested in the study of specific random point processes, called determinantal point processes (DPPs in short). They allow to model the repulsive nature of certain sets of points. These point processes capture negative correlations in the sense that the more similar two points are, the less likely they are sampled simultaneously: they tend to generate sets of points that are diverse or distant from each other. The purpose of this work was to apply DPPs to image processing. We have chosen two axes to realize this study: a definition on the set of pixels and a definition on the set of patches of an image. First, point processes defined on pixels are often used in image processing, for instance in order to synthesize textures, using shot noise models based on Poisson point processes [130, 48]. Due to their repulsive nature, DPPs provide an attractive alternative for these applications. We are hoping that, compared to a Poisson shot noise model, a shot noise model based on a DPP would be less affected by the averaging of the spot function. Second, this repulsive nature and their easy adaptability make them a useful tool to subsample sets of data, such as the patches of an image. Given the huge dimension of images, this set is very large and such selection is regularly needed in patch-based algorithms. In general, these strategies use a uniform random selection, which is easy to implement and fast, but DPPs offer the

opportunity to improve this selection and thus to improve the patch-based algorithm.

1.1 Discrete Point Processes

Some of the first studies of spatial statistics and random point processes were done to answer physics and astronomy questions, as for instance, in 1860, to know the probability that a certain number of stars lies in a given square [59], assuming that the stars are randomly and uniformly distributed in the sky. Since then, random point processes have emerged as powerful tools for modeling natural phenomena, such as monitoring a population [104], plant locations [47], or neural spiking activity [127]. Figure 1.1 displays the locations of 126 pine trees in a forest [10]. Because they need to share light and nutrients, trees often tend to be spaced from each other in a forest and thus to be modeled by repulsive point processes.

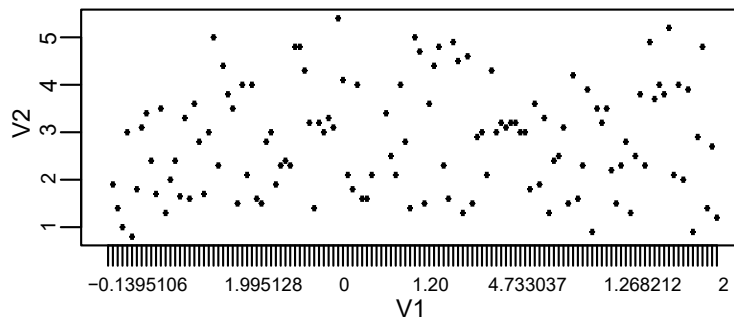


Figure 1.1: Locations of trees in a forest. These data come from the R library called “spatstat”.

A second use of point processes has recently gained influence as the number and size of data to be handle and analyze has increased: random subset selections. In that case, the aim of the point process is not to represent an existing phenomenon anymore but to randomly choose a small proportion of elements in an initial set. Applications are numerous, such as documents summarization [69] or recommendation systems [53]. These random selections are often able to provide results while the optimal selection is intractable, and they tend to produce different subsets at each trial. Furthermore, if the set of data to handle is huge, evaluating a function on it may be impossible. The solution can be to subsample the set of data using a point process to compute statistics on a large population [126, 93] or to estimate the empirical distribution of a large set of data [52]. On the other hand, if the dimension of the data is too high, a solution can be random features selection [17] or in another domain, stochastic sampling [31]. Indeed, in computer graphics, if a scene needs to

be subsampled, a random selection of points will provide perceptually better results and avoid aliasing compared to a subsampling on a regular grid.

At last, random selection using point processes has the major advantage of being flexible and easily adaptable depending on the data to handle and the desired selection, as a wide variety of models can be used.

Random point processes

Given a space \mathcal{Y} , a point process is a probability measure defined on the set of all subsets of \mathcal{Y} . It can be seen as a random countable subset $Y \subset \mathcal{Y}$, whose elements are called points. Its size, that is the number of points it contains, is called its cardinality and it is itself random. In this thesis, we will consider discrete point processes, meaning that the space \mathcal{Y} on which is defined the point process is discrete and finite (except in the subsection 3.3.3, where we will study a point process defined on \mathbb{Z}^2). When considering these general settings, the space \mathcal{Y} will be called the state space, the dataset or the ground set. Assuming it contains N elements, it will be denoted by $\mathcal{Y} = \{1, \dots, N\}$, identifying its elements with their index.

Such point processes can be characterized by their marginal probabilities of inclusion $\mathbb{P}(A \subset Y)$, which are the inclusion probabilities of any subset $A \subset \mathcal{Y}$. In the general continuous case, for instance when A contains n points, this quantity is called the n -th order product density function or the n -correlation function [85]. These probabilities give the correlations between the points of the state space.

These marginal probabilities of inclusion also provide various statistics to describe the point process. The intensity function gives the probability for the occurrence of any point of \mathcal{Y} . It is defined for all $x \in \mathcal{Y}$ by $\rho(x) = \mathbb{P}(x \in Y)$. If the intensity is constant, the point process is called homogeneous or first order stationary. A second statistic describes the interactions between pairs of points, it is called the pair correlation function. It is often denoted by g and it is defined, for all $x, y \in \mathcal{Y}$, by

$$g(x, y) = \frac{\mathbb{P}(\{x, y\} \subset Y)}{\mathbb{P}(x \in Y)\mathbb{P}(y \in Y)}. \quad (1.1)$$

This quantity is often used to describe local behaviours of attraction or repulsion. A point process is said simple if all the points of the process are almost surely distinct, meaning that an element of \mathcal{Y} has a zero probability to be selected twice in a realization. In that case, one can associate the subset Y with the vector of size N with ones in the places of the elements of Y and zeros elsewhere. As we consider point processes as random subsets $Y \subset \mathcal{Y}$, all point processes are implicitly simple in this thesis.

Different classes of point processes

As we have seen, the chosen model must be adapted to the dataset: the characteristics of the data, the natural phenomenon they can be related to and the goal of the analysis. We propose here to briefly and non-exhaustively review several classes of common point processes.

Bernoulli Point Processes

The discrete counterpart of a Poisson point process is called a Bernoulli point process. As Poisson point processes, Bernoulli point processes correspond to models without any interaction or of “complete spatial randomness” [101]. Indeed, given $\rho : \mathcal{Y} \rightarrow [0, 1]$ an intensity function, the elements of the set \mathcal{Y} are selected independently, each element $x \in \mathcal{Y}$ with probability $\rho(x)$. For Y a Bernoulli point process, with intensity ρ , we have

$$\forall x \in \mathcal{Y}, \mathbb{P}(x \in Y) = \rho(x) \text{ and } \forall A \subset \mathcal{Y}, \mathbb{P}(A \subset Y) = \prod_{x \in A} \rho(x). \quad (1.2)$$

The simulation of Bernoulli point processes is easy to implement and very fast, thus they are convenient to model different sorts of phenomena. Yet, some data may present dependence, for instance attraction or repulsion, or anisotropic structures, properties that Bernoulli point processes can not capture. Different models, more adapted to the variability of the situations, are needed.

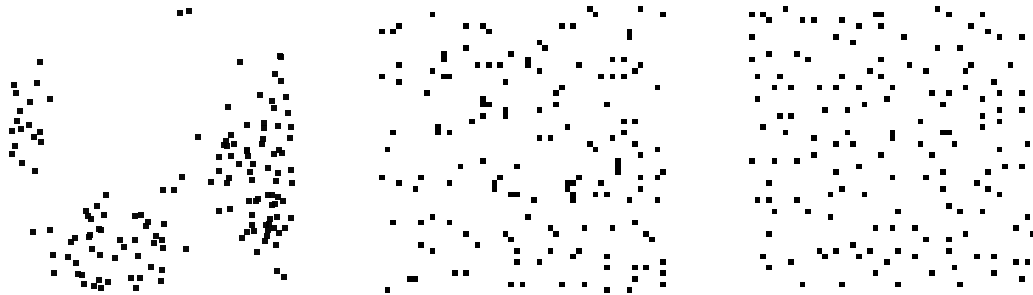


Figure 1.2: Realizations sampled from a clustering Cox process (left), from a Bernoulli point process (center) and from a Determinantal point process (right), with each 148 points.

As mentioned above, spatial dependency is often described using the pair correlation function. This statistic is used to characterize the attractive or repulsive nature of point processes. Notice that it is constant and equal to 1 for Poisson and Bernoulli point processes. Point processes with a pair correlation function above 1 are considered to be attractive point processes, while point processes with a pair correlation function below 1 are considered to be repulsive. Note that this notion of repulsion is sometimes associated with the

notion of regularity, that can be seen as a satisfying covering of the state space. Poisson and Bernoulli point processes stand for the pivot line between point processes generating regular and irregular realizations [59]. Figure 1.2 presents realizations of three point processes. From left to right, it shows a realization of a clustering Cox process which belongs to the class of attractive point processes, a realization of a Bernoulli point process, which model data with no interaction, and a realization of a determinantal point process, which belongs to the class of repulsive point processes and which is the object of this thesis.

Attractive point processes

According to Diggle [59], attractive point processes or models of points aggregation were first studied to describe the locations of insect larvae after hatching from eggs clusters by Neyman in 1939 [104]. The most studied class of attractive point processes is the class of continuous point processes named Cox point processes [101]. These processes generalize Poisson point processes, they are also called doubly stochastic Poisson point process. Given Λ a random locally finite measure on \mathcal{Y} , the point process X is said to be a Cox process if conditionally to Λ , it is distributed as a Poisson point process on \mathcal{Y} of intensity Λ . The realization to the left of Figure 1.2 is a sample of a specific case of Cox processes, called a Thomas point process [101], with parameters $\kappa = 7$, $\sigma = 0.09$ and $\alpha = 21$. It was generated using the R package named “spatstat” [10]. Another specific case of Cox processes is the class of permanental point processes [100, 44], which are the attractive dual form of determinantal point processes.

An interesting property of most attractive point processes, such as Cox processes, is the overdispersion of the counting random variable, which is counting the number of points of the point process in a given area. That means that the local number of points has a high variance. On the contrary, repulsive point processes tend to select points that are evenly distributed through space.

Repulsive point processes

Figure 1.2 illustrates an ambivalence: while one could expect that uniformity and independence would be the good conditions to cover a space, Poisson and Bernoulli point processes tend to generate realizations with clusters and large gaps in some regions. On the contrary, repulsive point processes, favoring negative correlations, tend to create sets of points well scattered in space. Furthermore, they are flexible: by choosing the repulsive model and defining the marginal probabilities of the point process, it is possible to adapt to the space structure and to the desired covering. Thus, for many point processes applications, one needs to use repulsive point processes.

Gibbs point processes are a classic category of repulsive point processes [34, 101, 38]. (Note that it is possible also to define attractive Gibbs point

processes.) Given U an energy function, a Gibbs point process Y is defined by the marginal probabilities

$$\mathbb{P}(Y = A) \propto \exp(-U(A)), \quad A \subset \mathcal{Y}. \quad (1.3)$$

The energy function is often supposed to be such that

$$\exp(-U(A)) = \prod_{B \subseteq A, |B| \leq k} \psi_{|B|}(B), \quad (1.4)$$

where the functions ψ are called potential functions and k is a small constant [34]. In the case where the energy functions can be decomposed into potential functions depending only on adjacent points, the point process is called a Markov point process [35].

The main advantages of Gibbs point processes are their easy interpretability and their flexibility as they are defined directly using the correlations between the points. Thus, they can easily adapt to the nature of the dataset and to the goal of the study. However, their normalization constant is often intractable, along with most of their describing statistics, and there is, in general, no exact algorithm to sample a Gibbs point process.

Matérn point processes [98] are another repulsive class of point processes, generated by the thinning of a Poisson point process. The sampling strategy is done to ensure that all points are spaced at least a given distance apart. The Matérn III process, also known as Poisson disk sampling, is particularly used by stochastic sampling strategies [31], to improve the rendering of pictures and avoid an aliasing effect, perceptually unpleasant. The method called random sequential adsorption [46] generates points samples using the same model to ensure a minimal distance between the points. Similarly, given any shape, for instance a circle or a rectangle, it consists in sequentially and randomly placing this shape on the space, keeping the current one only if it does not overlap with the shapes already selected.

These methods are popular in the computer graphics community, as they allow to randomly copy a given shape, with the certainty that these shapes won't overlap. Such a property, called “hard-core” repulsion will be investigated in Section 3.2.3 using determinantal point processes defined on the pixels of an image. While this property has major advantages, these two point processes classes lack theoretical definitions and computational guarantees.

Finally, determinantal point processes belong to the group of repulsive point processes. Unlike most of the classes we have described, these point processes have tractable densities and statistics, and exact sampling strategies.

1.2 Determinantal Point Processes (DPPs)

Determinantal point processes model the repulsion present in certain sets of points, which can be found in real-world situations: the position of trees in a forest [85] or the position of apples on a branch, for example. In contrast to Bernoulli point processes, DPPs tend to avoid the “bunching” phenomenon and as shown in Figure 1.2, the points generated by a DPP are more evenly distributed in space than those generated by the Bernoulli point process.

They naturally arose in random matrix theory [65] and they were analysed for the first time in 1975 by Macchi [96] to model fermions, a particle in quantum mechanics which exhibit natural repulsion. Ever since the work of Kulesza and Taskar [81], these processes have become more and more popular in machine learning, because of their ability to draw subsamples that account for the inner diversity of data sets and the theoretical computations this model allows. This repulsive nature has been used in many fields, such as summarizing documents [41], improving a stochastic gradient descent by drawing diverse subsamples at each step [133], extracting a meaningful subset of a large data set to estimate a cost function or some parameters [126, 12, 5], or to compute a Monte Carlo estimator to approximate integrals [11, 58].

Definition

In this manuscript, we will use the following notations. The initial discrete dataset, on which is defined the point process, is denoted by $\mathcal{Y} = \{1, \dots, N\}$. The cardinality, or the size, of a set A is denoted by $|A|$. When M is a $N \times N$ matrix, with real or complex entries, the complex conjugate matrix of M is denoted by \overline{M} . The conjugate transpose of the matrix M is denoted by $M^* = \overline{M}^t$ and the conjugate transpose of the vector v is denoted by v^* . We denote by $M_{A \times B}$, for all subsets $A, B \subset \mathcal{Y}$, the matrix $(M(x, y))_{(x, y) \in A \times B}$ and we use the short notation $M_A = M_{A \times A}$. When focusing on a specific couple of points, for instance $x, y \in \mathcal{Y}$, we sometimes identify $M(x, y)$ and M_{xy} for clarity purpose. If A and B are subsets of \mathcal{Y} such that $|A| = |B|$, the determinant $\det(M_{A \times B})$ is called a minor of M and in case $B = A$, $\det(M_A)$ is called a principal minor of M .

In this general, discrete and finite setting, the kernel function associated with a DPP is a matrix K that will be called its kernel, or kernel matrix. This kernel can be also called the marginal or the correlation kernel. We assume that K is a positive semidefinite Hermitian matrix, of size $N \times N$ indexed by the elements of \mathcal{Y} . A random subset $Y \subset \mathcal{Y}$ is called a determinantal point process with kernel K if,

$$\forall A \subset \mathcal{Y}, \quad \mathbb{P}(A \subset Y) = \det(K_A). \quad (1.5)$$

We will denote $X \sim \text{DPP}(K)$.

A $N \times N$ matrix K defines a determinantal point process on \mathcal{Y} if and only if

$$0 \preceq K \preceq I, \quad (1.6)$$

meaning that its eigenvalues are in $[0, 1]$. For a detailed presentation of discrete DPPs, their properties and some applications to machine learning, we recommend the article of Kulesza and Taskar [81].

The diagonal coefficients of K define the marginal probabilities of any singleton:

$$\forall x \in \mathcal{Y}, \quad \mathbb{P}(x \in Y) = K(x, x), \quad (1.7)$$

and the off-diagonal coefficients of K give the similarity between points. Notice that the repulsion property becomes clear when observing the marginal probability of couples of points. The more similar two points are, the less likely they are to belong to the DPP simultaneously:

$$\forall \{x, y\} \subset \mathcal{Y}, \quad \mathbb{P}(\{x, y\} \subset Y) = K(x, x)K(y, y) - |K(x, y)|^2. \quad (1.8)$$

If K is seen as a similarity matrix, then the point process tends to generate diverse sets of points. Similarly, this negative correlation is observable for any set of points since, according to Hadamard's inequality, we have for all $n \geq 2$, for all $\{i_1, \dots, i_n\} \subset \mathcal{Y}$,

$$\mathbb{P}(\{i_1, \dots, i_n\} \subset Y) \leq \mathbb{P}(i_1 \in Y) \mathbb{P}(i_2 \in Y) \dots \mathbb{P}(i_n \in Y). \quad (1.9)$$

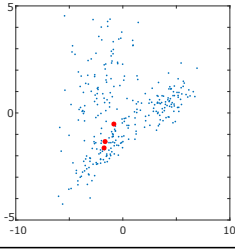
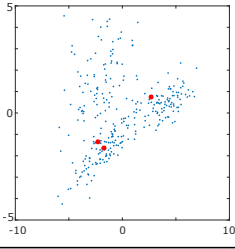
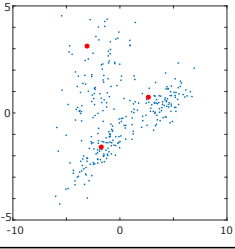
Let us take a simple example to highlight this property. We choose a set \mathcal{Y} of 300 points included in $[-10, 10] \times [-5, 5]$ and each point $i \in \mathcal{Y}$ is associated with its position p_i in \mathbb{R}^2 . We define a determinantal point process with kernel K depending on the distance between the points. Here, we take $K = I - (I + L)^{-1}$ with for all $i, j \in \mathcal{Y}$ by $L(i, j) = e^{-\|p_i - p_j\|_2^2}$: the closer two points $i, j \in \mathcal{Y}$ are, the higher the associated element $L_{\{i, j\}}$ is. This construction uses what is called an L -ensemble, that we present below. Note that the eigenvalues of such a kernel K are included in $[0, 1]$.

Table 1.1 shows that when the similarity given by K depends on the distance between points, subsets of points distant from each other have a significantly higher probability of occurrence.

Sampling

The sampling, also called the simulation, of a point process defined on \mathcal{Y} is the generation of a subset A of elements of \mathcal{Y} , distributed as the considered point process. The result of a sampling, the subset A , is called a realization, a selection or simply a sample. It is one of the major operations needed to use a point process however, and despite the fact that DPPs have been studied since the 1970s, the question of sampling DPPs seems still unsettled.

Table 1.1: DPPs tend to generate subsets of points far from one another.

Triplet $\{i, j, k\}$	$\{1, 2, 3\}$	$\{1, 50, 200\}$	$\{50, 100, 200\}$
Position			
$K_{\{i,j,k\}}$	$10^{-2} \times \begin{pmatrix} 7.4 & -0.4 & 6.5 \\ -0.4 & 15 & -0.5 \\ 6.5 & -0.5 & 10 \end{pmatrix}$	$10^{-2} \times \begin{pmatrix} 7.4 & 0.0 & 1.5 \\ 0.0 & 8.1 & -0.0 \\ 1.5 & -0.0 & 8.9 \end{pmatrix}$	$10^{-2} \times \begin{pmatrix} 8.1 & -0.1 & 0.0 \\ -0.1 & 33 & -0.0 \\ 0.0 & -0.0 & 8.9 \end{pmatrix}$
$\mathbb{P}(\{i, j, k\} \subset Y)$	4.8×10^{-4}	5.1×10^{-4}	23.7×10^{-4}

The main sampling algorithm is called the spectral algorithm. It was developed in 2008 by Hough et al. [72]. It has the significant advantage of being exact, meaning that it generates a sample which is distributed as the given DPP in a finite number of iterations. This spectral algorithm relies on the computation of the eigenvalues and the eigenvectors of the DPP's kernel matrix. When the state space \mathcal{Y} is large, the matrix is large too, and this computation is costly. Thus, one main drawback of DPP is that, in a general context, they take a long time to be exactly sampled.

Some authors have tried to adapt and speed up this algorithm by making assumptions on the kernel of the DPP such as a bounded rank [53], a decomposition into more tractable kernels [41] or the association of specific DPPs to uniform spanning trees [110].

On the other hand, some authors, such as Affandi et al. [2], Anari et al. or [6], have chosen to apply approximate methods to sample DPPs. Approximate strategies, such as Markov chain Monte Carlo methods, hope that after a certain number of simpler sampling iterations, the result is sufficiently close to the target distribution. The problem is twofold. First, one needs to decide when to stop the algorithm, and what does “sufficiently close” mean. This desired state is often called the equilibrium. Second, this equilibrium may need a high number of iterations to be (almost) reached.

Thus, it is important to develop an exact algorithm to sample DPPs in a general setting. In Chapter 2, we present two exact algorithms to sample general DPPs, which do not need the eigendecomposition of the kernel. While the first one, called the sequential algorithm, is very slow, the second, that we call the sequential thinning algorithm, provides competitive results with respect to the spectral algorithm.

Properties

Consider Y a determinantal point process with kernel K , defined on \mathcal{Y} . Denote the eigenvalues of K by $\{\lambda_1, \dots, \lambda_N\}$.

Cardinality

The cardinality $|Y|$ of the DPP is distributed as the sum of N independent Bernoulli random variables: $|Y| \sim \sum_{x \in \mathcal{Y}} \text{Ber}(\lambda_x)$, where the Bernoulli variables take the value 1 with probability λ_x . Different proofs of this proposition can be found in the papers [72] or [81]. One can easily note that

$$\mathbb{E}(|Y|) = \sum_{x \in \mathcal{Y}} \lambda_x = \text{Tr}(K) \text{ and } \text{Var}(|Y|) = \sum_{x \in \mathcal{Y}} \lambda_x(1 - \lambda_x). \quad (1.10)$$

The easy access to the expectation and the variance of the cardinality of any DPP is very useful when one needs to apply DPPs and to control the number of points to be sampled, or simply when one needs to compare several DPP kernels.

DPP defined from another DPP

The restriction of the DPP Y to a subset $A \subset \mathcal{Y}$, denoted by $Y \cap A$, is a DPP with kernel K_A . Thus, for all $B \subset A$,

$$\mathbb{P}(B \subset A \cap Y) = \det(K_B). \quad (1.11)$$

Furthermore, surprisingly, the complement of a DPP also favors repulsion. Consider $Y^c = \mathcal{Y} \setminus Y$, the complement of Y in \mathcal{Y} . This random subset is also a DPP, associated with the kernel $K^c = I - K$, where I is the identity matrix of size $N \times N$. Hence,

$$P(A \subset Y^c) = \mathbb{P}(A \cap Y = \emptyset) = \det((I - K)_A). \quad (1.12)$$

L-ensembles

We consider L a Hermitian matrix of size $N \times N$ such that

$$L \succeq 0, \quad (1.13)$$

then the random set $Y \subset \mathcal{Y}$ defined by

$$\forall A \subset \mathcal{Y}, \quad \mathbb{P}(Y = A) = \frac{\det(L_A)}{\det(I + L)} \quad (1.14)$$

is a determinantal point process with likelihood kernel L . We will denote $Y \sim \text{DPP}_L(L)$. This class of DPP is called L -ensembles and was developed

by Borodin and Rains [23]. To this point onward, the notation L denotes the kernel of an L -ensemble, which is positive semi-definite, while K denotes the correlation kernel of a general DPP, such that its eigenvalues are in $[0, 1]$.

Note that the matrices K and L define the same DPP if

$$K = L(L + I)^{-1} = I - (I + L)^{-1} \text{ and conversely } L = K(I - K)^{-1}. \quad (1.15)$$

In particular, if the spectral decomposition of K is $K = \sum_{n=1}^N \lambda_n v_n v_n^*$, then

$$L = \sum_{n=1}^N \frac{\lambda_n}{1 - \lambda_n} v_n v_n^*. \quad (1.16)$$

Nevertheless, if $\det(I - K) = 0$, or equivalently if any eigenvalue of the kernel K is equal to 1, the DPP can't be defined as an L -ensemble.

The definition of a DPP as an L -ensemble is convenient in practice, since, given a subselection problem, one only has to ensure that the likelihood kernel L is positive semidefinite. That is why this definition is often used in machine learning applications. Note that, contrary to specific DPPs called projection DPPs that we present right below, the cardinality of an L -ensemble cannot be fixed, it is random.

An interested reader should also be introduced to a related class of point processes called k-DPPs. A k-DPP is defined by conditioning a given DPP to generate samples with exactly k elements. This enables to preserve the repulsiveness of DPPs while ensuring that the samples have a fixed cardinality. This property can be very useful for some applications where the size of the realizations is crucial. However, in general, these k-DPPs don't share most of the appealing properties of DPPs, such as characterization through a marginal kernel, easy computation of marginal probabilities or explicit formulation of their moments. This is why we do not explore k-DPPs further in the remainder of this work.

Examples of determinantal point processes

Let us present specific cases of determinantal point processes that we will encounter several times in this manuscript. Suppose again that the set on which the point processes are defined is $\mathcal{Y} = \{1, \dots, N\}$. The first example is the (inhomogeneous) Bernoulli point process, which, as already introduced, corresponds to the case where the elements are selected independently from one another. This point process is also a particular case of DPP, associated with a diagonal kernel matrix K . Indeed, in that case,

$$P(A \subset Y) = \prod_{x \in A} K(x, x) = \prod_{x \in A} \mathbb{P}(x \in Y). \quad (1.17)$$

This is the least repulsive DPP, as there is no repulsion between the points.

A second common class of DPP is that of projection DPPs. They are characterized by a kernel matrix K with eigenvalues equal only to 0 or 1. Equivalently, denoting the eigenvalues of K by $\{\lambda_1, \dots, \lambda_N\}$, we have

$$\forall i \in \{1, \dots, N\}, \lambda_i(1 - \lambda_i) = 0. \quad (1.18)$$

Note that the cardinality of the point process is then fixed, equal to the rank of K as

$$\mathbb{E}(|X|) = \sum_{i=1}^N \lambda_i = \text{rank}(K) \text{ and } \text{Var}(|X|) = \sum_{i=1}^N \lambda_i(1 - \lambda_i) = 0. \quad (1.19)$$

These DPPs have two main advantages. The first one is the fixed cardinality of the generated samples. Their second advantage, depending of the number of non-zero eigenvalues, is that they may be associated with a low-rank matrix, which allows the use of faster sampling strategies, either exact [72] or approximate [56].

1.3 Applications to Image Processing

Point processes are often used in image processing, such as texture synthesis methods, for instance with shot noise models. These models, usually based on a Poisson process, generate textures [130, 48]. DPPs may provide an interesting alternative for these applications. This first question led us to adapt the determinantal point processes to the space of the pixels of an image: they become processes defined on a 2-dimensional grid, the image domain, discrete and under assumptions of stationarity and periodicity. Second, we were interested in the adaptation of the subsampling ability of DPPs to the set of patches of an image, which is as large as the size of the image itself, and often too large to be handled.

In this manuscript, on several instances, we will apply DPPs to methods of texture synthesis.

Texture synthesis

There is no formal and mathematical definition of texture images. A general definition was given by Wei in 2009 [131], considering textures as “images with repeated patterns”, allowing “a certain amount of randomness”. They can be roughly divided into two categories [48]. First, macro-textures can be seen as images made of repeated discernible objects. Second, micro-textures are texture images without geometric details or identifiable objects.

In computer graphics, the realistic rendering of a synthesized image highly depends on the textures covering the objects in the image. Depending on the

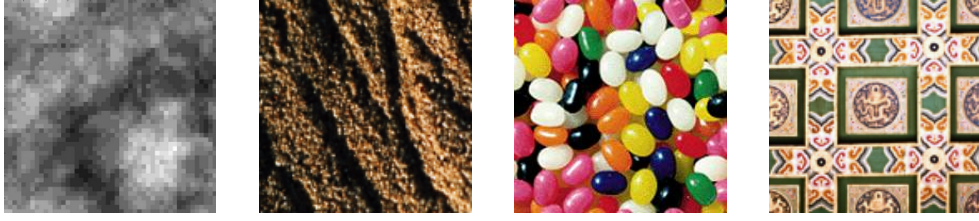


Figure 1.3: Examples of textures. It is difficult to formally characterize texture images as this term encompasses a wide variety of images, such as textures without identifiable elements, that can represent the surface of an object, or textures with repeated patterns and geometrical structures.

applications (video games, virtual reality, special effects in movies), it is crucial to develop algorithms for texture synthesis that generate efficiently potentially large images, with high perceptual quality. Discrete shot noise models are probabilistic models that consist in summing a given spot function translated around the points of a point process. Let us suppose the shot noise S is defined on an image domain Ω and it is driven by a spot function $g : \Omega \rightarrow \mathbb{R}$ and the point process X , containing n points. Then, it is defined by

$$\forall x \in \Omega, \quad S(x) = \sum_{x_i \in X} g(x - x_i). \quad (1.20)$$

In the case where $X = (X_i)_{1 \leq i \leq n}$ is a sequence of i.i.d. random points, the limit of this model when n tends to infinity is called the Asymptotic Discrete Spot Noise (ADSN) [48] and it is a Gaussian random vector whose covariance depends on the spot function. These models generate Gaussian textures visually related to the shape of the spot function, they are easy and fast to simulate. In Chapter 3, we study shot noise models based on a determinantal point process defined on the image domain.

Exemplar-based algorithms consist in synthesizing, from a given texture image, a texture visually equivalent to the initial one. For a review of the main exemplar-based texture synthesis algorithms, see the survey made by Raad et al. [112]. Two strategies are generally adopted: statistics-based methods [48, 68, 135, 108] and patch-based methods [43, 42, 89]. The first class methods rely on the extraction of statistics from the exemplar texture and, using a noisy image as initialization, they optimize a certain functional to enforce these statistics on the output. They are known to provide satisfying micro-texture synthesis. However, in general, these algorithms have trouble to generate more structured textures. On the contrary, the patch-based methods mainly consist in copy-paste strategies, meaning that they randomly re-arrange information, pixels or patches, already contained in the exemplar image, to generate the

output texture. In general, these methods are able to synthesize more complex textures than the previous class but they do not introduce innovative content and risk to create entire regions identical to the original texture. Moreover, they may be unstable and suffer from what is called “growing garbage”, meaning that the algorithm gets stuck and incoherently reproduces the same parts of the input texture.

These last few years, belonging to the first category, methods using neural networks statistics have emerged [54, 94, 18]. The method developed by Gatys et al. in 2015 [54] still provides state-of-the-art results, but it is computationally very costly, with a huge number of parameters to handle. Several algorithms [128, 74] tried to improve or speed up the synthesis but the perceptual quality of the result is impacted.

Let us mention also synthesis methods combining both previous classes, developing a model on the input texture but generating better synthesis from complex and structured textures than the statistics-based methods [111, 52]. Chapter 4 presents an attempt to accelerate and improve the method introduced by Galerne et al. in [52], using DPPs defined on the patches of the exemplar texture.

DPPs in computer vision and image processing

Several works have already tried to apply DPPs to computer vision and imaging issues. In that case, each point of the process is an image and the purpose of sampling from these DPPs is to generate a diverse subsample of images. Indeed, the amount of image and video contents available is overwhelming. To be handled, to be processed, it needs to be sorted and summarized. That is the purpose of recommendation systems. Some methods using DPPs have been developed to cope with this issue and to enforce diverse subsets, for images selection [79, 1, 27] or video recommendation [132]. Moreover, images and videos are now in very high resolution, but remain intrinsically redundant. The strategies for video summarization intend to extract meaningful and representative frames using sequential DPPs. This is a type of DPP taking into account the temporal dependencies of video frames [66, 97]. Besides, Chen et al. [28] prove that DPPs can be an appropriate tool to reduce the dimensionality of hyperspectral images, to select representative pixels from these images and be able to process such large-scale data.

Except this last paper dealing with hyperspectral images, these previous works applying DPPs to images define the DPP on a very large set of images, for instance a video to summarize or a corpus of pictures or videos. In Chapters 3 and 4 of this manuscript, we are given a single image and we define DPPs on the set of pixels or on the set of patches of this image.

1.4 Detailed Outline of the Manuscript

This section presents a detailed outline of the thesis. It describes the main contributions of this manuscript and the results obtained in the different chapters.

Chapter 2

Chapter 2 focuses on the methods used to sample a discrete determinantal point process. As we have seen, sampling a point process generates a subset of points, that can be used to reduce the size of an initial set of points, to illustrate the properties of a model or to synthesize an image for instance. Regardless to the purpose of the sample, the sampling algorithm must produce samples as close as possible to the target distribution and remain efficient, even when the size of the dataset grows. Concerning DPPs, the choice of the sampling strategy is crucial as it requires manipulating a kernel matrix K , which for most applications is very large. In Section 2.1, we present basic sampling strategies, starting with the classically used algorithm to sample general DPPs, the spectral algorithm. This algorithm relies on the fact that a general DPP can be considered as a mixture of projection DPPs, specific DPPs such that the eigenvalues of their kernel are either equal to 0 or to 1. The method is exact and it requires the computation of the eigenvalues and the eigenvectors of K [72]. As soon as the underlying space, on which the point process is defined, is large, this method is slow. We also present different algorithms, developed to sample DPPs more efficiently. In Section 2.2, we introduce a sampling strategy that does not use the eigendecomposition of the matrix K but a Cholesky decomposition, that we call the sequential algorithm. However, this algorithm involves computations to be done sequentially on each point of the initial space. Hence, it is very slow. Figure 1.4 illustrates how much slower the sequential algorithm is than the spectral algorithm.

To cope with this problem, we introduce in Section 2.3 a novel algorithm, called the sequential thinning algorithm. As a first step, it samples a dominating point process that contains the target DPP and in a second step, it applies the sequential algorithm on this reduced space. This strategy is called the thinning of a point process. If the maximum eigenvalue of K , λ_{\max} , is strictly smaller than 1, we obtain a bound on the cardinality of the dominating process, which is proportional to the cardinality of the target DPP. As the sequential sampling step is done on the subset given by the dominating process, this bound ensures that the overall running time is limited. This also highlights that the algorithm may have efficiency issues if λ_{\max} is equal to 1. Section 2.4 provides numerical experiments that illustrate the behavior of these three algorithms. In particular, they present competitive results for the sequential thinning algorithm with respect to the initial spectral algorithm.

Note that, contrary to the sequential algorithm, the running time of the

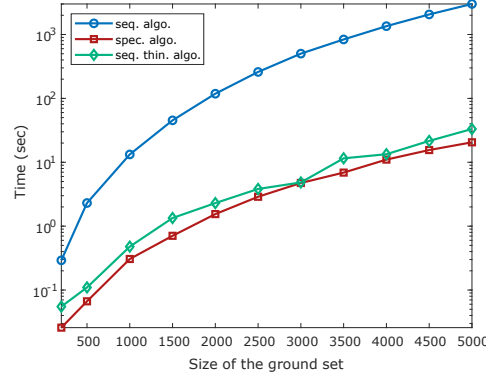


Figure 1.4: Running times of the 3 studied algorithms in function of the size of the ground set, using a patch-based kernel. The sequential algorithm is much slower than the two other sampling strategies.

sequential thinning algorithm is closer to that of the spectral algorithm (Figure 1.4). Moreover, Figure 1.5 compares the running times of these two algorithms in different situations, using a DPP kernel defined on the patches of an image. The spectral algorithm is more efficient when the expected size of the sample grows with the size of the dataset (left). Yet, when the dataset is large and the expected size of the sample is limited, one can observe that the sequential thinning algorithm seems to compete with the spectral algorithm. More illustrations are given in Section 2.4 to understand how the sequential thinning algorithm operates.

Chapter 3

In Chapter 3, we consider DPPs defined on a specific space, the set of the pixels of an image. Section 3.1 introduces these discrete DPPs that we call Determinantal Pixel Processes (DPiPs). In such a configuration, it is natural to assume that the point processes under study are stationary and periodic. The correlation between pairs of pixels no longer depends on the position of the pixels but on the difference between their position. As a consequence, the kernel K is a block-circulant matrix. Thus, the kernel can be characterized using a function C defined on the image domain, that we identify with the kernel of the DPiP in the following, so that $K(x, y) = C(x - y)$. Block-circulant matrices have the particularity to be diagonalized by the Fourier basis. Here, the eigenvalues of the matrix K are the Fourier coefficients of the function C . Thus, the 2D discrete Fourier transform plays a key role in this chapter. We study the consequences of the stationary and periodic hypotheses on basic properties of DPPs, in particular on the repulsion generated by these point processes. Whereas Gibbs point processes can generate hard-core repulsion,

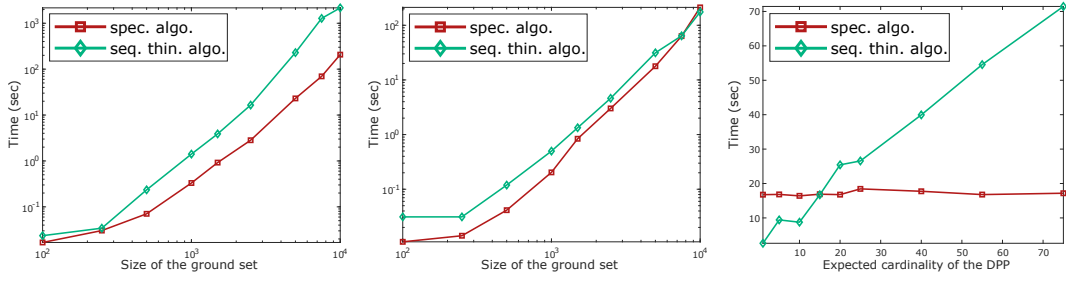


Figure 1.5: Running times in log-scale of the spectral and the sequential thinning algorithms as a function of the size of the ground set $|\mathcal{Y}|$ (two graphs on the left) or of the expected size of the sample $\mathbb{E}(Y)$ (right-hand graph), using a patch-based kernel. On the left, the expectation of the number of sampled points is set to 4% of $|\mathcal{Y}|$. In the middle, $\mathbb{E}(|Y|)$ is constant, equal to 20. On the right, the ground set $|\mathcal{Y}|$ is constant and contains 5000 points, while $\mathbb{E}(|Y|)$ grows.

that is imposing a minimal distance between the points of the point process, it is impossible to define DPixP with such a property. We prove that the only possible “hard-core” repulsion is directional, meaning that it is possible to define a DPixP kernel such that two points of the process can not be aligned along a given direction.

In Section 3.2, we investigate shot noise models based on DPixPs and on a given spot function. Consider X a DPP with intensity ρ defined on the image domain Ω and g a (deterministic) function, also defined on Ω . The shot noise random field S based on the points X and the spot g is defined by

$$\forall x \in \Omega, S(x) = \sum_{x_i \in X} g(x - x_i). \quad (1.21)$$

It appears that it is possible to adapt the kernel of a DPixP to the spot function g , in order to obtain particularly regular or irregular textures. This is related to an optimization problem based on the variance of the shot noise model. We are able to obtain the results presented in Figure 1.6. Whatever the spot function, the DPixP generating the least regular texture is the Bernoulli point process (Figure 1.6,b.). Given the spot g (Figure 1.6(a)), the DPixP generating the most regular texture is a projection DPixP (Figure 1.6(c)) whose Fourier coefficients are the solution of a combinatorial problem. An approximation of these Fourier coefficients is given ((d),(e) in Figure 1.6) using a greedy algorithm. Notice that the shot noise based on a Bernoulli point process produces many overlaps of the rectangle shape and regions without any rectangle, unlike the shot noise based on the projection DPixP.

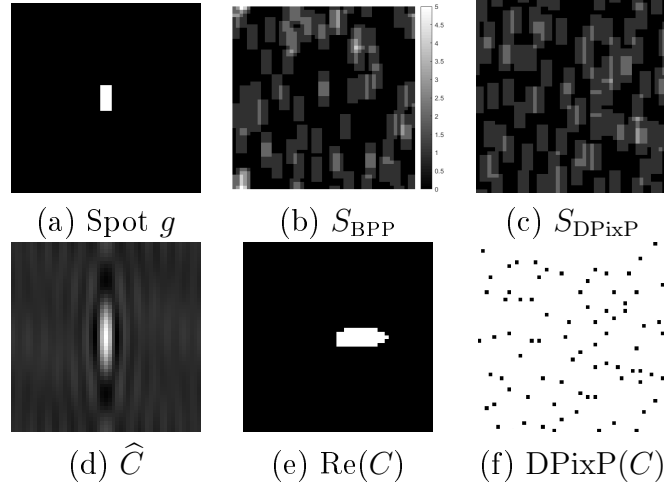


Figure 1.6: Realizations of the shot noise model based on a rectangle spot function and on a Bernoulli point process (b) or on projection DPiXP adapted to the spot (c). Both point processes have the same expected sample's size ($n = 80$).

We also prove that, in an appropriate framework, shot noise models based on any DPiXP and any spot function verify a Law of Large Number and a Central Limit Theorem characterizing their convergence to a Gaussian process (Figure 1.7).

Finally, in Section 3.3, to investigate inference on DPiXP kernels, we review the definition of equivalence classes of DPPs in different frameworks, this is a question called identifiability. Then, we develop an algorithm that uses the stationarity hypothesis to estimate the kernel of a DPiXP from one or several samples. This method is fast and provides satisfying results when the initial kernel is a projection kernel, a class of DPP kernels commonly considered as the most repulsive ones. Figure 1.8 illustrates these results obtained when we try to retrieve the Fourier coefficients of a complex projection DPiXP. Observe that while one realization is not sufficient to find the shape of the high Fourier coefficients, 10 realizations provide a satisfying approximation of the initial kernel.

Chapter 4

Chapter 4 examines DPPs defined on the patch space of an image. In Section 4.1, we study the choice of different kernels to subsample the set of patches of a given image. This can be useful to speed up or to improve a patch-based algorithm, by considering only the most significant patches in the image. Usually, if necessary, a uniform selection is performed to subsample the set of

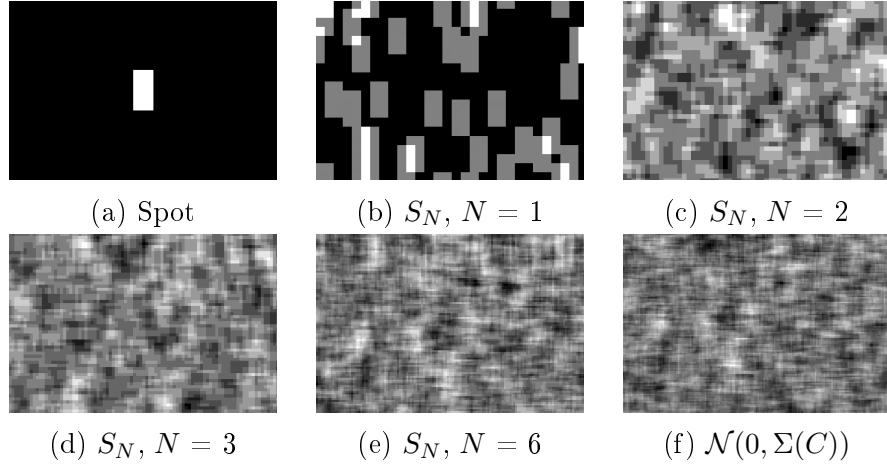


Figure 1.7: Determinantal shot noise realizations S_N as defined in Theorem 3.3.4 with various $N = 1, 2, 3, 6$ and a comparison with their associated limit Gaussian random field $\mathcal{N}(0, \Sigma(C))$ (f).

patches. However, this strategy may select points close to each other and miss some regions of the space. When considering patches, this amounts to select similar patches while possibly missing crucial regions of the image. In Section 4.1, we study five different types of DPP kernels, computed from the patches of the image. Numerical experiments show that these kernels behave very differently and that it is rather simple to adapt the kernel in function of the application that will be done with the selected patches.

Figure 1.9 presents an example of “image summarization” and shows several reconstructions of an image (a) from patches selected using different DPP kernels. Each reconstruction is done using the patches presented below such that each patch of the original image is replaced by the most similar patch in the selection. Thus, for each kernel, the original image is represented by a small number of patches and a vector connecting each patch to its nearest

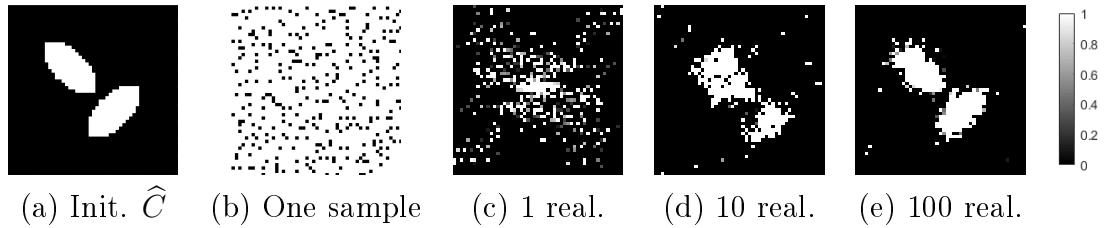


Figure 1.8: From left to right: the initial Fourier coefficients of the kernel, one realization of the associated DPixP, the estimation of the Fourier coefficients from one, from 10 and from 100 realizations.

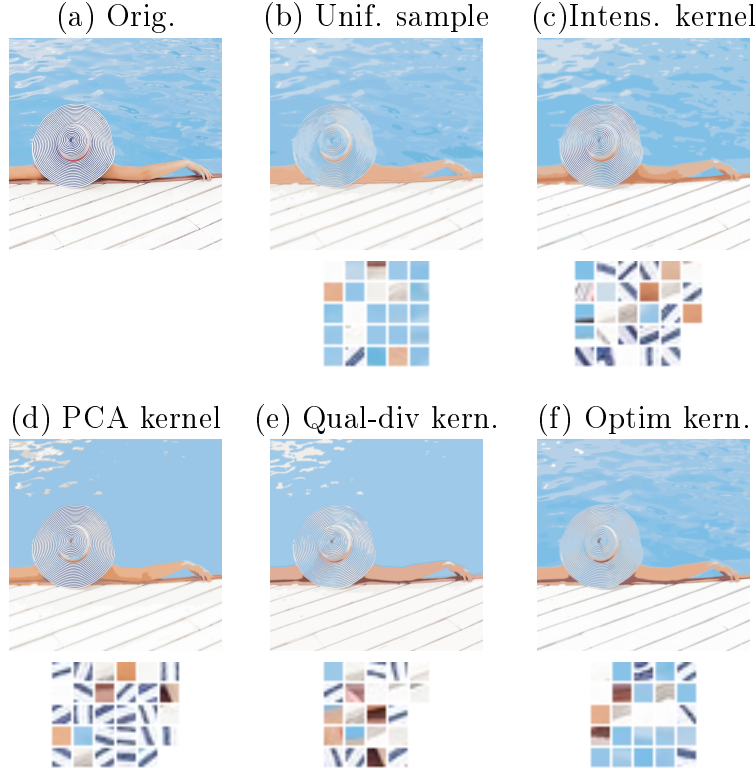


Figure 1.9: Image reconstructions comparing different DPP kernels. The first row presents the reconstruction of the image using only the patches selected by the corresponding kernel, given in the second row.

neighbor among the selection.

Section 4.2 applies this strategy to speed up a texture synthesis algorithm. This algorithm, presented in [52], uses the empirical distribution of the patches of an initial texture and heavily relies on semi-discrete optimal transport. This method enables to synthesize complex textures. The authors propose to uniformly subsample the set of patches of the image to approximate the empirical distribution of the patches, using 1000 patches.

After a presentation of this synthesis strategy, we show how using a DPP to subsample the distribution of patches enables us to reduce the number of patches (to 200 or 100) and thus to reduce the execution time of the algorithm while maintaining the quality of the synthesis. Figure 1.10 compares the strategies for two textures containing structures. The result using DPP is obtained using ten times less patches than the synthesis in column (b). The gain in computational time is significant. Once the model has been learned, for a synthesis of 1024×1024 images, using a MATLAB implementation of the algorithm on GPU, the algorithm runs in 0.47" using DPPs and 100 patches

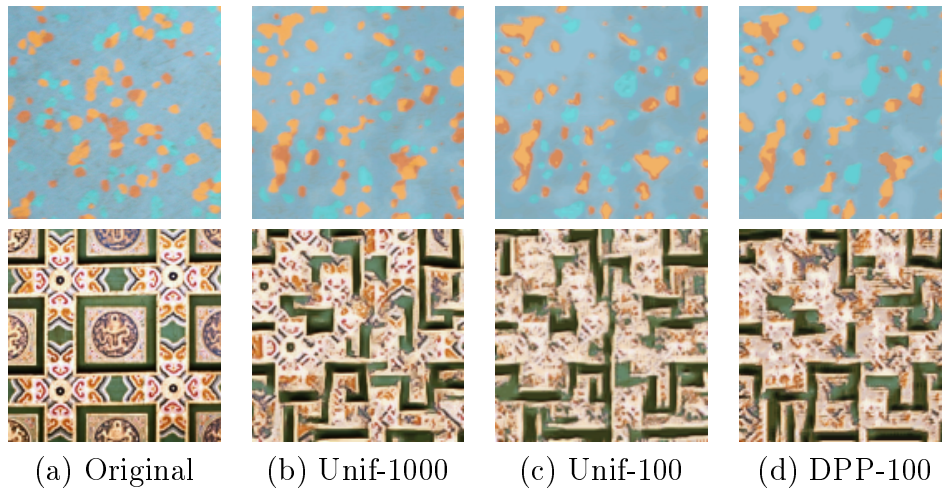


Figure 1.10: We compare the synthesis results when using either a target distribution with uniform subsampling (with cardinality 1000 or 100) or DPP subsampling (with expected cardinality 100).

and in 1.7" without DPPs, with 1000 patches.

Chapter 5

In Chapter 5, we conclude this manuscript. We summarize our main contributions and we discuss their limitations. We also present some perspectives and unanswered questions we would like to work on.

1.5 Contributions

The algorithms introduced in Chapter 2 is presented in an accepted paper for the journals of the Applied Probability trust, to appear in the Journal of Applied Probabilities 57.4 (December 2020)

Exact Sampling of Determinantal Point Processes without Eigendecomposition, Claire Launay, Bruno Galerne, Agnès Desolneux, preprint in Feb. 2018, <https://hal.archives-ouvertes.fr/hal-01710266/document>.

The content of Chapter 3 and of Chapter 4 Section 4.1, is presented in the submitted paper

Determinantal Point Processes for Image Processing, Claire Launay, Agnès Desolneux, Bruno Galerne, preprint in Mar. 2020, <https://hal.archives-ouvertes.fr/hal-02611259/document>.

A preliminary and French version of the work presented in Chapter 3, Sections 1 and 2, is introduced in the conference paper

Etude de la répulsion des processus pixelliques déterminantaux, Agnès Desolneux, Claire Launay, Bruno Galerne, proceedings of the GRETSI Conference, Sept. 2017, <https://hal.archives-ouvertes.fr/hal-01548767/document>

The application of DPPs to the texture synthesis algorithm [52] is discussed in

Determinantal Point Processes for Texture Synthesis, Claire Launay, Arthur Leclaire, proceedings of the GRETSI Conference, Aug. 2019, <https://hal.archives-ouvertes.fr/hal-02088725/document>.

Finally, MATLAB and Python implementations of the algorithms presented in Chapter 2 can be found on my webpage¹. A MATLAB implementation of the texture synthesis algorithm using (or not) DPPs can be found on Arthur Leclaire's webpage².

¹https://claunay.github.io/exact_sampling.html

²<https://www.math.u-bordeaux.fr/~aleclaire/texto/>

Chapter 2

Sampling Discrete DPPs

Contents

2.1	Introduction	37
2.2	Usual Sampling Method and Related Works	39
2.2.1	Spectral Algorithm	39
2.2.2	Other Sampling Strategies	41
2.3	Sequential Sampling Algorithm	44
2.3.1	Explicit General Marginal of a DPP	44
2.3.2	Sequential Sampling Algorithm of a DPP	46
2.4	Sequential Thinning Algorithm	47
2.4.1	General Framework of Sequential Thinning	47
2.4.2	Sequential Thinning Algorithm for DPPs	49
2.4.3	Computational Complexity	52
2.5	Experiments	53
2.5.1	DPP Models for Runtime Tests	53
2.5.2	Runtimes	54
2.6	Conclusion	60

2.1 Introduction

The simulation of a point process generates a subset of points, that can be used to reduce the size of an initial set of points, to illustrate the properties of a point process or to reduce the dimension of high-dimensional data. A sampling strategy must be efficient, especially when the size of the dataset grows. Concerning DPPs, the choice of the sampling method is crucial as it requires manipulating a kernel matrix K , which for most applications is very large. The classically used algorithm to sample general DPPs is called

the spectral algorithm. This algorithm relies on the fact that a general DPP can be considered as a mixture of projection DPPs, specific DPPs such that the eigenvalues of their kernel are either equal to 0 or to 1. The method, introduced in [72], is exact and it requires the computation of the eigenvalues and the eigenvectors of K . As soon as the underlying space on which the point process is defined is large, this method is slow. Many algorithms have been developed to sample DPPs more efficiently, by constraining the kernel to specific hypotheses [78, 41, 8], by approximating the kernel [64, 2] or by using Markov Chain Monte Carlo strategies [88, 56]. A few recent sampling methods are exact and apply to general DPP kernels [109, 63, 39].

In this chapter, we present a new exact algorithm to sample DPPs in discrete spaces, that avoids the eigenvalues and the eigenvectors computation. In Section 2.3, we introduce a sampling strategy that does not use the eigendecomposition of the matrix K but a Cholesky decomposition, that we call the sequential algorithm. However, this algorithm involves computations to be done sequentially on each point of the initial space. Hence, it is not efficient. To cope with this problem, we introduce in Section 2.4 a novel algorithm, called the sequential thinning algorithm. The proposed strategy relies on two new results: (i) the explicit formulation of the marginals of any determinantal point process and (ii) the derivation of an adapted Bernoulli point process containing a given DPP. As a first step, it samples a dominating point process that contains the target DPP and in a second step, it applies the sequential algorithm on this reduced space. This strategy is called the thinning of a point process. Finally, Section 2.5 presents numerical experiments to illustrate the behaviors of these algorithms.

This method was first presented in the preprint [83] and was, to our knowledge, the first exact sampling strategy without spectral decomposition. This paper has been accepted in the journals of the Applied probability trust. MATLAB and Python implementations of this algorithm (using the PyTorch library in the Python code) are available online¹ and hopefully soon in the repository created by Guillaume Gautier [57] gathering presentations and implementations of exact and approximate DPP sampling strategies.

In the following, we use the same notations as in the introduction. The state space, on which the DPP is defined, is supposed to be discrete, to contain N elements and is denoted by $\mathcal{Y} = \{1, \dots, N\}$. The DPP we want to sample from is characterized by the kernel K , which is a $N \times N$ matrix, whose eigenvalues are denoted by $\{\lambda_1, \dots, \lambda_N\}$.

¹https://claunay.github.io/exact_sampling.html

2.2 Usual Sampling Method and Related Works

2.2.1 Spectral Algorithm

The spectral algorithm is standard for drawing a determinantal point process. It relies on the eigendecomposition of its kernel K . It was first introduced by Hough et al. [72] and is also presented in a more detailed way by Scardicchio [118], Kulesza and Taskar [81] or Lavancier et al. [85].

This algorithm relies on the fact that DPPs can be written as mixtures of projection DPPs [72], also called *elementary* DPPs in [81]. We recall that a projection DPP is a DPP whose kernel has eigenvalues in $\{0, 1\}^N$. Let us consider a general discrete DPP kernel K , an eigendecomposition of the kernel $K = \sum_{j \in \mathcal{Y}} \lambda_j v_j v_j^*$, and denote $Y \sim \text{DPP}(K)$. We define the following random projection kernel

$$K^B = \sum_{j \in \mathcal{Y}} \text{Ber}(\lambda_j) v_j v_j^*. \quad (2.1)$$

where for all $j \in \mathcal{Y}$, $\text{Ber}(\lambda_j)$ is a Bernoulli variable with parameter $\lambda_j \in [0, 1]$. Hough et al. [72, Theorem 7] proved that this kernel K^B is a random analogue of K , in the sense that given $Y^B \sim \text{DPP}(K^B)$, we have

$$Y^B \stackrel{d}{=} Y. \quad (2.2)$$

The spectral algorithm takes advantage of this characterization. It proceeds in 3 steps. During the first step, the eigenvalues (λ_j) and the eigenvectors (v_j) of the matrix K are computed. The second step consists in randomly drawing N independent Bernoulli variables, each with parameter λ_j , for $j = 1, \dots, N$, and in storing the eigenvectors associated with the variables equal to 1 in a matrix V . Thus, the matrix VV^* (where V^* refers to the conjugate transpose of V) is an admissible DPP kernel, with every eigenvalue in $\{0, 1\}$. The third step consists in drawing the projection DPP associated to the kernel VV^* , using the relation between determinants and volumes of parallelotopes, which are the generalization of parallelograms in any dimension. This sampling sequentially selects the points, using a Gram-Schmidt procedure to compute pointwise conditional probabilities given the points already selected. Algorithm 1 presents this procedure.

This characterization impacts the distribution of the cardinality of the DPP. Consider $n \in \mathbb{N}$ such that $1 \leq n \leq N$ and suppose that the second step of the algorithm produced n Bernoulli variables equal to 1 (and thus $N - n$ Bernoulli variables equal to 0). The matrix VV^* has n non-zero eigenvalues equal to 1, it is the kernel of a projection process so it generates fixed size

samples with exactly n points. The size of the generated sample is determined by the drawing of the independent Bernoulli variables:

$$|Y| \sim \sum_{j \in \mathcal{Y}} \mathcal{Ber}(\lambda_j). \quad (2.3)$$

We can deduce that necessarily $|Y| \leq \text{rank}(K)$. Furthermore, we retrieve the properties given in the introduction: $\mathbb{E}(|Y|) = \sum_{j \in \mathcal{Y}} \lambda_j$ and $\text{Var}(|Y|) =$

$$\sum_{j \in \mathcal{Y}} \lambda_j(1 - \lambda_j).$$

Algorithm 1 The spectral sampling algorithm

1. Compute the orthonormal eigendecomposition (λ_j, v_j) of the matrix K .
2. Select a random set of eigenvectors: Draw a Bernoulli process $\mathbf{X} \in \{0, 1\}^N$ with parameter $(\lambda_j)_j$. Denote by n the number of Bernoulli samples equal to one, $\{\mathbf{X} = 1\} = \{j_1, \dots, j_n\}$. Define the matrix $V = (v_{j_1} \ v_{j_2} \ \dots \ v_{j_n}) \in \mathbb{R}^{N \times n}$ and denote by $V_k \in \mathbb{R}^N$ the k -th line of V , for $k \in \mathcal{Y}$.
3. Return the sequence $Y = \{y_1, y_2, \dots, y_n\}$ sequentially drawn as follows:
For $l = 1$ to n

- Sample a point $y_l \in \mathcal{Y}$ from the discrete distribution,

$$p_k^l = \frac{1}{n - l + 1} \left(\|V_k\|^2 - \sum_{m=1}^{l-1} |\langle V_k, e_m \rangle|^2 \right), \forall k \in \mathcal{Y}. \quad (2.4)$$

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

This algorithm is exact and relatively fast but it becomes slow when the size of the ground set grows. For a ground set of size N and a sample of size n , the third step costs $O(Nn^3)$ because of the Gram-Schmidt orthonormalisation. Tremblay et al. [125] propose to speed it up using optimized computations and they achieve the complexity $O(Nn^2)$ for this third step. Nevertheless, the eigendecomposition of the matrix K is the heaviest part of the algorithm, as it runs in time $O(N^3)$, and we will see in the numerical results that this first step represents in general more than 90% of the running time of the spectral algorithm. As nowadays the amount of data explodes, in practice the matrix K is very large so it seems relevant to try to avoid this costly operation. At the end of Section 2.4, we compare the time complexities of this spectral

algorithm with the algorithms we introduce in this chapter, the sequential algorithm (Algorithm 2) and the sequential thinning algorithm (Algorithm 3).

2.2.2 Other Sampling Strategies

As we have seen in the previous section, the main algorithm to sample DPPs is a spectral algorithm which uses the eigendecomposition of K to sample Y . This computation may be very costly when dealing with large-scale data. That is why numerous algorithms have been conceived to bypass this issue.

Sampling specific DPPs

Some authors have designed a sampling algorithm adapted to specific DPPs. For instance, it is possible to use an alternative algorithm, faster than the initial one, by assuming that K has a bounded rank [78, 81, 53]. These authors use a dual representation of the kernel so that the main computations in the spectral algorithm are reduced. In these articles, DPPs are L -ensemble, characterized by the positive semi-definite matrix L . Due to L 's positivity, there exists a $D \times N$ matrix B , such that $L = B^t B$, with $D \in \mathbb{N}^*$. It is possible to construct a dual representative $C = BB^t$, a matrix of size $D \times D$. In [78, 81], Kulesza and Taskar use this dual representation and prove that the computations needed for the sampling algorithm, to sample $\text{DPP}_L(L)$, can all be expressed in function of C , and be done on a $D \times D$ matrix instead of the $N \times N$ matrix L . They call this sampling algorithm, which has C as input, the dual sampling algorithm. Note that B_j , the j -th column of B can be considered as a feature vector associated to the point $j \in \mathcal{Y}$. The authors suppose that in general, $D \ll N$, meaning that the number of features representing the data is much smaller than the amount of data. In that case, L is low rank and one can use the dual algorithm detailed in [81, Algorithm 3] and sample the DPP faster, with a running complexity of order $O(D^3)$.

One can also deal with another class of DPPs associated to kernels K that can be decomposed into a sum of tractable matrices [41]. In this case, the sampling is much faster and the authors study the inference on these classes of DPPs. At last, Propp and Wilson [110] use Markov chains and the theory of coupling from the past to sample exactly particular DPPs: uniform spanning trees. Adapting Propp and Wilson's algorithm, Avena and Gaudillière [8] provide a similar algorithm to efficiently sample a parametric DPP kernel associated to random spanning forests.

Approximate algorithms

The second option to sample DPPs more efficiently is to use approximate methods. A first strategy is to approach the initial DPP kernel with another kernel, simpler to sample from. For instance, some authors approach the original DPP with a low rank matrix, using random projections [81, 64]. In these two papers, the authors use the decomposition of the L -ensemble kernel L seen previously, that is $L = B^t B$, with B a $D \times N$ matrix. Here, they suppose that D is not small, so they want to reduce the dimension of the feature vectors B_j associated to each point $j \in \mathcal{Y}$. To do so, they use a random projection matrix G , of size $d \times D$, with $d \ll D$. The coefficients of G are sampled independently from a Gaussian distribution and the authors prove that this model, applying random projection on the feature vectors, has a bounded approximation error.

If the previous decomposition of the L -ensemble kernel L is complicated, one can also use the Nyström approximation [2] to produce a low rank approximation of L . The main idea of the Nyström approximation is to select, with a suitable method, a proportion of elements of \mathcal{Y} called landmarks and to compute an approximation of L . In the end, this method produces an approximated low-rank decomposition $\tilde{L} = B_W^t B_W$, with B_W a $l \times N$ matrix and l the number of landmarks. Then, it applies the dual sampling algorithm to simulate the DPP.

A second strategy consists in using Monte Carlo Markov Chain (MCMC) methods. The method proposed by Anari et al. [6] and Li et al. [88] is based on iterative additions, deletions or exchanges of elements, until the mixing of the chain. In any step, associated to the selected set S , some elements $i \in S$ and $j \notin S$ are chosen independently and uniformly. Then i is deleted with a given probability, j is added with another one. Gautier et al. [56] developed a sampling algorithm based on MCMC strategies but from another perspective. They consider the initial state space as embedded into a continuous multi-dimensional polytope. This method consists in moving across this continuous domain by solving linear programs at each step of the chain. Unlike the previous MCMC methods modifying at most two elements of S , from one step of the algorithm to the other, the whole set S can be modified. This enables to explore the state space more easily but each step needs to solve a costly linear problem.

It is possible to obtain satisfying convergence guarantees for these strategies for particular DPPs, for instance for k-DPPs with fixed cardinality [6, 87] or projection DPPs [56]. Li et al. [88] even proposed a polynomial-time sampling algorithm for general DPPs.

Approximate strategies hope that after a certain number of simpler sam-

pling iterations, the result is sufficiently close to the target distribution. However, one needs to decide when to stop the algorithm, and what does “sufficiently close” mean. Second, this equilibrium may need a high number of iterations to be (almost) reached. These algorithms are commonly used as they save significant time but the price to pay is the lack of precision of the result.

Recent exact algorithms

Let us mention that three very recent preprints [109, 63, 39] also propose new algorithms to sample exactly general DPPs without spectral decomposition.

Poulson [109] presents factorization strategies of Hermitian and non-Hermitian DPP kernels to sample general determinantal point processes. As our sequential algorithm (Algorithm 2), it heavily relies on Cholesky decomposition and proceeds sequentially. It accepts or rejects each element of the state space according to pointwise conditional probabilities given the points already accepted. These sampling strategies generalize our own and adapt to non-Hermitian or sparse DPP kernels.

Gillenwater and al. [63] use the dual representation of L -ensembles presented previously to construct a binary tree. This tree contains enough information on the kernel to sample DPPs in sublinear time, after a preprocessing step done in $O(ND^2)$ time (where D is the size of the features vectors in the dual representation).

Dereziński et al. [39] apply a preprocessing step that preselects a portion of the points using a regularized DPP. This regularized DPP takes advantage of the connections between DPP’s marginal probabilities and ridge leverage scores of the L -ensemble kernel L , quantities that have already been used in sampling strategies. Then, a usual DPP sampling is done on the selection. Their preprocessing step is called intermediate sampling and is very related to our thinning procedure using a Bernoulli point process that contains the target DPP. However note that the authors report that the overall complexity of their sampling scheme is sublinear while ours is cubic due to Cholesky decomposition.

Finally, in [15], Blaszczyzyn and Keeler present a similar procedure based on a continuous space: they use discrete determinantal point processes to thin a Poisson point process defined on that continuous space. The generated point process offers theoretical guarantees on repulsion and is applied to fit network patterns.

In the next section, we show that any DPP can be exactly sampled by a sequential algorithm that does not require the eigendecomposition of K .

2.3 Sequential Sampling Algorithm

Our goal is to build a competitive algorithm to sample DPPs that does not involve the eigendecomposition of the matrix K . To do so, we first develop a “naive” sequential sampling algorithm and subsequently, we will accelerate it using a thinning procedure, presented in Section 2.4.

2.3.1 Explicit General Marginal of a DPP

First, we need to specify the marginals and the conditional probabilities of any DPP. When $I - K$ is invertible, a formulation of the explicit marginals already exists [81], it implies to deal with a L -ensemble matrix L instead of the matrix K . However, this hypothesis is reductive: among others, it ignores the useful case of projection DPPs, when the eigenvalues of K are either 0 or 1. We show below that general marginals can easily be formulated from the associated kernel matrix K . For all $A \subset \mathcal{Y}$, we denote I^A the $N \times N$ matrix with 1 on its diagonal coefficients indexed by the elements of A , and 0 anywhere else. We also denote $|A|$ the cardinality of any subset $A \subset \mathcal{Y}$ and $A^c \in \mathcal{Y}$ the complementary set of A in \mathcal{Y} .

Proposition 2.3.1 (Distribution of a DPP). *For any $A \subset \mathcal{Y}$, we have*

$$\mathbb{P}(Y = A) = (-1)^{|A|} \det(I^{A^c} - K). \quad (2.5)$$

Proof. We have that $\mathbb{P}(A \subset Y) = \sum_{B \supset A} \mathbb{P}(Y = B)$. Using the Möbius inversion formula (see Appendix A.1), for all $A \subset \mathcal{Y}$,

$$\begin{aligned} \mathbb{P}(Y = A) &= \sum_{B \supset A} (-1)^{|B \setminus A|} \mathbb{P}(B \subset Y) = (-1)^{|A|} \sum_{B \supset A} (-1)^{|B|} \det(K_B) \\ &= (-1)^{|A|} \sum_{B \supset A} \det((-K)_B). \end{aligned} \quad (2.6)$$

Furthermore, Kulesza and Taskar [81] state in Theorem 2.1 that for all $L \in \mathbb{R}^{N \times N}$, for all $A \subset \mathcal{Y}$, $\sum_{A \subset B \subset \mathcal{Y}} \det(L_B) = \det(I^{A^c} + L)$. Then we obtain

$$\mathbb{P}(Y = A) = (-1)^{|A|} \det(I^{A^c} - K). \quad (2.7)$$

□

We have by definition $\mathbb{P}(A \subset Y) = \det(K_A)$ for all A , and as a consequence $\mathbb{P}(B \cap Y = \emptyset) = \det((I - K)_B)$ for all B . The next proposition gives for any DPP the expression of the general marginal $\mathbb{P}(A \subset Y, B \cap Y = \emptyset)$, for any A, B

disjoint subsets of \mathcal{Y} , using K . In what follows, H^B denotes the symmetric positive semi-definite matrix

$$H^B = K + K_{\mathcal{Y} \times B}((I - K)_B)^{-1}K_{B \times \mathcal{Y}}. \quad (2.8)$$

Theorem 2.3.1 (General Marginal of a DPP). *Let $A, B \subset \mathcal{Y}$ be disjoint. If $\mathbb{P}(B \cap Y = \emptyset) = \det((I - K)_B) = 0$, then $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = 0$. Otherwise, the matrix $(I - K)_B$ is invertible and*

$$\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I - K)_B) \det(H_A^B). \quad (2.9)$$

Proof. Let $A, B \subset \mathcal{Y}$ disjoint such that $\mathbb{P}(B \cap Y = \emptyset) \neq 0$. Using the previous proposition,

$$\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \sum_{A \subset C \subset B^c} \mathbb{P}(Y = C) = \sum_{A \subset C \subset B^c} (-1)^{|C|} \det(I^{C^c} - K). \quad (2.10)$$

For any C such that $A \subset C \subset B^c$, one has $B \subset C^c$. Hence, by reordering the matrix coefficients, and using the Schur's determinant formula [70],

$$\begin{aligned} \det(I^{C^c} - K) &= \det \begin{pmatrix} (I^{C^c} - K)_B & (I^{C^c} - K)_{B \times B^c} \\ (I^{C^c} - K)_{B^c \times B} & (I^{C^c} - K)_{B^c} \end{pmatrix} \\ &= \det \begin{pmatrix} (I - K)_B & -K_{B \times B^c} \\ -K_{B^c \times B} & (I^{C^c} - K)_{B^c} \end{pmatrix} \\ &= \det((I - K)_B) \det((I^{C^c} - H^B)_{B^c}). \end{aligned} \quad (2.11)$$

Thus, $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I - K)_B) \sum_{A \subset C \subset B^c} (-1)^{|C|} \det((I^{C^c} - H^B)_{B^c})$.

According to Theorem 2.1 in Kulesza and Taskar [81], for all $A \subset B^c$,

$$\sum_{A \subset C \subset B^c} \det(-H_C^B) = \det((I^{A^c} - H^B)_{B^c}). \quad (2.12)$$

Then, Möbius inversion formula ensures that, $\forall A \subset B^c$,

$$\sum_{A \subset C \subset B^c} (-1)^{|C \setminus A|} \det((I^{C^c} - H^B)_{B^c}) = \det(-H_A^B) = (-1)^{|A|} \det(H_A^B). \quad (2.13)$$

Hence, $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) = \det((I - K)_B) \det(H_A^B)$. \square

With this formula, we can explicitly formulate the pointwise conditional probabilities of any DPP.

Corollary 2.3.1 (Pointwise conditional probabilities of a DPP). *Let $A, B \subset \mathcal{Y}$ be two disjoint sets such that $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) \neq 0$, and let $k \notin A \cup B$. Then,*

$$\begin{aligned} \mathbb{P}(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset) &= \frac{\det(H_{A \cup \{k\}}^B)}{\det(H_A^B)} \\ &= H^B(k, k) - H_{\{k\} \times A}^B (H_A^B)^{-1} H_{A \times \{k\}}^B. \end{aligned} \quad (2.14)$$

This is a straightforward application of the previous expression and the Schur determinant formula [70]. Note that these pointwise conditional probabilities are related to the Palm distribution of a point process [29] which characterizes the distribution of the point process under the condition that there is a point at some location $x \in \mathcal{Y}$. Shirai and Takahashi proved in [120] that DPPs on general spaces are closed under Palm distributions, in the sense that there exists a DPP kernel K^x such that the Palm measure associated to $\text{DPP}(K)$ and x is a DPP defined on \mathcal{Y} with kernel K^x . Borodin and Rains [23] also provide similar results on discrete spaces, using L -ensembles, that Kulesza and Taskar adapt in [81]. They condition the DPP not only on a subset included in the point process but also, similarly as Corollary 2.14, on a subset not included in the point process. As Shirai and Takahashi, they derive a formulation of the generated marginal kernel L .

Now, we have all the necessary expressions for the sequential sampling of a DPP.

2.3.2 Sequential Sampling Algorithm of a DPP

This sequential sampling algorithm simply consists in using Formula (2.14) and updating at each step the pointwise conditional probability, knowing the previous selected points. It is presented in Algorithm 2. We recall that this sequential algorithm is the first step toward developing a competitive sampling algorithm for DPPs: with this method, one doesn't need eigendecomposition anymore. The second strategy (presented in Section 2.4) will be to reduce its computational cost.

Algorithm 2 Sequential sampling of a DPP with kernel K

- Initialization: $A \leftarrow \emptyset$, $B \leftarrow \emptyset$.
 - For $k = 1$ to N :
 1. Compute $H_{A \cup \{k\}}^B = K_{A \cup \{k\}} + K_{A \cup \{k\} \times B}((I - K)_B)^{-1} K_{B \times A \cup \{k\}}$.
 2. Compute the probability p_k given by

$$p_k = \mathbb{P}(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset) = H^B(k, k) - H_{\{k\} \times A}^B (H_A^B)^{-1} H_{A \times \{k\}}^B. \quad (2.15)$$
 3. With probability p_k , k is included, $A \leftarrow A \cup \{k\}$, otherwise $B \leftarrow B \cup \{k\}$.
 - Return A .
-

The main operations of Algorithm 2 involve solving linear systems related to $(I - K)_B^{-1}$. Fortunately, here we can use the Cholesky factorization,

which alleviates the computational cost. Suppose that T^B is the Cholesky factorization of $(I - K)_B$, that is, T^B is a lower triangular matrix such that $(I - K)_B = T^B(T^B)^*$ (where $(T^B)^*$ is the conjugate transpose of T^B). Then, denoting $J^B = (T^B)^{-1}K_{B \times A \cup \{k\}}$, one simply has $H_{A \cup \{k\}}^B = K_{A \cup \{k\}} + (J^B)^*J^B$.

Furthermore, at each iteration where B grows, the Cholesky decomposition $T^{B \cup \{k\}}$ of $(I - K)_{B \cup \{k\}}$ can be computed from T^B using standard Cholesky update operations, involving the resolution of only one linear system of size $|B|$. See Appendix A.2 for the details of a typical Cholesky decomposition update.

In comparison with the spectral sampling algorithm of Hough et al. [72], one requires computations for each site of \mathcal{Y} , and not just one for each sampled point of Y . We will see at the end of Section 2.4 and in the experiments that it is not competitive.

2.4 Sequential Thinning Algorithm

In this section, we show that we can significantly decrease the number of steps and the running time of Algorithm 2: we propose to first sample a point process X containing Y , the desired DPP, and then make a sequential selection of the points of X to obtain Y . This procedure can be called a sequential thinning.

2.4.1 General Framework of Sequential Thinning

We first describe a general sufficient condition for which a target point process Y - it will be a determinantal point process in our case - can be obtained as a sequential thinning of a point process X . This is a discrete adaptation of the thinning procedure on the continuous line of Rolski and Szekli [114]. To do this, we will consider a coupling (X, Z) such that $Z \subset X$ will be a random selection of the points of X and that will have the same distribution as Y . From this point onward, we identify the set X with the vector of size N with 1 in the place of the elements of X and 0 elsewhere, and we use the notations $X_{1:k}$ to denote the vector (X_1, \dots, X_k) and $0_{1:k}$ to denote the null vector of size k . We want to define the random vector $(X_1, Z_1, X_2, Z_2, \dots, X_N, Z_N) \in \mathbb{R}^{2N}$ with the following conditional distributions for X_k and Z_k :

$$\begin{cases} \mathbb{P}(X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = \mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}) \\ \mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k} = x_{1:k}) = \mathbb{1}_{\{x_k=1\}} \frac{\mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1})}{\mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1})}. \end{cases} \quad (2.16)$$

Proposition 2.4.1 (Sequential thinning). *Assume that X, Y, Z are discrete point processes on \mathcal{Y} that satisfy for all $k \in \{1, \dots, N\}$, and all $z, x \in \{0, 1\}^N$,*

$$\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$$

implies

$$(2.17)$$

$$\mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \leq \mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}).$$

Then, it is possible to choose (X, Z) in such a way that (2.16) is satisfied. In that case, we have that Z is a thinning of X , that is $Z \subset X$, and Z has the same distribution as Y .

Proof. Let us first discuss the definition of the coupling (X, Z) . With the conditions (2.17), the ratios defining the conditional probabilities of Equation (2.16) are ensured to be between 0 and 1 (if the conditional events have non zero probabilities). Hence the conditional probabilities allow us to construct sequentially the distribution of the random vector $(X_1, Z_1, X_2, Z_2, \dots, X_N, Z_N)$ of length $2N$, and thus the coupling is well-defined. Furthermore, as Equation (2.16) is satisfied, $Z_k = 1$ only if $X_k = 1$, so one has $Z \subset X$.

Let us now show that Z has the same distribution as Y . By complementarity of the events $\{Z_k = 0\}$ and $\{Z_k = 1\}$, it is enough to show that for all $k \in \{1, \dots, N\}$, and z_1, \dots, z_{k-1} such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}) > 0$,

$$\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}) = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}). \quad (2.18)$$

Let $k \in \{1, \dots, N\}$, $(z_{1:k-1}, x_{1:k-1}) \in \{0, 1\}^{2(k-1)}$, such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$. Since $Z \subset X$, $\{Z_k = 1\} = \{Z_k = 1, X_k = 1\}$. Suppose first that $\mathbb{P}(X_k = 1 | X_1 = x_1, \dots, X_{k-1} = x_{k-1}) \neq 0$. Then

$$\begin{aligned} \mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \mathbb{P}(Z_k = 1, X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}, X_k = 1) \\ &= \mathbb{P}(X_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) \\ &= \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}), \text{ by Equations (2.16)}. \end{aligned} \quad (2.19)$$

If $\mathbb{P}(X_k = 1 | X_{1:k-1} = x_{1:k-1}) = 0$, then $\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = 0$ and using (2.17), $\mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) = 0$. Hence the identity

$$\mathbb{P}(Z_k = 1 | Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) = \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \quad (2.20)$$

is always valid. Since the values x_1, \dots, x_{k-1} do not influence this conditional probability, one can conclude that given (Z_1, \dots, Z_{k-1}) , Z_k is independent of X_1, \dots, X_{k-1} , and thus (2.18) is true. \square

The characterization of the thinning defined here allows both extreme cases: there can be no pre-selection of points by X , meaning that $X = \mathcal{Y}$ and that the DPP Y is sampled by Algorithm 2, or there can be no thinning at all, meaning that the final process Y can be equal to the dominating process X . Regarding sampling acceleration, a good dominating process X must be sampled quickly and with a cardinality as close as possible to $|Y|$.

2.4.2 Sequential Thinning Algorithm for DPPs

In this section, we use the sequential thinning approach, where Y is a DPP of kernel K on the ground set \mathcal{Y} , and X is a Bernoulli point process (BPP). BPPs are the fastest and easiest point processes to sample. The point process X is a Bernoulli process if the components of the vector (X_1, \dots, X_N) are independent. Its distribution is determined by the probability of occurrence of each point k , that we denote by $q_k = \mathbb{P}(X_k = 1)$. Due to the independence property, the conditions (2.17) simplifies to

$$\begin{aligned} \mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0 \\ \text{implies} \\ \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}) \leq q_k. \end{aligned} \quad (2.21)$$

The second inequality does not depend on x , hence it must be valid as soon as there exists a vector x such that $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}, X_{1:k-1} = x_{1:k-1}) > 0$, that is, as soon as $\mathbb{P}(Z_{1:k-1} = z_{1:k-1}) > 0$. Since we want Z to have the same distribution as Y , we finally obtain the conditions

$$\forall y \in \{0, 1\}^N, \mathbb{P}(Y_{1:k-1} = y_{1:k-1}) > 0 \text{ implies } \mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}) \leq q_k. \quad (2.22)$$

Ideally, we want the q_k to be as small as possible to ensure that the cardinality of X is as small as possible. So we look for the optimal values q_k^* , that is,

$$q_k^* = \max_{\substack{(y_{1:k-1}) \in \{0,1\}^{k-1} \text{ s.t.} \\ \mathbb{P}(Y_{1:k-1} = y_{1:k-1}) > 0}} \mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}). \quad (2.23)$$

A priori, computing q_k^* would raise combinatorial issues. However, due to the repulsive nature of DPPs, we have the following proposition.

Proposition 2.4.2. *Let $A, B \subset \mathcal{Y}$ be two disjoint sets such that $\mathbb{P}(A \subset Y, B \cap Y = \emptyset) \neq 0$, and let $k \neq l \in (A \cup B)^c$. If $\mathbb{P}(A \cup \{l\} \subset Y, B \cap Y = \emptyset) > 0$, then*

$$\mathbb{P}(\{k\} \subset Y | A \cup \{l\} \subset Y, B \cap Y = \emptyset) \leq \mathbb{P}(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset). \quad (2.24)$$

If $\mathbb{P}(A \subset Y, (B \cup \{l\}) \cap Y = \emptyset) > 0$, then

$$\mathbb{P}(\{k\} \subset Y | A \subset Y, (B \cup \{l\}) \cap Y = \emptyset) \geq \mathbb{P}(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset). \quad (2.25)$$

Consequently, for all $k \in \mathcal{Y}$, if $y_{1:k-1} \leq z_{1:k-1}$ (where \leq stands for the inclusion partial order) are two states for $Y_{1:k-1}$, then

$$\mathbb{P}(Y_k = 1 | Y_{1:k-1} = y_{1:k-1}) \geq \mathbb{P}(Y_k = 1 | Y_{1:k-1} = z_{1:k-1}). \quad (2.26)$$

In particular, $\forall k \in \{1, \dots, N\}$, if $\mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0$ then

$$\begin{aligned} q_k^* &= \mathbb{P}(Y_k = 1 | Y_{1:k-1} = 0_{1:k-1}) \\ &= K(k, k) + K_{k \times \{1:k-1\}}((I - K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\} \times k}. \end{aligned} \quad (2.27)$$

Proof. Recall that by Proposition 2.3.1, $P(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset) = H^B(k, k) - H_{\{k\} \times A}^B (H_A^B)^{-1} H_{A \times \{k\}}^B$. Let $l \notin A \cup B \cup \{k\}$. Consider T^B the Cholesky decomposition of the matrix H^B obtained with the following ordering the coefficients: A , l , the remaining coefficients of $\mathcal{Y} \setminus (A \cup \{l\})$. Then, the restriction T_A^B is the Cholesky decomposition (of the reordered) H_A^B and thus

$$\begin{aligned} H_{\{k\} \times A}^B (H_A^B)^{-1} H_{A \times \{k\}}^B &= H_{\{k\} \times A}^B (T_A^B (T_A^B)^*)^{-1} H_{A \times \{k\}}^B \\ &= \|(T_A^B)^{-1} H_{A \times \{k\}}^B\|_2^2. \end{aligned} \quad (2.28)$$

Similarly,

$$H_{\{k\} \times A \cup \{l\}}^B (H_{A \cup \{l\}}^B)^{-1} H_{A \cup \{l\} \times \{k\}}^B = \|(T_{A \cup \{l\}}^B)^{-1} H_{A \cup \{l\} \times \{k\}}^B\|_2^2. \quad (2.29)$$

Now note that solving the triangular system with $b = (T_{A \cup \{l\}}^B)^{-1} H_{A \cup \{l\} \times \{k\}}^B$ amounts solving the triangular system with $(T_A^B)^{-1} H_{A \times \{k\}}^B$ and an additional line at the bottom. Hence, one has $\|b\|_2^2 \geq \|(T_A^B)^{-1} H_{A \times \{k\}}^B\|_2^2$. Consequently, provided that $\mathbb{P}(A \cup \{l\} \subset Y, B \cap Y = \emptyset) > 0$,

$$\mathbb{P}(\{k\} \subset Y | A \cup \{l\} \subset Y, B \cap Y = \emptyset) \leq \mathbb{P}(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset). \quad (2.30)$$

The second inequality is obtained by complementarity in applying the above inequality to the DPP Y^c with $B \cup \{l\} \subset Y^c$ and $A \cap Y^c = \emptyset$. \square

As a consequence, an admissible choice for the distribution of the Bernoulli process is

$$q_k = \begin{cases} \mathbb{P}(Y_k = 1 | Y_{1:k-1} = 0_{1:k-1}) & \text{if } \mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (2.31)$$

Note that if for some index k , $\mathbb{P}(Y_{1:k-1} = 0_{1:k-1}) > 0$ is not satisfied, then for all the subsequent indexes $l \geq k$, $q_l = 1$, that is the Bernoulli process becomes degenerate and contains all the points after k . In the remaining of this section, X will denote a Bernoulli process with probabilities (q_k) given by (2.31).

As discussed in the previous section, in addition to being easily simulated, one would like the cardinality of X to be close to the one of Y , the final sample. The next proposition shows that this is verified if all the eigenvalues of K are strictly less than 1.

Proposition 2.4.3 ($|X|$ is proportional to $|Y|$). *Suppose that $P(Y = \emptyset) = \det(I - K) > 0$ and denote by $\lambda_{\max}(K) \in [0, 1)$ the maximal eigenvalue of K . Then,*

$$\mathbb{E}(|X|) \leq \left(1 + \frac{\lambda_{\max}(K)}{2(1 - \lambda_{\max}(K))}\right) \mathbb{E}(|Y|). \quad (2.32)$$

Proof. We know that $q_k = K(k, k) + K_{\{k\} \times \{1:k-1\}}((I - K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\} \times \{k\}}$, by Proposition 2.3.1. Since

$$\|((I - K)_{\{1:k-1\}})^{-1}\|_{\mathcal{M}_{k-1}(\mathbb{C})} = \frac{1}{1 - \lambda_{\max}(K_{\{1:k-1\}})} \quad (2.33)$$

and $\lambda_{\max}(K_{\{1:k-1\}}) \leq \lambda_{\max}(K)$, one has

$$K_{\{k\} \times \{1:k-1\}}((I - K)_{\{1:k-1\}})^{-1} K_{\{1:k-1\} \times \{k\}} \leq \frac{1}{1 - \lambda_{\max}(K)} \|K_{\{1:k-1\} \times \{k\}}\|_2^2. \quad (2.34)$$

Summing all these inequalities gives

$$\mathbb{E}(|X|) \leq \text{Tr}(K) + \frac{1}{1 - \lambda_{\max}(K)} \sum_{k=1}^N \|K_{\{1:k-1\} \times \{k\}}\|_2^2. \quad (2.35)$$

The last term is the Frobenius norm of the upper triangular part of K , hence it can be bounded by $\frac{1}{2} \|K\|_F^2 = \frac{1}{2} \sum_{j=1}^N \lambda_j(K)^2$. Since $\lambda_j(K)^2 \leq \lambda_j(K) \lambda_{\max}(K)$, $\sum_{j=1}^N \lambda_j(K)^2 \leq \lambda_{\max}(K) \text{Tr}(K) = \lambda_{\max}(K) \mathbb{E}(|Y|)$. \square

We can now introduce the final sampling algorithm that we call sequential thinning algorithm (Algorithm 3). It presents the different steps of our sequential thinning algorithm to sample a DPP of kernel K . The first step is a preprocess that must be done only once for a given matrix K . Step 2 is trivial and fast. The critical point is to sequentially compute the conditional probabilities $p_k = \mathbb{P}(\{k\} \subset Y | A \subset Y, B \cap Y = \emptyset)$ for each point of X . Recall that in Algorithm 2 we use a Cholesky decomposition of the matrix $(I - K)_B$ which is updated by adding a line each time a point is added in B . Here, the inverse of the matrix $(I - K)_B$ is only needed when visiting a point $k \in X$, so one updates the Cholesky decomposition by a single block, where the new block corresponds to all indices added to B in one iteration (see Appendix A.2). The MATLAB implementation used for the experiments is available online², together with a Python version of this code, using the PyTorch library. Note that, very recently, Guillaume Gautier [55] proposed an alternative computation of the Bernoulli probabilities q_k that generate the dominating point process in the first step of Algorithm 3, so that it only requires the diagonal coefficients of the Cholesky decomposition T of $I - K$. These simplified computations should improve the efficiency of the first step of the algorithm. We plan to test numerically how much this first step is sped up.

²https://claunay.github.io/exact_sampling.html

Algorithm 3 Sequential thinning algorithm of a DPP with kernel K

1. Compute sequentially the probabilities $\mathbb{P}(X_k = 1) = q_k$ of the Bernoulli process X :
 - Compute the Cholesky decomposition T of the matrix $I - K$.
 - For $k = 1$ to N :
 - If $q_{k-1} < 1$ (with the convention $q_0 = 0$),

$$q_k = K(k, k) + \|T_{\{1, \dots, k-1\}}^{-1} K_{\{1, \dots, k-1\} \times \{k\}}\|_2^2. \quad (2.36)$$
 - Else, $q_k = 1$.
2. Draw the Bernoulli process X . Let $m = |X|$ and $k_1 < k_2 < \dots < k_m$ be the points of X .
3. Apply the sequential thinning to the points of X :
 - Attempt to add sequentially each point of X to Y :

Initialize $A \leftarrow \emptyset$ and $B \leftarrow \{1, \dots, k_1 - 1\}$.

For $j = 1$ to m

 - If $j > 1$, $B \leftarrow B \cup \{k_{j-1} + 1, \dots, k_j - 1\}$.
 - Compute the conditional probability $p_{k_j} = \mathbb{P}(\{k_j\} \subset Y | A \subset Y, B \cap Y = \emptyset)$ (see Formula (2.14)):
 - * Update T^B the Cholesky decomposition of $(I - K)_B$ (see Appendix A.2).
 - * Compute $J^B = (T^B)^{-1} K_{B \times A \cup \{k_j\}}$.
 - * Compute $H_{A \cup \{k_j\}}^B = K_{A \cup \{k_j\}} + (J^B)^t J^B$.
 - * Compute $p_{k_j} = H^B(k_j, k_j) - H_{\{k_j\} \times A}^B (H_A^B)^{-1} H_{A \times \{k_j\}}^B$.
 - Add k_j to A with probability $\frac{p_{k_j}}{q_{k_j}}$ or to B otherwise.
 - Return A .

2.4.3 Computational Complexity

Recall that the size of the ground set \mathcal{Y} is N and the size of the final sample is $|Y| = n$. Both algorithms introduced in this chapter (Algorithms 2 and 3) have running complexities of order $O(N^3)$, as the spectral algorithm. Yet, if we get into the details, the most expensive task in the spectral algorithm is the computation of the eigenvalues and the eigenvectors of the kernel K . As this matrix is Hermitian, the common routine to do so is the reduction of K to some tridiagonal matrix to which the QR decomposition is applied,

meaning that it is decomposed into the product of an orthogonal matrix and an upper triangular matrix. When N is large, the total number of operations is approximately $\frac{4}{3}N^3$ [124]. In Algorithms 2 and 3, one of the most expensive operations is the Cholesky decomposition of several matrices. We recall that the Cholesky decomposition of a matrix of size $N \times N$ costs approximately $\frac{1}{3}N^3$ computations, when N is large [99]. Concerning the Sequential algorithm 2, at each iteration k , the number of operations needed is of order $|B|^2|A| + |B||A|^2 + |A|^3$, where $|A|$ is the number of selected points at step k so it's lower than n , and $|B|$ the number of unselected points, bounded by k . Then, when N tends to infinity, the total number of operations in Algorithm 2 is lower than $\frac{n}{3}N^3 + \frac{n^2}{2}N^2 + n^3N$ or $O(nN^3)$, as in general $n \ll N$. Concerning Algorithm 3, the sequential thinning from X , coming from Algorithm 2, costs $O(n|X|^3)$. Recall that $|X|$ is proportional to $|Y| = n$ when the eigenvalues of K are smaller than 1 (see Equation (2.32)) so this step costs $O(n^4)$. Then, the Cholesky decomposition of $I - K$ is the most expensive operation in Algorithm 3 as it costs approximately $\frac{1}{3}N^3$. In this case, the overall running complexity of the sequential thinning algorithm is of order $\frac{1}{3}N^3$, which is 4 times less than the spectral algorithm. When some eigenvalues of K are equal to 1, Equation (2.32) does not hold anymore so, in that case, the running complexity of Algorithm 3 is only bounded by $O(nN^3)$.

We will retrieve this experimentally as, depending on the application or on the kernel K , this Algorithm 3 is able to speed up the sampling of DPPs. Note that in the previous computations, we have not taken into account the possible parallelization of the sequential thinning algorithm. As a matter of fact, the Cholesky decomposition is parallelizable [61]. Incorporating this parallel computations would probably speed up the sequential thinning algorithm, since the Cholesky decomposition of $I - K$ is the most expensive operation when the expected cardinality $|Y|$ is low. The last part of the algorithm, the thinning procedure, operates sequentially, so it is not parallelizable. These comments on the complexity and running times highly depends on the implementation, on the choice of the programming language and speed up strategies, so they mainly serve as an illustration.

2.5 Experiments

2.5.1 DPP Models for Runtime Tests

In the following section, we use the common notation of L -ensembles, with matrix $L = K(I - K)^{-1}$. We present the results using four different kernels:

- (a) A random kernel: $K = Q^{-1}DQ$, where D is a diagonal matrix with uniformly distributed random values in $(0, 1)$ and Q an unitary matrix created from the QR decomposition of a random matrix.

- (b) A kernel similar to the continuous Ginibre kernel: $K = L(I + L)^{-1}$ with for all $x_1, x_2 \in \mathcal{Y} = \{1, \dots, N\}$,

$$L(x_1, x_2) = \frac{1}{\pi} e^{-\frac{1}{2}(|x_1|^2 + |x_2|^2) + x_1 x_2}, \quad (2.37)$$

- (c) A patch-based kernel: Let u be a discrete image and $\mathcal{Y} = \mathcal{P}$ a subset of all its patches, i.e. square sub-images of size $w \times w$ in u . Define $K = L(I + L)^{-1}$ where for all $P_1, P_2 \in \mathcal{P}$,

$$L(P_1, P_2) = \exp\left(-\frac{\|P_1 - P_2\|_2^2}{s^2}\right), \quad (2.38)$$

where $s > 0$ is called the bandwidth or scale parameter. We will detail the definition and the use of this kernel in Chapter 4.

- (d) A projection kernel: $K = Q^{-1}DQ$, where D is a diagonal matrix with the n first coefficients equal to 1, the others, equal to 0, and Q is a random unitary matrix as for model (a).

It is often essential to control the expected cardinality of the point process. For case (d) the cardinality is fixed to n . For the three other cases, we use a procedure similar to the one developed in [14]. Recall that if $Y \sim \text{DPP}(K)$ and $K = L(I + L)^{-1}$, $\mathbb{E}(|Y|) = \text{tr}(K) = \sum_{i \in \mathcal{Y}} \lambda_i = \sum_{i \in \mathcal{Y}} \frac{\mu_i}{1 + \mu_i}$, where $(\lambda_i)_{i \in \mathcal{Y}}$ are the eigenvalues of K and $(\mu_i)_{i \in \mathcal{Y}}$ are the eigenvalues of L [72, 81]. Given an initial matrix $L = K(I - K)^{-1}$ and a desired expected cardinality $\mathbb{E}(|Y|) = n$, we run a binary search algorithm to find $\alpha > 0$ such that $\sum_{i \in \mathcal{Y}} \frac{\alpha \mu_i}{1 + \alpha \mu_i} = n$. Then, we use the kernels $L_\alpha = \alpha L$ and $K_\alpha = L_\alpha(I + L_\alpha)^{-1}$.

2.5.2 Runtimes

For the following experiments, we ran the algorithms on a laptop HP Intel(R) Core(TM) i7-6600U CPU and we use the software MATLAB R2018b. Note that the computational time results depend on the programming language and the use of optimized functions by the software. Thus, the following numerical results are mainly indicative.

First, let us compare the sequential thinning algorithm (Algorithm 3) presented here with the two main sampling algorithms: the classic spectral algorithm (Algorithm 1) and the “naive” sequential algorithm (Algorithm 2). Figure 2.1 presents the running times of the three algorithms as a function of the total number of points of the ground set. Here, we have chosen a patch-based kernel (c). The expected cardinality $\mathbb{E}(|Y|)$ is constant, equal to 20.

As foreseen, the sequential algorithm (Algorithm 2) is far slower than the two others. Whatever the chosen kernel and the expected cardinality of the DPP, this algorithm is not competitive. Note that the sequential thinning algorithm uses this sequential method after sampling the particular Bernoulli process. But we will see that this first dominating step can be very efficient and lead to a relatively fast algorithm.

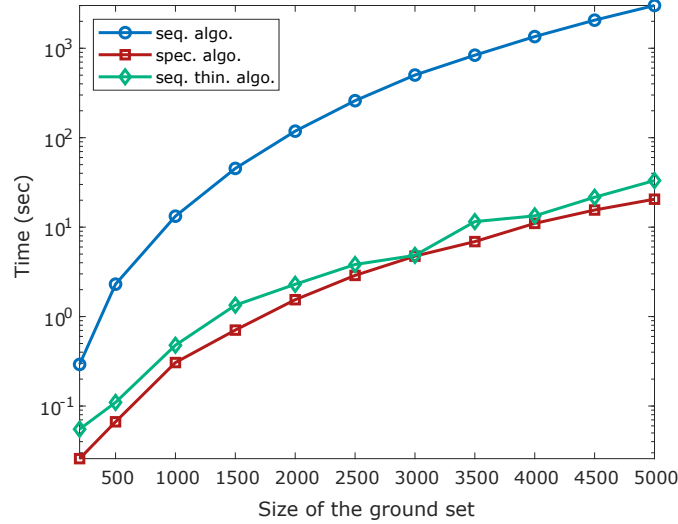


Figure 2.1: Running times of the 3 studied algorithms in function of the size of the ground set, using a patch-based kernel.

From now on, we restrict the comparison to the spectral and the sequential thinning algorithms (Algorithms 1 and 3). We present in Figure 2.2 the running times of these algorithms as a function of the size of $|\mathcal{Y}|$ in various situations. The first row shows the running times when the expectation of the number of sampled point $\mathbb{E}(|Y|)$ is equal to 4% of the size of \mathcal{Y} : it increases as the total number of points increases. In this case, we can see that whatever the chosen kernel, the spectral algorithm is faster as the complexity of sequential part of Algorithm 3 depends on the size $|X|$ that also grows. On the second row, as $|\mathcal{Y}|$ grows, $\mathbb{E}(|Y|)$ is fixed to 20. Except for the right-hand-side kernel, we are in the configuration where $|X|$ stays proportional to $|Y|$, then the Bernoulli step of Algorithm 3 is very efficient and this sequential thinning algorithm becomes competitive with the spectral algorithm. For these general kernels, we observe that the sequential thinning algorithm can be as fast as the spectral algorithm, and even faster, when the expected cardinality of the sample is small compared to the size of the ground set. The question is: when and up to which expected cardinality is Algorithm 3 faster?

Figure 2.3 displays the running times of both algorithms in function of the

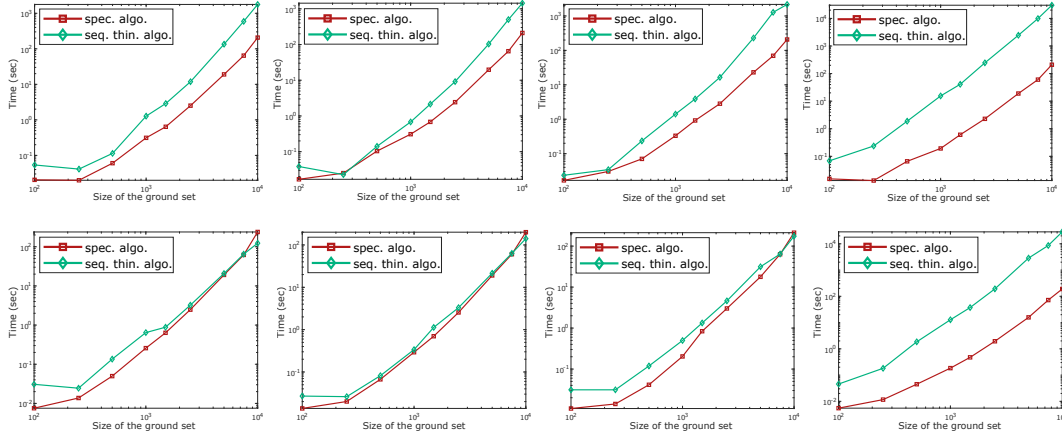


Figure 2.2: Running times in log-scale of the spectral and the sequential thinning algorithms as a function of the size of the ground set $|\mathcal{Y}|$, using “classic” DPP kernels. From left to right: a random kernel, a Ginibre-like kernel, a patch-based kernel and a projection kernel. On the first row, the expectation of the number of sampled points is set to 4% of $|\mathcal{Y}|$ and on the second row, $\mathbb{E}(|Y|)$ is constant, equal to 20.

expected cardinality of the sample when the size of the ground set is constant, equal to 5000 points. Notice that, concerning the three left-hand-side general kernels with no eigenvalue equal to one, the sequential thinning algorithm is faster under a certain expected number of points -which depends on the kernel. For instance, when the kernel is randomly defined and the range of desired points to sample is below 25, it is relevant to use this algorithm. To conclude, when the eigenvalues of the kernel are below one, Algorithm 3 seems relevant for large data sets but small samples. This case is quite common, for instance to summarize a text, to work only with representative points in clusters or to denoise an image with a patch-based method.

The projection kernel (when the eigenvalues of K are either 0 or 1) is, as expected, a complicated case. Figure 2.2 (bottom, right) shows that our algorithm is not competitive when using this kernel. Indeed, the cardinality of the dominating Bernoulli process X can be very large. In this case, the bound in Equation (2.32) isn’t valid (and even tends to infinity) as $\lambda_{\max} = 1$, and we necessarily reach the degenerated case when, after some index k , all the Bernoulli probabilities $q_l, l \geq k$, are equal to 1. Then the second part of the sequential thinning algorithm -the sequential sampling part- is done on a larger set which significantly increases the running time of our algorithm. Figure 2.3 confirms this observation as in that configuration, the sequential thinning algorithm is never the fastest.

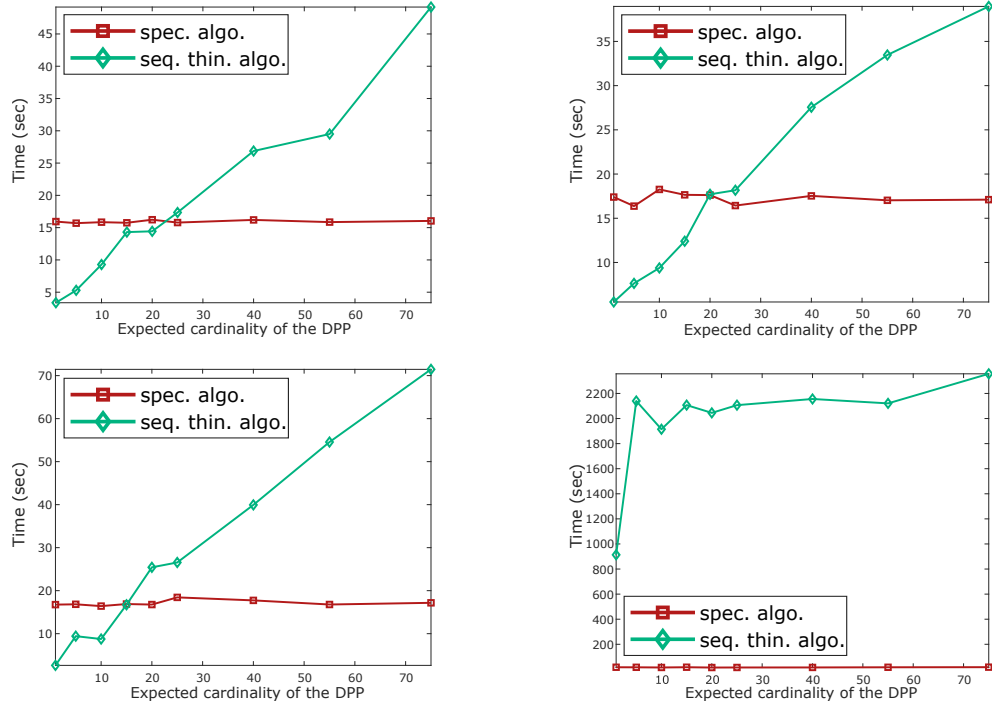


Figure 2.3: Running times of the spectral and sequential thinning algorithms in function of the expected cardinality of the process. From left to right, from top to bottom, using a random kernel, a Ginibre-like kernel, the patch-based kernel and a projection kernel. The size of the ground set is fixed to 5000 in all examples.

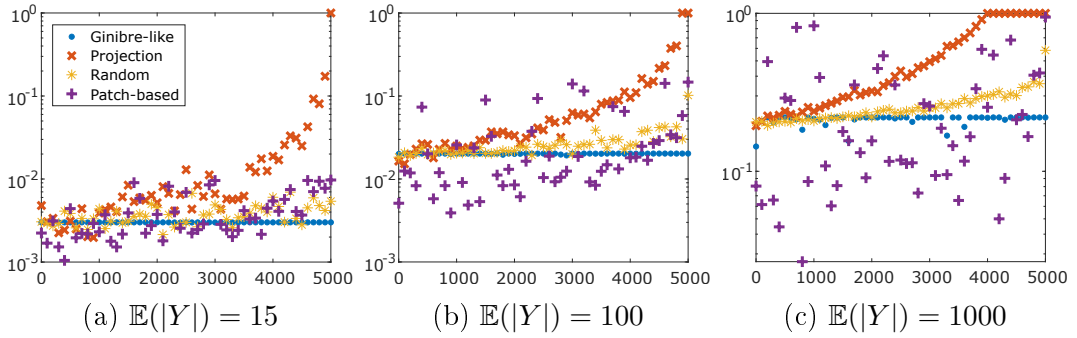


Figure 2.4: Behavior of the Bernoulli probabilities q_k , $k \in \{1, \dots, N\}$, for the kernels presented in Section 2.5.1, considering a ground set of $N = 5000$ elements and varying the expected cardinality of the DPP, $\mathbb{E}(|Y|) = 15, 100, 1000$.

Figure 2.4 illustrates how efficient the first step of Algorithm 3 can be to reduce the size of the initial set \mathcal{Y} . It displays Bernoulli probabilities $q_k, k \in \{1, \dots, N\}$ (Equation 2.31) associated to the previous kernels, for different expected cardinality $\mathbb{E}(|Y|)$. Observe that the probabilities are overall higher for a projection kernel. For such a kernel, we know that they necessarily reach the value 1, at the latest from the item $k = \mathbb{E}(|Y|)$. Indeed projection DPPs have a fixed cardinality (equal to $\mathbb{E}(|Y|)$) and q_k computes the probability to select the item k given that no other item has been selected yet. Notice that in general, considering the other kernels, the degenerated value $q_k = 1$ is rarely reached, even though in our experiments, the Bernoulli probabilities associated to the patch kernel (c) are sometimes close to one, when the expected size of the sample is $\mathbb{E}(|Y|) = 1000$. On the opposite, the Bernoulli probabilities associated to the Ginibre-like kernel remain rather close to a uniform distribution.

In order to understand more precisely to what extent high eigenvalues penalize the efficiency of the sequential thinning algorithm (Algorithm 3), Figure 2.5 compares its running times with that of the spectral algorithm (Algorithm 1) in function of the eigenvalues of the kernel K . For these experiments, we consider a ground set of size $|\mathcal{Y}| = 5000$ items and an expected cardinality equal to 15. In the first case (a), the eigenvalues are either equal to 0 or to λ_{\max} , with m non-zero eigenvalues so that $m\lambda_{\max} = 15$. It shows that above a certain λ_{\max} ($\simeq 0.65$), the sequential thinning algorithm is not the fastest anymore. In particular, when $\lambda_{\max} = 1$, the running time takes off. In the second case (b), the eigenvalues (λ_k) are randomly distributed between 0 and λ_{\max} so that $\sum_k \lambda_k = 15$. In practice, $(N - 1)$ eigenvalues are exponentially distributed, with expectation $\frac{15 - \lambda_{\max}}{N - 1}$, and the last eigenvalue is set to λ_{\max} . In this case, the sequential thinning algorithm remains faster than the spectral algorithm, even with high values of λ_{\max} , except when $\lambda_{\max} = 1$. This can be explained by the fact that, by construction of this kernel, most of the eigenvalues are very small. The average size of the Bernoulli process generated (light grey, right axes) also illustrates the influence of the eigenvalues.

Table 2.1 presents the individual weight of the main steps of the three algorithms. Concerning the sequential algorithm, logically, the matrix inversion is the heaviest part taking 74.25% of the global running time. These proportions remain the same when the expected number of points n grows. The main operation of the spectral algorithm is by far the eigendecomposition of the matrix K , counting for 83% of the global running time, when the expectation of the number of points to sample evolves with the size of \mathcal{Y} . Finally, the sequential sampling is the heaviest step of the sequential thinning algorithm. We have already mentioned that the thinning is very fast and that it produces a point process with a cardinality as close as possible to the final DPP. When the expected cardinality is low, the number of selected points by the thinning process

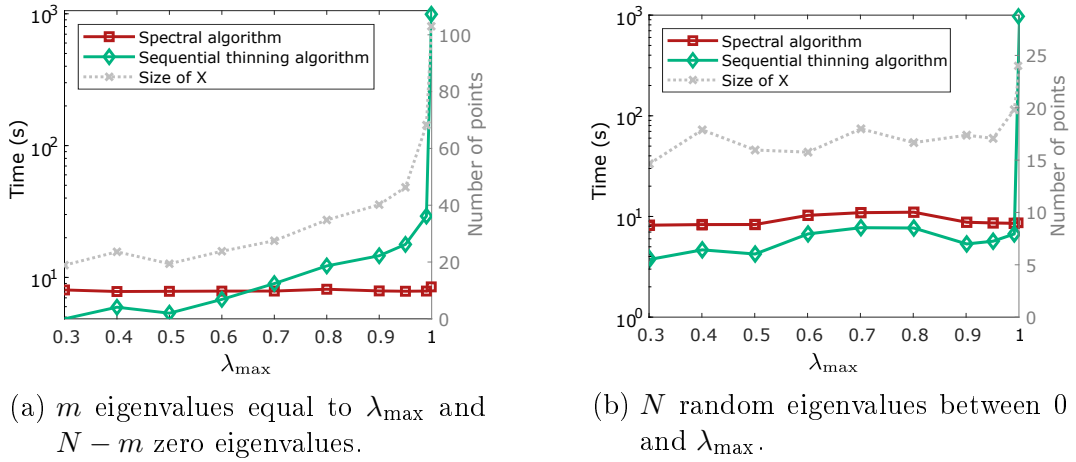


Figure 2.5: Running times of the spectral and sequential thinning algorithms (Algorithm 1 and 3) in function of λ_{\max} . The size of the Bernoulli process X is also displayed in light grey (right axis). Here, $|\mathcal{Y}| = 5000$ and $\mathbb{E}(|Y|) = 15$.

is low too, so the sequential sampling part remains bounded (86.53% when the expected cardinality $\mathbb{E}(|Y|)$ is constant). On the contrary, when $\mathbb{E}(|Y|)$ grows, the number of points selected by the dominated process rises as well so the running time of this step is growing (with a mean of 89.39%). As seen before, the global running time of the sequential thinning algorithm really depends on how good the domination is.

Algorithms	Steps	Expected cardinality	
		4% of $ \mathcal{Y} $	Constant (20)
Sequential	Matrix inversion	74.25%	72.71%
	Cholesky computation	22.96%	17.82%
Spectral	Eigendecomposition	83.34%	94.24%
	Sequential sampling	14.77%	4.95%
Sequential thinning	Preprocess to define q	10.07%	13.43%
	Sequential sampling	89.39%	86.53%

Table 2.1: Detailed running times of the sequential, spectral and sequential thinning algorithms for varying ground sets \mathcal{Y} with $|\mathcal{Y}| \in [100, 5000]$ using a patch-based kernel.

Thus, the main case when this sequential thinning algorithm (Algorithm 3) fails to compete with the spectral algorithm (Algorithm 1) is when the eigenvalues of the kernel are equal or very close to 1. This algorithm improves the sampling running times when the target size of the sample is very low

(below 25 in our experiments).

In cases when multiple samples of the same DPP have to be drawn, the eigendecomposition of K can be stored and the spectral algorithm is more efficient than ours. Indeed, in our case the computation of the Bernoulli probabilities can also be saved but the sequential sampling is the heaviest task and needs to be done for each sample.

2.6 Conclusion

In this chapter, we proposed a new sampling algorithm (Algorithm 3) adapted to general determinantal point processes, which doesn't use the spectral decomposition of the kernel and which is exact. It proceeds in two phases. The first one samples a Bernoulli process whose distribution is adapted to the target DPP. We know that the generated point process contains the DPP and it is constructed so that its size is the closest to the size of the target DPP. It is a fast and efficient step that reduces the initial number of points of the ground set. Moreover, if $I - K$ is invertible, the expectation of the cardinality of the Bernoulli process is proportional to the expectation of the cardinality of the DPP.

The second phase is a sequential sampling based on the points selected in the first step. This phase is made possible by the explicit formulations of the general marginals and the pointwise conditional probabilities of any DPP from its kernel K . The sampling is sped up using updated Cholesky decompositions to compute the conditional probabilities. This sequential strategy is not efficient, that is why it is crucial that the first step reduces the size of the initial state space as much as possible. MATLAB and Python implementations of the sequential thinning algorithm can be found online³.

In terms of running times, we have detailed the cases for which this algorithm is competitive with the spectral algorithm, in particular when the size of the ground set is high and the expected cardinality of the DPP is modest. This framework is common in machine learning applications. Indeed, DPPs are an interesting solution to subsample a data set, initialize a segmentation algorithm or summarize an image, examples where the number of datapoints needs to be significantly reduced, and where our algorithm would speed up the procedure.

As future works, we would like to investigate methods to further accelerate our algorithm. We are also interested in a potential adaptation of this strategy to continuous DPPs, defined on a continuous state space. Indeed, the thinning procedure we use comes from a continuous setting. We would like to examine

³https://claunay.github.io/exact_sampling.html

the modification of the rest of the algorithm to a continuous framework. Continuous DPPs appear in the distribution of the spectrum of Gaussian random matrices in probability or in the location of fermions in quantum mechanics, for instance. Note that sampling exactly a continuous DPPs models is a much more challenging problem than sampling discrete DPPs. The main reasons are that the domains are often infinite, and more importantly, because the eigen-decompositon of the kernel operator generally involves an infinite number of eigenvalues. Yet hope that adaptation of the sequential thinning procedure may provide an adequate sampling procedure for some continuous DPP models.

Chapter 3

Determinantal Point Processes on Pixels

Contents

3.1	Introduction	63
3.2	Determinantal Pixel Processes (DPixPs)	64
3.2.1	Notations and Definitions	65
3.2.2	Properties	67
3.2.3	Hard-core Repulsion	71
3.3	Shot Noise Models Based on DPixPs	73
3.3.1	Shot Noise Models and Micro-textures	73
3.3.2	Extreme Cases of Variance	76
3.3.3	Convergence to Gaussian Processes	78
3.4	Inference for DPixPs	82
3.4.1	Equivalence Classes of DPP and DPixP	83
3.4.2	Estimating a DPixP Kernel from One Realization	89
3.4.3	Estimating a DPixP Kernel From Several Realizations	93
3.5	Conclusion	97

3.1 Introduction

In this chapter, we consider DPPs defined on a specific space, the set of the pixels of an image. In such a framework, it seems natural to assume that the point processes under study are stationary and periodic. Thus, the correlation between pairs of pixels no longer depends on the position of the pixels but on the difference between their position. As a consequence, the kernel K is a block-circulant matrix. The kernel can be characterized using a function C

defined on the image domain, that we identify with the kernel of the DPP in the following. Circulant and block-circulant matrices have the particularity to be diagonalized by the Fourier basis. In this chapter, the eigenvalues of the matrix K are the Fourier coefficients of the function C . Thus, the discrete Fourier transform plays a key role in this chapter.

Section 3.2 introduces these discrete DPPs that we call Determinantal Pixel Processes (DPixPs). We study the consequences of the stationarity and periodicity hypotheses on basic properties of DPPs, in particular on the repulsion generated by these point processes. Gibbs point processes can generate hard-core repulsion, that is imposing a minimal distance between the points of the point process. We study the existence of a similar property for DPixPs.

In Section 3.3, we investigate shot noise models based on DPixPs and on a given spot function. These models consist in summing the spot function translated around the points of the point process. Usually based on Poisson point processes, they are fast and easy to simulate and they are used to generate micro-textures. After presenting these models based on DPixPs, we analyze the effect of the repulsion of DPPs on them. It appears that it is possible to adapt the kernel of a DPixP to the spot function g , in order to obtain particularly regular or irregular textures. This is related to an optimization problem based on the variance of the shot noise model. Usual Poisson shot noise models converge to a Gaussian texture when the intensity of the point process tends to infinity. Similarly, we prove that, in an appropriate framework, shot noise models based on any DPixP and any spot function verify a Law of Large Number and a Central Limit Theorem characterizing their convergence to a Gaussian process.

In Section 3.4, in order to investigate inference on DPixP kernels, we review the definition of equivalence classes of DPPs in different frameworks. This is a question called identifiability. A model is not identifiable if two different parametrizations produce equivalent distributions. Thus, for estimation purposes, it is crucial to characterize the equivalent kernels of a given DPP kernel. We develop an algorithm that uses the stationarity hypothesis to estimate the kernel of a DPixP from one or several samples. This method is fast and provides satisfying results.

3.2 Determinantal Pixel Processes (DPixPs)

In this section, let us present Determinantal Pixel Processes, DPPs defined on the set of pixels of an image, and the main properties of these point processes.

3.2.1 Notations and Definitions

In the following sections, we will consider DPPs defined on the pixels of an image. Let us first define any image as a function $u : \Omega \rightarrow \mathbb{R}^d$ ($d = 1$ for gray-scale images and $d = 3$ for color images), where $\Omega = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\} \subset \mathbb{Z}^2$ is a finite grid representing the image domain. The cardinality of Ω , that is the number of pixels in the image, is denoted by $N = |\Omega| = N_1 N_2$. Note that, if necessary, the pixels of an image are ordered and they are considered column by column. For any image $u : \Omega \mapsto \mathbb{R}^d$, and $y \in \mathbb{Z}^2$, the translation $\tau_y u$ of u by the vector y is defined by

$$\forall x = (x_1, x_2) \in \Omega, \tau_y u(x_1, x_2) := u(x_1 - y_1 \bmod N_1, x_2 - y_2 \bmod N_2).$$

In the following, we consider the Fourier domain $\widehat{\Omega} = \{-\frac{N_1}{2}, \dots, \frac{N_1}{2} - 1\} \times \{-\frac{N_2}{2}, \dots, \frac{N_2}{2} - 1\}$ if N_1 and N_2 are even (otherwise, for instance if N_i is odd, we consider $\{-\frac{N_i-1}{2}, \dots, \frac{N_i-1}{2}\}$), so that the frequency 0 is centered. We recall that the discrete Fourier transform of a function $f : \Omega \mapsto \mathbb{C}$ is given by, for all $\xi \in \widehat{\Omega}$,

$$\widehat{f}(\xi) = \mathcal{F}(f)(\xi) = \sum_{x \in \Omega} f(x) e^{-2i\pi \langle x, \xi \rangle}, \text{ with } \langle x, \xi \rangle = \frac{x_1 \xi_1}{N_1} + \frac{x_2 \xi_2}{N_2}. \quad (3.1)$$

This transform is inverted using the inverse discrete Fourier transform:

$$\forall x \in \Omega, f(x) = \mathcal{F}^{-1}(\widehat{f})(x) = \frac{1}{N} \sum_{\xi \in \widehat{\Omega}} \widehat{f}(\xi) e^{2i\pi \langle x, \xi \rangle}. \quad (3.2)$$

Note that given a function f defined on Ω , we consider it is extended by periodicity to \mathbb{Z}^2 . Thus, for any f defined on Ω , we set $f_-(x) := f(-x)$. The convolution of two functions f and g defined on Ω is given by

$$\forall x \in \Omega, f * g(x) = \sum_{y \in \Omega} f(x - y) g(y), \quad (3.3)$$

where the boundary conditions are considered periodic. Then, $f * g$ can be computed in the Fourier domain, since

$$\forall \xi \in \widehat{\Omega}, \widehat{f * g}(\xi) = \widehat{f}(\xi) \widehat{g}(\xi). \quad (3.4)$$

The autocorrelation of a function f is denoted by R_f . It is defined for all $x \in \Omega$ by $R_f(x) = f * f_-(x)$. Besides, the Parseval formula asserts that for any function $f : \Omega \rightarrow \mathbb{C}$,

$$\|f\|_2^2 = \sum_{x \in \Omega} |f(x)|^2 = \frac{1}{N} \sum_{\xi \in \widehat{\Omega}} |\widehat{f}(\xi)|^2 = \frac{1}{N} \|\widehat{f}\|_2^2. \quad (3.5)$$

Let us consider a DPP defined on Ω with kernel K . In this work, we will focus on the modeling of textures, which are often characterized by the repetition of a pattern, or small objects which may be indistinguishable individually. Their homogeneous aspect can be naturally modeled by a stationary random field. Thus we will suppose that the point processes under study are stationary and periodic. This hypothesis amounts to consider that the correlation between two pixels x and y only depends on the difference $x - y$: the distribution is invariant by translation, while assuming periodic boundary conditions. Thus the kernel matrix K is a block-circulant matrix with circulant blocks, entirely characterized by its first row. Note that in practice, the pixels are ordered column by column so that the ordered index of a pixel $x = (x_1, x_2) \in \Omega$ is $(x_1 - 1)N_2 + x_2$.

Definition 3.2.1. *A block-circulant matrix with circulant blocks K verifies for all $x = (x_1, x_2), y = (y_1, y_2) \in \Omega$, for all $\tau = (\tau_1, \tau_2) \in \Omega$,*

$$K(x + \tau, y + \tau) = K(x, y), \quad (3.6)$$

where we still consider periodic boundary conditions.

Let us define a correlation function $C : \Omega \rightarrow \mathbb{C}$ such that

$$K(x, y) = C(x - y), \quad \forall x, y \in \Omega. \quad (3.7)$$

Note that C is extended to \mathbb{Z}^2 by periodicity. As it entirely characterizes K , it also characterizes the associated DPP. Circulant matrices are diagonalized in the Fourier basis, thus the eigenvalues of K are the Fourier coefficients of C .

In this new framework, we can define DPPs from their correlation function C , they are now called determinantal pixel processes (DPixP). A DPixP kernel has two representations: C defined on Ω or the initial matrix K defined on $\Omega \times \Omega$ which corresponds to the block-circulant matrix with circulant blocks whose first row is C .

Definition 3.2.2 (Stationary DPixP). *Let $C : \Omega \rightarrow \mathbb{C}$ be a function defined on Ω , extended by periodicity to \mathbb{Z}^2 , such that*

$$\forall \xi \in \widehat{\Omega}, \quad \widehat{C}(\xi) \text{ is real and } 0 \leq \widehat{C}(\xi) \leq 1. \quad (3.8)$$

Such a function is called an admissible kernel. Any random subset $X \subset \Omega$ is called a (stationary) DPixP with kernel C and denoted $X \sim \text{DPixP}(C)$ if

$$\forall A \subset \Omega, \quad \mathbb{P}(A \subset X) = \det(K_A), \quad (3.9)$$

where $K_A = (C(x - y))_{x, y \in A}$ is a $|A| \times |A|$ matrix.

3.2.2 Properties

The next proposition is directly deduced from properties of general DPPs that were presented in the introduction.

Proposition 3.2.1 (Distribution of the cardinality). *The cardinality $|X|$ of a DPiXP is distributed as the sum $\sum_{\xi \in \widehat{\Omega}} B_{\xi}$, where for all $\xi \in \widehat{\Omega}$, B_{ξ} are independent*

Bernoulli random variables with parameters $\widehat{C}(\xi)$. In particular,

$$\mathbb{E}(|X|) = \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi) = NC(0) \text{ and } \text{Var}(|X|) = \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi)(1 - \widehat{C}(\xi)). \quad (3.10)$$

One can notice that it is easy to know and control the expected number of points in the point process. In the following, when comparing different DPiXP kernels, we will consider a fixed expected cardinality n , meaning that we will fix $C(0) = \frac{n}{N}$.

Proposition 3.2.2 (Separable kernel). *Let C_1 and C_2 be two discrete kernels, of dimension 1, defined respectively on $\{0, \dots, N_1 - 1\}$ and $\{0, \dots, N_2 - 1\}$, both verifying Equation (3.8) (for the 1D Fourier transform). Then the point process defined on Ω by the kernel C given by $\forall x = (x_1, x_2) \in \Omega$, $C(x) = C_1(x_1)C_2(x_2)$, is a DPiXP, that will be called separable.*

Proof. Notice that for all $\xi = (\xi_1, \xi_2) \in \widehat{\Omega}$,

$$\widehat{C}(\xi) = \sum_{x_1=0}^{N_1-1} \sum_{x_2=0}^{N_2-1} C_1(x_1)C_2(x_2)e^{-2i\pi\left(\frac{x_1\xi_1}{N_1} + \frac{x_2\xi_2}{N_2}\right)} = \widehat{C}_1(\xi_1)\widehat{C}_2(\xi_2). \quad (3.11)$$

Thus, clearly, for all $\xi \in \widehat{\Omega}$, $\widehat{C}(\xi)$ is real and $0 \leq \widehat{C}(\xi) \leq 1$. C is an admissible kernel. \square

Examples

Let us consider two fundamental examples of DPiXPs. The first one is the Bernoulli process. It corresponds to the discrete analogous of the Poisson point process: points are drawn independently and following a Bernoulli distribution of parameter $p \in [0, 1]$. This point process is the DPiXP characterized by the kernel C such that $C = p\delta_0$, or equivalently $\forall \xi \in \widehat{\Omega}$, $\widehat{C}(\xi) = p \in [0, 1]$. The second main example is the family of projection DPiXPs, that are determinantal processes defined by a kernel C which verifies for all $\xi \in \widehat{\Omega}$, $\widehat{C}(\xi)(1 - \widehat{C}(\xi)) = 0$. Thus, from Proposition 3.2.1, the number of points of projection DPiXPs is fixed and equal to the number of non-zero Fourier coefficients of C .

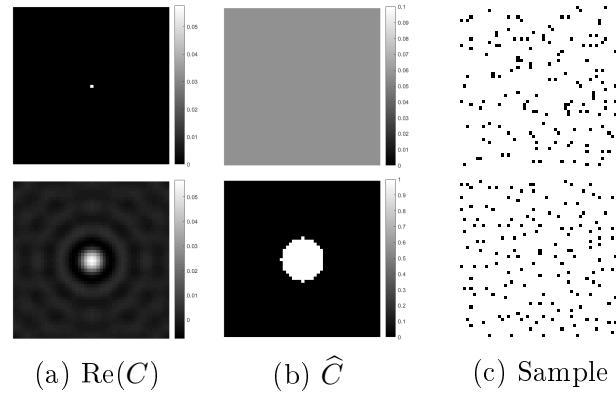


Figure 3.1: Comparison between two samples (both have 148 points) of a Bernoulli process (first line) and a projection DPixP defined by the kernel C such that \hat{C} is the indicator function of a discrete circle (second line). For both DPixPs, from left to right, the real part of the kernel function C , its Fourier coefficients \hat{C} and one associated sample.

As we have seen previously, notice that in the general discrete case, the first example corresponds to the case where K is diagonal and the second one corresponds to the case where the eigenvalues of K are either equal to 0 or to 1. It is also called a projection DPP and the cardinality of the point process is equal to the number of non-zero eigenvalues, i.e. the rank of K .

Figure 3.1 presents two samples of these particular cases. Clearly, the projection DPixP enables a more “regular” distribution of the points in the square, tends to avoid regions with holes and regions with clusters.

Sampling from DPixPs

The common algorithm to sample exactly general determinantal processes is the spectral algorithm, presented in Section 2.2.1. Remember that this is a two steps strategy which relies on an eigendecomposition $\{(\lambda_x, v_x)\}_{1 \leq x \leq N}$ of the matrix K . Indeed, define $(B_x)_{1 \leq x \leq N}$, N independent random variables such that $B_x \sim \mathcal{Ber}(\lambda_x)$ and $K_B = \sum_{x \in \Omega} B_x v_x v_x^*$. Such a matrix K_B is a random version of K and Hough and al. [73] proved that $\text{DPP}(K) = \text{DPP}(K_B)$. Hence, the spectral algorithm consists in first drawing N independent Bernoulli random variables of parameters λ_x : these variables select n eigenvalues and eigenvectors, where n is distributed as $\sum_{1 \leq x \leq N} B_x$. Then, it samples the n points from a projection DPP, obtained from the selected eigenvectors, thanks to a

Gram-Schmidt procedure.

In our discrete stationary periodic framework, the eigenvalues of the matrix K are the Fourier coefficients of C and its eigenvectors are the elements of the Fourier basis. Then an eigendecomposition of a DPixP of kernel C is computed using the 2D Fast Fourier Transform (FFT2) algorithm. Algorithm 4 presents the spectral algorithm adapted to sample a DPixP. In this algorithm, $(\varphi_\xi)_{\xi \in \widehat{\Omega}}$ denotes the columns of the discrete Fourier transform matrix:

$$\forall \xi \in \widehat{\Omega}, \forall x \in \Omega, \varphi_\xi(x) = e^{-2i\pi \langle x, \xi \rangle}. \quad (3.12)$$

Algorithm 4 Spectral simulation of $X \sim \text{DPixP}(C)$

- Sample a random field $U = (U_\xi)_{\xi \in \widehat{\Omega}}$ where the U_ξ are i.i.d. uniform on $[0, 1]$.
- Define the “active frequencies” $\{\xi_1, \dots, \xi_n\} = \{\xi \in \widehat{\Omega}; U(\xi) \leq \widehat{C}(\xi)\}$, and denote

$$\forall x \in \Omega, v(x) = (\varphi_{\xi_1}(x), \dots, \varphi_{\xi_n}(x)) \in \mathbb{C}^n. \quad (3.13)$$

- Sample X_1 uniform on Ω , and define $e_1 = v(X_1)/\|v(X_1)\|$.
- For $k = 2$ to n do:

– Sample X_k from the probability density p_k on Ω , defined by

$$\forall x \in \Omega, p_k(x) = \frac{1}{n - k + 1} \left(\frac{n}{N} - \sum_{j=1}^{k-1} |e_j^* v(x)|^2 \right) \quad (3.14)$$

– 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 $X = (X_1, \dots, X_n)$.
-

Because of the eigendecomposition of a matrix of size $|\Omega| \times |\Omega|$ the initial spectral algorithm runs in $\mathcal{O}(|\Omega|^3)$, yet thanks to the FFT2 algorithm, sampling DPixPs costs $\mathcal{O}(|\Omega| \log |\Omega|)$. Whereas in general the spectral algorithm is heavy when dealing with a huge data set, in this setting, it is very efficient. This allows us to handle large images. Thus, in addition to the explicit computation of marginals and of moments of a DPixP from its kernel, this exact sampler is one more asset of this family of point processes with respect to Gibbs processes.

Figure 3.2 presents the sampling of a projection DPixP. The Fourier coefficients of the kernel function are in $\{0, 1\}$, and the non-zero Fourier coefficients

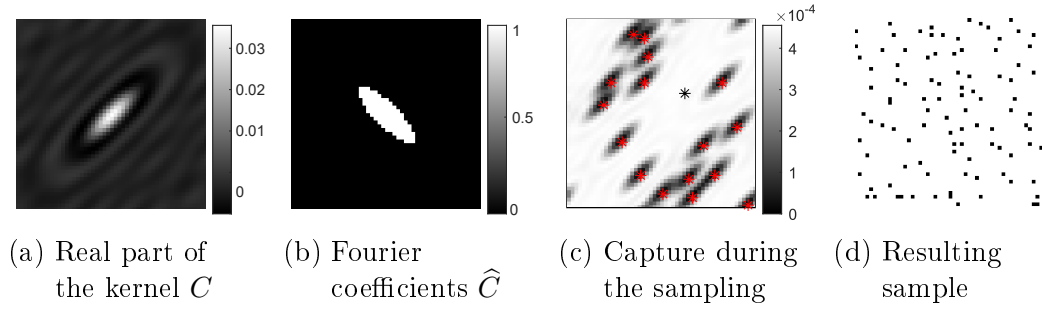


Figure 3.2: Sampling of a projection DPixP. From left to right, the real part of the kernel C , the Fourier coefficients of C , a capture of the conditional density during the simulation, the generated sample.

are shaped like a truncated anisotropic Gaussian distribution. Figure 3.2(c) shows a capture taken during the k -th iteration of the sequential step of the spectral algorithm. The red asterisks symbolize the $k - 1$ pixels already selected. The grey scale represents the values of the probability for each pixel to be selected next given the pixels already selected. The black asterisk symbolizes the k -th selected pixel. Observe as a “repulsion zone” is created around every selected pixel. This zone, where the conditional probability to select a new pixel is very low, reproduces exactly the shape of the kernel C . Thus, in the end, the pixels of the sample respect the repulsion imposed by the kernel.

Pair Correlation Function

In spatial statistics, the pair-correlation function (p.c.f.) g_X associated to a point process X is used to describe interactions between pairs of points. It characterizes the local repulsiveness of X [20]. For any discrete stationary point process on Ω , it is defined for all $x \in \Omega$ by

$$g_X(x) = \frac{\mathbb{P}(\{0, x\} \subset X)}{\rho^2}, \quad (3.15)$$

where ρ is the intensity of the point process, $\rho = \frac{\mathbb{E}(|X|)}{|\Omega|} = \mathbb{P}(0 \in X)$. It quantifies the degree of interaction between two points separated by a gap x : the closest g is to 1, the less correlated they are. If $g(x) > 1$, the points are considered to attract each other, whereas if $g(x) < 1$ the points are considered to repel each other. Notice that if $X \sim \text{DPixP}(C)$,

$$g_X(x) = \frac{C(0)^2 - |C(x)|^2}{C(0)^2} = 1 - \frac{|C(x)|^2}{|C(0)|^2}. \quad (3.16)$$

Thus, if X is a Bernoulli point process, for all $x \neq 0$, $g_X(x) = 1$: there is no interaction between the points. Note also that for any DPixP, $g_X \leq 1$.

During the sequential step of the sampling, each time a pixel is selected, a “repulsion zone” appears around it, where the probability for a pixel to be selected is low and whose shape depends on the kernel function C (Figure 3.2). This local “repulsion zone” is clearly retrieved in the pair correlation function computation.

3.2.3 Hard-core Repulsion

Gibbs processes are often used as their definition enables to precisely characterize the repulsion. Besides, they can provide hard-core repulsion, meaning that the points are prohibited from being closer than a certain distance. To compare with this family of point processes, we investigate the possibility of hard-core repulsion in the case of DPiXPs. First, we study a hard-core repulsion for pairs of points. Specifically, if $x \in \Omega$ and $e \in \Omega$ (for instance $e = (1, 0)$ or $(0, 1)$), is there a DPiXP kernel such that x and $x + e$ can't belong simultaneously to the sample? The following proposition answers to this question and characterizes the associated kernel.

Proposition 3.2.3. *Let us consider $X \sim \text{DPiXP}(C)$ 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 $\xi \in \widehat{\Omega}$ such that $\widehat{C}(\xi)$ is non-zero 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.

Proof. Let X be a DPiXP defined on Ω with kernel C . First, let us prove that $1 \Leftrightarrow 3$. Recall that for all $x \in \Omega$, $\mathbb{P}(\{x, x + e\} \subset X) = C(0)^2 - |C(e)|^2$. We deduce from the triangle inequality that

$$|C(e)| = \left| \frac{1}{|\Omega|} \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi) e^{2i\pi \langle e, \xi \rangle} \right| \leq \frac{1}{|\Omega|} \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi) = C(0), \quad (3.17)$$

and the equality holds if and only if all non-zero elements of the left-hand side sum have equal argument. Thus, $\mathbb{P}(\{x, x + e\} \subset X) = 0$ if and only if there

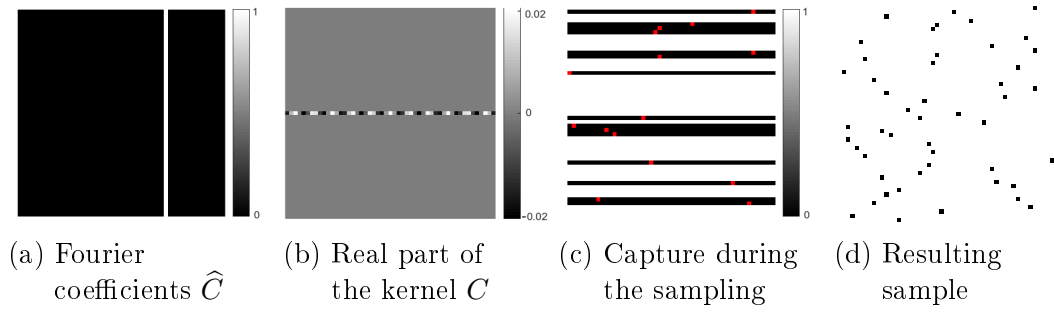


Figure 3.3: Example of a kernel associated with hard-core repulsion in the horizontal direction. From left to right, the Fourier coefficients of C , the real part of the kernel C , a capture of the conditional density during the simulation, the associated final sample.

exists $\theta \in \mathbb{R}$ such that for all $\xi \in \hat{\Omega}$, either $\hat{C}(\xi) = 0$, or $\langle e, \xi \rangle = \theta$. Hence, for all $x \in \Omega$, the probability that x and $x + e$ belong simultaneously to X is zero if and only if the only non-zeros Fourier coefficients of C are aligned in the orthogonal direction of e . Second, let us prove that $2 \Leftrightarrow 3$. Consider $\lambda \in \mathbb{Q}$ such that $\lambda e \in \Omega$. Similarly, $\mathbb{P}(\{x, x + \lambda e\} \subset X) = 0$ if and only if there exists $\theta \in \mathbb{R}$ such that for all $\xi \in \hat{\Omega}$, either $\hat{C}(\xi) = 0$, or $\langle \lambda e, \xi \rangle = \theta$, meaning that $\langle e, \xi \rangle = \frac{\theta}{\lambda}$, which also is the equation of a discrete line orthogonal to e . Finally, suppose that X contains almost surely at most one point on every discrete line of direction e . Then, for all $x \in \Omega$, the probability that x and $x + e$ belong to X is zero so $4 \Rightarrow 1 \Leftrightarrow 3$. Now assume that the only non-zero Fourier coefficients of C are aligned on a discrete line that is orthogonal to e . As $2 \Leftrightarrow 3$ for all $\lambda \in \mathbb{Q}$ such that $\lambda e \in \Omega$, $\mathbb{P}(\{x, x + \lambda e\} \subset X) = 0$. Hence, X contains at most one point on any line of direction e , which can be described as a hard-core repulsion of direction e . \square

Figure 3.3 illustrates this proposition: all non-zero Fourier coefficients are vertically aligned. The third figure presents a capture of the conditional density while the simulation is in progress, after 15 pixels already sampled. In each pixel, the probability that it is the next point selected is represented by the gray scale: the lighter a pixel is, the greater its probability of being the next point sampled. One can see that as soon as a pixel x is sampled, all the pixels belonging to the horizontal line passing through x have a zero probability of being sampled next.

Proposition 3.2.4. *Let $X \sim \text{DPixP}(C)$ verifying the properties of Proposition 3.2.3, with $e = (1, 0)$, meaning that X contains at most one point on any horizontal line and all non-zero Fourier coefficients of C are aligned on a vertical line. Then C is separable in the sense of Proposition 3.2.2. Besides, the associated vertical point process is a DPixP of dimension 1 and conditionally*

to the drawn ordinates, the associated horizontal point process consists of a single point chosen uniformly and independently from the other horizontal point processes. The same proposition holds for $e = (0, 1)$ and vertical hard-core repulsion (inverting the terms horizontal and vertical).

Proof. Consider an admissible DPiXP kernel C such that all its Fourier coefficients are either zero either aligned on a vertical line, positioned in $c \in \{-\frac{N_1}{2}, \dots, \frac{N_1}{2} - 1\}$ (here we assume that N_1 is even, the proof is similar if N_1 is odd). Thus we can define two functions $\hat{C}_1 = \mathbb{1}_c$ and $\hat{C}_2 = \hat{C}(c, \cdot)$ such that for all $\xi = (\xi_1, \xi_2) \in \hat{\Omega}$, $\hat{C}(\xi) = \hat{C}_1(\xi_1)\hat{C}_2(\xi_2) = \hat{C}_2(\xi_2)\mathbb{1}_c(\xi_1)$. Notice that $C = \mathcal{F}^{-1}(\hat{C}_1)\mathcal{F}^{-1}(\hat{C}_2) = C_1C_2$. Such a function C_1 corresponds to an admissible DPiXP projection kernel defined in one dimension, drawing one point and remember that the first point of a DPiXP is drawn uniformly. Furthermore, C is a separable kernel. \square

Note that as soon as a pair of points configuration is prohibited, the whole direction is prohibited. As imposing a minimum distance between points is equivalent to prohibiting pair of points configurations in all directions, we deduce that the only DPiXP imposing a minimum distance between the points is the degenerate DPiXP, consisting of a single pixel. Hence, we obtain the following proposition.

Proposition 3.2.5. *Let Ω be an image domain. There is no DPiXP kernel defined on Ω that generates a point process with hard core repulsion in the broad sense, except a degenerate DPiXP containing only one point.*

This property weakens the appeal of DPiXPs compared to Gibbs processes. Indeed, as we have seen before, hard core repulsion is a property appreciated by the computer graphics community and that Gibbs processes can introduce.

3.3 Shot Noise Models Based on DPiXPs

3.3.1 Shot Noise Models and Micro-textures

In the following section, we study discrete shot noise models driven by a DPiXP. Shot noise models naturally appear to model phenomena such as the superposition of impulses occurring at independent and random times or positions. These models have been introduced in the computer graphics field with the work of van Wijk [130]. Notice that van Wijk uses the expression spot noise texture as the spatial counterpart of 1D shot noise models yet the term shot noise is commonly employed for general models. Thus, in the rest of the section, we use this more general expression. Shot noise models are frequently used to approximate Gaussian textures as they are well-defined and simple

mathematical models that allows us for fast synthesis [82], [49], [51]. Here, we are interested in the discrete version of these models on the finite grid $\Omega = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\} \subset \mathbb{Z}^2$.

Definition 3.3.1 (Shot noise models based on a discrete point process). *Consider X a discrete point process with intensity ρ and g a (deterministic) function defined on Ω , periodically extended to \mathbb{Z}^2 . Then, the shot noise random field S based on the points X and the spot g is defined by*

$$\forall x \in \Omega, S(x) = \sum_{x_i \in X} g(x - x_i). \quad (3.18)$$

In general, discrete shot noise models are based on a set of n i.i.d. random variables: it amounts to summing n randomly shifted versions of the spot. These models are particularly interesting for Gaussian texture synthesis as they have a Gaussian limit [48]. Indeed, in that case, the shot noise is the sum of n i.i.d. random images so that thanks to the Central Limit Theorem, we obtain a Gaussian limit. We study here shot noise models based on DPixPs. At the end of the section, we prove that there is a similar Central Limit theorem for shot noise models based on DPixPs that needs a modified framework but that ensures a Gaussian limit.

From now on, we consider an admissible kernel C and we suppose that X is the DPixP of kernel C . We study the interactions between the kernel C and the spot function g . To compute the moments of a shot noise model S based on X and a given spot, we need a moment formula ([101], [9]), also known as the Campbell or Slivnyak-Mecke formula, adapted to our discrete setting in the following proposition.

Proposition 3.3.1 (Moments formula for DPixPs). *Let X be a DPixP of kernel C defined on Ω , let us consider $k \geq 1$ an integer and f a function defined on Ω^k . We have*

$$\mathbb{E} \left(\sum_{x_1, \dots, x_k \in X}^{\neq} f(x_1, \dots, x_k) \right) = \sum_{y_1, \dots, y_k \in \Omega} f(y_1, \dots, y_k) \det((C(y_i - y_j))_{1 \leq i, j \leq k}), \quad (3.19)$$

where $\sum_{x_1, \dots, x_k \in X}^{\neq}$ means that the (x_i) are all different. In particular, for $k = 1$,

$$\text{we have } \mathbb{E} \left(\sum_{x \in X} f(x) \right) = C(0) \sum_{y \in \Omega} f(y).$$

Proof. By definition of the DPixP of kernel C , for any y_1, \dots, y_k in Ω , we have

$$\mathbb{P}(\{y_1, \dots, y_k\} \subset X) = \det((C(y_i - y_j))_{1 \leq i, j \leq k}). \quad (3.20)$$

Therefore, by the Slivnyak-Mecke formula [9], as we have

$$\mathbb{E} \left(\sum_{x_{i_1}, \dots, x_{i_k} \in X}^{\neq} f(x_{i_1}, \dots, x_{i_k}) \right) = \sum_{y_1, \dots, y_k \in \Omega} f(y_1, \dots, y_k) \mathbb{P}(\{y_1, \dots, y_k\} \subset X), \quad (3.21)$$

we obtain the formula of the proposition. \square

Since $X \sim \text{DPiXP}(C)$ is stationary, S as defined in 3.3.1 is also stationary, so that $\mathbb{E}(S(x)^k) = \mathbb{E}(S(0)^k)$ for all $x \in \Omega$ and for all $k \geq 1$.

Proposition 3.3.2 (First and second order moments). *Let S be a shot noise model based on $X \sim \text{DPiXP}(C)$ and the spot g . We have $\mathbb{E}(S(0)) = C(0) \sum_{y \in \Omega} g(y)$, and for all $x \in \Omega$, $\Gamma_S(x) := \text{Cov}(S(0), S(x)) = C(0)R_g(x) - (R_g * |C|^2)(x)$. In particular,*

$$\text{Var}(S(0)) = C(0) \sum_{y \in \Omega} g(y)^2 - (R_g * |C|^2)(0), \quad (3.22)$$

and for all $\xi \in \widehat{\Omega}$, $\widehat{\Gamma}_S(\xi) = |\widehat{g}(\xi)|^2(C(0) - |\widehat{C}|^2(\xi))$, where $R_g = g * g_-$ is the autocorrelation of g .

Proof. First, let us compute the mean value of such a shot noise model S . Using the periodicity of g ,

$$\mathbb{E}(S(0)) = \mathbb{E} \left(\sum_{x \in X} g(-x) \right) = \sum_{y \in \Omega} g(-y)C(0) = C(0) \sum_{y \in \Omega} g(y). \quad (3.23)$$

Second, let us compute the covariance function of S for all $x \in \Omega$,

$$\begin{aligned} \Gamma_S(x) &= \text{Cov}(S(0), S(x)) = \mathbb{E}((S(0)S(x)) - \mathbb{E}(S(0))^2) \\ &= \mathbb{E} \left(\sum_{x_1 \in X} g(-x_1) \sum_{x_2 \in X} g(x - x_2) \right) - \mathbb{E}(S(0))^2 \\ &= \mathbb{E} \left(\sum_{x_1, x_2 \in X}^{\neq} g(-x_1)g(x - x_2) \right) + \mathbb{E} \left(\sum_{x_1 \in X} g(-x_1)g(x - x_1) \right) - \mathbb{E}(S(0))^2 \\ &= \sum_{y_1, y_2 \in \Omega} g(-y_1)g(x - y_2) (C(0)^2 - |C(y_2 - y_1)|^2) + \sum_{y \in \Omega} g(-y)g(x - y)C(0) \\ &\quad - \mathbb{E}(S(0))^2 \\ &= C(0)g * g_-(x) - (g * g_- * |C|^2)(x). \end{aligned} \quad (3.24)$$

\square

3.3.2 Extreme Cases of Variance

We set $N = |\Omega| = N_1 N_2 \in \mathbb{N}$ and \mathcal{C}_n the set of admissible kernels such that $C(0) = \frac{n}{N}$, where $n \in \mathbb{N}$. If $X \sim \text{DPixP}(C)$, with $C \in \mathcal{C}_n$, notice that $\mathbb{E}(|X|) = |\Omega|C(0) = n$. Given a spot function g , we are looking for admissible kernels $C \in \mathcal{C}_n$ that generate shot noise models S of maximal and minimal variance. Indeed, the value $\text{Var}(S(0))$ quantifies a repulsion “in the sense of g ” or the regularity of the shot noise. The case of a shot noise S based on a spot function g defined as an indicator function provides some intuition into this idea. If $\text{Var}(S(0))$ is low, the values taken by S are close to its mean value: there are few regions with no spot and few regions with many overlaps of the spot. Then the points sampled from $\text{DPixP}(C)$ tend to be far from one another, according to the shape of the function g and S appears more homogeneous. The repulsion is maximal. On the contrary, when $\text{Var}(S(0))$ is high, S may take high values, so there can be many points in the same region. In that case, the repulsion is minimal.

Proposition 3.3.3 (Extreme cases of variance). *Fix $g : \Omega \rightarrow \mathbb{R}^+$ and $n \in \mathbb{N}$. The variance of the shot noise model S is maximal if it is based on the Bernoulli DPixP that belongs to \mathcal{C}_n , meaning that its kernel C is such that $C(0) = \frac{n}{N}$ and for all $x \neq 0$, $C(x) = 0$.*

The variance of the shot noise model S is minimal when it is based on the projection DPixP of n points, such that the n frequencies $\{\xi_1, \dots, \xi_n\}$ associated with the non-zero Fourier coefficients of its kernel maximize

$$\sum_{\xi, \xi' \in \{\xi_1, \dots, \xi_n\}} |\widehat{g}(\xi - \xi')|^2. \quad (3.25)$$

Proof. Given a fixed $n \in \mathbb{N}$, let us consider $C \in \mathcal{C}_n$ that maximizes or minimizes

$$\begin{aligned} \text{Var}(S(0)) &= C(0)g * g_-(0) - (g * g_- * |C|^2)(0) \\ &= \frac{n}{|\Omega|^2} \sum_{\xi} |\widehat{g}(\xi)|^2 - \frac{1}{|\Omega|^2} \sum_{\xi, \xi'} |\widehat{g}(\xi - \xi')|^2 \widehat{C}(\xi) \widehat{C}(\xi'). \end{aligned} \quad (3.26)$$

If we identify the function \widehat{C} to a vector of \mathbb{R}^N , the question becomes finding $C \in \mathcal{C}_n$ that maximizes or minimizes $F : \mathbb{R}^N \rightarrow \mathbb{R}$, where

$$F(\widehat{C}) = \sum_{\xi, \xi'} |\widehat{g}(\xi - \xi')|^2 \widehat{C}(\xi) \widehat{C}(\xi'). \quad (3.27)$$

Maximal variance: We define a scalar product associated to g for all $v, w \in \mathbb{R}^N$, by $\langle v, w \rangle_g = \sum_{\xi, \xi' \in \Omega} |\widehat{g}(\xi - \xi')|^2 v_{\xi} w_{\xi'} = v^t G w$ where G is the $N \times N$ matrix such that $G = (|\widehat{g}(\xi - \xi')|^2)_{\xi, \xi' \in \Omega}$. This scalar product is well defined as it is bilinear,

symmetric and for all $v \in \mathbb{R}^N$, $\sum_{\xi, \xi'=1}^N |\widehat{g}(\xi - \xi')|^2 v_\xi v_{\xi'} = (g * g_- * |\widehat{v}|^2)(0) \geq 0$ and $\langle v, v \rangle_g = 0 \Leftrightarrow v = \mathbf{0}$. Notice that since G is symmetric positive definite then $F : \widehat{C} \mapsto \langle \widehat{C}, \widehat{C} \rangle_g$ is strictly convex. The case of maximal variance is achieved for the vector \widehat{C} that minimizes this strictly convex function on the convex set \mathcal{C}_n : the problem has at most one solution [24].

According to the Cauchy-Schwarz inequality, we have for all $v, w \in \mathbb{R}^N$, $|\langle v, w \rangle_g| \leq \|v\|_g \|w\|_g$. Let us pick $v = \widehat{C}$, the vector whose components are the Fourier coefficients of a kernel $C \in \mathcal{C}_n$ and $w = \mathbf{1}$ ($= (1, 1, \dots, 1)$ the constant vector of size N). We have $\|v\|_g^2 = F(\widehat{C})$ and $\|w\|_g^2 = \sum_{\xi, \xi'} |\widehat{g}(\xi - \xi')|^2 = \sum_{\xi, \xi'} \widehat{g * g_-}(\xi - \xi') = N^2 (g * g_-)(0)$. Hence $\|v\|_g \|w\|_g = \sqrt{N^2 F(\widehat{C}) (g * g_-)(0)}$ and

$$|\langle v, w \rangle_g| = \sum_{\xi, \xi'} |\widehat{g}(\xi - \xi')|^2 \widehat{C}(\xi) = \sum_{\xi} \widehat{C}(\xi) \sum_{\xi'} |\widehat{g}(\xi - \xi')|^2 = n N (g * g_-)(0). \quad (3.28)$$

Thus, $F(\widehat{C}) \geq n^2 (g * g_-)(0)$ and $F(\widehat{C})$ is minimal if and only if \widehat{C} is proportional to w : necessarily, for all $\xi \in \widehat{\Omega}$, $\widehat{C}(\xi) = \frac{n}{N}$. Hence, C is a Bernoulli process. This kernel maximizes the variance of any shot noise S , independently of the spot g . It is the least repulsive DPiXP.

Minimal variance: Let us characterize the kernel C that maximizes the function F on the convex set \mathcal{C}_n . F is quadratic so that solutions are on the boundaries of \mathcal{C}_n , meaning that for all kernel $\widehat{C}^* \in \widehat{\mathcal{C}}_F^* := \{\arg\max_{\widehat{C}} (F(\widehat{C}))\}$,

$\sum_{\xi} \widehat{C}^*(\xi) = n$ and $\forall \xi \in \widehat{\Omega}$, $\widehat{C}^*(\xi)(1 - \widehat{C}^*(\xi)) = 0$. Thus, the solutions are the projection DPiXP kernels C^* with exactly n frequencies $\{\xi_1, \dots, \xi_n\} \subset \widehat{\Omega}$ such that $\widehat{C}^*(\xi_i) = 1$ chosen so that $\sum_{\xi, \xi' \in \{\xi_1, \dots, \xi_n\}} |\widehat{g}(\xi - \xi')|^2$ is maximal. \square

In the end, to determine the kernel with minimal variance, one needs to maximize a quadratic function, which is NP-hard in general. In practice, it amounts to solve a combinatorial problem. It is possible to approximate the solution thanks to a glutton algorithm: first, one chooses two frequencies ξ_1, ξ_2 maximizing $|\widehat{g}(\xi_1 - \xi_2)|^2$ then, recursively, one chooses the k th frequency ξ_k , $2 < k \leq N$, such that it maximizes $\sum_{\xi \in \{\xi_1, \dots, \xi_{k-1}\}} |\widehat{g}(\xi - \xi_k)|^2$.

Figure 3.4 presents some results of this algorithm. This figure shows that a projection DPiXP adapted to g generates shot noise models with very few spot superpositions. Recall that in Section 3.2, we proved that it was impossible to

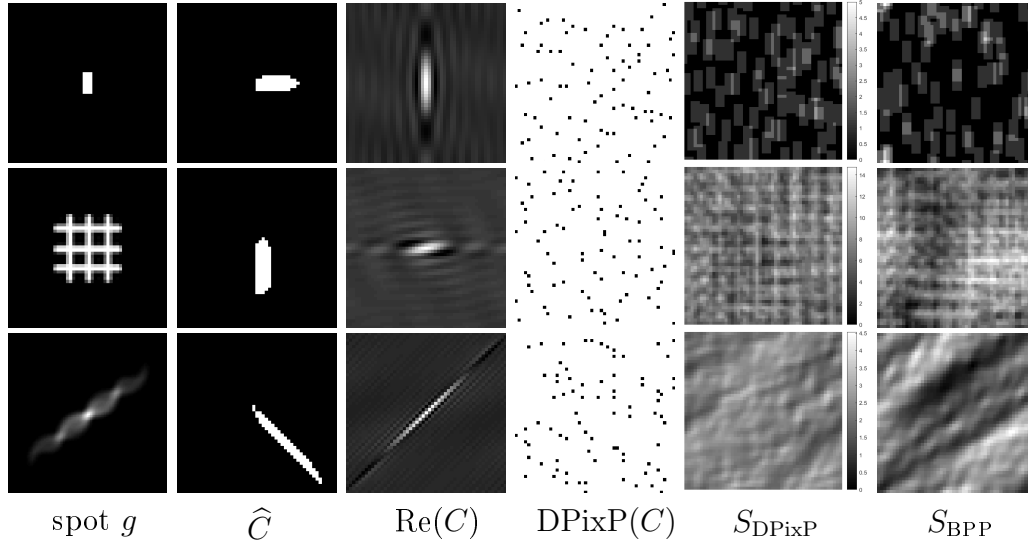


Figure 3.4: Realizations of the shot noise model driven by several spot functions and the most repulsive DPixP adapted to this spot. From left to right: the spot function, the Fourier coefficients obtained by our glutton algorithm, the real part of the associated kernel C , a sample of this most repulsive DPixP, a sample of the associated shot noise model and finally a Bernoulli shot noise model, both having the same expected number of points ($n = 80$).

completely prevent superpositions. Yet, it is possible to characterize the least and the most repulsive DPixPs according to a specific desired repulsion. These extreme cases are coherent with the results of Biscio and Lavancier [20] who quantified the repulsion of stationary DPPs defined on \mathbb{R}^d and stated that the least repulsive DPP is the Poisson point process whereas the most repulsive family of DPP contains the kernels C such that their Fourier transform $\mathcal{F}(C)$ is the indicator function of a Borel set, an analog to the projection DPixPs defined here.

3.3.3 Convergence to Gaussian Processes

Shot noise models driven by DPixP enable more diverse types of textures than the usual shot noise models, based on points drawn uniformly and independently. It takes into account this model based on Bernoulli processes yet it is important to notice that unlike usual discrete shot noise models, as defined in [48] for instance, here point processes are simple: the points can't coincide.

As with usual shot noise models based on discrete Poisson processes, it is appealing to study the behavior of the model when the density of the point

process increases and tends to infinity. Yet, as the points of the determinantal point process can't coincide, the framework needs to be adapted: if the intensity tends to infinity, we also need the size of Ω to tend to infinity. It is similar to consider Ω as a grid in $[0, 1]^2 = \mathbb{T}^2$, the torus of dimension 2, that is refined. The points are allowed to be increasingly close and the number of points inside $[0, 1]^2$ tends to infinity. In this configuration, it is possible to characterize asymptotic behaviors of these models and to derive limit theorems such as a Law of Large Numbers or a Central Limit Theorem. To this end, let us consider stationary determinantal point processes on \mathbb{Z}^2 [119], [95], that we will also call determinantal pixel processes. This point process is defined by a discrete bounded operator K on $\ell^2(\mathbb{Z}^2)$. That means that $K : \ell^2(\mathbb{Z}^2) \rightarrow \ell^2(\mathbb{Z}^2), f \mapsto Kf$ such that $\forall t \in \mathbb{Z}^2, Kf(t) = \sum_{s \in \mathbb{Z}^2} K(t, s)f(s)$. We suppose that this DPP is stationary: we define a kernel function $C : \mathbb{Z}^2 \rightarrow \mathbb{C}$, such that $K(t, s) = C(s - t)$ and $C \in \ell^2(\mathbb{Z}^2)$. Then for all $t \in \mathbb{Z}^2, Kf(t) = \sum_{s \in \mathbb{Z}^2} C(s - t)f(s)$: such a K is a convolution operator.

As C belongs to $\ell^2(\mathbb{Z}^2)$, there exists a function $\hat{C} \in L^2(\mathbb{T}^2)$ such that $\hat{C} : \mathbb{T}^2 \mapsto [0, 1], \forall t \in \mathbb{Z}^2, C(t) = \int_{\mathbb{T}^2} \hat{C}(x)e^{2i\pi t \cdot x} dx$ and $\hat{C} = \sum_{t \in \mathbb{Z}^2} C(t)e^{-2i\pi t \cdot \cdot}$ in the sense of $L^2(\mathbb{T}^2)$. Finally, the point process $X \sim \text{DPiXP}(C)$ is defined by $\forall A \subset \mathbb{Z}^2$, a finite subset,

$$\mathbb{P}(A \subset X) = \det(C_A), \text{ where } C_A = (C(x_i - x_j))_{x_i, x_j \in A}. \quad (3.29)$$

This new definition of DPiXPs on \mathbb{Z}^2 is simply an extension of the point process defined on Ω . The main properties of DPiXPs are preserved and it allows us to study the asymptotic behavior of shot noise models driven by DPiXPs, when the grid is refined or equivalently when the support of the spot is spread out. To do so, we need to consider spot functions defined on \mathbb{R}^2 .

Limit Theorems and DPiXPs

The following limit theorems are based on the works of Shirai and Takahashi [121], and Soshnikov [122]. Some guidelines for the proofs can be found in [121] for the \mathbb{Z}^2 case and in [119] and [120] for its continuous counterpart.

Proposition 3.3.4 (Limit theorems for DPiXPs [121]). *Let f be a bounded measurable function on \mathbb{R}^2 with compact support, and $X \sim \text{DPiXP}(C)$ with C some admissible kernel on \mathbb{Z}^2 . Then, we have the following Law of Large Numbers*

$$\frac{1}{N^2} \sum_{x \in X} f\left(\frac{x}{N}\right) \xrightarrow{N \rightarrow \infty} C(0) \int_{\mathbb{R}^2} f(x) dx, \text{ a.s and in } L^1. \quad (3.30)$$

Moreover, assume that f is continuous and $\int_{\mathbb{R}^2} f(x)dx = 0$. Then,

$$\lim_{N \rightarrow \infty} \mathbb{E} \left(\exp \left(\frac{i}{\sqrt{N^2}} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) \right) = \exp \left(-\frac{1}{2} \sigma(C)^2 \|f\|_2^2 \right) \quad (3.31)$$

where $\sigma(C)^2 = C(0) - \sum_{x \in \mathbb{Z}^2} |C(x)|^2$, and consequently, we obtain the following Central Limit Theorem

$$\frac{1}{\sqrt{N^2}} \sum_{x \in X} f \left(\frac{x}{N} \right) \xrightarrow[N \rightarrow \infty]{D} \mathcal{N}(0, \sigma(C)^2 \|f\|_2^2). \quad (3.32)$$

Appendices B.2 and B.3 provide a detailed proof of the previous proposition, specific to our image framework, using ergodic theory.

Convergence of Determinantal Shot Noise Models

In the following, let g be a spot function, that we assume continuous, with compact support, and $N > 0$. Denote the N -normalized shot noise S_N associated to g defined for all $y \in \mathbb{Z}^2$ by $S_N(y) = \frac{1}{N^2} \sum_{x \in X} g \left(y - \frac{x}{N} \right)$. We obtain a Law of Large Numbers for the shot noise driven by DPixPs:

$$S_N(0) = \frac{1}{N^2} \sum_{x \in X} g \left(-\frac{x}{N} \right) \xrightarrow[N \rightarrow \infty]{} C(0) \int_{\mathbb{R}^2} g(x)dx, \text{ a.s and in } L^1. \quad (3.33)$$

Finally, it is also possible to obtain a multidimensional central limit theorem thanks to the previous formulations.

Proposition 3.3.5 (Central Limit theorem for shot noise models). *Let g be a continuous function on \mathbb{R}^2 with zero mean and compact support, $X \sim \text{DPixP}(C)$ and the related shot noise S_N : $S_N(y) = \frac{1}{N^2} \sum_{x \in X} g \left(y - \frac{x}{N} \right)$, $\forall y \in \mathbb{Z}^2$.*

Then, $\forall x_1, \dots, x_m \in \mathbb{Z}^2$,

$$\sqrt{N^2} (S_N(x_1), \dots, S_N(x_m)) \xrightarrow[N \rightarrow \infty]{D} \mathcal{N}(0, \Sigma(C)) \quad (3.34)$$

where for all $k, l \in \{1, \dots, m\}$

$$\begin{aligned} \Sigma(C)(k, l) &= (C(0) - \|C\|_2^2) \int_{\mathbb{R}^2} g(x_k - t)g(x_l - t)dt \\ &= (C(0) - \|C\|_2^2) R_g(x_l - x_k). \end{aligned} \quad (3.35)$$

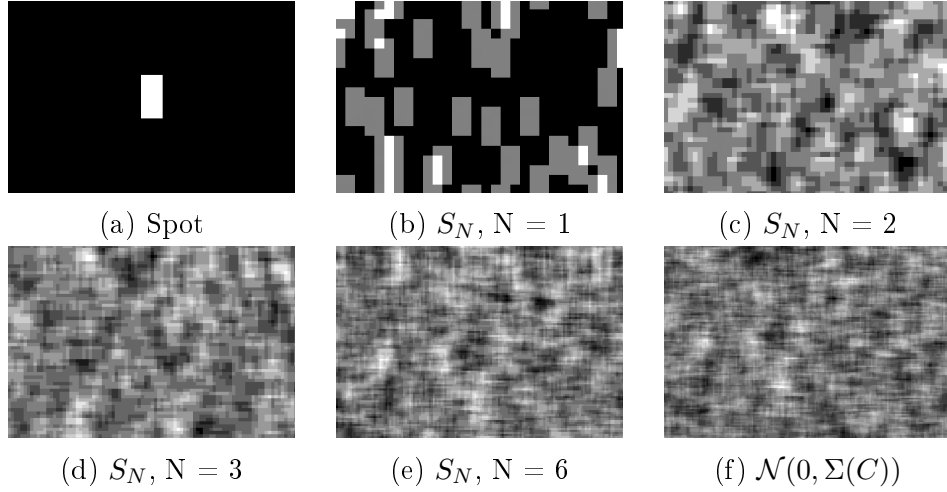


Figure 3.5: Determinantal shot noise realizations S_N as defined in Theorem 3.3.5 with various $N = 1, 2, 3, 6$ and a comparison with their associated limit Gaussian random field $\mathcal{N}(0, \Sigma(C))$ shown in (f). The shot noise is based on the spot (a) and the projection DPiXP with kernel C whose non-zero Fourier coefficients form a disk (Figure 3.1, bottom).

Proof. Consider the N -normalized shot noise S_N associated to g : $\forall y \in \mathbb{Z}^2$, $S_N(y) = \frac{1}{N^2} \sum_{x \in X} g\left(y - \frac{x}{N}\right)$. By setting $\forall u \in \mathbb{R}^m, \forall x_1, \dots, x_m \in \mathbb{Z}^2, \forall x \in \mathbb{R}^2$,

$$f(x) = u_1 g(x_1 - x) + u_2 g(x_2 - x) + \dots + u_m g(x_m - x), \quad (3.36)$$

f is continuous on \mathbb{R}^2 , with compact support such that $\int_{\mathbb{R}^2} f(x) dx = 0$ so it is possible to apply the limit theorem 3.3.4 and the Levy's continuity theorem. \square

Thus, shot noise models driven by a DPiXP also converge to a Gaussian limit whose covariance is related to the spot and to the kernel C of the point process. Note that, in the previous proposition, the limit variance $\Sigma(C)$ is equal to the product of a constant depending on the kernel C and the autocorrelation of the spot g . Similarly, a normalized Poisson shot noise associated to the spot g converges towards the distribution $\mathcal{N}(0, R_g)$, where R_g is the autocorrelation of g [48]. As the Bernoulli case corresponds to the kernel function $C = \delta_0$, we retrieve the same result here. Note also that there is no more interaction between the spot and the kernel in the limit. The higher the repulsion is, in the sense of the pair correlation function, involving high kernel coefficients, the lower the variance is. Let us mention the similar work in a continuous framework of Poinas et al. on the limit distribution of sums of functionals of DPPs defined on \mathbb{R}^d [107]. Figure 3.5 presents the asymptotic behavior of shot

noise models driven by a spot that is the indicator function of a rectangle and a projection DPixP on \mathbb{Z}^2 with a kernel whose Fourier coefficients are defined as the indicator function of a disk. When the grid is refined, the shot noise as defined in this section tends to a Gaussian texture associated to the spot and the kernel of the DPixP.

3.4 Inference for DPixPs

One of the purposes of statistical inference is to fit a predetermined model to data that can be represented by points, using information on their global or local behaviour. When the data are assumed independent and well represented by a homogeneous point process, one can use Poisson point processes. Yet, some data may present attraction or repulsion, they may also have an anisotropic structure. DPixP models can be suitable for representing 2-dimensional discrete data points with repulsion. For instance, the positions of plant seeds [101] or trees in a forest [85] often exhibit repulsion because of limited shared supply, but also anisotropy due to environmental factors as wind orientation or ground steepness. DPixPs can also be adapted to model samples of human cells [10] and the position of their nuclei, which present a certain shape of repulsion because of the structure of the cell around the nucleus. Knowledge on this repulsion can provide valuable information, for instance one could imagine comparing the blood cells from patients with sickle cell disease, provoking a sickle shape of blood cells, and from healthy patients. Once one has inferred the parameters of an appropriate model, it is possible to reproduce similar data, to detect anomalies or distinguish different regions by statistical testing.

Learning the parameters of a determinantal point process, either the whole underlying kernel K as in [77, 1] or a few parameters encoding the kernel as in [13, 21], is still considered as a difficult task, first because the likelihood is often non-convex, and most of all because it is complex to compute as it uses the determinant of a huge matrix. Most papers studying inference for DPPs overcome this difficult computation by using restrictive hypothesis on the kernel such as in the papers [80] or [1]. Bardenet and Titsias [13] develop bounds on the likelihood and use Markov Chain Monte Carlo methods to infer the parameters of the kernel. On the other hand, using descriptive statistics to fit the models to the data enables to cope with this difficult computation and to obtain more efficient inference algorithm. It is the approach that we choose in this chapter. Some authors try to infer first order characteristics such as the intensity of the point process [22], which provides the average number of points in a given area. In our finite and discrete setting, we can obtain a direct estimation of the intensity, as the ratio between the number of points and the size of the domain. Several second order characteristics

are used to describe a sample, for example the empty space distance, the cumulative nearest-neighbor function, the pair correlation function (p.c.f. in short), presented above, or the Ripley's K function, closely related to the p.c.f (see [101] for a detailed presentation). These statistics provide information on the interactions between points. Møller and Waagepetersen [101] present these different statistics and state that higher order characteristics may be less stable if the number of points is low. In the following, we choose to focus on a quantity related to the p.c.f. It has several advantages: it is easy to interpret, it is easy to compute and it provides insights on local interactions. Biscio and Lavancier [21] also use the p.c.f for a minimum contrast estimation in continuous settings.

The purpose of this section is to derive a DPixP kernel function C from one or several samples of points on a finite and discrete domain. This estimation is non-parametric as we focus on general DPixP even though it can be seen as a parametric estimation of a DPP kernel matrix K of size $|\Omega| \times |\Omega|$ that we suppose block-circulant and determined by $|\Omega|$ parameters, the values of C . Before we investigate this question, it is necessary to characterize the identifiability of DPixP models.

3.4.1 Equivalence Classes of DPP and DPixP

A model is not identifiable if two different parametrizations are equivalent. Here, it would correspond to several different kernel functions generating the same DPixP. Indeed, DPixPs, and DPPs in general, are not identifiable, as illustrates Figure 3.6. It is crucial, in particular for estimation purposes, to characterize these equivalence classes of kernels. Of course this question is also decisive in more general cases, when the kernel matrix K is Hermitian, with real or complex coefficients. We propose here a brief synthesis of what is known on this question, and we add a study on DPixP kernels.

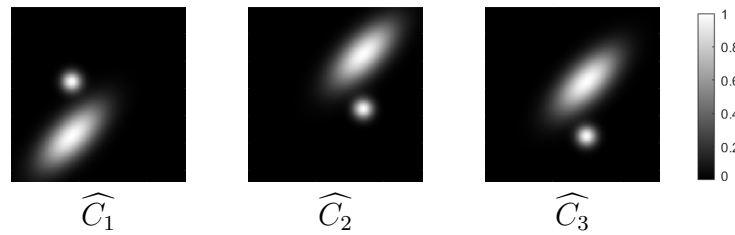


Figure 3.6: Three DPixP kernel functions, defined by their Fourier coefficients, generating the same DPixP.

The distribution of a DPP is entirely defined by all its principal minors (see Equation (1.5)), thus characterizing DPP kernel equivalences classes is

equivalent to understanding the consequences of equal principal minors on matrices, in the symmetric or Hermitian cases, and in the DPixP framework where the matrix is Hermitian circulant.

Notice that the characteristic polynomial of a matrix can be written as a function of its principal minors:

$$\det(tI + K) = \sum_{k=0}^N (-1)^k \left(\sum_{A \subseteq \mathcal{Y}, |A|=k} \det K_A \right) t^{n-k}. \quad (3.37)$$

Hence, two matrices with equal principal minors have equal characteristic polynomial so they have the same eigenvalues, with the same algebraic multiplicity. Two kernel matrices generating the same DPP have the same spectrum.

A key notion here is the diagonal similarity between two matrices: two square matrices M_1, M_2 are called diagonally similar if there exists a diagonal matrix D such that $M_2 = D^{-1}M_1D$. In the following, we also need the notion of the directed graph associated to a matrix [45, 67, 77]. Consider a matrix M of size $N \times N$. Its associated directed graph G_M contains the N vertices $\mathcal{Y} = \{1, \dots, N\}$ and an edge between the vertices x and y if and only if $M(x, y) \neq 0$. The matrix M is called irreducible if G_M is strongly connected, meaning that there exists a sequence a path from any vertex to any other one. In the opposite case, the matrix is called reducible, which is equivalent to being permutation-similar to a block upper triangular matrix. Besides, it is called completely reducible if it is permutation-similar to a block diagonal matrix with irreducible blocks, meaning that there exists a permutation matrix P such that $P^t M P = \begin{pmatrix} M_1 & & 0 \\ & \ddots & \\ 0 & & M_r \end{pmatrix}$, M_1, \dots, M_r irreducible. Notice that a Hermitian matrix is either irreducible or completely reducible.

Let us consider two general admissible DPP kernels K_1 and K_2 , admissible meaning that they are Hermitian and their eigenvalues are in $[0, 1]$. Thanks to basic determinant properties, notice that if there exists a diagonal matrix D such that $K_2 = D^{-1}K_1D$ or $K_2^t = D^{-1}K_1D$, then K_1 and K_2 have same principal minors, that is, the equivalence class of a DPP kernel contains all the admissible matrices of which the kernel matrix itself or its transpose is diagonally similar.

Real Symmetric DPPs

In the case where the DPP kernel is real and symmetric, Kulesza [77] proved the following proposition.

Proposition 3.4.1 (Equivalence classes of real symmetric kernels [77]). *Let K_1 and K_2 be two real positive symmetric $N \times N$ matrices with eigenvalues*

bounded by 1. Then $\text{DPP}(K_1) = \text{DPP}(K_2)$ if and only if there exists a $N \times N$ diagonal matrix D such that $K_2 = D^{-1}K_1D$, where the coefficients of D are either 1 or -1.

The proof of this proposition is in two parts. First, the author demonstrates the relation when all coefficients of the matrices are non-zero. Then, using graph theory, Kulesza extends this proof to matrices associated to a connected graph and finally to a disconnected graph, when the matrix is reducible. This equivalence property for real DPP kernels has impacted several learning strategies as in [113], [25], [129] or [26] which try to estimate real DPP kernels from several i.i.d. samples. In particular, the first two papers intend to solve the so-called principal minor assignment problem for symmetric matrices, and Brunel et al. [26] maximize a log-likelihood depending on the equivalence class of DPP kernels. Urschel et al. [129] obtain a bound on a distance between the estimated kernels L^* and the equivalence class of the original kernel: $\min_D \|L^* - D^{-1}LD\|_F$, on diagonal matrices D with coefficients only equal to 1 or -1.

Complex Hermitian DPPs

In the paper [123], Stevens characterizes equivalence classes of real or complex symmetric DPP kernels. We would like to characterize DPP equivalence classes in a more general setting, where the DPP kernels are no longer real or symmetric but complex and Hermitian. Schneider, Saunders and Engel [117, 45]) worked on the relation between equal principal minors and diagonal similarity through graph theory: see for instance [117] for links between equality of cyclic products and diagonal similarity, or [45] where they deal with real symmetric matrices. In 1986, Loewy [92] gives several sufficient conditions ensuring that if two square matrices have equal principal minors, one is diagonally similar to the other one or to the conjugate of the other one. We adapt these conditions to Hermitian DPP kernels in Theorem 3.4.1. In the following, we define $\mathcal{D}_N \subset \mathcal{M}_N(\mathbb{C})$ as the set of diagonal matrices of size $N \times N$ such that its coefficients are of modulus one.

Lemma 3.4.1. *Let K_1 and K_2 be two irreducible Hermitian matrices and assume that there exists an invertible diagonal matrix D such that $K_2 = D^{-1}K_1D$ or $K_2^t = D^{-1}K_1D$. Then all the coefficients of D have the same modulus so one can choose D in \mathcal{D}_N .*

Proof. Assume that K_1 and K_2 are two irreducible Hermitian matrices and there exists a diagonal matrix D such that $K_2 = D^{-1}K_1D$ or $K_2^t = D^{-1}K_1D$. First, let us suppose that $K_2 = D^{-1}K_1D$. For all $x, y \in \mathcal{Y}$ such that $K_1(x, y) \neq 0$, we have also $K_2(x, y) \neq 0$ and

$$K_2(x, y) = \frac{1}{d_x} K_1(x, y) d_y. \quad (3.38)$$

As K_2 is Hermitian, $K_2(x, y) = \overline{K_2(y, x)} = \overline{\frac{1}{d_y} K_1(y, x) d_x} = \frac{\overline{d_x}}{\overline{d_y}} K_1(x, y)$. Then $\frac{d_y}{d_x} = \frac{\overline{d_x}}{\overline{d_y}}$, hence for all $x, y \in \mathcal{Y}$ such that $K_1(x, y) \neq 0$, $|d_x| = |d_y|$. Now recall that K_1 is irreducible. Its associated graph is connected and every node is reachable from any other node so it is possible to propagate this equality so that for all $x, y \in \mathcal{Y}$, $|d_x| = |d_y| = \lambda$. Then without loss of generality, changing if necessary to $\frac{1}{\lambda} D$, we can choose D as the matrix such that $K_2 = D^{-1} K_1 D$ with diagonal coefficients of modulus equal to 1. The proof is similar if $K_2^t = D^{-1} K_1 D$. \square

Now we can prove the following theorem on the equivalence classes of Hermitian DPP kernels.

Theorem 3.4.1 (Identifiability for Hermitian DPP kernels). *Let N be a positive integer and let $\mathcal{Y} = \{1, \dots, N\}$. Suppose that $K_1, K_2 \in \mathcal{M}_N(\mathbb{C})$ are two Hermitian admissible DPP kernels and that K_1 is irreducible. If $N \geq 4$, suppose furthermore that, for every partition of \mathcal{Y} into subsets α, β such that $|\alpha| \geq 2, |\beta| \geq 2$, $\text{rank}(K_1)_{\alpha \times \beta} \geq 2$. Then, the following propositions are equivalent:*

- (i) $\text{DPP}(K_1) = \text{DPP}(K_2)$,
- (ii) There exists a diagonal matrix D such that $K_2 = D^{-1} K_1 D$ or $K_2^t = D^{-1} K_1 D$,
- (iii) There exists a diagonal matrix $D \in \mathcal{D}_N$ such that $K_2 = D^{-1} K_1 D$ or $K_2^t = D^{-1} K_1 D$.

Proof. Define K_1 and K_2 two admissible DPP kernels, such that K_1 verifies the hypothesis of Theorem 3.4.1. By definition, $\text{DPP}(K_1) = \text{DPP}(K_2)$ is equivalent to K_1 and K_2 having equal principal minors. In the papers [67] (Theorem 7) and [92] (Theorem 1), Hartfiel and Loewy prove that if K_1 is irreducible and for every partition of \mathcal{Y} into two subsets, α and β such that $|\alpha| \geq 2$ and $|\beta| \geq 2$, $\text{rank}(K_1)_{\alpha \times \beta} \geq 2$, then K_1 and K_2 have equal principal minors if and only if there exists a diagonal matrix D such that $K_2 = D^{-1} K_1 D$ or $K_2^t = D^{-1} K_1 D$. Notice that these two theorems, making the distinction between $\text{rank}(K_1)_{\alpha \times \beta}$ and $\text{rank}(K_1)_{\beta \times \alpha}$, are equivalent in this Hermitian setting. Then (i) is equivalent to (ii). Besides, clearly (iii) implies (ii) and under these assumptions, by Lemma 3.4.1, (ii) implies (iii). \square

In this general setting, assuming that K_1 is irreducible is crucial. Indeed, Hartfiel and Loewy [67] provide counterexamples of two admissible hermitian kernels generating the same DPP distribution without being diagonally similar.

Determinantal Pixel process

We now turn to the special case of DPiXP defined on Ω , the image domain of size $N_1 \times N_2$. Their kernel matrices are Hermitian block-circulant with circulant blocks. Recall that matrices generating DPiXPs have all the same eigenvectors, the vectors of the Fourier basis. We also know that two matrices generating the same DPiXP distribution have the same eigenvalues, so there is at most $N_1 N_2!$ different kernels associated to one DPiXP model. In the following proposition and remark, we prove that in most cases, the class of equivalence is much more constrained.

Proposition 3.4.2 (Identifiability for DPiXP). *Let Ω be a finite grid of size $N_1 \times N_2$, and C_1, C_2 be two admissible DPiXP kernels on Ω , generating the block-circulant matrices K_1 and K_2 that satisfy the hypothesis of Theorem 3.4.1. Then, $\text{DPiXP}(C_1) = \text{DPiXP}(C_2)$ if and only if there exists a translation mapping the Fourier coefficients of C_2 to the Fourier coefficient of C_1 or to their symmetry with respect to $(0, 0)$, meaning that*

$$\begin{aligned} \text{DPiXP}(C_1) = \text{DPiXP}(C_2) \iff \exists \tau \in \Omega \text{ s.t. either } \forall \xi \in \Omega, \widehat{C}_2(\xi) = \widehat{C}_1(\xi - \tau) \\ \text{or } \forall \xi \in \Omega, \widehat{C}_2(\xi) = \widehat{C}_1(-\xi - \tau). \end{aligned} \quad (3.39)$$

Proof. As K_1 and K_2 satisfy the hypothesis of Theorem 3.4.1, there exists an invertible diagonal matrix D such that $K_2 = D^{-1}K_1D$ or $K_2^t = D^{-1}K_1D$, where $D \in \mathcal{D}_N$, meaning that D is a diagonal matrix with coefficients of modulus equal to one. First, assume that $K_2 = D^{-1}K_1D$. Define for all $x \in \Omega, \theta_x \in [0, 2\pi[$ such that $D(x, x) = e^{i\theta_x}$. The goal is to prove that there exists τ such that $\theta_x = 2\pi\langle x, \tau \rangle$, for all $x \in \Omega$. Notice that, by changing D into $\frac{1}{D(0,0)}D$, we can assume that $\theta_0 = 0$, that is $D(0, 0) = 1$. By assumption, we obtain

$$\forall x, y \in \Omega, K_2(x, y) = C_2(y - x) = e^{-i\theta_x} K_1(x, y) e^{i\theta_y} = e^{i(\theta_y - \theta_x)} C_1(y - x),$$

$$\text{and } C_2(x) = C_2(x - 0) = e^{i\theta_x} C_1(x). \quad (3.40)$$

Recall, thanks to Equations (1.7) and (1.8), that $C_1(0) = C_2(0)$ and that, for all $x \in \Omega$, $|C_1(x)| = |C_2(x)|$. As $C_2(x) = 0$ if and only if $C_1(x) = 0$, for such $x \in \Omega$, any value θ_x is valid. Consider the set $\Omega_C^* = \{x \in \Omega; C_1(x) \neq 0\}$. For all $z \in \Omega$, and all $x \in \Omega$, we have

$$\begin{aligned} C_2(z) &= e^{i\theta_z} C_1(z) = C_2(z + x - x) = e^{i(\theta_{z+x} - \theta_x)} C_1(z + x - x) \\ &= e^{i(\theta_{z+x} - \theta_x)} C_1(z). \end{aligned} \quad (3.41)$$

Denote for all $x \in \Omega$, $\alpha(x) = e^{i\theta_x}$. Thus, for all $z \in \Omega_C^*$, for all $x \in \Omega$, $\alpha(z) = \alpha(z+x)\overline{\alpha(x)}$, meaning that $\alpha(x) = \alpha(z+x)\overline{\alpha(z)}$. For all $\xi \in \widehat{\Omega}$, for all $z \in \Omega_C^*$, we have

$$\hat{\alpha}(\xi) = \sum_{x \in \Omega} \alpha(x) e^{-2i\pi \langle x, \xi \rangle} = \sum_{x \in \Omega} \overline{\alpha(z)} \alpha(z+x) e^{-2i\pi \langle x, \xi \rangle} = \overline{\alpha(z)} e^{2i\pi \langle z, \xi \rangle} \hat{\alpha}(\xi). \quad (3.42)$$

As α is not the zero function, consider $\tau \in \widehat{\Omega}$ such that $\hat{\alpha}(\tau)$ is non-zero. Then, for all $z \in \Omega_C^*$, $\alpha(z) = e^{2i\pi \langle z, \tau \rangle}$. Thus, for all $z \in \Omega_C^*$, $C_2(z) = e^{2i\pi \langle z, \tau \rangle} C_1(z)$, which is also true for z such that $C_1(z) = 0$. To conclude, for all $z \in \Omega$, $C_2(z) = e^{2i\pi \langle z, \tau \rangle} C_1(z)$. In the second case when $K_2^t = D^{-1}K_1D$, the proof is identical. \square

Remark 3.4.1. Notice that when we consider two equivalent DPixP kernels C_1 and C_2 , generating the block-circulant matrices K_1 and K_2 , there are three possible configurations. The first one is when K_1 verifies the assumptions of Theorem 3.4.1, it leads to Proposition 3.4.2. In the second case, K_1 is irreducible, but $N = N_1N_2 \geq 4$ and there exists a partition α, β of \mathcal{Y} such that $|\alpha| \geq 2$, $|\beta| \geq 2$ and $\text{rank}(K_1)_{\alpha \times \beta} < 2$. In the third case, K_1 is not irreducible. Let us characterize the second and third cases. It appears that these configurations are “rare” in practice.

Case 2: Assume that K_1 is irreducible, $N = N_1N_2 \geq 4$ and that there exists a partition α, β of \mathcal{Y} such that $|\alpha| \geq 2$, $|\beta| \geq 2$ and $\text{rank}(K_1)_{\alpha \times \beta} < 2$. If $\text{rank}(K_1)_{\alpha \times \beta} = 0$, that is $(K_1)_{\alpha \times \beta} = 0$. There exists a permutation matrix such that K_1 is permutation similar to a block diagonal matrix, which is in contradiction with the irreducible hypothesis. Hence, $\text{rank}(K_1)_{\alpha \times \beta} = 1$. This means that there exist two vectors $u \in \mathbb{C}^{|\alpha|} \setminus \{0\}$ and $v \in \mathbb{C}^{|\beta|} \setminus \{0\}$ such that $(K_1)_{\alpha \times \beta} = u^t v$. In practice, as K_1 is Hermitian and the Fourier coefficients of C are real, the coefficients of the matrix K_1 are tightly constrained. The matrix is determined by a small number of modulus and arguments. Then, when assuming that K_1 and K_2 are equivalent, as DPixP kernels, the matrices are even more constrained. See Appendix C.1 for a simple example of this configuration. Notice that in the 1D case of dimension 5, two equivalent DPixP kernels K_1 and K_2 in this configuration still verify that there exists a diagonal matrix $D \in \mathcal{D}_N$ such that $K_2 = D^{-1}K_1D$ or $K_2^t = D^{-1}K_1D$. Our conjecture is that this is always the case, whatever the dimension of Ω . Thus, this assumption on the rank of the submatrix $(K_1)_{\alpha \times \beta}$ leads to degenerate kernels that are numerically “rare”.

Case 3: K_1 is not irreducible. Then, as a Hermitian or circulant matrix, K_1 is necessarily completely reducible, meaning that there exists a permutation matrix P such that K_1 is permutation similar to a block diagonal matrix with irreducible blocks. We prove in Appendix C.2 that these blocks are copies of one Hermitian block-circulant sub-matrix, that we can call the canonical block:

they all have equal size and the coefficients are identical. Note that restricting DPP to a subset A define also a DPP on this subset A [81, Section 2.3]. Furthermore, as each block matrix is still circulant, each one defines a sub-DPiP defined on the associated subset of pixels. By assumption, these blocks are irreducible so they are either in the first or in the second configuration. Let us consider K_2 a DPiP kernel equivalent to K_1 . Thanks to the modulus equality, K_2 is similar to a block diagonal matrix with blocks of same size, using the same permutation matrix. If the canonical block is in the first configuration, verifying the rank hypothesis of Theorem 3.4.1, the final diagonal matrix D is simply the concatenation and rearrangement of all the diagonal sub-matrices D_i associated to its respective i -th block. Notice that as the block submatrices are identical to the canonical block and each one concerns a different set of pixels, all submatrices are in the same configuration, meaning that either for all submatrices K_{1i} of K_1 , $K_{1i} = \overline{D_i} K_{2i} D_i$ or for all submatrices K_{1i} , $\overline{K_{1i}} = \overline{D_i} K_{2i} D_i$. On the other side, if the canonical block is in the second configuration, we can't conclude on the similarity of both matrices K_1 and K_2 in the general case yet. Notice that this completely reducible hypothesis is quite degenerate. It corresponds to a DPiP defined on an image domain that can be partitioned in groups of pixels evenly spaced with independence from one group to the other: that means that the pixels are independent to their immediate neighbors. A typical example of this model would be image domain partitioned following a grid. As DPiPs deals with spatial repulsion, there seems to be few applications of such models.

It is important to notice that the size of the equivalence classes we characterized in Proposition 3.4.2 is small and known: given a DPiP kernel verifying the appropriate hypothesis, it admits at most $2|\Omega|$ equivalent kernels, generating the same DPiP distribution. Moreover, we have shown previously how a kernel that does not verify the hypothesis of the proposition is quite degenerate: in practice, when dealing with kernels adapted to a given problem, these hypothesis are always verified. Characterizing equivalence classes of DPPs and DPiPs is crucial for the estimation of DPiP kernels from point process samples. This is what we investigate in the next subsection.

3.4.2 Estimating a DPiP Kernel from One Realization

First, we address the question of inference from one single realization. Consider one set of points Y on Ω , the finite and discrete grid of size $N_1 \times N_2 = N$ and assume that Y has been sampled from a certain DPiP of kernel C_0 . Note that in general, one realization does not provide enough information to characterize a model. Yet, due to the stationarity of the kernels we consider, all the translations of Y can also be seen as samples drawn by the same DPiP kernel C_0 .

Let $n = |Y|$ denotes the cardinality of Y . The problem is to find C_e an admissible DPixP kernel that estimates C_0 , the original one. Equivalently, we want to find the Fourier coefficients $\widehat{C}_e \in [0, 1]^N$ the closest to \widehat{C}_0 , in a sense defined below. In the following, we will work in Fourier domain.

Let C be any admissible kernel on Ω and $X \sim \text{DPixP}(C)$. As before, we will consider \widehat{C} either as a function from $\widehat{\Omega}$ to $[0, 1]$, or as a vector in $[0, 1]^N$. Recall that the intensity of the point process is given by $\frac{\mathbb{E}(|X|)}{\Omega} = \frac{1}{\Omega} \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi) = C(0)$.

In case of a kernel estimation from one sample, it is natural to consider that the expected cardinality of the point process to be estimated is the cardinality of this unique sample. Thus, a straightforward estimation of the intensity of the point process is

$$C_e(0) = \frac{n}{N} \quad (3.43)$$

or equivalently $\sum_{\xi \in \widehat{\Omega}} \widehat{C}_e(\xi) = n$. Now, we want to determine the estimator $C_e(x)$, for all $x \in \Omega \setminus \{0\}$ denoted Ω^* . Let us consider

$$p_C(x) = \begin{cases} \mathbb{P}(x \in X | 0 \in X) = \frac{\mathbb{P}(\{0, x\} \subset X)}{\mathbb{P}(0 \in X)} = C(0) - \frac{|C(x)|^2}{C(0)} & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases} \quad (3.44)$$

Now, from the realization Y , we can obtain $\theta(x)$ the empirical estimator of $p_C(x)$ by

$$\theta(x) = \begin{cases} \frac{1}{n} \sum_{y \in \Omega} 1_Y(y) 1_Y(y+x) & \text{if } x \neq 0 \\ 0 & \text{if } x = 0. \end{cases} \quad (3.45)$$

For optimization purposes, we express all the quantities in function of \widehat{C}_e . In the following computations, we consider that the vectors are column vectors. Let us denote the set of admissible functions by

$$\widehat{C}_n = \{\widehat{C} \in \mathbb{R}^N \text{ such that } \sum_{\xi \in \widehat{\Omega}} \widehat{C}(\xi) = n \text{ and } \forall \xi \in \widehat{\Omega}, 0 \leq \widehat{C}(\xi) \leq 1\}. \quad (3.46)$$

We are looking for \hat{C}_e such that

$$\begin{aligned}
\hat{C}_e &\in \operatorname{argmin}_{\hat{C} \in \hat{\mathcal{C}}_n} \|p_C - \theta\|_2^2 \\
&= \operatorname{argmin}_{\hat{C} \in \hat{\mathcal{C}}_n} \sum_{x \in \Omega^*} \left(\frac{n}{N} - \frac{N}{n} |\mathcal{F}^{-1}(\hat{C})(x)|^2 - \frac{1}{n} \sum_{y \in Y} 1_Y(y) 1_Y(y+x) \right)^2 \\
&= \operatorname{argmin}_{\hat{C} \in \hat{\mathcal{C}}_n} \sum_{x \in \Omega^*} \left(\frac{n^2}{N^2} - \frac{1}{N} \sum_{y \in Y} 1_Y(y) 1_Y(y+x) - |\mathcal{F}^{-1}(\hat{C})(x)|^2 \right)^2 \\
&= \operatorname{argmin}_{\hat{C} \in \hat{\mathcal{C}}_n} \sum_{x \in \Omega^*} \left(b(x) - g(\hat{C})(x) \right)^2 = \operatorname{argmin}_{\hat{C} \in \hat{\mathcal{C}}_n} E(\hat{C}),
\end{aligned} \tag{3.47}$$

where, for all $\hat{C} \in \mathbb{R}^N$, and for all $x \in \Omega^*$,

$$g(\hat{C})(x) = |\mathcal{F}^{-1}(\hat{C})(x)|^2 \quad \text{and} \quad b(x) = \frac{n^2}{N^2} - \frac{1}{N} \sum_{y \in \Omega} 1_Y(y) 1_Y(y+x). \tag{3.48}$$

We want to minimize E on $\hat{\mathcal{C}}_n$ a non empty closed convex set so we can use the projected gradient algorithm. To project on the set of constraints, we use a classic adapted version of the algorithm to project onto the simplex [30], integrating a maximum bound constraint, denoted “proj”. Let us compute the gradient of the energy E we want to minimize.

As $g : \mathbb{R}^N \rightarrow \mathbb{R}^{\Omega^*}$, $\hat{C} \mapsto \left(|\mathcal{F}^{-1}(\hat{C})(x)|^2 \right)_{x \in \Omega^*}$, we have

$$\begin{aligned}
\forall x \in \Omega^*, \forall \xi \in \hat{\Omega}, \quad \frac{\partial g(\hat{C})(x)}{\partial \hat{C}(\xi)} &= \frac{1}{N} \overline{\mathcal{F}^{-1}(\hat{C})(x)} e^{2i\pi \langle x, \xi \rangle} + \frac{1}{N} \mathcal{F}^{-1}(\hat{C})(x) e^{-2i\pi \langle x, \xi \rangle} \\
&= \frac{2}{N} \operatorname{Re} \left(\mathcal{F}^{-1}(\hat{C})(x) e^{-2i\pi \langle x, \xi \rangle} \right),
\end{aligned} \tag{3.49}$$

and moreover $\nabla E(\hat{C}) = \left(-Dg(\hat{C}) \right)^t 2 \left(b - g(\hat{C}) \right)$.

Notice that given a vector $u = (u_0, \dots, u_{N-1})^t \in \mathbb{R}^N$, we let u^* be equal to $(u_1, \dots, u_{N-1})^t$ the restriction of u to Ω^* . For all $\xi \in \hat{\Omega}$,

$$\begin{aligned}
\left(\left(-Dg(\hat{C}) \right)^t u^* \right)_\xi &= \frac{2}{N} \sum_{x \in \Omega^*} u_x \operatorname{Re} \left(\mathcal{F}^{-1}(\hat{C})(x) e^{-2i\pi \langle x, \xi \rangle} \right) \\
&= \frac{2}{N} \operatorname{Re} \left(\sum_{x \in \Omega} \left(u_x \mathcal{F}^{-1}(\hat{C})(x) \right) e^{-2i\pi \langle x, \xi \rangle} - u_0 C(0) \right).
\end{aligned}$$

$$\text{Then } \left(-Dg(\hat{C}) \right)^t u^* = \frac{2}{N} \operatorname{Re} \left(\mathcal{F} \left(u \odot \mathcal{F}^{-1}(\hat{C}) \right) \right) - \frac{2n}{N^2} u_0, \tag{3.50}$$

where \odot refers to the componentwise product of vectors. Finally we obtain

$$\nabla E(\hat{C}) = \frac{4}{N} \operatorname{Re} \left(\mathcal{F} \left(\left(|\mathcal{F}^{-1}(\hat{C})|^2 - b \right) \mathcal{F}^{-1}(\hat{C}) \right) \right) - \frac{4n^3}{N^4}, \text{ by setting } b(0) = 0. \quad (3.51)$$

In particular, computing $\nabla E(\hat{C})$ only requires two FFT calls. The projected gradient descent algorithm is recalled and adapted to this problem in Algorithm 5.

Algorithm 5 Projected gradient descent algorithm used to minimize E .

Input: Y the input realization, step size t , k_{\max} ,

- Compute for all $x \in \Omega^*$, $b(x) = \frac{n^2}{N^2} - \frac{1}{N} \sum_{y \in Y} 1_Y(y) 1_Y(y+x)$, $b(0) = 0$ (3.48).
- Set $\hat{C}_0 = \hat{C}_{\text{init}}$ (3.52).
- for $k = 1, \dots, k_{\max}$
 - Compute $\nabla E(\hat{C}_{k-1})$ (3.51).
 - Set $\hat{C}_k = \operatorname{proj} \left(\hat{C}_{k-1} - t \nabla E(\hat{C}_{k-1}) \right)$.

Output: \hat{C}_K .

Note that the energy we want to minimize is not convex and it has several local minima: the initialization of the algorithm is crucial. Indeed, if the algorithm is initialized with a random matrix \hat{C}_{init} , the results can be far from the original target. We propose to initialize the algorithm with

$$\hat{C}_{\text{init}} = \operatorname{proj} \left(\mathcal{F} \left(\sqrt{b} \right) \right), \quad (3.52)$$

which is believed to be quite close to a solution of the optimization and provides good results, as observed in the experiments. Note that b can be negative, so applying a square root to b may produce complex coefficients to which we apply the Fourier transform. This enables the initialization kernel \hat{C}_{init} to be asymmetric.

Figures 3.9 and 3.10 (column 3) provides some results of this algorithm, from one realization generated by different DPixP kernels. One realization seems enough to retrieve the Fourier coefficients of a simple symmetric projection kernel (see Figure 3.9, a, b whose non-zero Fourier coefficients form a convex set). Even though for most projection kernels a predominant shape appears in the estimation, as soon as the kernel is more complex, one sample does not provide enough information.

3.4.3 Estimating a DPixP Kernel From Several Realizations

A unique realization may not provide enough information for our proposed algorithm to estimate the Fourier coefficients of a DPixP kernels but if several realizations are available, combining them provides better results. Assume that we have J realizations, $J \in \mathbb{N}^*$, each of cardinality n_j , that we suppose independently generated by the same DPixP kernel.

Method by Average

The first strategy to take advantage of these multiple realizations is to apply independently the previous estimation process to each realization and then to average the estimated kernels. This method requires to handle the issue of identifiability: the realizations can lead to different kernels belonging to the same equivalence class. In section 3.4.1, we prove that the equivalence class of a DPixP kernel C_1 includes the set of DPixP kernels C_2 such that there exists a translation mapping the Fourier coefficients of C_2 to the Fourier coefficient of C_1 or to their symmetry with respect to $(0,0)$. In order to look for an admissible canonical kernel and to deal with the equivalence under translation of Fourier coefficients, for each estimated kernel, we ensure that the gravity center of its Fourier coefficients is centered. Concerning the symmetry equivalence, we propose to consider the first estimator as the canonical one and, for any subsequent estimation, we try both orientations and keep the closest to the first one.

Figure 3.7 shows some estimated kernel using this strategy. The kernels we want to retrieve are projection DPixP kernels. For display purpose, we projected the estimated kernel on the set of projection DPixP kernels. The results are satisfying if the kernel is simple, meaning that for instance the high Fourier coefficients form a convex shape, or if the Fourier coefficients are symmetric with respect to $(0,0)$, but as soon as the kernel is more complex, the algorithm only retrieve a weak approximation of the target kernel. Moreover, estimating J different kernels does not seem to be the most efficient method and it requires the handling of the identifiability issue.

Method by Combination

We propose a second strategy which combines all the realizations to produce a better empirical estimator θ_J of p_C . First, the expected number of points is approximated by the mean number of points in the realizations,
$$n = \frac{n_1 + \dots + n_J}{J}.$$

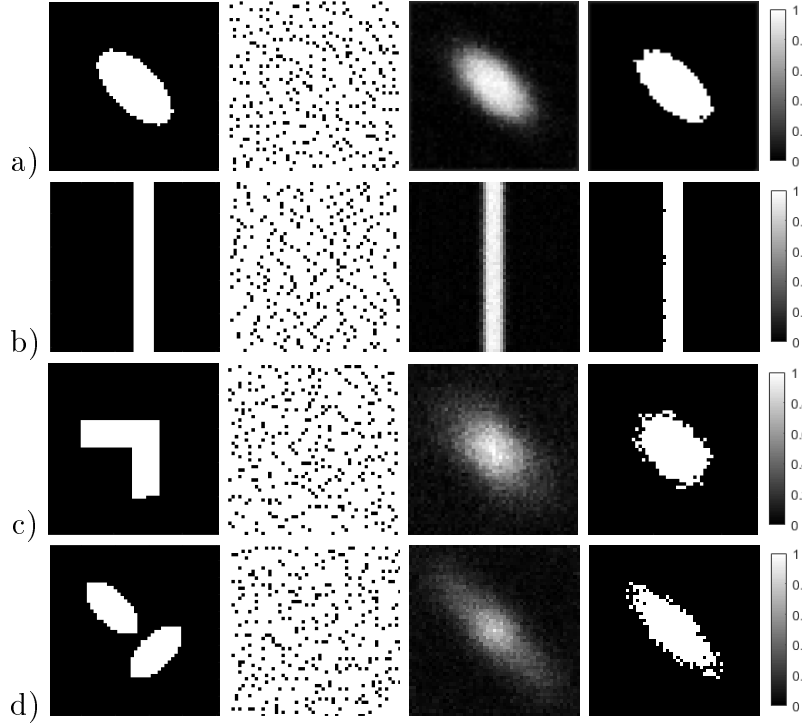


Figure 3.7: Estimation of a DPixP kernel from 100 realizations, using a method by average. From left to right: the target DPixP kernel, one sample generated from this DPixP, the average of 100 independent estimations done on every sample, its projection on the set of projection DPixP kernels.

If we have J realizations $(Y_i)_{i \in \{1, \dots, J\}}$, Equation (3.45) is replaced by:

$$\forall x \in \Omega, \theta_J(x) = \begin{cases} \frac{1}{nJ} \sum_{i=1}^J \sum_{y \in \Omega} 1_{Y_i}(y) 1_{Y_i}(y+x) & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases} \quad (3.53)$$

The rest of the procedure remains similar as we want to minimize the function $\|p_C - \theta_J\|_2^2$, in particular, the initialization kernel is

$$\hat{C}_{\text{init}} = \text{proj} \left(\mathcal{F} \left(\sqrt{\frac{n^2}{N^2} - \frac{1}{NJ} \sum_{i=1}^J \sum_{y \in \Omega} 1_{Y_i}(y) 1_{Y_i}(y+x)} \right) \right). \quad (3.54)$$

Figure 3.8 presents several initialization kernels computed from one, 10 and 100 realizations. As one can see, the initialization is very noisy but already contains information on the target kernel.

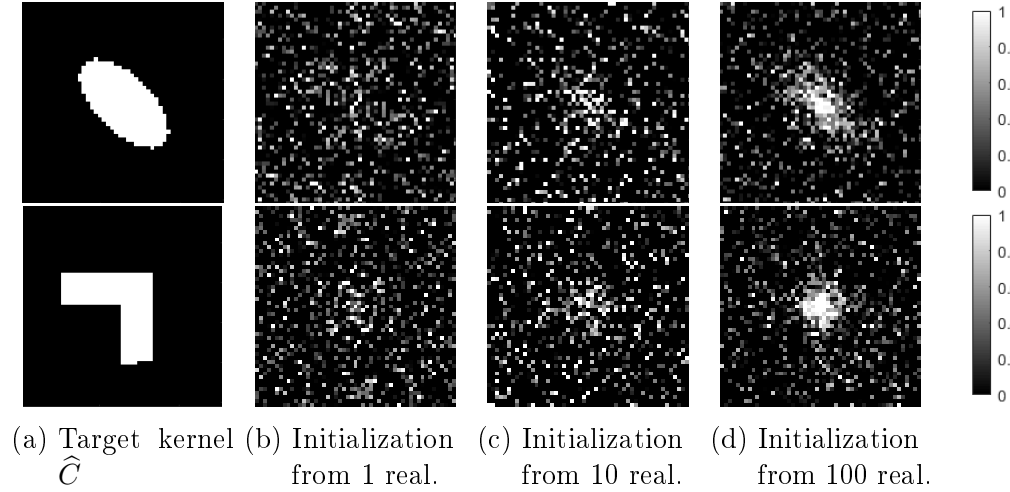


Figure 3.8: Two examples of initialization of our estimation algorithm. From left to right: the Fourier coefficients of the target kernel (a), the initialization from 1, 10 and 100 realizations.

Figures 3.9 and 3.10 present some experiments on several DPiXP kernels, using the second strategy presented here and combining all the samples in one estimation process. We have seen in the previous subsection that any translation of the estimated Fourier coefficients or a symmetry with respect to $(0, 0)$ of the estimated Fourier coefficients generate the same DPiXP. Thus, in Figures 3.9 and 3.10, we display a centered version of the estimation. First, Figure 3.9 presents the results of this estimation procedure with projection kernels, meaning that the Fourier coefficients of these kernels are zero or one. It shows how 10 realizations provide enough information to retrieve a kernel close to the original one. Using 100 realizations enables us to obtain satisfying results. This algorithm is able to retrieve the shape formed by non-zero Fourier coefficients, even when it is intricate (for instance (g),(h) in Figure 3.9).

Figure 3.10 presents some results of this algorithm for non-projection DPiXP kernels. Kernel (a) is a Bernoulli kernel: all the Fourier coefficients are equal to $\frac{n}{N}$. As expected, no specific structure appears from the estimation, regardless of the number of samples used. The estimations (b) and (c) are much noisier than their projection equivalent (Figure 3.9(a,e)) even if the shape formed by the Fourier coefficients (which directly impacts the local repulsion of the point process) seems retrieved.

To conclude, the algorithm presented in this section provides satisfying estimations if the original kernel is a projection DPiXP kernel, in particular when we have more than 10 samples. Indeed, as we have seen in Section 3.3.2 and as the authors of [20] noted, projection determinantal processes can be seen as the most repulsive DPPs. Thus, within a sample, the characteristics of

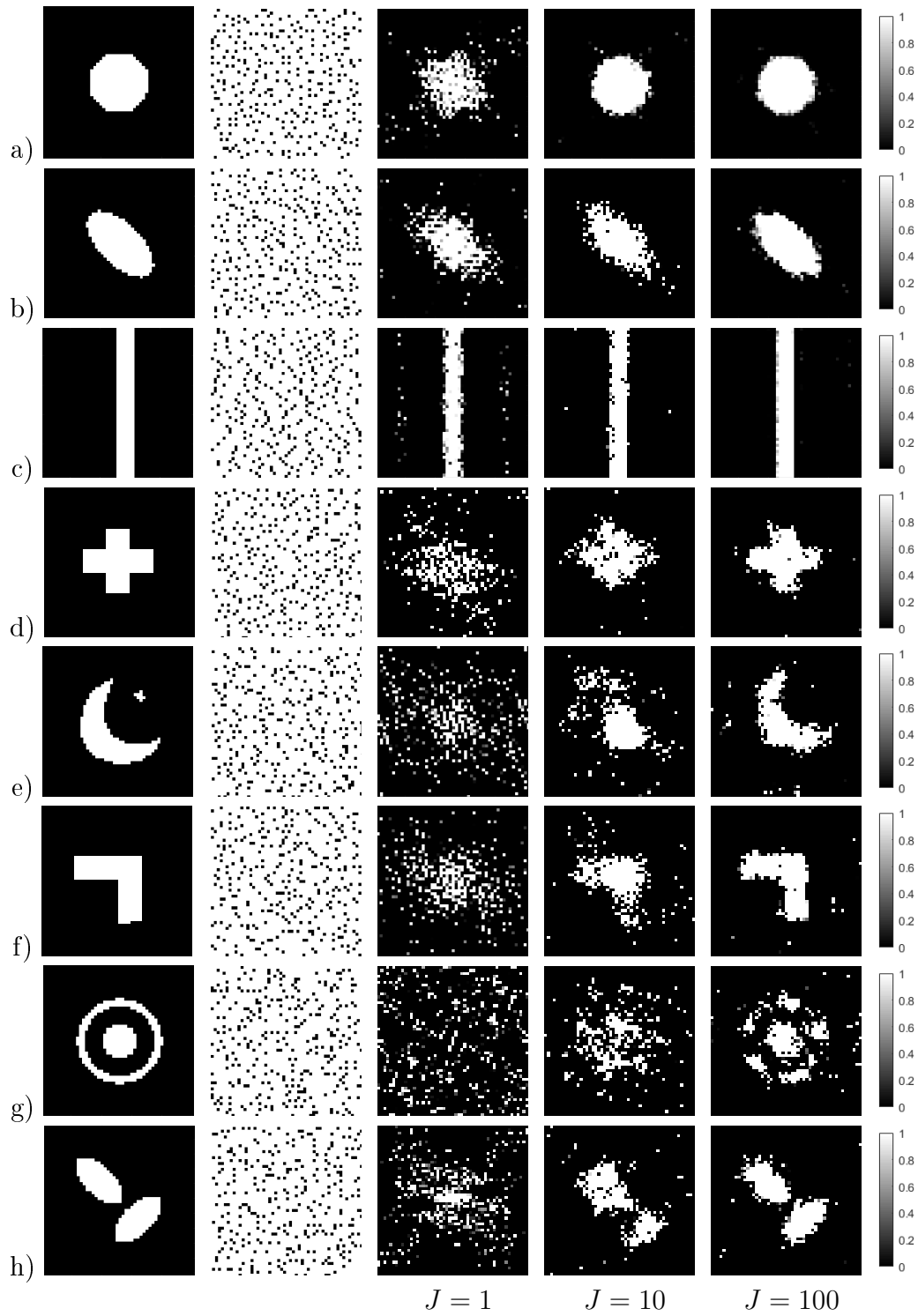


Figure 3.9: Experiments on several projection kernels. From left to right: the target Fourier coefficients of the kernel we want to recover, one realization of this DPixP, the estimation of the Fourier coefficients from one, from 10 and from 100 realizations, with $k_{\max} = 2000$.

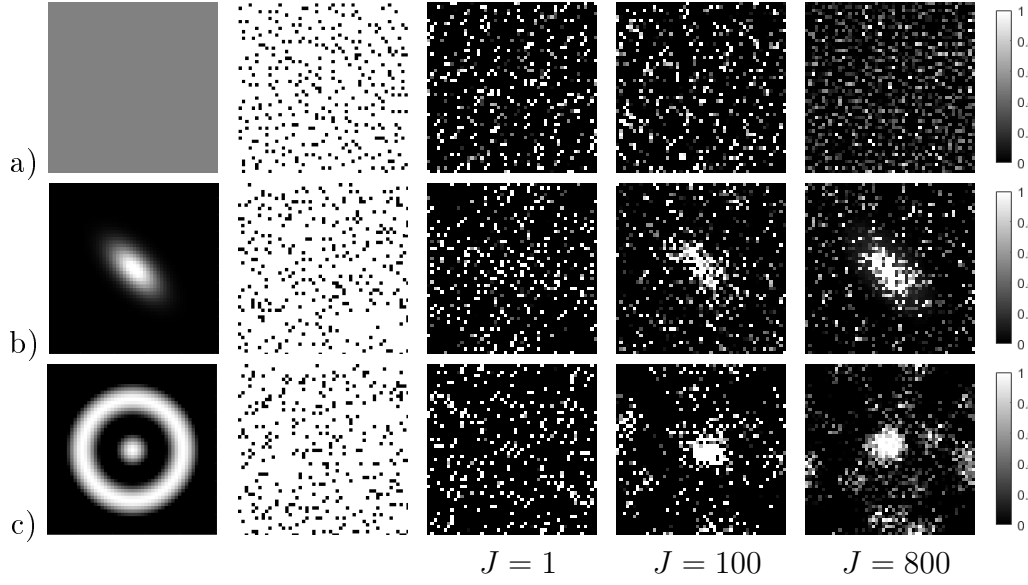


Figure 3.10: Experiments on general DPixP kernels. From left to right: the target Fourier coefficient of the kernel we want to recover, one realization of this DPixP, the estimation of the Fourier coefficients from one, from 100 and from 800 realizations, with $k_{\max} = 2000$.

the repulsion, and of the kernel, are more accessible. Nevertheless, if we deal with a general complex kernel, the algorithm retrieves fewer information.

3.5 Conclusion

In this chapter, we introduced a new type of DPPs defined on the pixels of an image that we call determinantal pixel processes. In this setting, we showed that the only possible hard-core repulsion for DPixP is directional. Given a direction, it is possible to impose to select at most one pixel on any discrete line with this direction in the image, but any further hard-core constraint leads to a degenerate kernel. We studied shot noise models based on a DPixP as a method to sample micro-textures and we adapted the choice of DPixP kernel in function of a given spot function of the shot noise and of the regularity one is looking for. It appears that the least repulsive DPixP, generating the least regular textures, is a homogeneous Bernoulli process while the most repulsive DPixP kernel, generating regular textures, is a projection kernel, which enables getting closer to a hard-core repulsion.

Thus, in Section 3.2, we proved that it is not possible to avoid overlaps if we randomly copy and place a given shape using a DPixP, unlike particular

Gibbs processes. However, in Section 3.3, we saw that, given a shape, it is possible to derive a DPixP kernel so that there are as few overlaps as possible. This property may be interesting for computer graphics issues especially since DPixPs have elegant theoretical properties. Notice that our algorithm to retrieve the “minimal variance” kernel, a kernel minimizing the number of overlaps, is greedy, it is not optimal. As a future work, we would like to investigate the development of an algorithm more efficient and look for a theoretical bound on the number of overlaps in shot noise models based on this DPixP and on a given shape.

We also investigated the DPP and DPixP equivalence classes, that is families of kernels generating the same point process. In the DPixP case, two kernels are equivalent if the Fourier coefficients of one of them is a translation and possibly a symmetry of the Fourier coefficients of the second. We developed an algorithm to infer the Fourier coefficients of a DPixP kernel from one sample or from a set of samples. This algorithm takes advantage of the stationarity of DPixPs and provides satisfying results, particularly when the target kernel is a projection kernel, with Fourier coefficients either equal to 0 or to 1.

We plan to investigate the joint estimation, from a texture image, of the spot function and of the DPixP kernel associated to a shot noise that could have generated the texture. As a result, we would be able to reproduce micro-textures and retrieve the properties of the input texture.

Chapter 4

Determinantal Point Processes on Patches

Contents

4.1	Introduction	99
4.2	Determinantal Patch Processes	101
4.2.1	DPP Kernels to Sample in the Space of Image Patches	101
4.2.2	Minimizing the Selection Error	104
4.2.3	Experiments	107
4.3	Application to a Method of Texture Synthesis	111
4.3.1	Texture Synthesis with Semi-Discrete Optimal Transport	112
4.3.2	DPP Subsampling of the Target Distribution	114
4.3.3	Results	118
4.4	Conclusion	122

4.1 Introduction

As datasets to analyze and to process keep being larger and more complex, strategies to subsample these sets or to reduce the dimension of data have recently flourished. As we have seen before, DPP subsampling is part of these approaches, as it enables capturing the structure of data and produce a representative subset of the whole initial set, taking into account its inner diversity. In image processing and computer vision, DPPs have raised interest through video summarization ([66], [134]). The authors of [66] introduce sequential DPPs to take into account both the diversity of the frames and the chronology of the video. To represent the diversity of the frames they use a decomposition

similar to the quality-diversity decomposition that is introduced in [81] and that we recall below. Furthermore, the paper [134] proposes a strategy enhanced by DPPs which makes it one of the state of the art methods for video summarization. This method also uses a decomposition similar to a quality-diversity decomposition to describe the diversity in the video.

In this chapter, we focus on subsampling the set of patches \mathcal{P} of an image. This procedure can be useful for compression purpose for instance. It can also be necessary in order to fit a model on the patch set using only a proportion of the set, to increase the efficiency of the algorithm. For example, several patch-based denoising methods represent the patch distribution as a Gaussian mixture model ([136], [71]). These methods rely on the estimation of the parameters of such models thanks to the Expectation-Maximization (EM) algorithm. To do so, in general, they randomly and uniformly select a subset of patches, to reduce the cost of the estimation. This random selection is fast but, as we have seen in the previous chapters, this strategy may select points close to each other and miss some regions of the space. When considering patches, this amounts to select similar patches while possibly missing crucial areas of the image. Thus, the subset needs to be large enough so that it captures the patches diversity. The size of this selection impacts the running time of the estimation process, so a smaller selection, representative of the patches of the image, would ensure a faster and more accurate estimation. DPPs offer the opportunity to select a reduced subset of patches that captures the whole image.

Agarwal et al. [3] propose to adapt the k -Means algorithm by using a DPP initialization: the authors sample an appropriate DPP to select the initial centroids for the clustering strategy. The authors prove that this initialization compares favorably with k -Means++, the most popular adaptation of the k -Means algorithm, with a deterministic initialization. One advantage of this algorithm using DPPs over the second is its adaptability concerning the number of clusters. Similarly, in the previous example with denoising methods, DPPs could also provide a satisfying initialization to the EM algorithm.

This chapter examines DPPs defined on the patch space of an image. We investigate here the possible choices of DPP kernels for such applications, in order to subsample the patch space of an image. This can be useful to speed up or to improve a patch-based algorithm, by considering only the most significant patches in the image. In Section 4.2, we study several classes of DPP kernels, computed from the patches of the image. Numerical experiments show that these kernels behave very differently and that it is rather simple to adapt the kernel in function of the application that will be done with the selected patches.

Section 4.3 applies this strategy to speed up a texture synthesis algorithm. This algorithm, presented by Galerne et al. in [52], uses the empirical distribution of the patches of an initial texture and heavily relies on semi-discrete optimal transport. This method enables to synthesize complex textures. The authors propose to uniformly subsample the set of patches of the image to approximate the empirical distribution of the patches, using 1000 patches.

After a presentation of this synthesis strategy, we show how using a DPP to subsample the distribution of patches enables us to reduce the number of patches (to 200 or 100) and thus to significantly reduce the execution time of the algorithm while maintaining the quality of the synthesis.

4.2 Determinantal Patch Processes

4.2.1 DPP Kernels to Sample in the Space of Image Patches

When considering determinantal point processes on patches, that can be called determinantal patch processes, the framework is more general than in Chapter 3: We are no longer dealing with stationary periodic point processes. We consider a Hermitian kernel K adapted to select diverse subsets of patches from an image, as set in Equation (1.5). The definition of this diversity depends on the problem we want to solve: for instance, compression, reconstruction of the image or initialization of the centroids of a clustering or of the EM algorithm.

As we have seen in Section 1.2, there exists a second characterization of DPPs, using a positive semi-definite matrix L . These DPPs are called L -ensembles.

Definition 4.2.1. *We consider $\mathcal{Y} = \{1, \dots, N\}$ and L a Hermitian matrix of size $N \times N$ such that $L \succeq 0$, then the random set $X \subset \mathcal{Y}$ defined by*

$$\forall A \subset \mathcal{Y}, \quad \mathbb{P}(X = A) = \frac{\det(L_A)}{\det(I + L)} \quad (4.1)$$

is a DPP with likelihood kernel L . We will denote $X \sim \text{DPP}_L(L)$.

Recall that the initial definition using the kernel denoted by K , requires that $0 \preceq K \preceq I$. This L -ensemble definition doesn't need the constraint of bounding the eigenvalues of the kernel by one. This property is convenient to define a kernel, and a diversity model adapted to a specific problem. So this characterization is increasingly used in the machine learning community. That is also the definition we mostly use in this chapter.

We recall here the relation between the correlation kernel K and the likelihood kernel L of a DPP. Consider the following spectral decomposition of a DPP kernel K , $K = \sum_{k=1}^N \lambda_k v_k v_k^*$. Note that the definitions using the kernels K and the likelihood kernel L characterize the same DPP if and only if for all $k \in \{1, \dots, N\}$, $0 \leq \lambda_k < 1$ and if

$$K = L(L + I)^{-1} = I - (I + L)^{-1} \text{ and conversely } L = K(I - K)^{-1}. \quad (4.2)$$

Hence, in this case, $L = \sum_{k=1}^N \frac{\lambda_k}{1 - \lambda_k} v_k v_k^*$. Note that if K has any eigenvalue equal to 1, the DPP can't be associated to an L -ensemble.

In the following, consider an image u and the initial set $\mathcal{P} = \{P_i, i = 1, \dots, N\}$, the set of its patches of size $(2\rho + 1) \times (2\rho + 1) \times d$, where $\rho \in \mathbb{N}$ and d is the number of color channels. Let us present some kernels that can be used to subsample the patches of this image.

A first type of DPP likelihood kernels that are regularly used ([126],[84]) is the class of Gaussian kernels (sometimes called exponential kernels). Let us consider a Gaussian kernel based on the intensity of the patches, that we call the Intensity Gaussian kernel, defined by

$$\forall P_i, P_j \in \mathcal{P}, \quad L_{ij} = \exp\left(-\frac{\|P_i - P_j\|_2^2}{s^2}\right), \quad (4.3)$$

where s is called the bandwidth or scale parameter. This kernel depends on the squared Euclidean distance between the intensity values of pairs of patches. It is often used as a similarity measure on patches. Despite its natural limitations, this similarity measure provides good results.

The value of the parameter s has a direct impact on how repulsive the DPP is. Notice that if s is small, due to the exponential function, L_{ij} converges very quickly to zero as soon as $i \neq j$ and the distinction between patches is not very subtle. Thus, if s is small, L is close to the identity matrix and the DPP selection of patches is similar to a random uniform selection. On the contrary, for the same reason, the larger s is, the more repulsive the DPP is. However, this scale parameter should not be set too large because this would cause high numerical instability. As noticed in [4] and [126], the median of the interdistances between the patches is a satisfying choice for setting the value of s .

We propose to compare this kernel with another Gaussian kernel that we call the PCA kernel, which depends on the squared distance between patches in the space given by keeping only the k principal components after a Principal Component Analysis (PCA). Set \mathbf{P} the matrix gathering all the patches of

the image reshaped in column so that the size of \mathbf{P} is $d(2\rho + 1)^2 \times N$. We assume that \mathbf{P} has been centered, by subtracting the average patch to all the patches. It has not been reduced, meaning that patches with high variance, for instance patches with edges, will highly influence the decomposition. Thanks to a singular value decomposition, consider U, V two unitary matrices and Σ a diagonal matrix storing the sorted principal values of \mathbf{P} such that $\mathbf{P} = U\Sigma V^t$. We choose to keep only k principal components and we obtain the matrix $\mathbf{P}^k = V_k \mathbf{P}$, where we kept only the k first rows of the matrix V in V_k of size $k \times d(2\rho + 1)^2$ and the matrix $\mathbf{P}^k = \{P_i^k, i = 1, \dots, N\}$ is $k \times N$. Every initial patch $P_i \in \mathcal{P}$ is associated with a projected vector $P_i^k \in \mathbf{P}^k$. Thus, the PCA kernel is defined by

$$\forall P_i, P_j \in \mathcal{P}, \quad L_{ij} = \exp \left(-\frac{\|P_i^k - P_j^k\|_2^2}{s^2} \right). \quad (4.4)$$

This method discards principal vectors associated to small singular values and projects the patches on a low-dimensional space associated with the large singular values. This enables to find the components that best represent the variance of the patches and ignores mainly noise (depending on the number of dimension discarded). Thus, comparing patches in this low-dimensional space seems relevant to capture more precisely their dissimilarity.

A second type of common likelihood kernels uses a quality-diversity decomposition of the data. Kulesza and Taskar present in [81] this decomposition that uses a given quality measure computed on each element of the set and a dissimilarity computed between pairs of elements. Here, each patch P_i is associated with a quality measure, which is a non-negative number $q_i = q(P_i, \mathcal{P}) \in \mathbb{R}^+$, depending on the patch itself and on the other patches. Each patch P_i is also associated with a feature vector $\phi_i = \phi(P_i) \in \mathbb{R}^D$, such that $\|\phi_i\|_2 = 1$, which depends only on the patch itself. The quality/diversity likelihood kernel L is defined by

$$\forall P_i, P_j \in \mathcal{P}, \quad L_{ij} = q_i \phi_i^t \phi_j q_j. \quad (4.5)$$

This class of kernels presents several advantages. The first advantage of this definition is its interpretability. Each patch is associated with a quality measure, that one can adapt depending on the characteristics one wants to favor. The comparison between patches is also accessible and adjustable to obtain the most adapted kernel. This decomposition has a second advantage: the likelihood kernel becomes a low-rank matrix, with a rank equal at most to D , the number of features. In case of low-rank kernels, Kulesza and Taskar [80] propose a dual representation and a dual sampling algorithm. This sampling scheme is equivalent to the original algorithm but it takes advantage of the low-rank kernel and becomes much faster. We recall that, whatever the DPP

likelihood kernel, the cardinality of a sample generated from $\text{DPP}_L(L)$ will necessarily be lower than the rank of L . This low-rank definition imposes to sample subsets of size smaller than D , the number of features computed from the patches. Thus, this kernel is adapted when small and very small subsets of patches are needed. In these cases, it is very important to precisely control the selection process so such kernels are particularly relevant.

For this kernel that we call Qual-div kernel, we associate each patch with a feature vector given by a discrete cosine transform of the patch. Thus, each feature vector is of size $d(2\rho + 1)^2$. Note that in the experiments, we use color images (with 3 color channels) and patches of size 7×7 (meaning that $\rho = 3$) so the feature vectors of length 147. We define the quality measure such that it attributes a high value to patches whose intensity is far from that of its neighbors in the pixel grid. This choice gives further priority to singular patches, that can be seen as the outliers of the set of patches. As experiments will show, it highly favors textures and edges.

4.2.2 Minimizing the Selection Error

The question is to choose the best kernel, such that the sampled DPP on the patches minimizes an error computed as a distance between the selected patches and the initial set of patches \mathcal{P} . This problem is similar to discrete optimal quantization problems [106] where the aim is to find the best subset of patches \mathcal{Q} such that $\mathbb{E}_{\mathcal{Q} \sim \mu}(d(\mathcal{Q}, \mathcal{P}))$ is minimal, for a given distance d . Yet, this computation is often costly and hardly tractable. In the following, we suppose that the patches are of size $(2\rho + 1) \times (2\rho + 1)$ for some positive integer ρ and we denote by $\omega \subset \mathbb{Z}^2$ the patch domain $\{-\rho, \dots, \rho\}^2$.

First, the error, or the distance between the sample and the initial set of points, we want to minimize depends on the application. The mean square error (MSE in short) is commonly used to compare an image and its reconstruction. Here, we use a similar distance, the squared L^2 norm between the patches of the image and their nearest neighbor in the selection given by the DPP sampling on the patches. Consider \mathcal{Q} a subset of patches. This error is defined by

$$E_1 = \frac{1}{N} \sum_{i=1}^N d_{L^2}(P_i, \mathcal{Q})^2 = \frac{1}{N} \sum_{i=1}^N \min_{Q \in \mathcal{Q}} \sum_{x \in \omega} (P_i(x) - Q(x))^2, \quad (4.6)$$

where ω is the patch domain. One hopes that using a DPP to generate \mathcal{Q} will prevent from concentrating only on the most common patches and select singular patches. The following error can be useful to verify this property:

$$E_2 = \max_{i \in \{1, \dots, N\}} d_{L^2}(P_i, \mathcal{Q})^2 = \max_{i \in \{1, \dots, N\}} \min_{Q \in \mathcal{Q}} \sum_{x \in \omega} (P_i(x) - Q(x))^2. \quad (4.7)$$

A low error value asserts that the outlier patches (non redundant) are selected.

Given an expected cardinality $n \in \mathbb{N}^*$ and a kernel K_n , we will consider $\mathcal{Q} \sim \text{DPP}(K_n)$. We would like to find the DPP kernel minimizing the expectation of the errors: $\mathbb{E}_{\mathcal{Q} \sim \text{DPP}(K_n)}(E_1)$ and $\mathbb{E}_{\mathcal{Q} \sim \text{DPP}(K_n)}(E_2)$. Yet, this optimization problem depending on a DPP matrix K_n is intractable. As in the papers by Kulesza and Taskar [81] and Affandi et al. [1], we would like to have a closed-form minimization problem to obtain optimal parameters. These strategies are based on the quality-diversity decomposition of an L -ensemble kernel described in the previous section. Given predetermined features vectors, they determine an appropriate quality measures from the data. Here, we use a similar parametrization, using the first definition of DPPs, with a kernel matrix K . We suppose that its eigenvectors are fixed (given by features computed from the patches of the image) and we want to determine the optimal spectrum so that the associated matrix K minimizes a tractable error. Furthermore, thanks to the Campbell Formula (3.19), we know that the expectation of some functionals defined on point processes are tractable. That is what we use in the following.

Suppose we select a subset of patches using a DPP of kernel K : $\mathcal{Q} \sim \text{DPP}(K)$. We would like to study the following measure:

$$R(\mathcal{Q}) = \sum_{P \in \mathcal{P}} \sum_{Q \in \mathcal{Q}} f_P(Q). \quad (4.8)$$

It can be seen as a reconstruction evaluation, if the function f_P involves a distance between the input patch and the patch P . With the appropriate function f_P , R can represent how well a patch $P \in \mathcal{P}$ is represented by the selection \mathcal{Q} . For instance, by considering the functions $f_{\alpha, P}(Q) = \mathbf{1}_{\|P-Q\|^2 \leq \alpha}$ or $f_P(Q) = e^{-\|P-Q\|^2}$, R will return a high value if the selection \mathcal{Q} encompasses the set of patches. Notice that if we use a function f_p which depends on the L^2 distance between patches, maximizing R will favor selections similar to the ones minimizing the MSE. Thus, contrary to the previous error quantities, E_1 and E_2 , we want to generate a subset \mathcal{Q} such that R is large. From the Campbell Formula (3.19) adapted to general discrete DPPs, we have

$$\begin{aligned} \mathbb{E}(R(\mathcal{Q})) &= \mathbb{E} \left(\sum_{P \in \mathcal{P}} \sum_{Q \in \mathcal{Q}} f_P(Q) \right) = \sum_{j=1}^N \mathbb{E} \left(\sum_{Q \in \mathcal{Q}} f_{P_j}(Q) \right) \\ &= \sum_{j=1}^N \sum_{i=1}^N f_{P_j}(P_i) K(P_i, P_j). \end{aligned} \quad (4.9)$$

Assume that K admits the eigendecomposition

$$K(P_i, P_j) = \sum_{k=1}^D \lambda_k \phi_k(P_i) \phi_k^*(P_j), \quad (4.10)$$

with $D \leq N$, fixed eigenvectors and unknown eigenvalues $(\lambda_k)_{k \in \{1, \dots, D\}}$. Then the previous expectation becomes

$$\mathbb{E}(R(\mathcal{Q})) = \sum_{k=1}^D \lambda_k \sum_{i=1}^N |\phi_k(P_i)|^2 \sum_{j=1}^N f_{P_j}(P_i). \quad (4.11)$$

The maximization of this quantity with respect to $(\lambda_1, \dots, \lambda_D)$ is a linear problem under the linear constraints: $\sum_{P \in \mathcal{P}} K(P, P) = \sum_{k=1}^D \lambda_k = n$, and for all $k \in \{1, \dots, D\}$, $0 \leq \lambda_k \leq 1$. The advantage of solving such a problem is that the solution $(\lambda_k^*)_{k \in \{1, \dots, D\}}$ is explicit. It is on the boundary of the constraints, meaning that is a kernel K with only n non-zero eigenvalues, each one equal to 1: the solution is a projection DPP. Given any function f_p , any integer $n \leq D$, let us consider I_n the set of the indices associated to the n largest coefficients of the vector ψ of size D defined by $\psi_k = \sum_{i=1}^N |\phi_k(P_i)|^2 \sum_{j=1}^N f_{P_j}(P_i)$. The solution of the problem

$$\operatorname{argmax}_{(\lambda_k)} \mathbb{E}(R(\mathcal{Q})) \text{ such that } \sum_{k=1}^D \lambda_k = n \text{ and } \forall k, 0 \leq \lambda_k \leq 1, \quad (4.12)$$

is the set of eigenvalues $(\lambda_k^*)_{k=1, \dots, D}$ defined by

$$\lambda_k^* = \begin{cases} 1 & \text{if } k \in I_n \\ 0 & \text{otherwise} \end{cases}. \quad (4.13)$$

For instance, if we choose $f_{\alpha, P_i}(P_j) = \mathbf{1}_{\|P_i - P_j\|^2 \leq \alpha}$, then we need to maximize the function

$$\begin{aligned} \mathbb{E}(R(\mathcal{Q})) &= \sum_{k=1}^D \lambda_k \sum_{i=1}^N |\phi_k(P_i)|_2^2 \sum_{j=1}^N \mathbf{1}_{\|P_i - P_j\| \leq \alpha} \\ &= \sum_{k=1}^D \lambda_k \sum_{i=1}^N |\phi_k(P_i)|^2 |\mathcal{B}(P_i, \alpha)|, \end{aligned} \quad (4.14)$$

where $\mathcal{B}(P, \alpha)$ is the ball inside \mathcal{P} with center P and radius α for the Euclidean distance between patch intensities and $|A|$ is the cardinality of the subset A . Thus, $|\mathcal{B}(P_i, \alpha)|$ denotes the number of patches in the image that are within a

distance of P_i smaller than α . In the experiments, we use this function and we choose α to be half the median of interdistances between patches. Note that this maximization problem will favor patches similar to many others. This creates an interesting compromise: the DPP will tend to select diverse subsets of redundant patches. As anticipated, we will see in the experiments that this method tends to miss singular patches.

4.2.3 Experiments

The following figures present some results of subsampling in the space of image patches, for different cardinality. First notice that the cardinality is fixed for the uniform sampling. It is also fixed for the last optimized kernel, as we obtain a projection kernel from the maximization problem. Concerning the three other kernels, they are defined using the L -ensemble definition in Equations (4.3), (4.4) and (4.5). We used a common normalization strategy, formalized in [14], using a likelihood kernel L whose eigenvalues are denoted $(\lambda_k)_{k \in \{1, \dots, N\}}$. Given a desired expected cardinality n , we normalize L to obtain a kernel $L_c = cL$, where c is chosen such that

$\sum_{k=1}^N \frac{c\lambda_k}{1 + c\lambda_k} = n$. Note also

that the Qual-div kernel (4.5) and the optimized kernel (4.13) are low-rank, with a rank equal at most to the number of features that we use to defined the kernels. In these experiments, the feature vector associated to each patch (ϕ in Equations (4.5) and (4.9)) is obtained from the discrete cosine transform of the patch. Note that a DPP kernel can't generate samples with more items than its rank and in the following experiments, we use patches of size $7 \times 7 \times 3$. Thus, the rank of the two previous kernels is 147 and we can observe the results, with a step of 50, up to a cardinality equal to 100 in Figure 4.4.

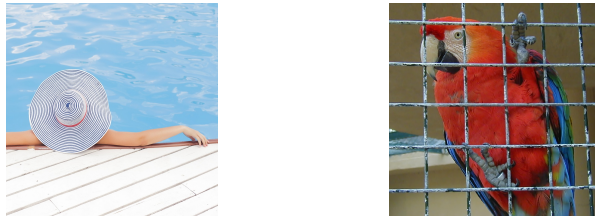


Figure 4.1: Original images considered in Figures 4.2 to 4.4.

Figures 4.2 and 4.3 show images reconstructed using the associated selected patches presented below the reconstruction. Each patch in the initial image is replaced by its nearest neighbor in the DPP selection. The final image is obtained by average: given a pixel, all the overlapping patches con-

taining this pixel are averaged. This is a common strategy to aggregate the patches. Several other methods are proposed in the literature, such as using a weighted average [33, 116] or implicitly including the reconstruction in a global variational problem [136]. An average considering uniform weights on all the patches is often used as it does not require any other computation or information to store. Thus, after subsampling the set of patches, the initial image can be represented by its size $N_1 \times N_2 = N$, the small set of patches of size $(2\rho + 1) \times (2\rho + 1) \times d$ and a vector of indices of length N , associating each initial patch to its nearest neighbor in the selection.





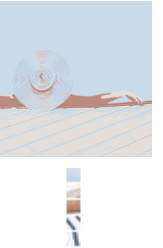


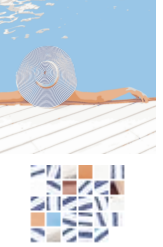







Card	Unif. sample	Intens. kernel	PCA kernel	Qual-div kern.	Optim kern.
5					
25					
100					

Figure 4.2: Image reconstruction comparing different expected cardinality and the DPP kernels presented in the previous subsections. For each cardinality, the first row presents the reconstruction of the image using only the patches selected by the corresponding kernel, given in the second row.


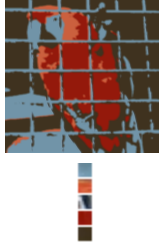

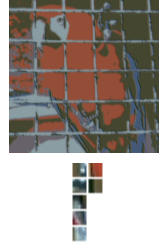
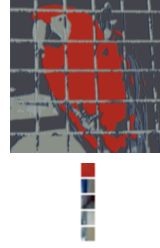
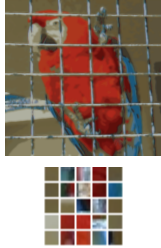
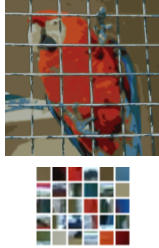
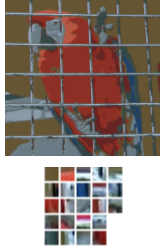
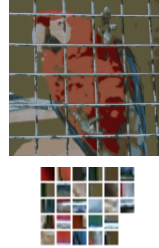
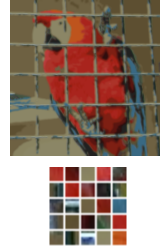
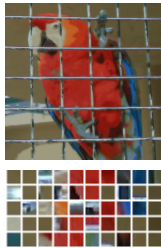
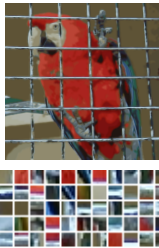
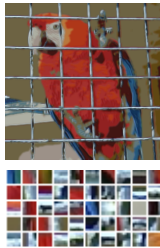
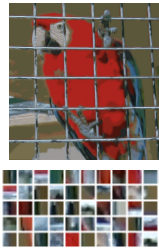
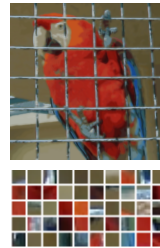
Card	Unif. sample	Intens. kernel	PCA kernel	Qual-div kern.	Optim kern.
5					
25					
100					

Figure 4.3: Same as Figure 4.2 for the Parrot image.

Figure 4.4 compares the errors E_1 (4.6), E_2 (4.7) and the peak signal-to-noise ratio (PSNR) of the reconstruction images generated from samples given by the different kernels. The PSNR is a metric commonly used to evaluate the quality of the reconstruction of an image. Consider an initial image I_0 and a reconstruction I_1 , both having d color channels and N pixels with a value between 0 and 1. Then,

$$\text{PSNR} = 10 \log_{10} \frac{Nd}{\sum_{c=1}^d \sum_{i=1}^N (I_0(i, c) - I_1(i, c))^2}. \quad (4.15)$$

First, as expected, a uniform sampling can produce samples which contain many similar patches. The first image (Pool) has several large and regular regions that could be represented by a few patches and these regions are often

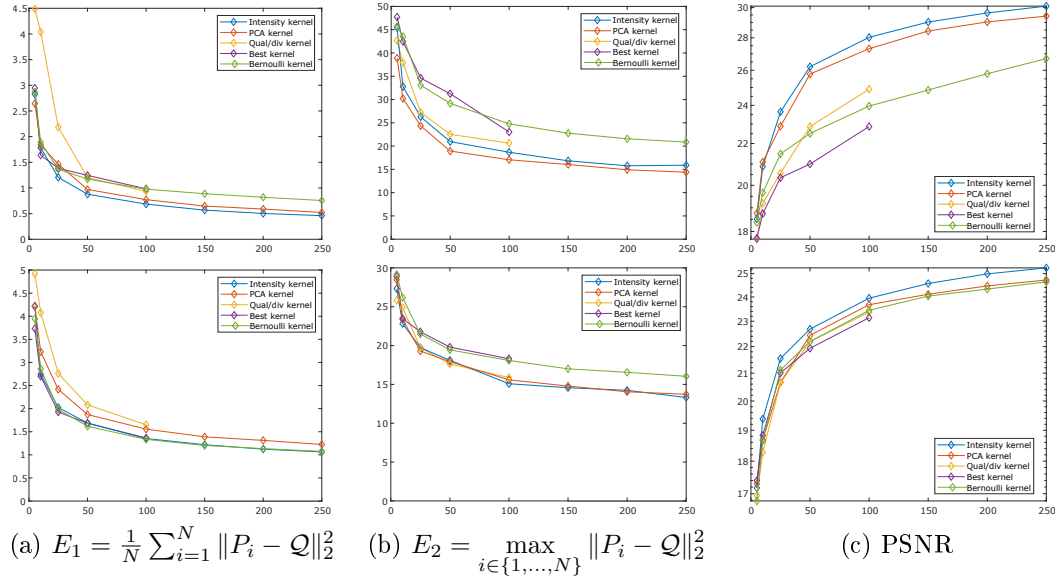


Figure 4.4: Reconstruction errors E_1 and E_2 and the PSNR for the Pool image (top) and the Parrot image (bottom), comparing several DPP kernels and a uniform selection (Bernoulli kernel) in function of different expected cardinality, from 5 to 250, with a step of 50. Note that the curves associated to the “Qual/div” and the “Best” kernels stop at an expected cardinality equal to 100 selected patches due to the rank of their kernel matrix equal to 147.

over-represented in the results. Note that when we compare the kernels using the error E_1 , in particular for the second image (Parrot), the uniform selection provides satisfying results. On the contrary, small and rare details are often missed by the uniform sampling, and the second graph of Figure 4.4 shows that this sampling strategy compares badly with the others when considering this criteria. Furthermore, the graph presenting the PSNR results illustrates how this uniform strategy provides overall poorer reconstructed images.

Note that the optimized kernel, making a compromise between the diversity induced by DPPs and the redundancy imposed by maximizing the chosen reconstruction error (4.8), produces quantitative results similar to a uniform sampling. When observing the patches selected by this kernel in Figures 4.2 and 4.3, one can see that this kernel tends to select slightly more diverse patches than a uniform sampling.

Second, the PCA kernel and the Qual-div kernel behave rather similarly. They tend to favor singular patches and patches containing edges, even sometimes over-representing them. Thus, they provide good results when looking at the second error measuring the distance between the selection and the furthest

patch, especially the PCA kernel. Yet, they can provide even worst results than the uniform selection when we look at the average distance between the selection and the initial set of patches (Error E_1 (4.6)).

Finally, the Intensity kernel, using only the squared Euclidean distance between intensities, seems to be the most stable kernel. It provides small average error and tends to include singular patches in the selection. For both images, whatever the expected cardinality, the samples generated by this kernel produce visually satisfying reconstructions.

Thus, the choice of subsampling strategy in the patch space of an image highly depends on the purpose of the generated selection. The most stable strategy seems to be using the Intensity kernel (4.3), which provides a selection close in average to the initial patches and which selects also singular patches. If the priority of the application is efficiency, the best strategy may remain to use a uniform selection with a high number of selected patches. Yet if the size of the selection needs to be low or if the selection needs to contain mainly structure and texture information, the good choice may be to use a PCA kernel or a kernel using the quality-diversity decomposition.

4.3 Application to a Method of Texture Synthesis

The study carried out in this section is a joint work with Arthur Leclaire and is presented in the proceedings [84]. We build on the texture model proposed in [52], which exploits optimal transport (OT) in the patch space in order to reimpose statistics of local features at several resolutions. This model is based on semi-discrete OT, meaning that it uses transformations of the patch space that are designed to optimally transport an absolutely continuous source measure onto a discrete target measure. The chosen discrete target measure in [52] is the subsampled empirical patch distribution of the exemplar texture, so that these OT maps help to reimpose the patch statistics of the exemplar. These OT maps are given by weighted nearest neighbor (NN) assignment on the points of the target measure support. Therefore, the computational time for synthesis highly depends on the discrete sampling of the target distribution. For 3×3 patch distributions, a naive 1000-uniform subsampling gives good results in general. But more accurate subsampling strategies could be used by taking profit of the structure in the patch point cloud.

Here we propose to use a different subsampling strategy based on determinantal point processes (DPPs) defined on patches. We propose to integrate the DPP subsampling strategy in the OT-based texture model of [52]. We show that because of the repulsion property of the DPP, it is able to cover

efficiently the original patch cloud with a low number of samples. As a result, the obtained transport maps can be applied faster, thus allowing to synthesize very large textures with competitive computational time. We also discuss the parameters of the model, in particular the expected cardinality of the DPP, which should depend on the complexity of the input texture.

4.3.1 Texture Synthesis with Semi-Discrete Optimal Transport

In this section, we will recall the definition given in [52] of the texture model based on semi-discrete optimal transport. Let $u : \Omega \rightarrow \mathbb{R}^d$ be the exemplar texture defined on a domain $\Omega \subset \mathbb{Z}^2$. As before, the patch domain will be denoted by $\omega = \{-\rho, \dots, \rho\}^2$ and the associated patch space by \mathbb{R}^D where $D = d(2\rho + 1)^2$.

Monoscale Model

The model is based on a coarse synthesis obtained with a Gaussian random field U , which is called the asymptotic discrete spot noise (ADSN) associated with the texture u [48]. We have seen this model before, in Section 3.3.3, as the limit distribution of Poisson discrete shot noise models. Associated to the texture u , it is defined by

$$\forall x \in \mathbb{Z}^2, \quad U(x) = \bar{u} + \sum_{y \in \mathbb{Z}^2} t_u(y) W(x - y) \quad (4.16)$$

where $\bar{u} = \frac{1}{|\Omega|} \sum u(x)$, $t_u = \frac{1}{\sqrt{|\Omega|}}(u - \bar{u})\mathbf{1}_\Omega$ and W is a normalized Gaussian white noise on \mathbb{Z}^2 . However, this Gaussian random field model U is only adapted to the synthesis of unstructured textures. Figure 4.5 shows the ADSN associated to several textures. Note that the first one, which belongs to the micro-textures family, is the only well synthesized texture.

For that reason, the authors of [52] proposed to apply local modifications to reinforce geometric structures in a statistically coherent way. In other words, a transformation $T : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is applied to all the patches of U , an image is recomposed by simple averaging, thus obtaining the transformed random field

$$\forall x \in \mathbb{Z}^2, \quad V(x) = \frac{1}{|\omega|} \sum_{h \in \omega} T(U_{[x-h+\omega]})(h). \quad (4.17)$$

The map T is chosen to solve a semi-discrete optimal transport problem between the probability distribution μ of the patches of U and a discrete target distribution $\nu = \sum_{j=1}^J \nu_j \delta_{Q_j}$ representing the patches of u (that we define in

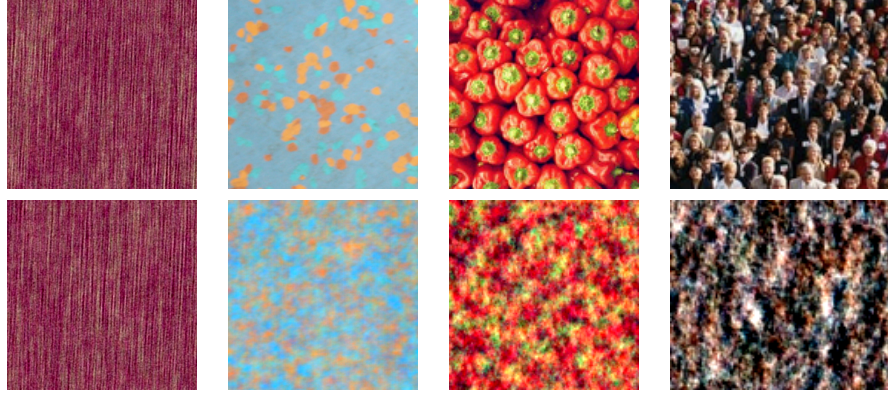


Figure 4.5: Examples of asymptotic discrete spot noise synthesis (4.16). First row: Input textures. Second row: Synthesis.

Section 4.3.2). This problem can be written as

$$\inf \int_{\mathbb{R}^D} \|P - T(P)\|^2 d\mu(P) \quad (4.18)$$

where the infimum is taken over all measurable maps T for which the image measure of μ is ν . As proved in [7, 76], the solution can be obtained as a weighted Nearest Neighbour (NN) assignment

$$T_v(P) = Q_{j(P)} \quad \text{where} \quad j(P) = \underset{j}{\operatorname{argmin}} \|P - Q_j\|^2 - v_j \quad (4.19)$$

where $v \in \mathbb{R}^J$ solves a concave maximization problem. Solving for v relies on a costly stochastic gradient procedure (see the details in [60, 52]) which is more and more difficult when the number J of points in the target distribution increases. This is a first reason to look for a simplification of the target measure ν with the least possible points. Another reason, which will be highlighted in the experimental section, is that once the map T_v is estimated, applying it to all patches of U amounts to applying a weighted NN projection on a set of J patches; thus the required computational time for synthesis also depends on the number J of points in the target distribution.

This monoscale model (only one scale of patches) is summarized in Figure 4.6. Given the input texture u , and a discrete distribution ν representing its patches, a Gaussian random field U is generated, providing a coarse approximation of the texture. The continuous distribution of the patches of U is denoted by μ . A transformation T_v is estimated so that the image distribution of μ is ν . After applying T_v to the patches of U , they are aggregated by averaging, to obtain the texture V .

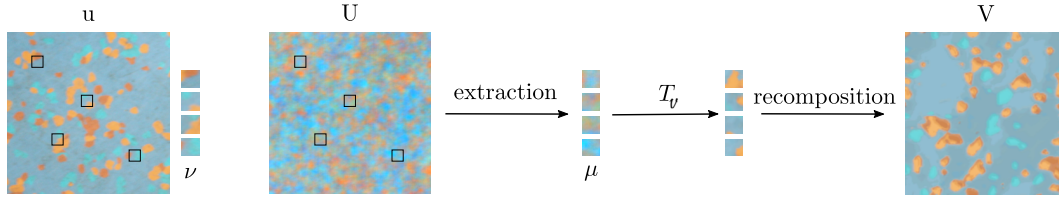


Figure 4.6: Monoscale model defined in [52] using semi-discrete optimal transport for texture synthesis, from u the input texture.

Multiscale Model

One drawback of the stochastic algorithm for semi-discrete OT is that it gets slower when the dimension D increases. In practice, it is thus only applicable for patches of size 3×3 . A multiscale extension was proposed in [52] in order to deal with larger structures. It consists in working with subsampled versions $u^\ell, \ell = 0, \dots, L-1$ of the original texture defined on coarser grids $\Omega^\ell = \Omega \cap 2^\ell \mathbb{Z}^2$, and with discrete target patch distributions $\nu^\ell, \ell = 0, \dots, L-1$.

Starting from a Gaussian random field U^{L-1} estimated from u^{L-1} as in (4.16), for $\ell = L-1, \dots, 0$, we apply a transport map T^ℓ to all patches of U^ℓ

$$V^\ell(x) = \frac{1}{|\omega|} \sum_{h \in 2^\ell \omega} T^\ell(U^\ell_{|x-h+2^\ell \omega})(h), \quad x \in 2^\ell \mathbb{Z}^2 \quad (4.20)$$

and we get $U^{\ell-1}$ by exemplar-based upsampling (taking the same patches than T^ℓ but twice larger). The transport map T^ℓ is designed to solve a semi-discrete OT problem between a source measure μ^ℓ (a GMM estimated from the patches of the current synthesis) and a discrete target distribution ν^ℓ representing the patches of u^ℓ . The output texture is V^0 .

One strong feature of this multiscale model is that the maps T^ℓ can be estimated once and for all. Once the model estimated, it can be sampled efficiently since applying the map T^ℓ at each scale consists in a simple weighted NN projection on 3×3 patches.

4.3.2 DPP Subsampling of the Target Distribution

In this subsection, we discuss how to choose the discrete target distribution ν in order to represent efficiently the patches of the original texture u .

Choosing the Target Distribution

One natural choice to represent all the patches of u is of course to consider the empirical distribution

$$\nu_{\text{emp}} = \frac{1}{N} \sum_{i=1}^N \delta_{P_i} \quad (4.21)$$

where $\mathcal{P} = \{P_i, 1 \leq i \leq N\}$ is the set of all patches of u . Unfortunately, this choice must often be discarded because the number N of patches is in general very large ($N \gg 10^5$) and thus unsuitable for the stochastic algorithm for semi-discrete OT.

The authors of [52] coped with this problem by considering the simple subsampling

$$\nu_{\text{unif}} = \frac{1}{J} \sum_{j=1}^J \delta_{Q_j} \quad (4.22)$$

where the patches (Q_j) are chosen at random (uniformly) among the patches \mathcal{P} . Although naive, this solution proved to be sufficient for many textures, with a value of J set as a ground rule to $J = 1000$ for subsampling 3×3 patch distributions.

However, as mentioned above, the size J of the support of the target distribution highly impacts the execution times of the estimation of the model and of the synthesis step. That is the reason why we propose here to consider alternative choices in order to use even lower values of J while maintaining the visual quality of the output texture.

We want to approximate the empirical distribution with a discrete distribution with support of size J

$$\nu = \sum_{j=1}^J \nu_j \delta_{x_j} \quad (4.23)$$

where $x_j \in \mathbb{R}^D$, for all $j = 1, \dots, J$ and whose weights (ν_j) belong to the probability simplex, meaning that $\forall j \leq J, \nu_j \geq 0$ and $\sum_{j=1}^J \nu_j = 1$. One can formulate this problem using the L^2 -Wasserstein distance between discrete probability distributions $\mu = \sum_{i=1}^N \mu_i \delta_{y_i}$ and $\nu = \sum_{j=1}^J \nu_j \delta_{x_j}$ defined by

$$W_2^2(\mu, \nu) = \inf_{(\pi_{i,j})} \sum_{i,j} \pi_{i,j} \|y_i - x_j\|^2 \quad (4.24)$$

where the infimum is taken on $(\pi_{i,j}) \in \mathbb{R}_+^{N \times J}$ such that for all i , $\sum_j \pi_{i,j} = \mu_i$ and for all j , $\sum_i \pi_{i,j} = \nu_j$. Approximating ν_{emp} with a discrete distribution amounts to find ν minimizing the Wasserstein distance

$$\nu^* = \underset{\nu}{\operatorname{argmin}} W_2^2(\nu_{\text{emp}}, \nu). \quad (4.25)$$

Note that solving this optimization problem is actually equivalent to solving a k -Means clustering problem [105, 32]. In [32], the authors propose an algorithm to solve, among more general issues, the optimization problem (4.25)

and state that this method is equivalent to Lloyd's algorithm [91], the common k -Means clustering algorithm. Note that this problem is non convex and that Lloyd's algorithm only provides a local minimum. More importantly, in this image framework, we have a supplementary constraint: we want the points $x_i \in \mathbb{R}^D$ defining the support of ν^* to be part of the initial patches of the texture. Indeed, the k -Means algorithm may create blurry patches, that do not belong to the input texture and that would be unsuited to represent it.

Thus, in the following, we propose to fix the support of the distribution ν and to define it as the realization of a DPP, so that the resulting support represents the set of patches of the input texture.

Setting the Weights

In the following, we select a subset of patches of the input texture u using a DPP. Given a DPP kernel K , we denote by $\mathcal{Q} \sim DPP(K)$, a random subset of patches. The choice of the DPP kernel K is our main concern here and it will be discussed in the next paragraphs.

Once the support $\mathcal{Q} = \{Q_j, 1 \leq j \leq J\}$ has been fixed, one must build a measure ν supported on \mathcal{Q} that accurately represents the patches of u . This amounts to adjusting the masses (ν_j) associated with (Q_j) such that

$$\nu = \sum_{j=1}^J \nu_j \delta_{Q_j} \quad (4.26)$$

realizes a good approximation of ν_{emp} .

As before, one can use the L^2 -Wasserstein distance to determine ν . Finding the masses (ν_j) that minimizes $W_2^2(\nu_{\text{emp}}, \nu)$ is equivalent to solving

$$\pi_{i,j}^* = \underset{(\pi_{i,j})}{\operatorname{argmin}} \sum_{i,j} \pi_{i,j} \|P_i - Q_j\|^2 \quad (4.27)$$

such that $\forall(i, j), \pi_{i,j} \geq 0$ and $\sum_j \pi_{i,j} = \frac{1}{N}$, which is similar to the original OT problem, but relaxing the second marginal constraint. The solution ν can thus be obtained with

$$\forall j \in \{1, \dots, J\}, \nu_j^* = \sum_i \pi_{i,j}^*. \quad (4.28)$$

This is simply a linear programming problem with the projection on a simplex that can be solved with the ‘‘Interior point’’ or ‘‘Dual simplex’’ algorithms. Finally we approximate the empirical distribution with the (random) distribution

$$\nu_{\text{DPP}} = \sum_{j=1}^J \nu_j^* \delta_{Q_j}, \quad (4.29)$$

where $\mathcal{Q} = \{Q_j, 1 \leq j \leq J\}$ is a realization of the DPP with kernel K .

Choice of a DPP kernel

One needs to choose a DPP kernel such that the selected subset of patches provides a good approximation of the empirical distribution of the patches of u . To do so, we compare the different kernels presented in the previous section, using texture images.

Let us define one more evaluation measure, using the Wasserstein distance between the empirical distribution ν_{emp} and the approximation ν_{DPP} presented above. In practice, we want the DPP kernel that minimizes the error:

$$E_3 = W_2^2(\nu_{\text{emp}}, \nu_{\text{DPP}}) = \sum_{i,j} \pi_{i,j}^* \|P_i - Q_j\|^2. \quad (4.30)$$

Figure 4.7 compares the kernels introduced in Section 4.2 and applied to subsample the set of patches of several textures (used in the experiment section). These graphs display the errors E_2 (4.7), E_3 (4.30) and the PSNR (4.15), computed, for each kernel, by averaging the results obtained from 9 texture images and, for each image, from several samples. One can notice that the PCA kernel (4.4) and the Intensity kernel (4.3) seem to behave in a more satisfying way than the other kernels and in general their quantitative results are similar. As we have seen before, in general, the PCA kernel produces more diverse subsets, with singular patches. For most textures, this kernel is the one minimizing E_2 (4.7) the error computing the maximum distance between the selection and the rest of the patches.

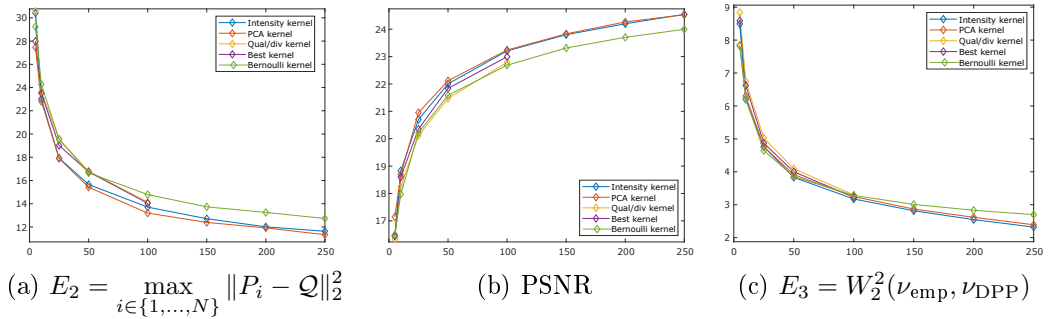


Figure 4.7: Error E_2 (4.7), PSNR (4.15) and Error E_3 (4.30) comparing several DPP kernels, using 9 different texture images.

Thus, in the following, we choose to use a DPP generated by the PCA kernel introduced previously. Let us recall that every patch $P_i \in \mathcal{P}$ is associated with a vector $P_i^k \in \mathbf{P}^k$ given by keeping only the k principal components after a Principal Component Analysis (PCA), and we define the likelihood kernel by

$$\forall P_i, P_j \in \mathcal{P}, \quad L_{ij} = \exp \left(-\frac{\|P_i^k - P_j^k\|_2^2}{s^2} \right), \quad (4.31)$$

where s is the median of the interdistances between the patches and $k = 10$.

As we have seen in Chapter 2, the exact algorithms to sample DPPs presented in this manuscript cost $\mathcal{O}(N^3)$, which is very costly since in general N is large. Yet, we only need to perform this sampling once (at every scale) and as it enables to significantly reduce the number of patches used to estimate the target distribution, we will see in the next section that this cost can be afforded. Algorithm 6 presents the steps of the whole texture synthesis algorithm using semi-discrete optimal transport and DPP subsampling to synthesize textures. Note that, given a texture, once a first synthesis has been done, the model is estimated and stored. For all subsequent synthesis of the same texture, one only needs to do the steps written in *italic* in Algorithm 6.

Algorithm 6 Semi-discrete OT algorithm for texture synthesis, using DPPs.

Input: Exemplar u , number of scales L .

1. Preprocessing:
 - Define subsampled versions of u , u^0, \dots, u^{L-1} .
 - At each scale l , select a subset of patches \mathcal{Q}^l using $\text{DPP}(K^l)$ (4.31) defined on u^l .
 - At each scale l , compute ν^l , representing the patch distribution of u^l (4.29).
2. *Define U^{L-1} a Gaussian synthesis (4.16).*
3. *At each scale $l = L - 1, \dots, 0$,*
 - Estimate μ^l as a Gaussian mixture model from U^l (except at scale $L - 1$ where we already know the Gaussian distribution of U^{L-1}).
 - Compute the weights v^l (4.19) using a stochastic gradient descent algorithm and compute the optimal transport map T_v^l .
 - *Apply the map to the patches of U^l , which consists in a weighted nearest neighbor projection on \mathcal{Q}^l , to obtain V^l .*
 - *If $l \neq 0$, exemplar-based upsampling of V^l to obtain U^{l-1} .*

Output: Synthesized texture V^0 .

4.3.3 Results

We now comment the synthesis results obtained by subsampling the target patch measures with DPPs. All parameters of the texture model are set to the default values listed in [52] (4 scales, patches of size 3×3). The only

difference lies in the subsampling strategy. At each scale, a first naive subsampling is performed by drawing (uniformly) 1000 patches in the exemplar texture. Then, a second subsampling step is performed with either another uniform subsampling to cardinality J or a DPP subsampling with expected cardinality J . Let us mention that we cannot use a direct DPP subsampling of ν_{emp} because the total number of patches N is often very large ($\approx 10^6$) and it would be very slow to sample from a DPP kernel that large. In the following experiments, $J \in \{50, 100, 200\}$.

First, note that the evaluation of the quality of a texture synthesis relies usually on human visual assessment. Unlike denoising methods, that can be evaluated using the PSNR (4.15) for instance, it is difficult to objectively and systematically quantify the quality of a generated texture. This is partly due to the wide diversity of texture images. Thus, in the following, we are only able to visually assess the quality of the syntheses.

In Figure 4.8, one can observe a predictable loss of quality when going from 1000 to 100 patches. However, one can see that for many textures, the visual quality can be maintained to a reasonable level while using 10 times less patches. This will help us to reach a compromise between visual quality and execution time for synthesis (see below). One can also observe on Figure 4.8 that uniform and DPP subsampling behave quite differently. In particular, DPP subsampling seems to favor patches with sharper edges and less noise. Also, on several textures (like the last example of Figure 4.8), the output seems statistically closer to the input texture; but it would require a more involved analysis to precisely assess this fact. Let us remark that this statistical consistency crucially relies on the precise estimation of the weights explained in Section 4.3.2.

In Figure 4.9, we analyze the influence of the cardinality of the target discrete distribution. One can observe that for each texture there is a cardinality value, which mainly depends on the complexity and the geometric components in the texture, under which results get visually degenerate and over which the visual quality is maintained to a reasonable level.

Finally, let us highlight the main benefit obtained with the proposed subsampling strategy, which lies in the gain in computation time for synthesis. Once the texture model is estimated, it is indeed very fast to sample large pieces of it, and since it relies on weighted NN assignments at each scale, the execution time depends quasi-linearly on the cardinality J of the target measures. Using a CPU Intel i7-5600U (4 cores at 2.6GHz), for $J \leq 200$ we are able to synthesize 512×512 images in $\approx 0.4''$ and 1024×1024 in $\approx 1.6''$. This execution time can be improved using a GPU implementation: Table 4.1 provides the running times for the synthesis of 1024×1024 textures, for several

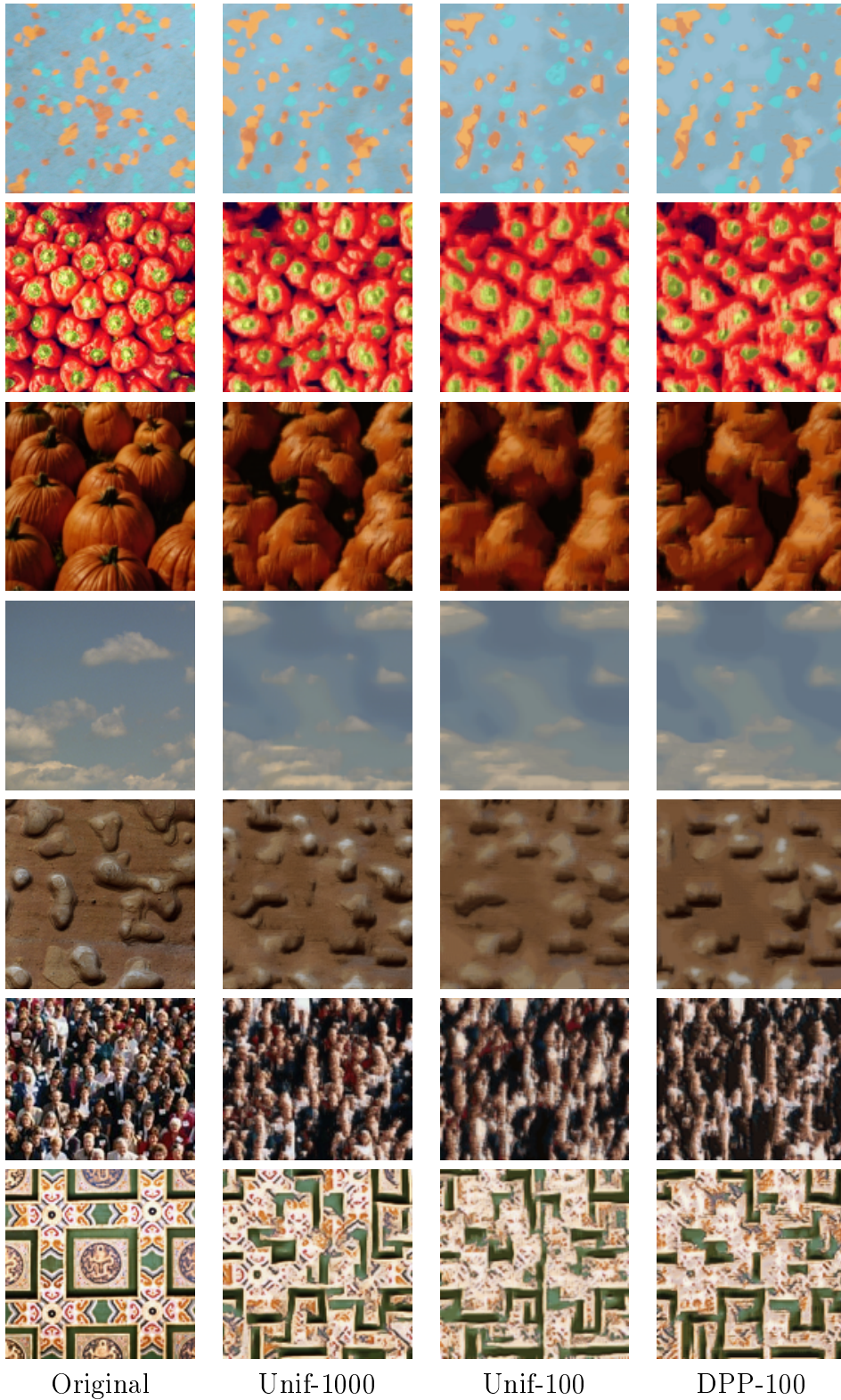


Figure 4.8: Visual comparison of the synthesis results when using either a target distribution with uniform subsampling (with cardinality 100) and DPP subsampling (with expected cardinality 100). See the text for comments.

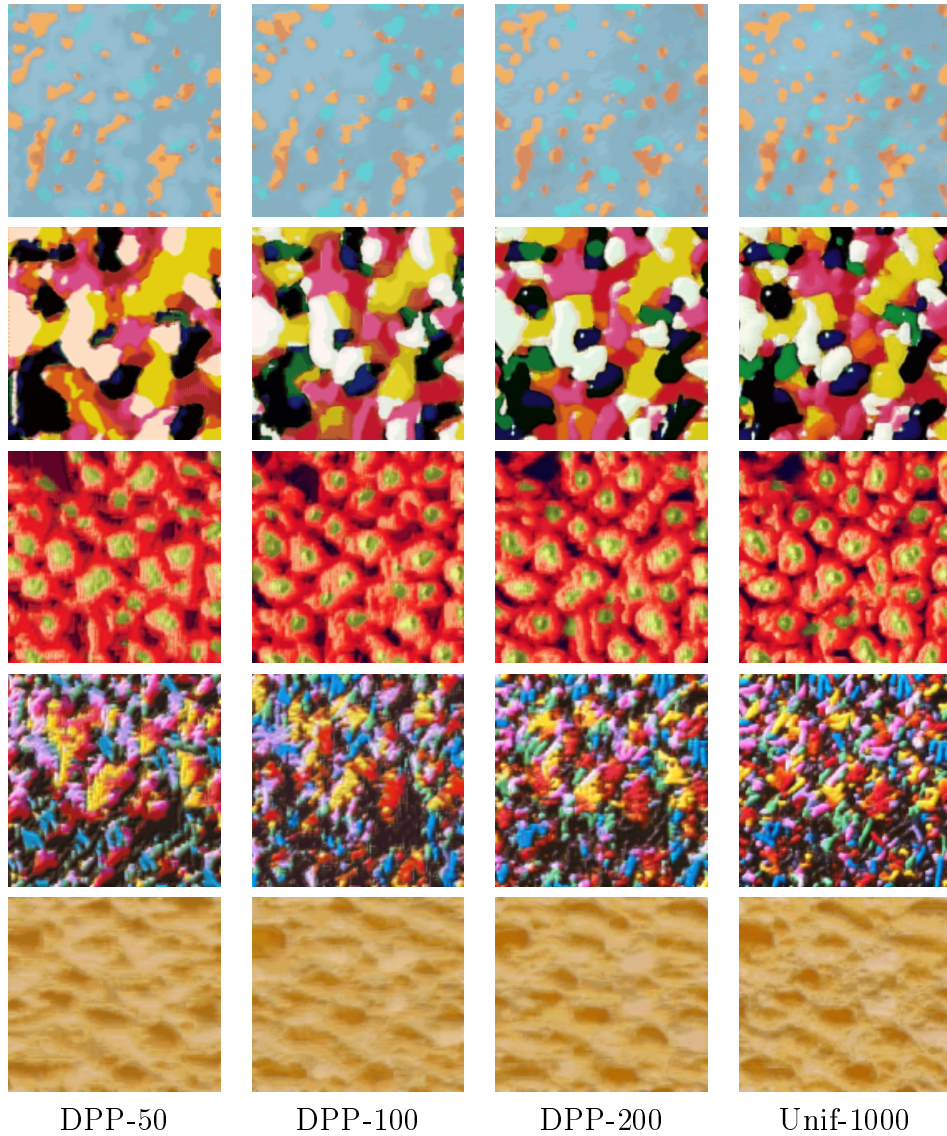


Figure 4.9: We display the visual impact of the expected cardinality of the DPP on the results. See the text for comments.

values of J . One can notice that these computation times are close to the state of the art values obtained in [62] for structured textures.

Figures 4.10 and 4.11 present some experiments comparing the synthesis of 720×512 textures using the initial algorithm [52], using 1000 patches to represent the patch distribution, and our adaptation using a DPP subsampling of the set of patches. Observe that for most textures the visual quality seems satisfying. Yet, one can notice a loss of quality between the uses of 1000 and 100 patches, concerning the syntheses from the third and fourth textures of Figure 4.10. These textures contains larger geometric structures or large

J	50	100	200	1000
Running time	0.19"	0.28"	0.47"	1.7"

Table 4.1: **Execution time for synthesis** depending on the number J of points in the patch target distributions. These execution times have been obtained with a GPU implementation.

repeated patterns, and a selection of 100 patches appears to be too small to retrieve such content. The suggested approach thus allows to accelerate the synthesis algorithm of [52] while maintaining the quality of synthesis. Note that a MATLAB implementation of this adapted algorithm (for CPU and GPU) is available online¹.

4.4 Conclusion

In this chapter, we investigated the use of determinantal point processes to subsample the set of patches of an image. We presented several DPP kernels adapted to the representation of an image and compared them using several evaluation measures. It appears that the choice of the kernel highly depends on the purpose of the generated selection. The most stable strategy seems to be using the Intensity kernel, which provides a selection both close in average to the initial patches and containing singular patches.

We proposed an alternative strategy to subsample the set of patches of a texture and to approximate its empirical distribution. This method was applied to a texture synthesis model using semi-discrete optimal transport. The resolution of this OT problem involves a weighted nearest neighbor assignment, computed using a slow stochastic gradient procedure. Thus, the execution times of the estimation of the OT map as well as its application highly depend on the size of the support of the discrete patch distribution. That is why we proposed here to approximate the patch distribution using DPP subsampling. Considering textures, the PCA kernel, along with the Intensity kernel, provides appealing subsets of patches. As it also tends to select more singular patches, we choose to use this PCA kernel in the texture synthesis algorithm. The execution time of the synthesis is significantly shortened because of the possibility for the estimated patch distribution to have a reduced support. This strategy proposes a compromise between synthesis quality and execution speed.

Because of the stochastic gradient descent needed to solve the OT problem,

¹<https://www.math.u-bordeaux.fr/~aleclaire/texto/>

the patches can't be too large. In practice, Galerne et al. [52] use 3×3 patches and, in this study, so do we. However, Leclaire and Rabin [86] recently developed a multi-layer version of the optimal transport resolution. This method enables the use of patches of size 7×7 , which improves the synthesis of textures with geometry and large scale structures. We would like to adapt the DPP subsampling done here to this multilayer algorithm to speed it up and analyze more precisely the consequences of the estimation of the textured patch distribution using DPPs.

Notice also that whereas some textures can be represented and synthesized using very few patches, for some complex textures, 100 or 200 patches may not be enough to accurately approximate them. It would be interesting to develop a criterion related to the complexity of the texture, determining the approximate number of patches needed to represent it.

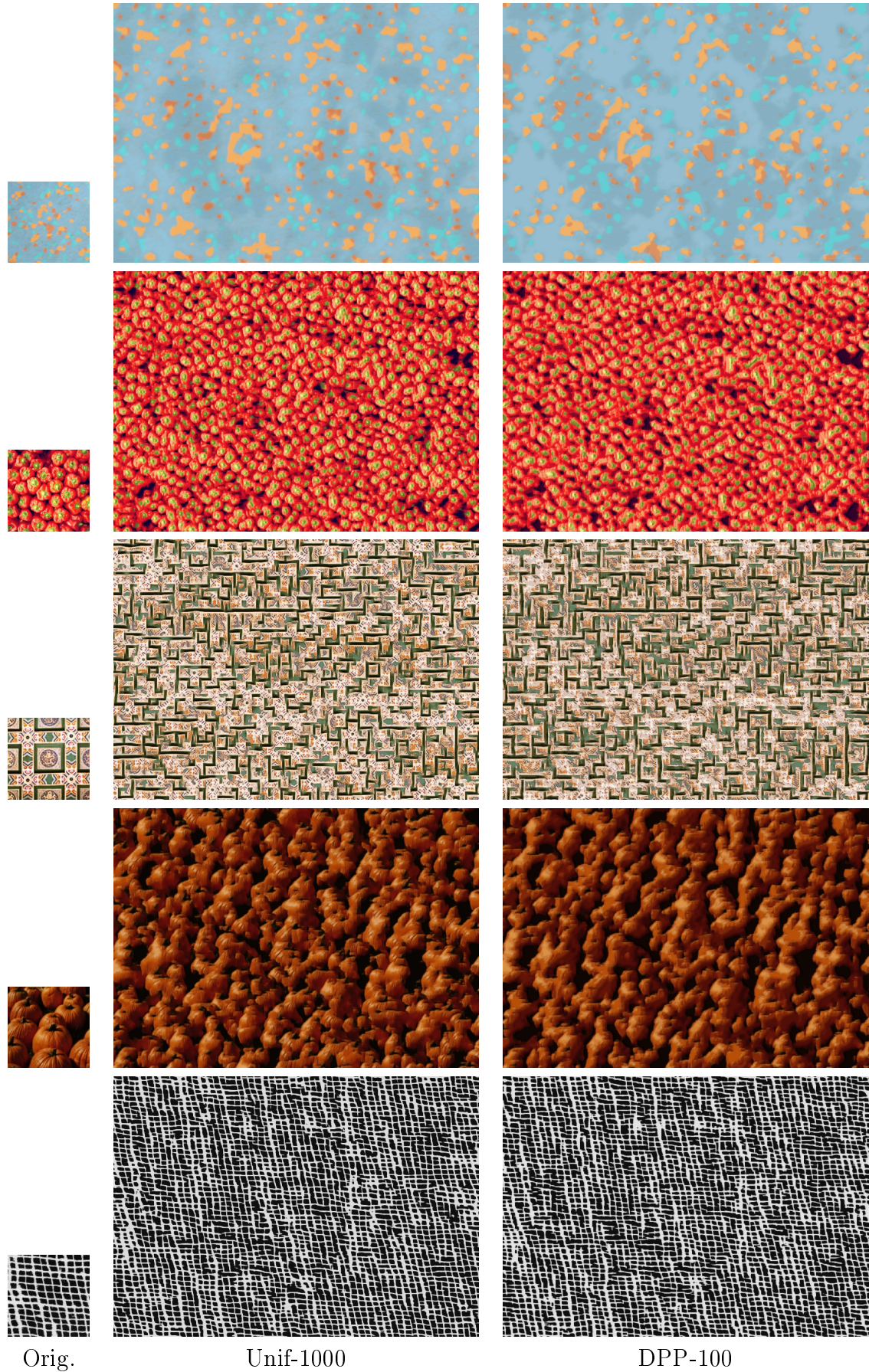


Figure 4.10: We compare the synthesis results when using either a uniform subsampling (with cardinal 1000) or a DPP subsampling (with expected cardinal 100).

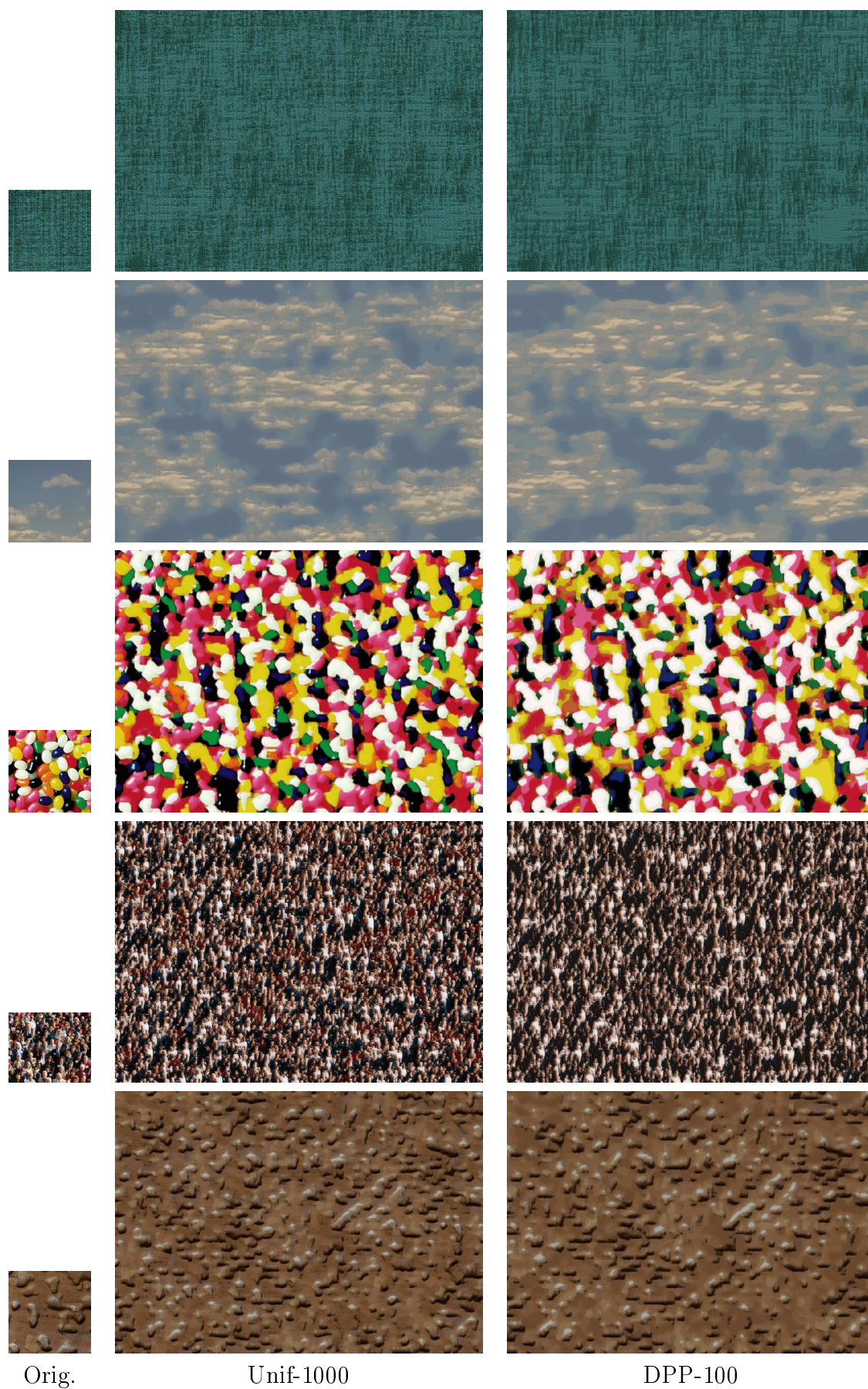


Figure 4.11: Same as Figure 4.10.

Chapter 5

Conclusion and Perspectives

Contents

5.1	Exact Determinantal Point Processes Sampling	127
5.2	Determinantal Pixel Processes	129
5.3	Determinantal Point Processes on Patches	131

This thesis focused on discrete determinantal point processes and on their application to image processing. We wanted to use the ability of DPPs to model repulsive phenomena or to subsample sets of data while enforcing diversity in the sample. These properties have been explored when the point process is defined on the pixels or the patches of an image. This chapter presents a synthesis of the main contributions of this manuscript. We also mention perspectives that we would like to explore for future research.

5.1 Exact Determinantal Point Processes Sampling

In Chapter 2, we focused on sampling general determinantal point processes. We developed two new sampling algorithms, that we call the sequential sampling algorithm (Algorithm 2) and the sequential thinning algorithm (Algorithm 3). Both algorithms are exact, adapted to general determinantal point processes and, unlike the usual exact sampling algorithm, they don't use the spectral decomposition of the kernel. MATLAB and Python implementations of the sequential thinning algorithm can be found online¹. Algorithm 2 relies

¹https://www.math-info.univ-paris5.fr/~claunay/exact_sampling.html

on the sequential computation of pointwise conditional probabilities from a DPP kernel. The sampling is sped up using updated Cholesky decompositions to compute the conditional probabilities. This strategy is simple but it is not competitive with usual sampling methods.

We use the *thinning* of a point process to reduce the execution time of the sampling. This new sampling algorithm proceeds in two phases. The first one draws a Bernoulli process whose distribution is adapted to the target DPP. We ensured that the generated point process contains the DPP and it is constructed so that its cardinality is the closest to the cardinality of the target DPP. This step is fast and efficient and it significantly reduces the initial number of points of the ground set. Moreover, if $I - K$ is invertible, the expectation of the cardinality of the Bernoulli process is proportional to the expectation of the cardinality of the DPP. The second phase uses the previous sequential sampling based on the points selected by the Bernoulli point process. This sequential strategy is not efficient, that is why it is crucial that the first step reduces the size of the initial state space as much as possible.

We have illustrated the behavior of these two algorithms with numerical experiments and compared their running times with the spectral algorithm. We have detailed the cases for which the sequential thinning algorithm is competitive with the spectral algorithm, in particular when the size of the ground set is high and the expected cardinality of the DPP is modest. This framework is common in machine learning applications.

To pursue this work, we would like to explore new methods to further accelerate our sampling algorithm. In his thesis [55], Guillaume Gauthier proposed an alternative computation of the Bernoulli probabilities (2.31), defining the distribution of the dominating Bernoulli process used in the first step of the sequential thinning algorithm. His formula avoids the inversion of a triangular matrix and thus accelerates the first part of the algorithm. Furthermore, using specific matrix factorization techniques and parallelizations, Poulson [109] developed an efficient sampling algorithm that relies on same conditional probabilities as our sequential algorithm (Algorithm 2). The author states that these speedups bring important gains in terms of running times to our sequential thinning algorithm (Algorithm 3). We would like to further investigate these speedups and similar factorization strategies, to understand to what extent a modified sequential thinning algorithm would be more efficient and to study other possible improvements.

Another promising perspective would be to extend this strategy to continuous DPPs, defined on a continuous state space. Indeed, the thinning procedure we use comes from a continuous setting. We would like to examine the adaptation of the rest of the algorithm to a continuous framework. Continuous DPPs appear in the distribution of the spectrum of Gaussian random matrices in probability or in the location of fermions in quantum mechanics, for instance.

The common exact sampling algorithm for continuous DPPs is given by Hough et al. in [72] and still relies on the characterization of a DPP as a mixture of projection DPP. Scardicchio et al [118] and Lavancier et al. [85] provide more efficient implementations based on the previous sampling algorithm, in particular for the simulation of the Bernoulli variables. These strategies still use the eigendecomposition of the kernel. Furthermore, some authors, such as Decreusefont et al. [37], use a MCMC strategy and the method called “coupling from the past” to draw a continuous DPP. They call this method “perfect simulation” as it reaches the target distribution in a finite time.

Sampling exactly continuous DPP models is a much more challenging problem than sampling discrete DPPs. The main reasons that the domains are often infinite, and more importantly, because the eigendecomposition of the kernel operator generally involves an infinite number of eigenvalues. Yet we hope that adapting the sequential thinning procedure (Algorithm 3) may provide an adequate and efficient sampling procedure for some continuous DPP models.

5.2 Determinantal Pixel Processes

In Chapter 3, we adapted the definition of DPPs to the set of the pixels of an image. Such a DPP is defined on the image domain Ω and is called a determinantal pixel process (DPixP). In this setting, and with the application to texture synthesis in mind, the stationarity and the periodicity of the point process are natural hypotheses. We showed that the only possible hard-core repulsion for DPixP is directional. Given a direction, it is possible to impose to select at most one pixel on any discrete line with this direction in the image. In Section 3.3, we studied shot noise models based on DPixP as a method to sample micro-textures. We developed a method to adapt the DPixP kernel to a given spot function and to the regularity one is looking for. The regularity of the shot noise, that can be seen as a specific type of repulsion adapted to the spot function, is related to the variance of the shot noise. This quantity depends on the spot function and on the DPixP kernel. It appears that the least repulsive DPixP, which generates the least regular textures and which maximizes the variance of the shot noise, is the homogeneous Bernoulli process. In that case, the kernel is independent of the spot function. On the other hand, the most repulsive DPixP kernel, generating regular textures and minimizing the variance of the shot noise, is a projection kernel which is solution to a combinatorial problem depending on the spot function. Considering the associated shot noise models enables getting closer to a hard-core repulsion.

Thus, in Section 3.2, we proved that it is not possible to avoid overlaps if we randomly copy and place a given shape using a DPixP, unlike particular Gibbs processes. However, in Section 3.3, we saw that, given a shape (the

spot function), it is possible to derive a DPixP kernel so that there are as few overlaps as possible. This property may be interesting for computer graphics issues especially since DPixPs have elegant theoretical properties. Notice that our algorithm to retrieve the “minimal variance” kernel, a kernel minimizing the number of overlaps, is greedy and is not optimal. Further research would be needed to develop an algorithm more efficient. Furthermore, we would like to look for a theoretical bound on the number of overlaps in shot noise models based on this DPixP and on a given shape.

Note that one of our initial motivations was to reduce the number of spot overlaps in the shot noise model. This goal is achieved using DPixPs and their repulsive nature, by choosing a kernel adapted to the spot. Another motivation could be to generate more “contrasted” textures from shot noise models containing clusters of patterns. As a future work, we would like to explore shot noise models based on attractive point processes, such as Cox processes. It would be interesting to derive properties similar to those we obtain with DPixPs, for instance while studying shot noise models based on permanental point processes, which are considered as the attractive counterpart to determinantal point processes. As for DPPs, it is possible to compute the moments of these point processes. In the continuous case, Blaszczyzyn and Yogeshwaran [16] study shot noise models based on different point process, sorting them according to their repulsiveness. They use these results on shot noise models and Cox processes for wireless networks. Shirai and Takahashi obtain in [120] a law of large numbers, a central limit theorem and a large deviation result for point processes that they call α -determinantal point processes, which gather determinantal and permanental point processes. Thus, one may retrieve similar convergence conclusions for shot noise models based on permanental processes, as the ones proved in Section 3.3.3, and apply those results to texture synthesis. As we have seen in Section 3.3.2, shot noise models based on attractive processes could enhance the contrast of the textures generated, by creating regions with high amount of spot overlaps and regions without any point. We could define an objective function to optimize, such as the variance of the shot noise models, in order to find the optimal kernel of the permanental process in function of the spot function.

In Section 3.4, we endeavored to characterize the equivalence classes of DPP and DPixP kernels, that are families of kernels generating the same distribution. In the DPixP case, the equivalence classes involve translation and symmetry with respect to $(0,0)$ of the Fourier coefficients of the kernels. This question is crucial when dealing with inference, in order to understand what can be retrieved by an estimation algorithm and in order to assess the uniqueness of the solution. We developed an algorithm to estimate the Fourier coefficients of a DPixP kernel from one sample or from a set of samples. This algorithm takes advantage of the stationarity of DPixPs and provides satis-

fying results, particularly when the target kernel is a projection kernel. For instance, we have seen that the algorithm is able to retrieve most of the kernel information using only one sample, for some simple projection kernels.

We plan to investigate the joint estimation, from a texture image, of the spot function and of the DPixP kernel associated to a shot noise that could have generated the texture. Such an algorithm would allow for the reproduction of Gaussian textures or the inference of the model underlying the input texture, in order to retrieve some of the texture properties. Several approaches [40, 50, 51] have focused on this question as they intend to generate, given an input texture image, what they call a *texton*. A texton is a compact representation of the texture, a small texture image, containing the frequency content of the input. In fact, this texton can be seen as a spot function and it is used to reproduce the initial Gaussian texture using a discrete shot noise model, based on a Poisson point process. This whole strategy enables efficient exemplar-based texture synthesis for Gaussian textures. A similar algorithm, retrieving both the texton and the DPixP kernel underlying a given texture could be a promising method to adapt the previous strategies to a wider family of textures.

5.3 Determinantal Point Processes on Patches

In Chapter 4, we studied the use of determinantal point processes to subsample the set of patches of an image. In Section 4.2, we introduced different DPP kernels adapted to the representation of an image and compared them using several evaluation measures. The choice of kernel highly depends on the purpose of the generated selection as each kernel favors different types of selection. The most stable strategy seems to be using the Intensity kernel, which provides a selection both close in average to the initial patches and containing singular patches. On the other hand, the PCA kernel, involving the principal components given by a PCA on the matrix gathering the patches, highly favors patches with edges or textures. Such selections of key patches can serve to represent an image using little memory, if the image is reduced to its size, the small set of selected patches and the vector of indices associating each initial patch to its nearest neighbor in the selection. Such diverse selections can also be applied to initialize the centroids of a clustering algorithm or to estimate the parameters of a model defined on the image, by evaluating them on a small but representative proportion of patches.

Section 4.3 presents an application of these subsampling strategies to a texture synthesis model [52] using semi-discrete optimal transport (OT). We developed an alternative strategy to select a small subset of patches of a texture and to approximate the empirical distribution of the whole set of patches of the image. The initial texture synthesis algorithm begins with the synthesis

of a Gaussian random field adapted to the input texture, having the same second order statistics. Then, it uses semi-discrete optimal transport to impose local features, at several resolutions, to the patches of the Gaussian random field. To do so, the authors need to approximate the discrete distribution of the input texture's patches. Solving the OT problem involves a stochastic gradient descent and in the end, the solution is given by a weighted nearest neighbor assignment between the patches of the Gaussian random field and the considered patches of the input texture.

This algorithm needs to subsample the set of patches of the texture and to approximate as precisely as possible the distribution of the patches. Using a DPP instead of a uniform selection allows for the use of much less patches to represent the texture. Considering textures, the PCA kernel, along with the Intensity kernel, provide appealing subsets of patches. As it also tends to select more singular patches, we chose to use this PCA kernel in the texture synthesis algorithm. Even though sampling a DPP is more costly than sampling a Bernoulli point process, the DPP sampling is done only once, offline, during the analysis part of the algorithm. Moreover, the final reduction of the number of considered patches is decisive both in the analysis part of the algorithm, estimating the model, but most of all in the online part of the algorithm, synthesizing the output texture. The execution time of the synthesis is significantly shortened because of the possibility for the estimated discrete patch distribution to have a reduced support. The experiments show that this strategy propose a compromise between synthesis quality and execution speed. Using ten times less patches than in the initial algorithm allows for accelerating the synthesis by a factor six on a GPU, while for many textures the visual quality of the result is maintained. Note that MATLAB implementations of the initial synthesis algorithm and of the DPP acceleration on CPU and GPU can be found online².

During the computation of the OT solution, the definition of the weights associated to the nearest neighbor assignment needs the use of stochastic optimization strategies. However, these methods are very slow, particularly in high dimension. That is the reason why the authors of [52] use 3×3 patches and, in this study, so did we. Leclaire and Rabin [86] recently developed a multi-layer version of the OT resolution. They approximate the real OT solution by using a hierarchical clustering of the patches and estimate the weights of each cluster and each layer using a tree search strategy, which is very fast. This enables performance gain during the estimation of the model and during the synthesis of the texture. This method allows for the use of large patches (for instance of size 7×7) which capture larger structures in the texture. Thus, this algorithm is able to synthesize complex textures, with large geometric features. To pursue the work done in Chapter 4, we would like to adapt the DPP subsampling

²<https://www.math.u-bordeaux.fr/~aleclaire/texto/>

studied here to this multi-layer strategy to accelerate the synthesis algorithm and analyze more precisely the consequences of the estimation of the textured patch distribution using DPPs. It would also be important to investigate the behavior of the DPP kernels when using larger patches, capturing much more information.

Notice also that whereas some textures can be represented and synthesized using few patches, for some complex textures with geometric structures, 100 or 200 patches may not be enough to accurately approximate their patch distribution. It would be interesting to develop a criterion related to the complexity of the texture, determining the approximate number of patches needed to represent it, so that it is set to the minimum value while maintaining a good visual synthesis. Unfortunately, this issue is as complex as the evaluation of the quality of a texture synthesis. As we have seen previously, there is no widely accepted measure to objectively and systematically assess a texture synthesis. Several papers [90, 36] propose strategies to automatically sort textures, for instance by considering the regularity and the repetition of patterns [90] or the periodicity of the texture [36]. We could rely on similar sorting strategies to evaluate the complexity of a texture, the amount of geometrical structures, the nature of the periodicity, and adapt the synthesis algorithm accordingly.

Appendix A

Explicit General Marginal of a DPP

Contents

A.1	Möbius Inversion Formula	135
A.2	Cholesky Decomposition Update	136
A.2.1	Add a Line	136
A.2.2	Add a Bloc	136

The following appendix is related to Chapter 2 and the computation of general marginals of a DPP. We introduce here the Möbius inversion formula, needed in Section 2.3 and we present several strategies to update a Cholesky decomposition. These methods are used to implement the sequential sampling algorithm (Algorithm 2).

A.1 Möbius Inversion Formula

Proposition A.1.1 (Möbius inversion formula). *Let V be a finite subset and f and g be two functions defined on the power set $\mathcal{P}(V)$ of subsets of V . Then,*

$$\forall A \subset V, \quad f(A) = \sum_{B \subset A} (-1)^{|A \setminus B|} g(B) \iff \forall A \subset V, \quad g(A) = \sum_{B \subset A} f(B), \quad (\text{A.1})$$

and

$$\forall A \subset V, \quad f(A) = \sum_{B \supset A} (-1)^{|B \setminus A|} g(B) \iff \forall A \subset V, \quad g(A) = \sum_{B \supset A} f(B). \quad (\text{A.2})$$

Proof. The first equivalence is proved in [102] for instance. The second equivalence corresponds to the first applied to $\tilde{f}(A) = f(A^c)$ and $\tilde{g}(A) = g(A^c)$. You will find more details on this matter in the book of Rota [115]. \square

A.2 Cholesky Decomposition Update

We describe below various updates for Cholesky decompositions.

A.2.1 Add a Line

We describe here how a Cholesky decomposition of symmetric semi-definite matrix M is computed given the Cholesky decomposition of its largest top left submatrix.

Let M be a symmetric semi-definite matrix of the form

$$M = \begin{pmatrix} A & b \\ b^t & c \end{pmatrix} \quad (\text{A.3})$$

where A is a square matrix, b a column vector, and c a real positive number. We suppose that the Cholesky decomposition of the matrix A is known, that is, $A = TT^t$ where T is lower triangular. The goal is to compute the Cholesky decomposition of the matrix M given T . Set

$$v = T^{-1}b \quad (\text{A.4})$$

$$x = \sqrt{c - v^t v}. \quad (\text{A.5})$$

Then the Cholesky decomposition of M is

$$\begin{pmatrix} T & 0 \\ v^t & x \end{pmatrix}. \quad (\text{A.6})$$

Indeed,

$$\begin{pmatrix} T & 0 \\ v^t & x \end{pmatrix} \begin{pmatrix} T & 0 \\ v^t & x \end{pmatrix}^t = \begin{pmatrix} T & 0 \\ v^t & x \end{pmatrix} \begin{pmatrix} T^t & v \\ 0 & x \end{pmatrix} = \begin{pmatrix} TT^t & Tv \\ v^t T^t & v^t v + x^2 \end{pmatrix} = \begin{pmatrix} A & b \\ b^t & c \end{pmatrix} = M. \quad (\text{A.7})$$

A.2.2 Add a Bloc

To be efficient, the sequential algorithm relies on Cholesky decompositions that are updated step by step to save computations. Let M be a symmetric semi-definite matrix of the form $M = \begin{pmatrix} A & B \\ B^t & C \end{pmatrix}$ where A and C are square

matrices. We suppose that the Cholesky decomposition T_A of the matrix A has already been computed and we want to compute the Cholesky decomposition T_M of M . Then, set

$$V = T_A^{-1}B \quad \text{and} \quad X = C - V^tV = C - B^tA^{-1}B \quad (\text{A.8})$$

the Schur complement of the block A of the matrix M . Denote by T_X the Cholesky decomposition of X . Then, the Cholesky decomposition of M is given by

$$T_M = \begin{pmatrix} T_A & 0 \\ V^t & T_X \end{pmatrix}. \quad (\text{A.9})$$

Indeed,

$$T_M T_M^t = \begin{pmatrix} T_A & 0 \\ V^t & T_X \end{pmatrix} \begin{pmatrix} T_A^t & V \\ 0 & T_X^t \end{pmatrix} = \begin{pmatrix} T_A T_A^t & T_A V \\ V^t T_A^t & V^t V + T_X T_X^t \end{pmatrix} = \begin{pmatrix} A & B \\ B^t & C \end{pmatrix}. \quad (\text{A.10})$$

Appendix B

Convergence of Shot Noise Models Based on DPixP

Contents

B.1	Ergodic Theory	139
B.2	Proof of Proposition 3.3.4 - Law of Large Numbers	141
B.3	Proof of Proposition 3.3.4 - Central Limit Theorem	144

This appendix is related to Chapter 3. Its goal is to prove Proposition 3.3.4, providing convergence results for shot noise based on DPixP defined on \mathbb{Z}^2 , when the grid is refined. We obtain a law of large numbers and a central limit theorem adapted to this framework. Proposition 3.3.4 is adapted from the work of Shirai and Takahashi in [121, Propositions 3.3 and 3.4].

In order to prove these limit theorems, let us recall some results of ergodic theory.

B.1 Ergodic Theory

The following definitions and theorems, along with more details on ergodic theory, can be found in the book of Kallenberg [75].

We will denote a measurable space (S, \mathcal{S}, μ) , T a measurable transformation on S , ξ a random element of S with probability measure μ and θ a shift on S defined by, $\forall x_0, x_1, \dots \in S$, $\theta(x_0, x_1, \dots) = (x_1, x_2, \dots)$. The transformation T is said to be measure-preserving if and only if $T\xi \stackrel{d}{=} \xi$. Moreover, a random element of S ξ is stationary if and only if $\theta\xi \stackrel{d}{=} \xi$.

Definition B.1.1 (Invariant sets and ergodicity). *A set $I \subset S$ is said to be invariant if $T^{-1}I = I$. The class \mathcal{I} of invariant sets in S form a σ -field in S called the invariant σ -field.*

A measure-preserving transformation T is ergodic with respect to μ or μ -ergodic if \mathcal{I} the class of T -invariant sets is μ -trivial, that is if $\mu I = 0$ or $1, \forall I \in \mathcal{I}$. Any random element ξ with distribution μ is said to be ergodic if and only if $\mathbb{P}(\xi \in I) = 0$ or 1 , for any $I \in \mathcal{I}$.

We can now state the ergodic theorems in general cases and under our hypothesis.

Theorem B.1.1 (Ergodic theorem - Von Neumann [103], Birkhoff [19]). *Consider a measurable space S , a measurable transformation T on S with associated invariant σ -field \mathcal{I} and a random element ξ in S where $T\xi \stackrel{d}{=} \xi$. Let $f : S \rightarrow \mathbb{R}$ be a measurable function with $f(\xi) \in L^p$ for some $p \geq 1$. Then*

$$\frac{1}{n} \sum_{k < n} f(T^k \xi) \xrightarrow{n \rightarrow \infty} \mathbb{E}(f(\xi) | \xi \in \mathcal{I}) \text{ a.s. and in } L^p. \quad (\text{B.1})$$

Theorem B.1.2 (Multivariate ergodic theorem - Kallenberg [75] Thm 9.9). *As before, consider a measurable space S and a random element ξ with measure μ in S . Let T_1, \dots, T_d be some measurable, commuting, μ -preserving transformations on S , and some measurable function $f : S \rightarrow \mathbb{R}$ with $f(\xi) \in L^p$ for some $p \geq 1$. Denote \mathcal{I} for the (T_1, \dots, T_d) -invariant σ -field in S . Then*

$$\frac{1}{n_1 \dots n_d} \sum_{k_1 < n_1} \dots \sum_{k_d < n_d} f(T_1^{k_1} \dots T_d^{k_d} \xi) \xrightarrow{n_1, \dots, n_d \rightarrow \infty} \mathbb{E}(f(\xi) | \xi \in \mathcal{I}) \text{ a.s and in } L^p. \quad (\text{B.2})$$

Our framework is 2D and discrete. Here, the random element X is a DPiXP of some kernel C . The measure-preserving transformations we are interested in are the vertical shift or translation of a , T_1 , defined by $T_1(x) = T_1(x_1, x_2) = (x_1 - a, x_2)$ and the vertical shift of b , T_2 , such that $T_2(x) = T_2(x_1, x_2) = (x_1, x_2 - b)$. In both directions, the invariant sets associated with the transformation is $\{\emptyset, \mathbb{Z}\}$. The associated (T_1, T_2) -invariant σ -field is $\mathcal{I} = \{\emptyset, \mathbb{Z}^2\}$ and we can state the following result, for any function $f : \mathbb{Z}^2 \rightarrow \mathbb{R}$, such that $f(\xi) \in L^p$,

$$\frac{1}{n_1 n_2} \sum_{k_1 < n_1} \sum_{k_2 < n_2} f(T_1^{k_1} T_2^{k_2} X) \xrightarrow{n_1, n_2 \rightarrow \infty} \mathbb{E}(f(X)) \text{ a.s. and in } L^p. \quad (\text{B.3})$$

B.2 Proof of Proposition 3.3.4 - Law of Large Numbers

Consider f a given function on \mathbb{R}^2 , and $X \sim \text{DPixP}(C)$ with C some admissible kernel on \mathbb{Z}^2 . We want to prove the following Law of Large Numbers

$$\frac{1}{N^2} \sum_{x \in X} f\left(\frac{x}{N}\right) \xrightarrow[N \rightarrow \infty]{} C(0) \int_{\mathbb{R}^2} f(x) dx, \text{ a.s and in } L^1. \quad (\text{B.4})$$

This proof proceeds in 3 steps: first, we prove the Law of Large Numbers (Equation (B.4)) given f is an indicator function. Then, we prove the convergence considering f is a simple function and finally we prove the proposition for mesurable functions with compact support.

Let us start by proving the convergence in the case of an indicator function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, $x = (x_1, x_2) \mapsto \mathbf{1}_{[0,a[\times [0,b[}(x)$, with $a, b \in \mathbb{N}$. We have $\forall n_1, n_2 \in \mathbb{N}$,

$$\begin{aligned} f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) &= \mathbf{1}_{[0,n_1 a[\times [0,n_2 b[}(x_1, x_2) \\ &= \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} \mathbf{1}_{[k_1 a, (k_1+1)a[\times [k_2 b, (k_2+1)b[}(x_1, x_2) \\ &= \sum_{k_1=0}^{n_1-1} \sum_{k_2=0}^{n_2-1} \mathbf{1}_{[0,a[\times [0,b[}(T_1^{k_1} T_2^{k_2}(x_1, x_2)) \\ &= \sum_{k_1 < n_1} \sum_{k_2 < n_2} f(T_1^{k_1} T_2^{k_2}(x_1, x_2)). \end{aligned}$$

Then, using the bivariate ergodic theorem (Theorem B.1.2), g a mesurable function defined by $g(X) = \sum_{x \in X} f(x)$ and the moment formula (Equation (3.19)),

$$\begin{aligned} \frac{1}{n_1 n_2} \sum_{x \in X} f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) &= \frac{1}{n_1 n_2} \sum_{k_1 < n_1} \sum_{k_2 < n_2} g(T_1^{k_1} T_2^{k_2} X) \text{ and then,} \\ \frac{1}{n_1 n_2} \sum_{x \in X} f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) &\xrightarrow[n_1, n_2 \rightarrow \infty]{a.s., L^p} \mathbb{E}(g(X)) = \mathbb{E}\left(\sum_{x \in X} f(x)\right) \\ &= \sum_{x \in \mathbb{Z}^2} f(x) C(0) = \int_{\mathbb{R}^2} f(x) C(0) dx \text{ because } a, b \in \mathbb{N}. \end{aligned}$$

Let us now consider $k_1, k_2 \in \mathbb{N}^*$. We define $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, as $f(x) = \mathbf{1}_{[0, \frac{1}{k_1}[\times [0, \frac{1}{k_2}[}(x)$, T_1 and T_2 as the translation of 1 unit in the vertical and hori-

zontal directions. Then, $\forall n_1, n_2 \in \mathbb{N}^*$,

$$\begin{aligned}
f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) &= \mathbf{1}_{[0, \frac{1}{k_1}] \times [0, \frac{1}{k_2}]} \left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) = \mathbf{1}_{[0, \frac{n_1}{k_1}] \times [0, \frac{n_2}{k_2}]}(x_1, x_2) \\
&= \mathbf{1}_{[0, \lfloor \frac{n_1}{k_1} \rfloor] \times [0, \lfloor \frac{n_2}{k_2} \rfloor]}(x) + \mathbf{1}_{[0, \lfloor \frac{n_1}{k_1} \rfloor] \times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}]}(x) \\
&\quad + \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}] \times [0, \lfloor \frac{n_2}{k_2} \rfloor]}(x) + \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}] \times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}]}(x) \\
f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) &= \sum_{l_1=0}^{\lfloor \frac{n_1}{k_1} \rfloor - 1} \sum_{l_2=0}^{\lfloor \frac{n_2}{k_2} \rfloor - 1} \mathbf{1}_{[l_1, l_1+1[\times [l_2, l_2+1[}(x) + \sum_{l_1=0}^{\lfloor \frac{n_1}{k_1} \rfloor - 1} \mathbf{1}_{[l_1, l_1+1[\times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}]}(x) \\
&\quad + \sum_{l_2=0}^{\lfloor \frac{n_2}{k_2} \rfloor - 1} \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}] \times [l_2, l_2+1[}(x) + \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}] \times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}]}(x) \\
&= \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \sum_{l_2 < \lfloor \frac{n_2}{k_2} \rfloor} \mathbf{1}_{[0, 1[\times [0, 1[}(T_1^{l_1} T_2^{l_2} x) \quad (1) \\
&\quad + \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \mathbf{1}_{[0, 1[\times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}]}(T_1^{l_1} x) \quad (2) \\
&\quad + \sum_{l_2 < \lfloor \frac{n_2}{k_2} \rfloor} \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}] \times [0, 1[}(T_2^{l_2} x) \quad (3) \\
&\quad + \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}] \times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}]}(x). \quad (4)
\end{aligned} \tag{B.5}$$

Now, we are going to study the limit of each part of the term above when we sum it for $x \in X$ and multiply it by $\frac{1}{n_1 n_2}$. First, we have

$$\begin{aligned}
&\frac{1}{n_1 n_2} \sum_{x \in X} \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \sum_{l_2 < \lfloor \frac{n_2}{k_2} \rfloor} \mathbf{1}_{[0, 1[\times [0, 1[}(T_1^{l_1} T_2^{l_2} x) \\
&= \frac{\lfloor \frac{n_1}{k_1} \rfloor \lfloor \frac{n_2}{k_2} \rfloor}{n_1 n_2} \frac{1}{\lfloor \frac{n_1}{k_1} \rfloor \lfloor \frac{n_2}{k_2} \rfloor} \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \sum_{l_2 < \lfloor \frac{n_2}{k_2} \rfloor} g(T_1^{l_1} T_2^{l_2} X),
\end{aligned}$$

where $g(X) = \sum_{x \in X} \mathbf{1}_{[0, 1[\times [0, 1[}(x)$. Since $\forall y \in \mathbb{R}$, $\lfloor y \rfloor \underset{+\infty}{\sim} y$, we have $\frac{\lfloor \frac{n_1}{k_1} \rfloor \lfloor \frac{n_2}{k_2} \rfloor}{n_1 n_2} \underset{+\infty}{\sim} \frac{1}{k_1 k_2}$. Moreover, thanks to the multivariate ergodic theorem,

$$\frac{1}{\lfloor \frac{n_1}{k_1} \rfloor \lfloor \frac{n_2}{k_2} \rfloor} \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \sum_{l_2 < \lfloor \frac{n_2}{k_2} \rfloor} g(T_1^{l_1} T_2^{l_2} X) \xrightarrow[n_1, n_2 \rightarrow \infty]{a.s., L^p} \mathbb{E}(g(X)) \quad (B.6)$$

$$\text{and } \mathbb{E}(g(X)) = \mathbb{E}\left(\sum_{x \in X} \mathbf{1}_{[0,1[\times [0,1[}(x)\right) = C(0) \sum_{x \in \mathbb{Z}^2} \mathbf{1}_{[0,1[\times [0,1[}(x) = C(0).$$

Finally, we obtain for this part

$$\frac{1}{n_1 n_2} \sum_{x \in X} \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \sum_{l_2 < \lfloor \frac{n_2}{k_2} \rfloor} \mathbf{1}_{[0,1[\times [0,1[}(T_1^{l_1} T_2^{l_2} x) \xrightarrow[n_1, n_2 \rightarrow \infty]{a.s., L^p} \frac{1}{k_1 k_2} C(0) = \int_{\mathbb{R}^2} f(x) C(0) dx. \quad (\text{B.7})$$

Second, we need to prove that the 3 other positive terms of the sum tends to 0. For (2) and (3), the proof is identical:

$$\frac{1}{n_1 n_2} \sum_{x \in X} \sum_{l_1 < \lfloor \frac{n_1}{k_1} \rfloor} \mathbf{1}_{[0,1[\times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}[}(T_1^{l_1} x) \leq \frac{1}{n_1 n_2} |X| \left\lfloor \frac{n_1}{k_1} \right\rfloor \xrightarrow[n_1, n_2 \rightarrow \infty]{+\infty} \frac{|X|}{n_2 k_1} \xrightarrow[n_1, n_2 \rightarrow \infty]{} 0. \quad (\text{B.8})$$

Similarly, concerning the last term, we have

$$\frac{1}{n_1 n_2} \sum_{x \in X} \mathbf{1}_{[\lfloor \frac{n_1}{k_1} \rfloor, \frac{n_1}{k_1}[\times [\lfloor \frac{n_2}{k_2} \rfloor, \frac{n_2}{k_2}[}(x) \leq \frac{1}{n_1 n_2} |X| \xrightarrow[n_1, n_2 \rightarrow \infty]{} 0. \quad (\text{B.9})$$

$$\text{Thus, } \frac{1}{n_1 n_2} \sum_{x \in X} f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) \xrightarrow[n_1, n_2 \rightarrow \infty]{a.s., L^p} \int_{\mathbb{R}^2} f(x) C(0) dx.$$

We have proved this property for all indicator functions on intervals of types $[0, a[\times [0, b[$, for all $a, b \in \mathbb{N}$ and $[0, \frac{1}{k_1}[\times [0, \frac{1}{k_2}[$ for all $k_1, k_2 \in \mathbb{N}^*$. As we made a translation invariance hypothesis, and thanks to the linearity of limits and integrals, this property is also verified for any indicator function on $[p_1, q_1[\times [p_2, q_2[$, $\forall p, q \in \mathbb{Q}^2$. As the set of 2D-rational sets generates the Borel set, this property is verified for all indicator functions on half-open intervals of \mathbb{R}^2 .

Now, let us prove it when f is a simple function, that is, given A_1, \dots, A_p half-open disjoint intervals of \mathbb{R}^2 , $f : \mathbb{R}^2 \rightarrow \mathbb{R}, x \mapsto f(x) = \sum_{k=1}^p c_k \mathbf{1}_{A_k}(x)$.

We can use the following results. Let $(X_n)_n, (Y_n)_n$ be two sequences of random variables on \mathbb{Z}^2 and X and Y be two random variables defined on \mathbb{Z}^2 . If $X_n \xrightarrow[n \rightarrow \infty]{a.s.} X$ and $Y_n \xrightarrow[n \rightarrow \infty]{a.s.} Y$, then $X_n + Y_n \xrightarrow[n \rightarrow \infty]{a.s.} X + Y$. Similarly, $X_n \xrightarrow[n \rightarrow \infty]{L^p} X$ and $Y_n \xrightarrow[n \rightarrow \infty]{L^p} Y$, then $X_n + Y_n \xrightarrow[n \rightarrow \infty]{L^p} X + Y$.

Hence,

$$\begin{aligned}
\frac{1}{n_1 n_2} \sum_{x \in X} f\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) &= \frac{1}{n_1 n_2} \sum_{x \in X} \sum_{k=1}^p c_k \mathbf{1}_{A_k}\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) \\
&= \sum_{k=1}^p c_k \frac{1}{n_1 n_2} \sum_{x \in X} \mathbf{1}_{A_k}\left(\frac{x_1}{n_1}, \frac{x_2}{n_2}\right) \\
&\xrightarrow[n_1, n_2 \rightarrow \infty]{a.s., L^p} \sum_{k=1}^p c_k \int_{\mathbb{R}^2} \mathbf{1}_{A_k}(x) C(0) dx = \int_{\mathbb{R}^2} f(x) C(0) dx.
\end{aligned} \tag{B.10}$$

Finally, we need to prove the a.s.-convergence and the L^1 -convergence for any bounded measurable function with a compact support. As it is bounded, there exists an increasing sequence of simple functions $(\phi_n)_{n \in \mathbb{N}}$ defined on \mathbb{R}^2 such that $\phi_n \xrightarrow[n \rightarrow \infty]{} f$, and the convergence is uniform.

Using this uniform convergence and common dominated convergence theorems, we can prove that the limit in Equation (B.4) holds when f is a measurable function with a compact support.

B.3 Proof of Proposition 3.3.4 - Central Limit Theorem

Consider f a bounded continuous function, with compact support, such that $\int_{\mathbb{R}^2} f(x) dx = 0$. We want to prove the following result

$$\frac{1}{\sqrt{N^2}} \sum_{x \in X} f\left(\frac{x}{N}\right) \xrightarrow[N \rightarrow \infty]{D} \mathcal{N}(0, \sigma(C)^2 \|f\|_2^2). \tag{B.11}$$

The proof of the Central Limit Theorem will be done in three steps. First, we need to compute the limit of the variance of $\sqrt{N^2} S_N$, where S_N is defined by

$$S_N(y) = \frac{1}{N^2} \sum_{x \in X} f\left(y - \frac{x}{N}\right), \forall y \in \mathbb{Z}^2. \tag{B.12}$$

Then, we rewrite the characteristic function of $\sqrt{N^2} S_N$. At last, we compute its limit.

Let us start by computing the limit of the variance of $\sqrt{N^2} S_N$ when N tends to infinity. We need the following lemma.

Lemma B.3.1. *Let f be a bounded continuous function on \mathbb{R}^2 with compact*

support and $\int_{\mathbb{R}^2} f(x)dx = 0$ and $X \sim \text{DPixP}(C)$ on \mathbb{Z}^2 . Then, $\forall N \in \mathbb{N}$,

$$\text{Var} \left(\frac{1}{N} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) = \frac{C(0)}{N^2} \sum_{x \in \mathbb{Z}^2} f \left(\frac{x}{N} \right)^2 - \frac{1}{N^2} \sum_{x, y \in \mathbb{Z}^2} |C(x)|^2 f \left(\frac{y}{N} \right) f \left(\frac{y+x}{N} \right). \quad (\text{B.13})$$

Proof. Suppose that f is a bounded continuous function on \mathbb{R}^2 with compact support such that $\int_{\mathbb{R}^2} f(x)dx = 0$ and that X is a determinantal pixel process of kernel K , associated with the kernel function C , on \mathbb{Z}^2 .

Thanks to moments formulas [9] on DPPs on Γ with measure μ (here $\Gamma = \mathbb{Z}^2$ and μ is the DPP distribution), we know that, $\forall f, h$, functions on Γ ,

$$\begin{aligned} \text{Cov} \left(\sum_{x \in X} f(x), \sum_{x \in X} h(x) \right) &= \int_{\Gamma} f(x)h(x)K(x, x)\mu(dx) \\ &\quad - \int_{\Gamma^2} f(x)h(y)K(x, y)K(y, x)\mu(dx)\mu(dy). \end{aligned} \quad (\text{B.14})$$

Then, $\forall N \in \mathbb{N}$,

$$\begin{aligned} \text{Var} \left(\frac{1}{N} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) &= \frac{1}{N^2} \text{Var} \left(\sum_{x \in X} f \left(\frac{x}{N} \right) \right) \\ &= \frac{1}{N^2} \left(\sum_{x \in \mathbb{Z}^2} f^2 \left(\frac{x}{N} \right) C(0) - \sum_{z, y \in \mathbb{Z}^2} f^2 \left(\frac{z}{N} \right) f^2 \left(\frac{y}{N} \right) |C(z - y)|^2 \right) \quad (\text{B.15}) \\ &= \frac{C(0)}{N^2} \sum_{x \in \mathbb{Z}^2} f \left(\frac{x}{N} \right)^2 - \frac{1}{N^2} \sum_{x, y \in \mathbb{Z}^2} |C(x)|^2 f \left(\frac{y}{N} \right) f \left(\frac{y+x}{N} \right). \end{aligned}$$

□

Notice that $C(0) \frac{1}{N^2} \sum_{x \in \mathbb{Z}^2} f \left(\frac{x}{N} \right)^2 \xrightarrow{N \rightarrow \infty} C(0) \int_{\mathbb{R}^2} f(z)^2 dz$, thanks to the Riemman sums theory. To compute the limit of the second part of the variance, we need to use the dominated convergence theorem.

(1) Let us prove first that $\forall x \in \mathbb{Z}^2, |C(x)|^2 \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f \left(\frac{y}{N} \right) f \left(\frac{y+x}{N} \right)$ has a

limit. Let us consider $x \in \mathbb{Z}^2$ and $\epsilon > 0, \forall N \in \mathbb{N}$,

$$\begin{aligned}
& \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) - \int_{\mathbb{R}^2} f(z) dz \right| \\
& \leq \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) - \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right)^2 \right| + \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right)^2 - \int_{\mathbb{R}^2} f(z) dz \right| \\
& = \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) \left(f\left(\frac{y+x}{N}\right) - f\left(\frac{y}{N}\right) \right) \right| + \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right)^2 - \int_{\mathbb{R}^2} f(z) dz \right|. \tag{B.16}
\end{aligned}$$

Concerning the first part, as f has compact support, there exists $A \in \mathbb{N}$ such that its support is included in $\Lambda = [-A, A] \times [-A, A]$ and then f_N 's support is included in $N\Lambda = [-NA, NA] \times [-NA, NA]$. $\forall x \in \mathbb{Z}^2$, the support of the function $f_N(\cdot)f_N(\cdot+x)$ is also included in Λ_N .

As f is bounded, there exists $M > 0$ s.t. $|f| \leq M$ and it is uniformly continuous: $\exists \eta > 0$, such that $\forall y, z \in \mathbb{Z}^2, |z - y| \leq \eta \Rightarrow |f(z) - f(y)| \leq \epsilon$. As here $x \in \mathbb{Z}^2$ is set, there exists $N_x \in \mathbb{N}$ such that $\forall N \geq N_x, \left| \frac{x}{N} \right| \leq \eta$ and then $\forall y \in \mathbb{Z}^2, |f(\frac{y+x}{N}) - f(\frac{y}{N})| \leq \epsilon$.

Concerning the second part, as we have the Riemann sum of a continuous function on compact support, $\exists N_2 \in \mathbb{N}$ such that $\forall N \geq N_2$,

$$\left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right)^2 - \int_{\mathbb{R}^2} f(z)^2 dz \right| < \epsilon. \tag{B.17}$$

Let us consider $N \geq \max(N_x, N_2)$,

$$\begin{aligned}
& \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) - \int_{\mathbb{R}^2} f(z)^2 dz \right| \\
& \leq \frac{1}{N^2} \sum_{y \in \Lambda_N} \left| f\left(\frac{y}{N}\right) \right| \left| f\left(\frac{y+x}{N}\right) - f\left(\frac{y}{N}\right) \right| + \left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right)^2 - \int_{\mathbb{R}^2} f(z)^2 dz \right| \\
& \leq \frac{1}{N^2} M \epsilon (2NA)^2 + \epsilon = (4MA^2 + 1)\epsilon. \tag{B.18}
\end{aligned}$$

Then $\left| \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) - \int_{\mathbb{R}^2} f(z)^2 dz \right| \xrightarrow{N \rightarrow \infty} 0$. We can conclude that $\forall x \in \mathbb{Z}^2, \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) \xrightarrow{N \rightarrow \infty} \int_{z \in \mathbb{R}^2} f(z)^2 dz$.

(2) Second, let us prove that, $\forall N \in \mathbb{N}$, $|C(x)|^2 \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right)$

is dominated by a sequence that does not depend on N and that is summable. Using the same notations as before, we can notice that $\forall N \in \mathbb{N}$,

$$\begin{aligned} \left| |C(x)|^2 \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) \right| &\leq |C(x)|^2 \frac{1}{N^2} \sum_{y \in \Lambda_N} \left| f\left(\frac{y}{N}\right) \right| \left| f\left(\frac{y+x}{N}\right) \right| \\ &\leq |C(x)|^2 \frac{M^2(2NA)^2}{N^2} \\ &= |C(x)|^2 4(MA)^2, \end{aligned} \tag{B.19}$$

and $\sum_{x \in \mathbb{Z}^2} |C(x)|^2 4(MA)^2 = 4(MA)^2 \sum_{x \in \mathbb{Z}^2} |C(x)|^2 < \infty$ as $C \in \ell^2(\mathbb{Z}^2)$.

To conclude, we can interchange the limit and the sum and:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{x, y \in \mathbb{Z}^2} |C(x)|^2 f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) \\ = \sum_{x \in \mathbb{Z}^2} |C(x)|^2 \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{y \in \mathbb{Z}^2} f\left(\frac{y}{N}\right) f\left(\frac{y+x}{N}\right) \tag{B.20} \\ = \sum_{x \in \mathbb{Z}^2} |C(x)|^2 \int_{z \in \mathbb{R}^2} f(z)^2 dz. \end{aligned}$$

$$\text{Thus, } \lim_{N \rightarrow \infty} \text{Var} \left(\frac{1}{N} \sum_{x \in X} f\left(\frac{x}{N}\right) \right) = \left(C(0) - \sum_{x \in \mathbb{Z}^2} |C(x)|^2 \right) \int_{\mathbb{R}^2} f(z)^2 dz. \tag{B.21}$$

Now, let us compute the characteristic function of our studied sum.

As, $\forall N \in \mathbb{N}$, f_N is defined on Λ_N ,

$$\mathbb{E} \left(\exp \left(\frac{i}{N} \sum_{x \in X} f\left(\frac{x}{N}\right) \right) \right) = \mathbb{E} \left(\exp \left(\frac{i}{N} \sum_{x \in X \cap \Lambda_N} f\left(\frac{x}{N}\right) \right) \right). \tag{B.22}$$

Then, we can consider the process $X_N = X \cap \Lambda_N$ which becomes a finite DPiP on Λ_N . We introduce the matrix K_N -the restriction of K to Λ_N - and the associated kernel function C_N . Now let us denote P_N of size $\Lambda_N \times \Lambda_N$ as $P_N = \Phi_N K_N$ where Φ_N is the diagonal matrix with coordinate $\Phi_N(x, x) = \phi_N(x) = 1 - e^{\frac{i}{N} f(\frac{x}{N})}$, $\forall x \in \Lambda_N$.

As we defined K_N , we know that there exists a L-ensemble L such that $L = K_N(I - K_N)^{-1}$ and $K_N = L(L + I)^{-1}$ (where I is the $\Lambda_N \times \Lambda_N$ -identity matrix).

Then, $\forall N \in \mathbb{N}$,

$$\begin{aligned} \mathbb{E} \left(\exp \left(\frac{i}{N} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) \right) &= \sum_{A \subset \Lambda_N} e^{\frac{i}{N} \sum_{y \in A} f \left(\frac{y}{N} \right)} \mathbb{P}(X \cap \Lambda_N = A) \\ &= \sum_{A \subset \Omega} e^{\frac{i}{N} \sum_{y \in A} f \left(\frac{y}{N} \right)} \frac{\det(L_A)}{\det(I + L)} \\ &= \frac{1}{\det(I + L)} \sum_{A \subset \Omega} \det((D_N L)_A) \end{aligned} \quad (\text{B.23})$$

where D_N is the diagonal matrix of size $\Lambda_N \times \Lambda_N$ with $D_N(y, y) = e^{\frac{i}{N} f \left(\frac{y}{N} \right)}$.

$$\begin{aligned} \mathbb{E} \left(\exp \left(\frac{i}{N} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) \right) &= \frac{1}{\det(I + L)} \det(I + D_N L) \\ &= \det(L^{-1} L(I + L)^{-1}) \det(I + D_N K_N(I - K_N)^{-1}) \\ &= \det(K_N^{-1}(I - K_N)K_N) \det((I - K_N)(I - K_N)^{-1} + D_N K_N(I - K_N)^{-1}) \\ &= \det(I - K_N) \det(I - K_N + D_N K_N) \det(I - K_N)^{-1} \\ &= \det(I - (I - D_N)K_N) \\ &= \det(I - \Phi_N K_N) \\ &= \det(I - P_N) \\ &= \exp(\text{tr}(\ln(I - P_N))). \end{aligned} \quad (\text{B.24})$$

On the other hand, we can find a relation between this quantity and the limit of the variance of $S_N(0)$ by computing $\text{tr}(P_N)$ and $\text{tr}(P_N^2)$.

$$\text{tr}(P_N) = \sum_{x \in \Lambda_N} P_N(x, x) = \sum_{x \in \Lambda_N} \phi_N(x) K_N(x, x) = \sum_{x \in \Lambda_N} C(0) \left(1 - e^{\left(\frac{i}{Nd/2} f \left(\frac{x}{N} \right) \right)} \right). \quad (\text{B.25})$$

As $1 - \exp(x) \stackrel{0}{=} -x - \frac{x^2}{2} + o(x^2)$, for sufficiently large N we have

$$\begin{aligned}
\text{tr}(P_N) &= \sum_{x \in \Lambda_N} C(0) \left(-\frac{i}{N} f\left(\frac{x}{N}\right) + \frac{1}{2N^2} f^2\left(\frac{x}{N}\right) + o\left(\frac{1}{N^2}\right) \right) \\
&= -C(0)iN \frac{1}{N^2} \sum_{x \in \Lambda_N} f\left(\frac{x}{N}\right) + \frac{C(0)}{2} \frac{1}{N^2} \sum_{x \in \mathbb{Z}^2} f^2\left(\frac{x}{N}\right) + o\left(\frac{1}{N^2}\right) \\
&= -C(0)iN \left(\frac{1}{N^2} \sum_{x \in \Lambda_N} f\left(\frac{x}{N}\right) - \int_{\mathbb{R}^2} f(t) dt \right) + \frac{C(0)}{2} \frac{1}{N^2} \sum_{x \in \Lambda_N} f^2\left(\frac{x}{N}\right) + o\left(\frac{1}{N^2}\right) \\
&\xrightarrow{N \rightarrow \infty} \frac{1}{2} \int_{\mathbb{R}^2} C(0) f^2(t) dt.
\end{aligned} \tag{B.26}$$

On the other hand, when N is large,

$$\begin{aligned}
\text{tr}(P_N^2) &= \sum_{n \in \Lambda_N} P_N^2(n, n) = \sum_{n \in \Lambda_N} \sum_{m \in \Lambda_N} \phi_N(n) K(n, m) \phi_N(m) K(m, n) \\
&= \sum_{n, m \in \mathbb{Z}^2} \phi_N(n) \phi_N(m) |C(n - m)|^2 \\
&= \sum_{n, m \in \mathbb{Z}^2} \left(1 - e^{\frac{i}{N} f\left(\frac{n}{N}\right)} \right) \left(1 - e^{\frac{i}{N} f\left(\frac{m}{N}\right)} \right) |C(n - m)|^2
\end{aligned} \tag{B.27}$$

$$\begin{aligned}
\text{tr}(P_N^2) &= \sum_{x, y \in \mathbb{Z}^2} \left(1 - e^{\frac{i}{N} f\left(\frac{x+y}{N}\right)} \right) \left(1 - e^{\frac{i}{N} f\left(\frac{y}{N}\right)} \right) |C(x)|^2 \\
&= \sum_{x, y \in \mathbb{Z}^2} |C(x)|^2 \left(-\frac{i}{N} f\left(\frac{x+y}{N}\right) + o\left(\frac{1}{N}\right) \right) \left(-\frac{i}{N} f\left(\frac{y}{N}\right) + o\left(\frac{1}{N}\right) \right) \\
&= -\frac{1}{N^2} \sum_{x, y \in \mathbb{Z}^2} |C(x)|^2 f\left(\frac{x+y}{N}\right) f\left(\frac{y}{N}\right) + o\left(\frac{1}{N^2}\right) \\
&\xrightarrow{N \rightarrow \infty} - \sum_{x \in \mathbb{Z}^2} |C(x)|^2 \int_{\mathbb{R}^d} f(x)^2 dx, \text{ by the same arguments as in the} \\
&\text{previous computation of the variance's limit.}
\end{aligned} \tag{B.28}$$

We have shown that

$$\lim_{N \rightarrow \infty} \text{Var} \left(\frac{1}{N} \sum_{x \in X} f\left(\frac{x}{N}\right) \right) = \lim_{N \rightarrow \infty} (2 \text{tr}(P_N) + \text{tr}(P_N^2)) = \sigma(C)^2 \|f\|_2^2. \tag{B.29}$$

Now, let us consider a sufficiently large N ,

$$\begin{aligned}
& \left| -\log \mathbb{E} \left(\exp \left(\frac{i}{N} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) \right) - \text{tr}(P_N) - \frac{1}{2} \text{tr}(P_N^2) \right| \\
&= \left| -\log(\det(I - P_N)) - \text{tr}(P_N) - \frac{1}{2} \text{tr}(P_N^2) \right| \\
&= \left| -\sum_{n \geq 1} \frac{(-1)^{n+1}}{n} \text{tr}(P_N^n) (-1)^n - \text{tr}(P_N) - \frac{1}{2} \text{tr}(P_N^2) \right| \\
&\leq \sum_{n \geq 3} \frac{|\text{tr}(P_N^n)|}{n} \leq \sum_{n \geq 3} \frac{\text{tr}(|P_N^n|)}{n} \\
&\leq \sum_{n \geq 3} \frac{1}{n} \text{tr}(|P_N|^2) \|P_N\|^{n-2}, \text{ as, given a bounded operator } S \text{ and a trace class} \\
&\text{operator } T, \quad \text{tr}(|ST|) \leq \|S\| \text{tr}(|T|) \text{ [120, Lemma 2.1], and } P_N \text{ is bounded.} \\
&\leq \text{tr}(|P_N|^2) \sum_{n \geq 1} \frac{1}{n+2} \|P_N\|^n \leq \text{tr}(|P_N|^2) \sum_{n \geq 1} \frac{1}{n} \|P_N\|^n = -\text{tr}(|P_N|^2) \ln(1 - \|P_N\|) \\
&\text{because } \forall x < 1, \ln(1 - x) = -\sum_{n \geq 1} \frac{x^n}{n} \text{ and as } N \text{ is large, } \|P_N\| \text{ is small,} \\
&\leq -\ln(1 - \|\phi_N K_N\|) \text{tr}(|P_N|^2) \\
&\leq -\ln(1 - \|\phi_N\|_\infty \|K_N\|) \text{tr}(|P_N|^2) \xrightarrow[N \rightarrow \infty]{} 0, \text{ using the fact that} \\
&\|\phi_N\|_\infty \leq \|f\|_\infty / N, \|K_N\| \leq \|K\| \text{ and } \text{tr}(|P_N|^2) \leq C(0) \|K\| \|f\|_\infty |\text{supp } f|.
\end{aligned} \tag{B.30}$$

Thus, we have

$$\mathbb{E} \left(\exp \left(\frac{i}{N} \sum_{x \in X} f \left(\frac{x}{N} \right) \right) \right) \xrightarrow[N \rightarrow \infty]{} \exp \left(-\frac{1}{2} \left(C(0) - \sum_{x \in \Omega} C(x)^2 \right) \|f\|_2^2 \right). \tag{B.31}$$

Notice that if we use the function tf instead of the function f , $\forall t \in \mathbb{R}$, then we can apply the Levy's continuity theorem which leads to the following Central Limit theorem:

$$\frac{1}{\sqrt{N^2}} \sum_{x \in X} f \left(\frac{x}{N} \right) \xrightarrow[N \rightarrow \infty]{D} \mathcal{N}(0, \sigma(C)^2 \|f\|_2^2), \tag{B.32}$$

with $\sigma(C)^2 = C(0) - \sum_{x \in \mathbb{Z}^2} |C(x)|^2$.

Appendix C

Identifiability of a DPixP

Contents

C.1 Remark 3.4.1, Case 2	151
C.2 Remark 3.4.1, Case 3: K_1 is not irreducible	153

This appendix is related to Chapter 3. It provides some details on the question of equivalence classes for DPixP kernels, presented in Section 3.4.1. Proposition 3.4.2 and Remark 3.4.1 offer several results on these equivalence classes depending on the DPixP kernel, dividing the kernels into three categories. The first category corresponds to the DPixP kernels so that the kernel matrix K_1 is irreducible and verifies the rank hypothesis given in Theorem 3.4.1, meaning that $N \leq 4$ or that $N \geq 4$ and for every partition of \mathcal{Y} into subsets α, β such that $|\alpha| \geq 2, |\beta| \geq 2$, $\text{rank}(K_1)_{\alpha \times \beta} \geq 2$. The second category concerns DPixP kernels such that K_1 is irreducible but does not verify the rank hypothesis in Theorem 3.4.1. Section C.1 gives an insight into this category by developing the case where the DPixP is defined on Ω of size 1×5 . The third case is when the kernel matrix K_1 is not irreducible. Section C.2 discusses the consequences of this hypothesis on DPixP equivalence classes.

C.1 Remark 3.4.1, Case 2

Let us study the equivalence class of a DPixP of kernel C_1 such that its associated matrix K_1 is irreducible and it does not verify the rank hypothesis given in Theorem 3.4.1, in the case Ω of size 1×5 . That means that there exists a partition α, β of \mathcal{Y} such that $\text{rank}(K_1)_{\alpha \times \beta} = 1$. As an admissible kernel matrix on Ω , K_1 is such that

$$K_1 = \text{circulant} \left(C_1(0), C_1(1), C_1(2), \overline{C_1(2)}, \overline{C_1(1)} \right). \quad (\text{C.1})$$

Define $r_{11}, \theta_{11}, r_{12}, \theta_{12}$ the respective modulus and argument of $C_1(1)$ and $C_1(2)$. Whatever α, β , the partition of \mathcal{Y} such that $\text{rank}(K_1)_{\alpha \times \beta} = 1$, due to rows proportionality, one obtains $r_{11} = r_{12}$ and $\theta_{12} = -3\theta_{11} \bmod 2\pi$. Now, assume that C_2 is an admissible DPiXP kernel such that $\text{DPiXP}(C_2) = \text{DPiXP}(C_1)$. Then the matrices K_1 and K_2 have equal principal minors. Necessarily, K_2 is irreducible and there exists a partition such that $\text{rank}(K_2)_{\alpha \times \beta} = 1$, otherwise K_2 would verify the assumptions of Theorem 3.4.1 and so would K_1 . Then, as C_1, C_2 is fully determined by $C_2(0)$, one modulus r_{21} and one argument θ_{21} . Once again, we know that $C_1(0) = C_2(0) = C_0$ and thanks to the equality of principal minors of size 2, the modulus are equal so $r_{21} = r_{11} = r$. One of the principal minors of size 3 for C_1 is equal to

$$C_0^3 + \overline{C_1(1)C_1(1)}C_1(2) + C_1(1)C_1(1)\overline{C_1(2)} - C_0C_1(2)\overline{C_1(2)} - 2C_0C_1(1)\overline{C_1(1)}, \quad (\text{C.2})$$

so by equality of principal minors, we obtain

$$\begin{aligned} \text{Re} \left(C_1(1)C_1(1)\overline{C_1(2)} \right) &= \text{Re} \left(C_2(1)C_2(1)\overline{C_2(2)} \right) \\ \Leftrightarrow \text{Re} \left(r^3 e^{2i\theta_{11}+3i\theta_{11}} \right) &= \text{Re} \left(r^3 e^{2i\theta_{21}+3i\theta_{21}} \right) \\ \Leftrightarrow r^3 \cos(5\theta_{11}) &= r^3 \cos(5\theta_{21}) \\ \Leftrightarrow \exists k \in \mathbb{Z} \text{ s.t. } \theta_{11} &= \begin{cases} \theta_{21} + \frac{2}{5}k\pi & (\text{case 1}) \\ -\theta_{21} + \frac{2}{5}k\pi & (\text{case 2}). \end{cases} \end{aligned} \quad (\text{C.3})$$

Finally, let us assume we are in the first case, K_1 can be written

$$\begin{aligned} K_1 &= \text{circulant} \left(C_0, r e^{i(\theta_{21} + \frac{2}{5}k\pi)}, r e^{-3i(\theta_{21} + \frac{2}{5}k\pi)}, r e^{3i(\theta_{21} + \frac{2}{5}k\pi)}, r e^{-i(\theta_{21} + \frac{2}{5}k\pi)} \right) \\ &= DK_2D^{-1} \end{aligned} \quad (\text{C.4})$$

with $D = \text{diag} \left(1, e^{i\frac{2}{5}k\pi}, e^{i\frac{4}{5}k\pi}, e^{-i\frac{4}{5}k\pi}, e^{-i\frac{2}{5}k\pi} \right)$, which corresponds to a translation of the Fourier coefficients of C of k pixels. The second case yields to $K_1 = D\overline{K_2}D^{-1}$ which corresponds to the symmetry and the translation of k pixels of the Fourier coefficients of C .

Thus, in that case, even if K_1 does not verify the rank hypothesis of Theorem 3.4.1, its equivalence class is defined as that of a kernel which does: K_2 is equivalent to K_1 if and only if the Fourier coefficients of K_2 are a translation or a symmetry with respect to $(0,0)$ of the Fourier coefficients of K_1 .

Here, this study is limited to the case 1×5 . We have not been able to generalize this result to all sizes of image domain yet. We would like to demonstrate that the equivalence class of a kernel belonging to this second category, such that it is irreducible and such that there exists a partition α, β of \mathcal{Y} such that $\text{rank}(K_1)_{\alpha \times \beta} = 1$, is characterized as in the first category: DPiXP kernels are equivalent if and only if they have translated and/or symmetrized Fourier coefficients. This question remains open.

C.2 Remark 3.4.1, Case 3: K_1 is not irreducible

In this section, we consider a kernel that belongs to the third category mentioned in Remark 3.4.1. That means that its associated matrix is a Hermitian block-circulant matrix K_1 of size $N \times N$ that is completely reducible, meaning that it is permutation similar to a block diagonal matrix with irreducible blocks. We want to prove that in that case, the blocks are identical, that is they are of equal size and they are composed of the same coefficients. Moreover, we prove that these blocks are not only irreducible but also Hermitian and circulant. First, let us study the 1D case, meaning that K_1 is a kernel defined on the points of $\mathcal{Y} = \{0, \dots, N-1\}$ (to be consistent with our 2D representation) and it is circulant. Therefore, for all $i, j \in \mathcal{Y}$, there exists c_{j-i} such that $K_1(i, j) = c_{j-i} = \overline{c_{i-j}}$. As K_1 is not irreducible, there exist $i, j \in \mathcal{Y}$, such that $K_1(i, j) = c_{j-i} = 0$. Let us denote $k = \inf\{l > 0 \text{ such that } c_l \neq 0\}$, hence $c_1 = \dots = c_{k-1} = 0 = c_{-1} = \dots = c_{-k+1}$. Notice that k is necessarily larger or equal to 2, otherwise K_1 would not have any zero coefficient, it would be possible to access to any index from any other, and it would be irreducible. Similarly, k necessarily divides N and the only non-zero coefficients c_m are multiples of k , as otherwise, once again, the non-zero elements of K_1 would be located such that it would be possible to access to any index from any other by traveling only through non-zero coefficients: K_1 would be irreducible. Then, if we define l such that $N = k \times l$, there are k cycles of size l in the graph associated to K_1 , each block with the same l coefficients $\{c_k, c_{2k}, \dots, c_{lk}\}$, or equivalently, $\forall i_0 = 0, \dots, N-1$,

$$K(i_0, j) = \begin{cases} c_{kp}, & \text{if } j = kp + i_0 \pmod{N}, \text{ with } p = 0, \dots, l-1, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{C.5})$$

Thus it is possible to define the permutation matrix P which gathers the cycles, and which associates K_1 with a block diagonal matrix:

$$\forall p = 0, \dots, l-1, \quad \forall r = 0, \dots, k-1, \quad P(p+lr, r+pk) = 1. \quad (\text{C.6})$$

In other words, the matrix P associates the index $r+pk$ of K_1 to the index $p+lr$ (r -th block, p -th coefficient) of the permuted block matrix. Moreover, these blocks $(B_r)_{r \in \{0, \dots, k-1\}}$ are circulant: for all $r = 0, \dots, k-1$, for all $i, i' = 0, \dots, l-1$,

$$B_r(i, i') = K(r+ik, r+i'k) = c_{(i-i')k}, \quad (\text{C.7})$$

for all $\tau \in \mathcal{Y}$ such that $(i+\tau \pmod{N})$ and $(i'+\tau \pmod{N})$ are in the r -th cycle,

$$B_r(i+\tau, i'+\tau) = K_1(r+(i+\tau)k, r+(i'+\tau)k) = c_{(i-i')k} = B_r(i, i'). \quad (\text{C.8})$$

To conclude, K_1 is permutation similar to a block-diagonal matrix, which is the repetition of one irreducible, circulant and Hermitian block.

Now let us consider the 2D case, when K_1 is a kernel matrix defined on $\Omega = \{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\}$ and assume that K_1 is Hermitian, block-circulant with circulant blocks and completely reducible. Define C_1 the function such that for all $(i, j), (i', j') \in \Omega$, $K_1((i, j), (i', j')) = C_1(i' - i, j' - j)$. As in the 1D case, define $(e_1, e_2) \in \mathbb{Z}^2 \cap \Omega$ the two generating vectors such that $C_1(r, s) = 0, \forall (r, s)$ inside the elementary cell generated by (e_1, e_2) . These two vectors generate a subgroup of \mathbb{Z}^2 and it contains $\mathbb{Z}(0, N_2) + \mathbb{Z}(N_1, 0)$, as K_1 is not irreducible and similarly as in the 1D case. Then e_1 divides N_1 , e_2 divides N_2 . As before, the only non-zero coefficients of C_1 belong to $\{\mathbb{Z}e_1 + \mathbb{Z}e_2\} \cap \Omega$. The size of the elementary cell determines the number of cycles (and future blocks) and $l = \#\{\mathbb{Z}e_1 + \mathbb{Z}e_2\} \cap \Omega$ defines the size of each cycle. It is possible to define the permutation matrix that transforms K_1 into a block-diagonal matrix with irreducible blocks. For all $(i, j) \in \Omega$, let us define (r, s) its representative element in the elementary cell such that there exists p, q such that $(i, j) = (pe_1 + qe_2) + (r, s) \bmod (N_1, N_2)$. We define P such that it associates the index (i, j) of K_1 to the index $(p, q) + (r, s)$ (block (r, s) , coefficient (p, q)) of the permuted block matrix. As before, the blocks $(B_{(r,s)})$ have the same size and have an identical structure. Let us consider the block (r, s) , consider $(i, j), (i', j') \in \Omega$,

$$\begin{aligned} B_{(r,s)}((i, j), (i', j')) &= K_1((pe_1 + qe_2) + (r, s) \bmod (N_1, N_2), (p'e_1 + q'e_2) + (r, s) \bmod (N_1, N_2)) \\ &= C_1((p' - p)e_1 + (q' - q)e_2) \end{aligned} \tag{C.9}$$

Let $(\tau_1, \tau_2) \in \Omega$ be such that $(i + \tau_1, j + \tau_2), (i' + \tau_1, j' + \tau_2)$ both belong to the cycle (r, s) . Then $(\tau_1, \tau_2) \in \mathbb{Z}e_1 + \mathbb{Z}e_2$, we can write $(\tau_1, \tau_2) = t_1e_1 + t_2e_2$.

$$\begin{aligned} B_{(r,s)}((i + \tau_1, j + \tau_2), (i' + \tau_1, j' + \tau_2)) &= K_1((pe_1 + qe_2) + (r, s) + (t_1e_1 + t_2e_2) \bmod (N_1, N_2), \\ &\quad (p'e_1 + q'e_2) + (r, s) + (t_1e_1 + t_2e_2) \bmod (N_1, N_2)) \\ &= C_1((p' - p)e_1 + (q' - q)e_2) = B_{(r,s)}((i, j), (i', j')). \end{aligned} \tag{C.10}$$

Thus, for all (r, s) , the associated bloc $B_{(r,s)}$ is block circulant with circulant blocks. Similarly, it is Hermitian. To conclude, K_1 is permutation similar to a block diagonal matrix defined by only one repeated irreducible, circulant, Hermitian block.

Bibliography

- [1] AFFANDI, R. H., FOX, E. B., ADAMS, R. P., AND TASKAR, B. Learning the parameters of determinantal point process kernels. In *ICML (2014)*, vol. 32 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 1224–1232.
- [2] AFFANDI, R. H., KULESZA, A., FOX, E. B., AND TASKAR, B. Nystrom approximation for large-scale determinantal processes. In *AISTATS (2013)*, vol. 31 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 85–98.
- [3] AGARWAL, A., CHOROMANSKA, A., AND CHOROMANSKI, K. Notes on using determinantal point processes for clustering with applications to text clustering. *CoRR abs/1410.6975* (2014).
- [4] AGGARWAL, C. C. *Outlier Analysis*, 2nd ed. Springer Publishing Company, Incorporated, 2016.
- [5] AMBLARD, P.-O., BARTHELME, S., AND TREMBLAY, N. Subsampling with k-determinantal point processes for estimating statistics in large data sets. In *2018 IEEE workshop on Statistical Signal Processing (SSP 2018)* (Freiburg, Germany, June 2018).
- [6] ANARI, N., GHARAN, S. O., AND REZAEI, A. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *COLT (2016)*, vol. 49 of *JMLR Workshop and Conference Proceedings*, JMLR.org, pp. 103–115.
- [7] AURENHAMMER, F., HOFFMANN, F., AND ARONOV, B. Minkowski-type theorems and least-squares clustering. *Algorithmica* 20, 1 (1998), 61–76.
- [8] AVENA, L., AND GAUDILLIÈRE, A. Two applications of random spanning forests. *Journal of Theoretical Probability* 31, 4 (Dec. 2018), 1975–2004.

-
- [9] BACCELLI, F., AND BLASZCZYSZYN, B. *Stochastic Geometry and Wireless Networks, Volume I - Theory*, vol. 1 of *Foundations and Trends in Networking Vol. 3: No 3-4*, pp 249-449. NoW Publishers, 2009. Stochastic Geometry and Wireless Networks, Volume II - Applications; see <http://hal.inria.fr/inria-00403040>.
 - [10] BADDELEY, A., RUBAK, E., AND TURNER, R. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC Press, London, 2015.
 - [11] BARDENET, R., AND HARDY, A. Monte Carlo with determinantal point processes. *The Annals of Applied Probability* 30, 1 (Feb 2020), 368–417.
 - [12] BARDENET, R., LAVANCIER, F., MARY, X., AND VASSEUR, A. On a few statistical applications of determinantal point processes. *ESAIM: Procs* 60 (2017), 180–202.
 - [13] BARDENET, R., AND TITSIAS, M. Inference for determinantal point processes without spectral knowledge. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3393–3401.
 - [14] BARTHELMÉ, S., AMBLARD, P.-O., AND TREMBLAY, N. Asymptotic equivalence of fixed-size and varying-size determinantal point processes. *Bernoulli* 25, 4B (11 2019), 3555–3589.
 - [15] BLASZCZYSZYN, B., AND KEELER, H. P. Determinantal thinning of point processes with network learning applications. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)* (April 2019), pp. 1–8.
 - [16] BLASZCZYSZYN, B., AND YOGESHWARAN, D. Directionally convex ordering of random measures, shot noise fields, and some applications to wireless communications. *Advances in Applied Probability* 41, 3 (Sep 2009), 623–646.
 - [17] BELHADJI, A., BARDENET, R., AND CHAINAIS, P. A determinantal point process for column subset selection. *CoRR abs/1812.09771* (2018).
 - [18] BERGMANN, U., JETCHEV, N., AND VOLLGRAF, R. Learning texture manifolds with the periodic spatial GAN. *arXiv preprint arXiv:1705.06566* (2017).
 - [19] BIRKHOFF, G. D. Proof of the ergodic theorem. *Proceedings of the National Academy of Science* 17, 12 (Dec. 1931), 656–660.

- [20] BISCIO, C., AND LAVANCIER, F. Quantifying repulsiveness of determinantal point processes. *Bernoulli* 22, 4 (11 2016), 2001–2028.
- [21] BISCIO, C., AND LAVANCIER, F. Contrast estimation for parametric stationary determinantal point processes. *Scandinavian Journal of Statistics* 44, 1 (2017), 204–229.
- [22] BISCIO, C. A., AND COEURJOLLY, J.-F. Standard and robust intensity parameter estimation for stationary determinantal point processes. *Spatial Statistics* 18 (2016), 24 – 39. Spatial Statistics Avignon: Emerging Patterns.
- [23] BORODIN, A., AND RAINS, E. M. Eynard–Mehta theorem, Schur process, and their Pfaffian analogs. *Journal of Statistical Physics* 3 (2005), 291–317.
- [24] BOYD, S., AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, March 2004.
- [25] BRUNEL, V. Learning signed determinantal point processes through the principal minor assignment problem. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*. (2018), pp. 7376–7385.
- [26] BRUNEL, V., MOITRA, A., RIGOLLET, P., AND URSCHER, J. Rates of estimation for determinantal point processes. In *COLT (2017)*, vol. 65 of *Proceedings of Machine Learning Research*, PMLR, pp. 343–345.
- [27] CELIS, E., KESWANI, V., STRASZAK, D., DESHPANDE, A., KATHURIA, T., AND VISHNOI, N. Fair and diverse DPP-based data summarization. In *Proceedings of the 35th International Conference on Machine Learning* (10–15 Jul 2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 716–725.
- [28] CHEN, W., YANG, Z., CAO, F., YAN, Y., WANG, M., QING, C., AND CHENG, Y. Dimensionality reduction based on determinantal point process and singular spectrum analysis for hyperspectral images. *IET Image Processing* 13, 2 (2019), 299–306.
- [29] CHIU, S., STOYAN, D., KENDALL, W., AND MECKE, J. *Stochastic Geometry and Its Applications*. Wiley Series in Probability and Statistics. Wiley, 2013.
- [30] CONDAT, L. Fast Projection onto the Simplex and the l_1 Ball. *Mathematical Programming, Series A* 158, 1 (July 2016), 575–585.

- [31] COOK, R. Stochastic sampling in computer graphics. *ACM Trans. Graph.* 5, 1 (jan 1986), 51–72.
- [32] CUTURI, M., AND DOUCET, A. Fast computation of Wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning* (Bejing, China, 22–24 Jun 2014), E. P. Xing and T. Jebara, Eds., vol. 32 of *Proceedings of Machine Learning Research*, PMLR, pp. 685–693.
- [33] DABOV, K., FOI, A., KATKOVNIK, V., AND EGIAZARIAN, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing* 16, 8 (2007), 2080–2095.
- [34] DALEY, D. J., AND VERE-JONES, D. *An introduction to the theory of point processes. Vol. I*, second ed. Probability and its Applications (New York). Springer-Verlag, New York, 2003. Elementary theory and methods.
- [35] DALEY, D. J., AND VERE-JONES, D. *An introduction to the theory of point processes. Vol. II*, second ed. Probability and its Applications (New York). Springer, 2008. General theory and structure.
- [36] DE BORTOLI, V., DESOLNEUX, A., GALERNE, B., AND LECLAIRE, A. Patch redundancy in images: A statistical testing framework and some applications. *SIAM Journal on Imaging Sciences* 12, 2 (2019), 893–926.
- [37] DECREUSEFOND, L., FLINT, I., AND LOW, K. C. Perfect simulation of determinantal point processes. *ArXiv e-prints* (Nov. 2013).
- [38] DEREUDRE, D. Introduction to the theory of Gibbs point processes. In *Stochastic Geometry*. Springer, 2019, pp. 181–229.
- [39] DEREZIŃSKI, M., CALANDRIELLO, D., AND VALKO, M. Exact sampling of determinantal point processes with sublinear time preprocessing. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 11546–11558.
- [40] DESOLNEUX, A., MOISAN, L., AND RONSIN, S. A compact representation of random phase and Gaussian textures. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Kyoto, Japan, Mar. 2012), proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1381–1384.

- [41] DUPUY, C., AND BACH, F. Learning determinantal point processes in sublinear time. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Spain* (2018), pp. 244–257.
- [42] EFROS, A., AND FREEMAN, W. Image quilting for texture synthesis and transfer. *ACM TOG* (August 2001), 341–346.
- [43] EFROS, A. A., AND LEUNG, T. K. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision* (1999), vol. 2, pp. 1033–1038 vol.2.
- [44] EISENBAUM, N., AND KASPI, H. On permanent processes. *Stochastic Processes and their Applications* 119, 5 (2009), 1401–1415.
- [45] ENGEL, G. M., AND SCHNEIDER, H. Matrices diagonally similar to a symmetric matrix. *Linear Algebra and its Applications* 29 (Feb. 1980), 131–138.
- [46] FEDER, J. Random sequential adsorption. *Journal of Theoretical Biology* 87, 2 (1980), 237–254.
- [47] FISHER, R. A. Design of experiments. *Br Med J* 1, 3923 (1936), 554–554.
- [48] GALERNE, B., GOUSSEAU, Y., AND MOREL, J.-M. Random phase textures: Theory and synthesis. *IEEE Trans. Image Process.* 20, 1 (2011), 257 – 267.
- [49] GALERNE, B., LAGAE, A., LEFEBVRE, S., AND DRETTAKIS, G. Gabor noise by example. *ACM Trans. Graph.* 31, 4 (jul 2012), 73:1–73:9.
- [50] GALERNE, B., LECLAIRE, A., AND MOISAN, L. A texton for fast and flexible Gaussian texture synthesis. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)* (2014), pp. 1686–1690.
- [51] GALERNE, B., LECLAIRE, A., AND MOISAN, L. Texton noise. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 205–218.
- [52] GALERNE, B., LECLAIRE, A., AND RABIN, J. A texture synthesis model based on semi-discrete optimal transport in patch space. *SIAM Journal on Imaging Sciences* 11, 4 (2018), 2456–2493.
- [53] GARTRELL, M., PAQUET, U., AND KOENIGSTEIN, N. Low-rank factorization of determinantal point processes. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (2017), AAAI’17, AAAI Press, pp. 1912–1918.

- [54] GATYS, L., ECKER, A. S., AND BETHGE, M. Texture synthesis using convolutional neural networks. In *Proc. of NIPS* (2015), pp. 262–270.
- [55] GAUTIER, G. *On sampling determinantal point processes*. Phd thesis, Ecole Centrale de Lille, March 2020. <https://guilgautier.github.io/>.
- [56] GAUTIER, G., BARDENET, R., AND VALKO, M. Zonotope hit-and-run for efficient sampling from projection DPPs. In *Proceedings of the 34th International Conference on Machine Learning* (Aug. 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1223–1232.
- [57] GAUTIER, G., BARDENET, R., AND VALKO, M. DPPy: Sampling determinantal point processes with Python. *CoRR abs/1809.07258* (2018).
- [58] GAUTIER, G., BARDENET, R., AND VALKO, M. On two ways to use determinantal point processes for Monte Carlo integration. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 7770–7779.
- [59] GELFAND, A., FUENTES, M., GUTTORP, P., AND DIGGLE, P. *Handbook of Spatial Statistics*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis, 2010.
- [60] GENEVAY, A., CUTURI, M., PEYRÉ, G., AND BACH, F. Stochastic optimization for large-scale optimal transport. In *Proc. of NIPS* (2016), pp. 3432–3440.
- [61] GEORGE, A., HEATH, M. T., AND LIU, J. Parallel Cholesky factorization on a shared-memory multiprocessor. *Linear Algebra and its Applications* 77 (may 1986), 165–187.
- [62] GILET, G., SAUVAGE, B., VANHOEY, K., DISCHLER, J., AND GHAZANFARPOUR, D. Local random-phase noise for procedural texturing. *ACM Transactions on Graphics* 33, 6 (2014), 195:1–195:11.
- [63] GILLENWATER, J., KULESZA, A., MARIET, Z., AND VASSILVTISKII, S. A tree-based method for fast repeated sampling of determinantal point processes. In *Proceedings of the 36th International Conference on Machine Learning* (Long Beach, California, USA, 09–15 Jun 2019), K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97 of *Proceedings of Machine Learning Research*, PMLR, pp. 2260–2268.

- [64] GILLENWATER, J., KULESZA, A., AND TASKAR, B. Discovering diverse and salient threads in document collections. In *EMNLP-CoNLL (2012)*, ACL, pp. 710–720.
- [65] GINIBRE, J. Statistical ensembles of complex: Quaternion, and real matrices. *Journal of Mathematical Physics Vol: 6* (Mar 1965).
- [66] GONG, B., CHAO, W., GRAUMAN, K., AND SHA, F. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada* (2014), pp. 2069–2077.
- [67] HARTFIEL, D. J., AND LOEWY, R. On matrices having equal corresponding principal minors. *j-LINEAR-ALGEBRA-APPL* 58 (Apr. 1984), 147–167.
- [68] HEEGER, D. J., AND BERGEN, J. R. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 229–238.
- [69] HONG, K., CONROY, J., FAVRE, B., KULESZA, A., LIN, H., AND NENKOVA, A. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (Reykjavik, Iceland, May 2014), European Language Resources Association (ELRA), pp. 1608–1616.
- [70] HORN, R. A., AND JOHNSON, C. R. *Matrix Analysis*. Cambridge University Press, 1990.
- [71] HOUDARD, A., BOUYEYRON, C., AND DELON, J. High-dimensional mixture models for unsupervised image denoising (HDMI). *SIAM Journal on Imaging Sciences* (2018).
- [72] HOUGH, J. B., KRISHNAPUR, M., PERES, Y., AND VIRÁG, B. Determinantal processes and independence. *Probability Surveys* (2006), 206–229.
- [73] HOUGH, J. B., KRISHNAPUR, M., PERES, Y., AND VIRÁG, B. *Zeros of Gaussian Analytic Functions and Determinantal Point Processes*, vol. 51 of *University Lecture Series*. American Mathematical Society, Providence, RI, 2009.

- [74] JOHNSON, J., ALAHI, A., AND FEI-FEI, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision* (2016).
- [75] KALLENBERG, O. *Foundations of modern probability*, second ed. Probability and its Applications (New York). Springer-Verlag, New York, 2002.
- [76] KITAGAWA, J., MÉRIGOT, Q., AND THIBERT, B. A Newton algorithm for semi-discrete optimal transport. *Journ. of the Europ. Math Soc.* (2017).
- [77] KULESZA, A. *Learning with Determinantal Point Processes*. PhD thesis, University of Pennsylvania, 2012.
- [78] KULESZA, A., AND TASKAR, B. Structured determinantal point processes. In *NIPS* (2010), Curran Associates, Inc., pp. 1171–1179.
- [79] KULESZA, A., AND TASKAR, B. k-DPPs: Fixed-size determinantal point processes. In *Proceedings of the 28th International Conference on International Conference on Machine Learning* (USA, 2011), ICML’11, Omnipress, pp. 1193–1200.
- [80] KULESZA, A., AND TASKAR, B. Learning determinantal point processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence* (2011), pp. 419–427.
- [81] KULESZA, A., AND TASKAR, B. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning* 5, 2-3 (2012), 123–286.
- [82] LAGAE, A., LEFEBVRE, S., DRETTAKIS, G., AND DUTRÉ, P. Procedural noise using sparse Gabor convolution. *ACM Transactions on Graphics* 28, 3 (2009), 54–64.
- [83] LAUNAY, C., GALERNE, B., AND DESOLNEUX, A. Exact sampling of determinantal point processes without eigendecomposition. *ArXiv e-prints* (Feb 2018), arXiv:1802.08429.
- [84] LAUNAY, C., AND LECLAIRE, A. Determinantal patch processes for texture synthesis. In *GRETSI 2019* (Lille, France, Aug 2019).
- [85] LAVANCIER, F., MØLLER, J., AND RUBAK, E. Determinantal point process models and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 77, 4 (2015), 853–877.
- [86] LECLAIRE, A., AND RABIN, J. A fast multi-layer approximation to semi-discrete optimal transport. In *SSVM* (2019), pp. 341–352.

- [87] LI, C., JEGELKA, S., AND SRA, S. Efficient sampling for k -determinantal point processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (Cadiz, Spain, 09–11 May 2016), A. Gretton and C. C. Robert, Eds., vol. 51 of *Proceedings of Machine Learning Research*, PMLR, pp. 1328–1337.
- [88] LI, C., SRA, S., AND JEGELKA, S. Fast mixing Markov chains for strongly Rayleigh measures, DPPs, and constrained sampling. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4188–4196.
- [89] LI, C., AND WAND, M. Combining Markov random fields and convolutional neural networks for image synthesis. In *Proc. the IEEE CVPR* (2016), pp. 2479–2486.
- [90] LIU, Y., COLLINS, R., AND TSIN, Y. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 3 (Mar. 2004), 354–371.
- [91] LLOYD, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137.
- [92] LOEWY, R. Principal minors and diagonal similarity of matrices. *Linear Algebra and its Applications* 78 (June 1986), 23–64.
- [93] LOONIS, V., AND MARY, X. Determinantal sampling designs. *Journal of Statistical Planning and Inference* 199 (2019), 60 – 88.
- [94] LU, Y., ZHU, S.-C., AND WU, Y. N. Learning FRAME models using CNN filters. In *31th conference on artificial intelligence* (2016).
- [95] LYONS, R., AND STEIF, J. E. Stationary determinantal processes: Phase multiplicity, Bernoullicity, entropy, and domination. *Duke Math. J.* 120, 3 (12 2003), 515–575.
- [96] MACCHI, O. The coincidence approach to stochastic point processes. *Advances in Applied Probability* 7 (1975), 83–122.
- [97] MAHASSENI, B., LAM, M., AND TODOROVIC, S. Unsupervised video summarization with adversarial LSTM networks. In *CVPR* (2017), IEEE Computer Society, pp. 2982–2991.
- [98] MATÉRN, B. *Spatial variation*, vol. 36 of *Lecture notes in statistics*. Springer-Verlag, 1986.

- [99] MAYERS, D., AND SÜLI, E. *An introduction to numerical analysis*. Cambridge Univ. Press, Cambridge, 2003.
- [100] MCCULLAGH, P., AND MØLLER, J. The permanental process. *Advances in Applied Probability* 38 (12 2006), 873–888.
- [101] MØLLER, J., AND WAAGEPETERSEN, R. *Statistical Inference and Simulation for Spatial Point Process*, vol. 100. Chapman and Hall/CRC, Boca Raton, 2003.
- [102] MUMFORD, D., AND DESOLNEUX, A. *Pattern Theory: The Stochastic Analysis of Real-World Signals*. Ak Peters Series. Taylor & Francis, 2010.
- [103] NEUMANN, J. V. Physical applications of the ergodic hypothesis. *Proceedings of the National Academy of Science* 18, 3 (Mar. 1932), 263–266.
- [104] NEYMAN, J. On a new class of “contagious” distributions, applicable in entomology and bacteriology. *Ann. Math. Statist.* 10, 1 (03 1939), 35–57.
- [105] NG, M. A note on constrained k -Means algorithms. *Pattern Recognition* 33 (03 2000), 515–519.
- [106] PAGÈS, G. Introduction to vector quantization and its applications for numerics. *ESAIM: Proceedings and Surveys* 48, 1 (2015), 29–79. Proceedings of CEMRACS 2013 - Modelling and simulation of complex systems: stochastic and deterministic approaches. : T. Lelièvre et al. Editors.
- [107] POINAS, A., DELYON, B., AND LAVANCIER, F. Mixing properties and central limit theorem for associated point processes. *Bernoulli* 25, 3 (2019), 1724–1754.
- [108] PORTILLA, J., AND SIMONCELLI, E. A parametric texture model based on joint statistics of complex wavelet coefficients. *IJCV* 40, 1 (2000), 49–70.
- [109] POULSON, J. High-performance sampling of generic determinantal point processes. *Philosophical Transactions of the Royal Society* 378, 2166 (Jan 2020), arXiv:1905.00165.
- [110] PROPP, J. G., AND WILSON, D. B. How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *J. Algorithms* 27, 2 (1998), 170–217.
- [111] RAAD, L., DESOLNEUX, A., AND MOREL, J. A conditional multiscale locally Gaussian texture synthesis algorithm. *J. Math. Imaging Vision* 56, 2 (2016), 260–279.

- [112] RAAD CISA, L., DAVY, A., DESOLNEUX, A., AND MOREL, J.-M. A survey of exemplar-based texture synthesis. *Annals of Mathematical Sciences and Applications* 3 (07 2017).
- [113] RISING, J., KULESZA, A., AND TASKAR, B. An efficient algorithm for the symmetric principal minor assignment problem. *Linear Algebra and its Applications* 473 (May 2015), 126–144.
- [114] ROLSKI, T., AND SZEKLI, R. Stochastic ordering and thinning of point processes. *Stochastic Processes and their Applications* 37, 2 (1991), 299–312.
- [115] ROTA, G.-C. On the foundations of combinatorial theory I. Theory of Möbius functions. *Z. Wahrscheinlichkeitstheorie und verw 2* (1964), 340–368.
- [116] SALMON, J., AND STROZECKI, Y. From patches to pixels in non-local methods: Weighted-average reprojection. In *2010 IEEE International Conference on Image Processing* (2010), IEEE, pp. 1929–1932.
- [117] SAUNDERS, B. D., AND SCHNEIDER, H. Flows on graphs applied to diagonal similarity and diagonal equivalence for matrices. *Discrete Mathematics* 24, 2 (1978), 205 – 220.
- [118] SCARDICCHIO, A., ZACHARY, C. E., AND TORQUATO, S. Statistical properties of determinantal point processes in high dimensional Euclidean spaces. *Phys. Rev. E* 79, 4 (2009).
- [119] SHIRAI, T., AND TAKAHASHI, Y. *Fermion Process and Fredholm Determinant*. Springer US, Boston, MA, 2000, pp. 15–23.
- [120] SHIRAI, T., AND TAKAHASHI, Y. Random point fields associated with certain Fredholm determinants. I. Fermion, Poisson and boson point processes. *Journal of Functional Analysis* 205, 2 (2003), 414–463.
- [121] SHIRAI, T., AND TAKAHASHI, Y. Random point fields associated with certain Fredholm determinants II: Fermion shifts and their ergodic and Gibbs properties. *Ann. Probab.* 31, 3 (07 2003), 1533–1564.
- [122] SOSHIKOV, A. Determinantal random point fields. *Russian Mathematical Surveys*, 55 (2000), 923–975.
- [123] STEVENS, M. Equivalent symmetric kernels of determinantal point processes. *arXiv e-prints* (May 2019), arXiv:1905.08162.
- [124] TREFETHEN, L. N., AND BAU, D. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, June 1997.

- [125] TREMBLAY, N., BARTHELMÉ, S., AND AMBLARD, P.-O. Optimized algorithms to sample determinantal point processes. *CoRR abs/1802.08471* (2018).
- [126] TREMBLAY, N., BARTHELMÉ, S., AND AMBLARD, P.-O. Determinantal point processes for coresets. *Journal of Machine Learning Research* (Nov. 2019).
- [127] TRUCCOLO, W., EDEN, U., FELLOWS, M., DONOGHUE, J., AND BROWN, E. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology* 93 2 (2005), 1074–89.
- [128] ULYANOV, D., LEBEDEV, V., VEDALDI, A., AND LEMPITSKY, V. Texture networks: feed-forward synthesis of textures and stylized images. In *Proc. of the Int. Conf. on Machine Learning* (2016), vol. 48, pp. 1349–1357.
- [129] URSCHER, J., BRUNEL, V., MOITRA, A., AND RIGOLLET, P. Learning determinantal point processes with moments and cycles. In *ICML (2017)*, vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 3511–3520.
- [130] VAN WIJK, J. J. Spot noise texture synthesis for data visualization. In *SIGGRAPH '91* (New York, NY, USA, 1991), ACM, pp. 309–318.
- [131] WEI, L., LEFEBVRE, S., KWATRA, V., AND TURK, G. State of the art in example-based texture synthesis. In *Eurographics 2009, State of the Art Report, EG-STAR* (Munich, Germany, 2009), Eurographics Association, pp. 93–117.
- [132] WILHELM, M., RAMANATHAN, A., BONOMO, A., JAIN, S., CHI, E. H., AND GILLENWATER, J. Practical diversified recommendations on Youtube with determinantal point processes. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (New York, NY, USA, 2018), CIKM '18, ACM, pp. 2165–2173.
- [133] ZHANG, C., KJELLSTRÖM, H., AND MANDT, S. Balanced mini-batch sampling for SGD using determinantal point processes. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence* (Aug. 2017).
- [134] ZHANG, K., CHAO, W., SHA, F., AND GRAUMAN, K. Video summarization with long short-term memory. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII* (2016), pp. 766–782.

-
- [135] ZHU, S., WU, Y., AND MUMFORD, D. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *Int. J. Comput. Vis.* 27, 2 (1998), 107–126.
 - [136] ZORAN, D., AND WEISS, Y. From learning models of natural image patches to whole image restoration. In *Proceedings of the 2011 International Conference on Computer Vision (USA, 2011)*, ICCV '11, IEEE Computer Society, p. 479–486.