TSE FISE3 & Master AIMA - PSRS
Markov models and applications to imagery.

SABIR ILYASS.

1 Introduction to Markov Point Process (MPP)

1.1 General definition of Point Process

On a mathematical space, Point process is a collection of points randomly located.

In a mathematical point of view, Let $(\Omega, \tau, \mathbb{P})$ a probability space, and $(S, \mathcal{F})$ a measurable space, and consider a kernel function $\xi: (\Omega, \tau) \rightarrow (S, \mathcal{F})$ that is, a mapping $\Omega \times S \rightarrow \mathbb{N}$ such that:.

1. For every $\omega \in \Omega$, $\xi(\omega, .)$ is locally finite measure on S.
2. For every $B \in S, \xi(., B): \Omega \rightarrow \mathbb{N}$ is a random variable over $\mathbb{N}$.

This kernel defines a random measure. We would like to think of ξ as defining a mapping which maps $\omega \in \Omega$ to a measure $\xi_m \in \mathcal{M}(\mathcal{S})$, where $\mathcal{M}(\mathcal{S})$ is the set of all locally finite measures on $\mathcal{S}$.

Remark 1. $\mathbb{N}$ can be replaced by any countable set.

1.2 Markov Point Process

Now, we will define the Markov Point Process, for that let $n \in \mathbb{N}$ and $\mathbf{x} = \{x_1, \ldots, x_n\}$ a set of objects (where $x_i = ((p_i, q_i), m_i) \in \mathbb{R}^2 \times \mathbb{R}^+$, $\forall i = 1, \ldots, n$). Writing Ω for the set of all configurations, and let μ be a finite non-atomic measure on U. Then, under the Poisson model, the total number of objects has a Poisson distribution with mean $\mu(U)$.

To construct spatial processes that do exhibit dependence between '***neighbouring***' objects, we specify the probability density of the new process with respect to the Poisson process . A density is a measurable and integrable function $p: \Omega \rightarrow [0, +\infty[$. For the new process, the distribution of the total number of objects is

$$\mathbb{P}(N = n) = \frac{\exp(-\mu(U))}{n!} \int_U \ldots \int_U p(\{x_1, \ldots, x_n\}) d\mu(x_1) \ldots d\mu(x_n)$$

Than the joint probability desnity is

$$p(\mathbf{x}) = p_n(x_1, \ldots, x_n) = \frac{\exp(-\mu(U))}{n!} \mu(U)^n \frac{p(\{x_1, \ldots, x_n\})}{\mathbb{P}(N = n)}$$

To introduce interactions, let $\backsim$ be any symmetric, reflexive relation on U. For instance two objects represented by $u, v \in U$ are **'neighbours'** if their intersection is non-empty. A widely used class is that of pairwise interaction models

$$p(\mathbf{x}) \propto \beta^{n(\mathbf{x})} \prod_{x_i \backsim x_j} \phi(x_i, x_j) \tag{1}$$

Here $\alpha, \beta > 0$ are constants, $n(\mathbf{x})$ is the number of objects in $\mathbf{x}$, and $\phi : U \times U \rightarrow [0, +\infty[$. the interaction function. The product is over all pairs of neighbouring objects $x_i \backsim x_j$ with $i < j$.

2 Simulation of the homogeneous Poisson Point Process

The simulation of the homogeneous Poisson Point Process of a set of points $(x_1, \ldots, x_{n(x)})$ with only to possible moves (birth or death with a priori probability P_b and P_d). Let $N_{\mathbf{x}}$: the random variable associated to the distribution of the points.

Via (1), we know that $p(\mathbf{x}) \propto e^{n(x).\log\beta}$. Our objectif is to estimate the confidence artival $[\bar{n} - b, \bar{n} + b]$, where $\bar{n}$ is the average of $n(x)$ and b is the bound value. this average will be estimated by $\mathbb{E}[N_{\mathbf{x}}] = \beta S$. and usually we take b so as to have a confidence interval of risk equal to 5%.

The ratios R_b and R_d are respectively for birth and death moves

$$R_b = \frac{P_d}{P_b} . \frac{\beta S}{N_{\mathbf{x}} + 1} \text{ and } R_d = \frac{P_b}{P_d} . \frac{N_{\mathbf{x}}}{\beta S}$$

From a theoretical point of view, the higher P_b, the higher the number of points, and consequently its average will also be high.

Also the greater p_b is, the greater the stability of the number of resulting points, and therefore there will be a faster convergence of the number of points and the number of cumulative points.

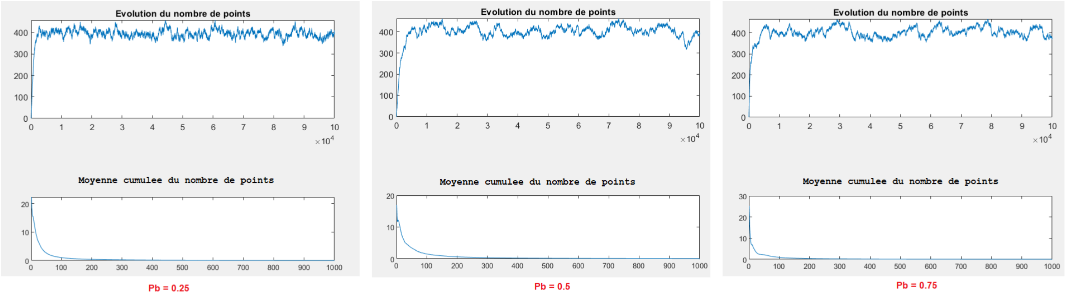

Figure 1. Graphs represent the evolution of the number of points $n(x)$ and the cumulative average of $n(x)$ at each iteration, for $P_b = 0.25$, $P_b = 0.5$ and $P_b = 0.75$

3 Simulation of the Strauss point process

The Strauss Point Process take in consideration the the number of neighboring points. In fact let $\sim$ a neighbourhood relation (the proximity relation for exemple defined by: for a metric d, and a real $R > 0$, $x \sim y$ if and only if $d(x, y) \leqslant R$.

The density of the Strauss process is then defined, with β always representing the intensity, as:

$$p(\mathbf{x}) \propto \beta^{n(\mathbf{x})} \prod_{i<j} \phi(x_i\,, x_j) \text{ where } \phi(x_i, x_j) = \begin{cases} \gamma \text{ if } x_i \sim x_j \\ 1 \text{ else} \end{cases}$$

At the birth or death of a point y, its number of neighbours is $n_b(y)$, so the ratios R_b and R_d respectively of birth and death moves are:

$$R_b = \frac{P_d}{P_b}\beta.\gamma^{n_b(y)}\frac{S}{N_{\mathbf{x}}+1} \quad \text{and} \quad R_d = \frac{P_b}{P_d}\frac{1}{\beta.\gamma^{n_b(y)}}\frac{N_{\mathbf{x}}}{S}$$

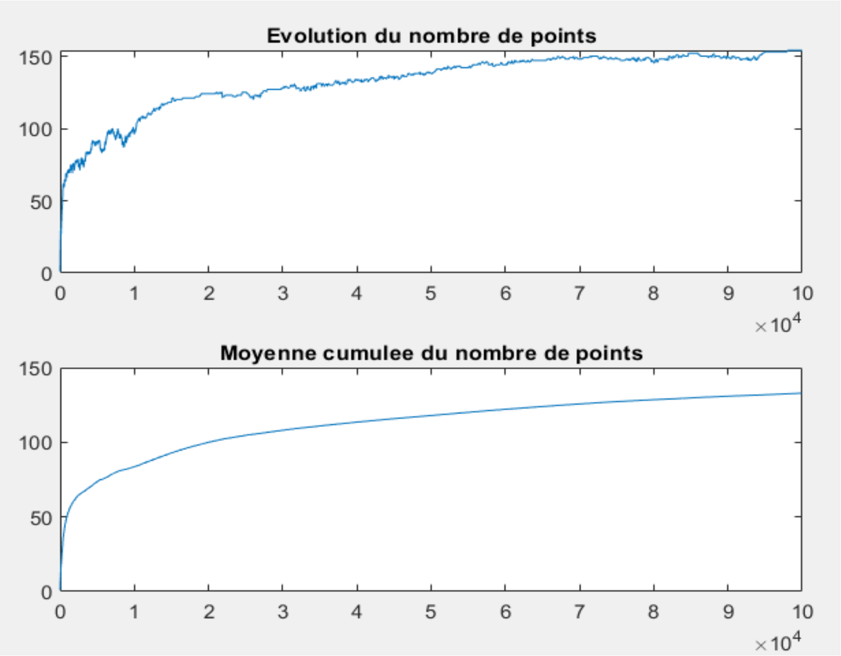

Figure 2. Evolution of the number of points $n(\mathbf{x})$ and the cumulative average of the number of points are respectively represented here regarding the Strauss process. For $\gamma = 0.2$

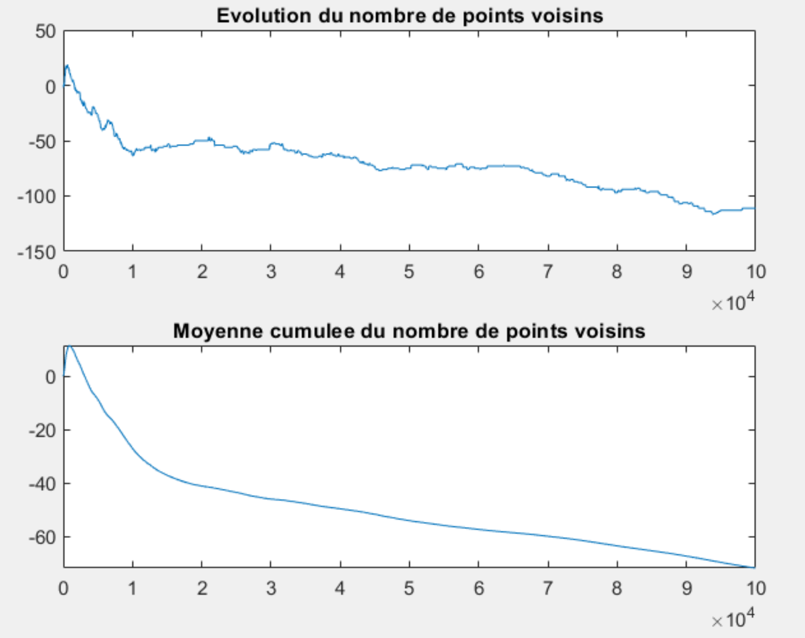

Figure 3. Evolution of the number of neighbour points $n_b(\mathbf{x})$ and its cumulative average for the Strauss process. For $\gamma = 0.2$

Figures 2 and 3 illustrate the evolution of the number of points, number of cumulative points, the number of neighboring points and the number of neighboring points, we notice a decrease in these numbers according to the iterations, which is normal since we have taken $\gamma = 0.2 < 1$.

For $\gamma = 1.02$ for exemple, we will obtain graphs with an increase in the number of points and the number of neighboring points (because $\gamma > 1$):

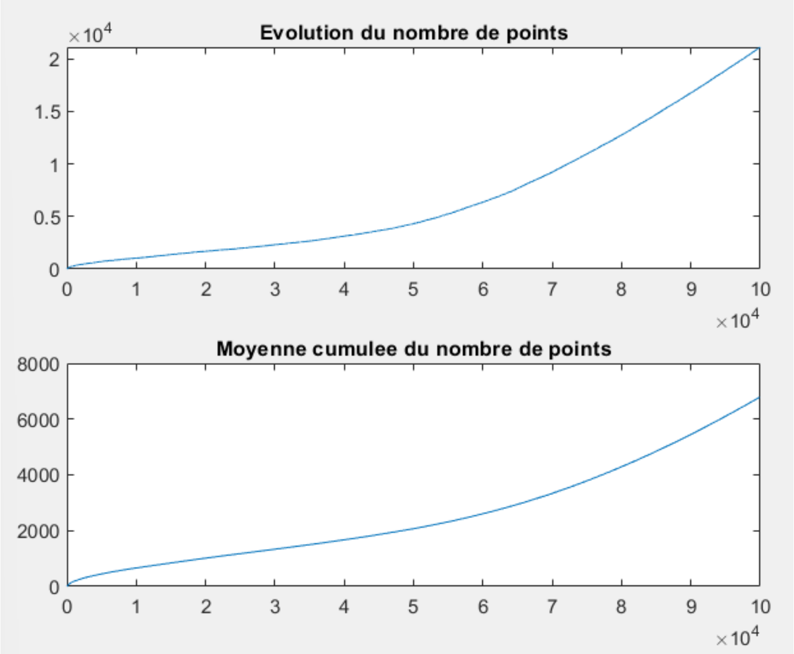

Figure 4. Evolution of the number of points $n(\mathbf{x})$ and the cumulative average of the number of points are respectively represented here regarding the Strauss process. For $\gamma = 1.02$

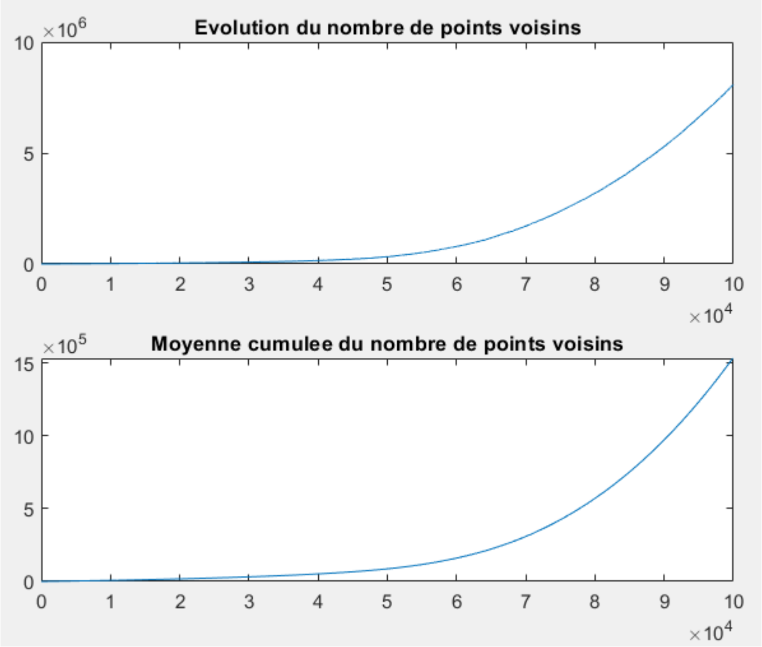

Figure 5. Evolution of the number of neighbour points $n_b(\mathbf{x})$ and its cumulative average for the Strauss process. For $\gamma = 1.02$

4 Simulation of the Strauss point process with image data term

Adapting the Strauss point process simulation to image problems, we need to add a new term $\gamma_d^{n_d(\mathbf{x})}$ (the data term) corresponding to existing data in order generate a fitting model, where $n_d(\mathbf{x})$ is the number of points considered badly localised, and $\gamma_d \in [0,1]$ is the weight of penalisation.

The density of probability becomes:

$$p(\mathbf{x}) \propto \beta^{n(\mathbf{x})} \gamma^{n_b(\mathbf{x})} \gamma_d^{n_d(\mathbf{x})}$$

And the ratios R_b and R_d become for birth and death moves of a point y:

$$R_b = \frac{P_d}{P_b}.\beta.\gamma^{n_b(y)}.\gamma_d^{n_d(y)}.\frac{S}{N_{\mathbf{x}}+1} \quad \text{and} \quad R_d = \frac{P_b}{P_d}.\frac{1}{\beta\gamma^{n_b(y)}\gamma_d^{n_d(y)}}.\frac{N_{\mathbf{x}}}{S}$$

In the following Figures, we illustrate the different graphs obtained with the simulation of the strauss point process with image data term.

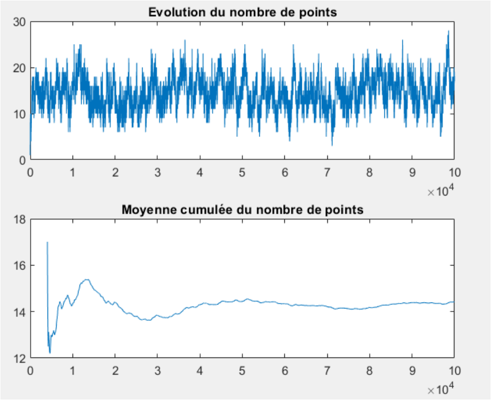

Figure 6. Number of points $n(\mathbf{x})$ and the cumulative average of the number of points represented in function of the number of iterations

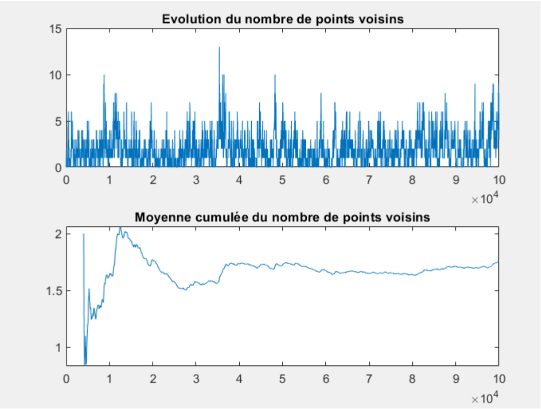

Figure 7. Number of neighbourhood point $n_b(\mathbf{x})$ relationships and its cumulative average in function of the number of iterations

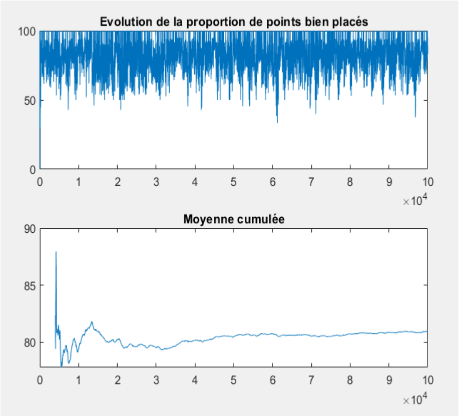

Figure 8. Percentage of well-placed points over the total number of points $\frac{n_d(\mathbf{x})}{n(\mathbf{x})}$ and the cumulative average of that number in function of the number of iterations

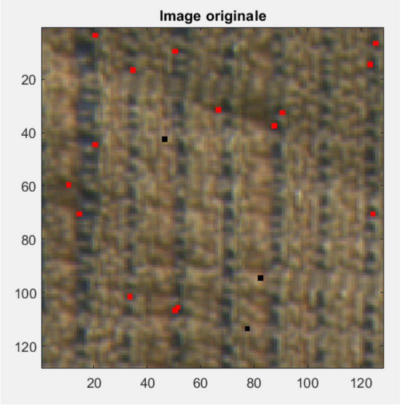

Figure 9. Generated point set over the original image : well-placed points appear in red, badly-placed points in black.

The global behavior of the graph is increased, which means that the number of points placed by the algorithm is increased, also relative to the neighbors ns(x). However, the quality of the points is not satisfactory. In fact, after the initial increase in energy due to the higher number of points placed, there is a small drop due to the increase in the percentage of good points placed.

5 Simulated annealing to estimate a population of well-placed points

Simulated annealing is a stochastic technique for global approximation of a given function. We used the Metropolis-Hasting algorithms. It implements a temperature T which "cools down" with iteration number, and which influences the calculation of the enegry.

Here,leading to the following density function where T_i is the temperature at the i$^{\text{th}}$ iteration :

$$p(\mathbf{x}) \propto \beta^{n(\mathbf{x})} \gamma^{\frac{n_b(\mathbf{x})}{T_i}} \gamma_d^{\frac{n_d(\mathbf{x})}{T_i}}$$

Inclusion of lower temperatures causes the Strauss terms and the data terms to increase in weight as the simulation progresses. Thus, it allows to start the process by simplifying the generation of points before stepping through the criteria.

the ratios R_b and R_d become for birth and death moves of a point y:

$$R_b = \frac{P_d}{P_b}.\beta.\gamma^{\frac{n_b(y)}{T_i}} \gamma_d^{\frac{n_d(y)}{T_i}}.\frac{S}{N_{\mathbf{x}}+1} \text{ and } R_d = \frac{P_d}{P_b}.\frac{1}{\beta.\gamma^{\frac{n_b(y)}{T_i}}.\gamma^{\frac{n_d(y)}{T_i}}}.\frac{N_{\mathbf{x}}}{S}$$

In the figures below, we observe that the number of points in the system decreases after a certain number of iterations, we see that the number of well-placed points actually increases until a maximum value, which means that all remaining points are well placed.

Also, we can see a large decrease of the energy, leading to good points being chosen and leading to negative values, optimizing the function. The upper bound is a bit better than the former case due to the actual large number of dislocation points in the simulation, but it is still higher than the Poisson and Strauss point processes because those models don't have the data to compare.

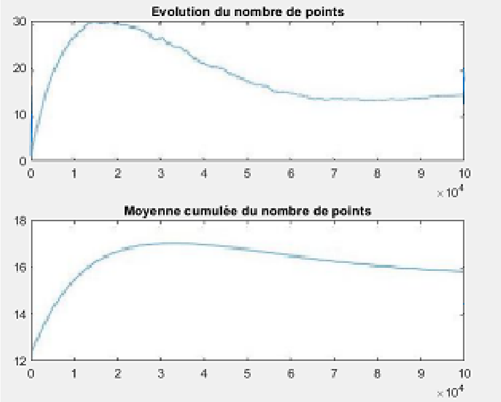

Figure 10. Number of points $n(\mathbf{x})$ and the cumulative average of the number of points represented in function of the number of iterations

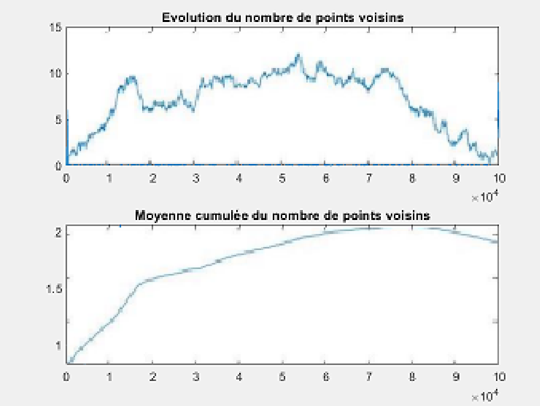

Figure 11. Number of neighbourhood point $n_b(\mathbf{x})$ relationships and its cumulative average in function of the number of iterations

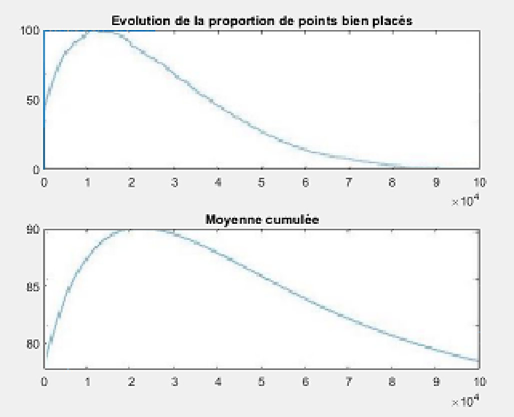

Figure 12. Percentage of well-placed points over the total number of points $\frac{n_d(\mathbf{x})}{n(\mathbf{x})}$ and the cumulative average of that number in function of the number of iterations

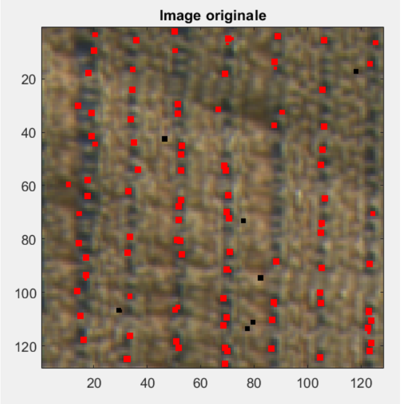

Figure 13. Generated point set over the original image : well-placed points appear in red, badly-placed points in black.