# Image Inpainting using Particle Filters and Markov Random Fields

Nikolas Borrel-Jensen (exam number 27)
*Department of Computer Science*
*University of Copenhagen*
*Copenhagen, Denmark*
*Email: mail@nikolasborrel.com*

***Abstract*—We present an algorithm for solving the image inpainting problem based on a field of locally interacting particle filters. A Markov Random Field connecting neighbour particle filters are introduced to ensure global consistency. The benefit for using global interacting particle filters is that multiple solutions are kept alive and hence not freeze the solution in a possible non-optimal state.**

**The theory behind Markov Random Fields and particle filtering are examined to support the review of the inpainting algorithm. A Matlab implementation has been done and four experiments are conducted on diverse images.**

**Finally, the results from the experiments are compared with an open source inpainting application.**

***Keywords*-Inpainting, Markov Random Field, Particle Filter.**

## I. Introduction

Inpainting is the process of reconstructing missing parts of a picture caused by for example damage to the image. The problem is how to define visually plausible or pleasant solutions.

The method used and implemented for this report is the method presented in [1] and is using a combination of exemplar based [2] and statistical methods [3][4]. The advantage of the proposed method is, that some of the drawbacks of exemplar based methods and statistical methods to some extend are resolved. That is, for large scale deterministic geometric structures statistical methods tends to fail whereas exemplar based methods can fail when two structures is flowing from oppposite sides of the the hole, making the reconstruction difficult in the area where the two structures meet [1]. The new method proposed by [1] is to maintain multiple local hypotheses about solutions at each pixel position in the hole and for that particle filters is used to approximate the probability distribution over image patches centered at the position. A Markov Random Field describing local interactions between pixels centered at each patch in the region to inpaint is used together with the Iterated Conditional Modes (ICM) algorithm for a globally consistent solution.

The aim for this project is two-fold: To describe the theory of Markov Random Fields and particle filtering; and to implement and explain the inpainting method in [1] and compare the results with GIMP 2.6.8[1] using the Resynthesizer inpainting plug-in[2].

[1]http://www.gimp.org/ (Wed June 23 13:35:37 CET 2010)

[2]http://www.logarithmic.net/pfh/resynthesizer (Wed June 23 13:35:37 CET 2010)

The paper is organised as follows: Section II introduces the theory behind Markov random fields, Section III introduces non-linear Baysian tracking by defining a generic non-linear system described in state-space form. Section IV is describing sampling in general and more specific importance sampling and resampling, Section V is describing the two sequential sequential Monte Carlo methods Sequential Importance Sampling and particle filters. In Section VI the inpainting method proposed in [1] is described, including the definition of the MRF image model and the spatial consistency between pathes, how to sample an inpainting solution from the MRF image model and how to update the image model with respect to the distribution over image patches. In Section VII an overview of the implementation is given, in Section VIII the method is applied on different images and compared with an open source inpainting application and finally a conclusion is made in Section IX.

## II. Markov Random Fields

A *Markov Random Field* (MRF) is a graphical model with nodes $V$ connected with undirected edges $E$, where each node correspond to a variable or a group of variables. Let $G = (V, E)$ be an undirected graph and $\mathbf{X} = \mathbf{x}$ be a set of random variables, then $G$ forms a Markov random field with respect to the joint probability distribution $p(\mathbf{X} = \mathbf{x}) \geq 0$ if and only if graph separation in $G$ implies conditional independence: if a set of nodes $\mathbf{Z}$ is removed from $G$ and separates $\mathbf{X}_v \in \mathbf{X}$ and $\mathbf{X}_w \in \mathbf{X}$, $v \neq w$, then $p(\mathbf{X} = \mathbf{x})$ must assert that $\mathbf{X}_i$ and $\mathbf{X}_j$ are conditionally independent given the random variables corresponding to $\mathbf{Z}$. This can be written as

$$p(\mathbf{x}_v|\{\mathbf{x}_w : w \in \mathcal{D} \wedge w \neq v\}) = p(\mathbf{x}_v|\{\mathbf{x}_w : w \in \mathcal{N}(v)\}) \tag{1}$$

where $\mathcal{D}$ is the domain for $\mathbf{x}$ and $\mathcal{N}(\mathbf{v}) \subset \mathcal{D}$ is the neigbourhood system for $\mathbf{x}_v$. Since it is usually very difficult to describe a MRF by the local characteristics in (1), the equivalence can be described through the Hammersley-Clifford theorem: $\mathbf{X}$ is a MRF with respect to the neighbour system $\mathcal{N}$ if and only if $p(\mathbf{x})$ is a Gibbs distribution

$$p(\mathbf{x}) = \frac{1}{Z}\exp(-U(\mathbf{x})) = \frac{1}{Z}\exp\left(-\sum_C \psi_C(\mathbf{x})\right) \tag{2}$$

where Z is the normalisation constant

$$Z = \sum_w \exp(-U(\mathbf{x}))$$

$U(\mathbf{x}) = \sum_C \psi_C(\mathbf{x})$ is the energy function and $\psi_C$ denotes the clique potentials [5].

We want to find the value $\hat{\mathbf{x}}$ that optimise each of the local probabilies $p(\mathbf{x}_v|\{\mathbf{x}_w : w \in \mathcal{N}(v)\})$ such that the overall joint probability $p(\hat{\mathbf{x}})$ is maximised. Since it is difficult to maximise the joint probability of an MRF, several optimisation schemes has been proposed. For the inpainting problem, we use the *Iterated Conditional Modes* (ICM) as proposed in [1].

The ICM algorithm is simply an application of coordinate-wise gradient ascent and uses the "greedy" strategy in the iterative local approximation. Denote a configuration of $\mathbf{x}$ by $\mathcal{C}_{\mathbf{x}_{t-1}}$. Then the energy function $U$ is then computed for a new configuration that differs at most in one coordinate as

$$\mathbf{x}_t = \arg \min_{\eta \in \mathcal{C}_{\mathbf{x}_{t-1}}} U(\eta) \tag{3}$$

This iterative approach is continued until convergence is obtained.

## III. Non-linear Bayesian Tracking

For many application areas, it is important to model a dynamical system where elements of nonlinearity and non-gaussianity is included. Moreover, it is important to process data on-line as it arrives both because of time usage and because of adaption for rapid signal changes. We will derive the state-space model for the Sequential Importance Sampling (SIS) algorithm and the particle filter, but before going into these specific algorithms, we define the general problem of tracking [6] by considering the following nonlinear dynamic system, where $t$ is the time evolution:

- System model

$$\mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}) \tag{4}$$

- Measurement model

$$\mathbf{z}_t = \mathbf{g}_t(\mathbf{x}_t, \mathbf{v}_t) \tag{5}$$

The functions $\mathbf{f}_t(\cdot)$ and $\mathbf{g}_t(\cdot)$ are possibly nonlinear functions, where $\mathbf{f}$ is generating the hidden state $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$ and the noise disturbances $\mathbf{u}_{t-1}$ and $\mathbf{g}$ is generating the observed data $\mathbf{z}$ given the hidden state $\mathbf{x}_t$ and the noise disturbances $\mathbf{v}_t$. The system model describes the evolution of the state with time and the measurement model is relating the noisy measurements to the state.

From a Bayesian perspective, the tracking problem is to recursively calculate the probability for a state $\mathbf{x}_t$ given the data $\mathbf{z}_{1:t}$, thus it is required to construct the probability density function $p(\mathbf{x}_k|\mathbf{z}_{1:t})$. It is assumed that $\mathbf{x}_t$ is Markovian, since (4) describes a Markov process of order one and is defined by the system equation and the known statistics of $\mathbf{v}_{k-1}$. It is further assumed that the initial state $\mathbf{x}_0$ is distributed according to a density function $p(\mathbf{x}_0|\mathbf{z}_0) = p(\mathbf{x}_0)$. Then the density function may be obtained recursively by a prediction and an correction step:

**Prediction stage.** Suppose the density function $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ is available, we can then use the system model (4) to obtain the prior of the state at time $t$ via the Chapman-Kolmogorov equation

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{x}_{k-1} \tag{6}$$

**Correction stage.** At time step $t$ a measurement $\mathbf{z}_t$ becomes available and can be used to update the prior via Bayes' rule

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{7}$$

where

$$p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{z}_{1:t-1}) d\mathbf{x}_t \tag{8}$$

is the normalization factor depending on the likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ defined by the measurement model (5) and the known statistics of $\mathbf{v}_t$. The measurement $\mathbf{z}_t$ is used to modify the prior density to obtain the required posterior density of the current stage.

The recursive relationship in (6) and (7) can be used to find the optimal posterior solution but cannot in general be solved analytically. The Kalman filter can be used under restrictive assumptions, whereas the particle filtering methods can be used less restrictive assumptions and solves the problem by approximates the optimal Bayesian solution.

## IV. Importance Sampling and Resampling

In this section we will explain sampling and more specific *importance sampling* [5].

In many problems, we would like to make predictions from data. For the Monte-Carlo methods we need the posterior probability for evaluating expectations which we can use to make predictions. The expectation of some function $h(\mathbf{x})$ with respect to a possibly highly complex probability distribution $p(\mathbf{x})$, is given by

$$\mathbb{E}[h] = \int h(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \tag{9}$$

In the case where this cannot be achieved analytically, sampling methods can then be used to obtain a set of samples $\mathbf{x}^{(i)}, i = 1, \ldots, N$ drawn independently from the distribution $p(\mathbf{x})$. We can approximate the distribution $p$ by point masses by a finite sum

$$\tilde{h} = \frac{1}{N} \sum_{i=1}^{N} h(\mathbf{x}^{(i)}) \tag{10}$$

Suppose, however, that we cannot sample directly from the distribution $p(\mathbf{x})$ but that we can evaluate $p(\mathbf{x})$ easily for any given value $\mathbf{x}$. *Importance sampling* is then based on sampling from another distribution $q$, having a support larger that $p$, from which it is easy to draw samples. So we make $N$ random draws $\mathbf{x}^{(i)}, i = 1, \ldots, N$ from $q$ instead of $p$. A correction has to be made to ensure that the obtained estimate is an unbiased estimate of $\mathbb{E}[h]$ and involves assigning a positive weight to each of the random

points. The expectation $\mathbb{E}[h]$ can be estimated using a *weighted average*:

$$\mathbb{E}[h] = \int h(\mathbf{x}) \frac{q(\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})} d\mathbf{x} \quad (11)$$

$$= \int h(\mathbf{x}) r(\mathbf{x}) q(\mathbf{x}) d\mathbf{x} \quad (12)$$

$$\approx \sum_{i=1}^{N} \frac{\tilde{w}^{(i)}}{\sum_{j=1}^{N} \tilde{w}^{(j)}} h(\mathbf{x}^{(i)}) \quad (13)$$

where $r(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ is the *importance function* and $\tilde{w}^{(i)} = r(\mathbf{x}^{(i)})$.

## V. Sequential Importance Sampling and Particle Filter

We now specialise the sampling techniques in IV to the sequential setting in the model defined by (4) and (5) and the corresponding prediction and correction recursion given in (6) and (7). We would like to sample from the posterior distribution $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})$, but since this is in general imposible due to the complexity of the distribution, we stick to a sequential version of the importance sampling procedure. The idea is to sample *N particle paths* $\tilde{\mathbf{x}}_t^{(i)}, i = 1, \ldots, N$ from an importance distribution $q(\mathbf{x}_{1:t}|\mathbf{z}_{1:t})$ and compute the unnormalised importance weights [7]

$$\tilde{w}_t^{(i)} = \frac{p(\tilde{\mathbf{x}}_{1:t}^{(i)}|\mathbf{z}_{1:t})}{q(\tilde{\mathbf{x}}_{1:t}^{(i)}|\mathbf{z}_{1:t})}, i = 1, \ldots, N \quad (14)$$

Using the self-normalised importance samling estimator, we can approximate the expectation of a function $h$ defined on the path space by

$$\mathbb{E}(h) = \int h(\mathbf{x}_{1:t}) p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}) d\mathbf{x}_t \quad (15)$$

$$\approx \sum_{i=1}^{N} \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}} h(\tilde{\mathbf{x}}_{1:t}^{(i)}) \quad (16)$$

We will refer to the *normalised weights* by $w_t^{(i)} = \tilde{w}_t^{(i)} / \sum_{j=1}^{N} \tilde{w}_t^{(j)}$.

We need to carry out the importance distribution in a sequencial manner for the importance sampling procedure, and the trick is now to write the importance distribution such that it factorises in a form similar to that of the target posterior distribution:

$$q(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}) = \overbrace{q(\mathbf{x}_{1:t-1}|\mathbf{z}_{1:t-1})}^{\text{keep existing path}} \overbrace{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)}^{\text{extend path}} \quad (17)$$

The importance weights then takes the form

$$\tilde{w}_t^{(i)} = \frac{p(\tilde{\mathbf{x}}_t^{(i)}|\mathbf{z}_{1:t})}{q(\tilde{\mathbf{x}}_t^{(i)}|\mathbf{z}_{1:t})} \quad (18)$$

$$\propto w_{t-1}^{(i)} \times \frac{p(\tilde{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)}) p(\mathbf{z}_t|\tilde{\mathbf{x}}_t^{(i)})}{q(\tilde{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)}, \mathbf{y}_t) p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \quad (19)$$

This decomposition implies that we only need the previous weight to time $t-1$ to compute the weight to time $t$ and these can be computed recursively. Since we are using the self-normalising form of the importance sampling in (16), it does not matter that the weights can only be computed up to a normalisation constant as indicated by the proportionality sign $\propto$ nor do we need to compute term $p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ since it does not depend on the state sequence and hence is only a normalisation constant.

The sequential importance sampling method is summarised in Algorithm 1 [7].

The described algorithm so far is known as the SIS algorithm, but because we are sampling from a very high dimensional state space, namely the entire path history up to time $t$, the weights will become highly degenerate after a few time steps, meaning that a small portion of the weights will contain nearly all of the probability mass and hence most of the particles will not contribute to the expection function in (16). The solution is to sample $N$ draws from the current set of particles using the normalised weights $w$ as probabilities of selection, eliminating small importance weights, whereas those with large importance weights are replicated. The remaining importance weights are then reset to $\frac{1}{N}$. Applying the resampling step to the SIS algorithm leads to the particle filter, but we will not investigate this further, since the inpainting algorithm is not using resampling for reasons that will be clarified in the next section.

If the importance distribution $q(\tilde{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)}, \mathbf{y}_t)$ is set to the prediction density $p(\mathbf{x}_{v,t}|\mathbf{x}_{v,t-1}^{(i)})$, the recursive formulation simplifies as

$$\tilde{w}_{v,t}^{(i)} \propto \tilde{w}_{v,t-1}^{(i)} p(\mathbf{z}_{v,t}|\mathbf{x}_{v,t}^{(i)}) \quad (20)$$

known as the *bootstrap filter*. For the inpainting method we will use this simplified formulation.

## VI. Inpainting

We will in the following describe the inpainting method from [1] which uses MRF and particle filtering as described above. The inpainting problem is to restore areas in pictures where the data is missing and the problem is how to define visually plausible or pleasing solutions.

The method can be characterised as lying in the intersection between exemplar based (see e.g. [2]) and statistical methods (see e.g. [3][4]), resolving the problem from exemplar based methods where structures flowing from opposite sides of the boundary do not meet correctly and the problem from statistical methods where stochastic textures tend to produce blurred reconstructions and not to reproduce the texture[1]. This is done by maintaining multiple local hypoteses about solutions at each pixel position in the hole by maintaining an estimate of the probability distribution over image patches centered at the position. Since the distribution in general is non-trivial, we use the particle filter framework. Besides that, we need to find global optimal solutions, since the particle filter only concerns local path distributions. By introducing a MRF model describing local interaction between overlapping image patches and specified in terms of local patch distribution, we reach for global spatially consistency. The ICM algorithm is applied on the MRF where we locally pick a patch which maximises the conditional probability

**Algorithm 1** Sequential Importance Sampling

**for** $i = 1, \ldots, N$ **do**
  Sample $\tilde{\mathbf{x}}_0^{(i)} \backsim q(\mathbf{x}_0|\mathbf{z}_0)$
  Assign initial importance weights

$$\tilde{w}_0^{(i)} = \frac{p(\mathbf{z}_0|\tilde{\mathbf{x}}_0^{(i)})p(\tilde{\mathbf{x}}_0^{(i)})}{q(\tilde{\mathbf{x}}_0^{(i)})|\mathbf{z}_0}$$

**end for**
**for** $t = 1, \ldots, T$ **do**

  **for** $i = 1, \ldots, N$ **do**
    Propagate particles:

$$\tilde{\mathbf{x}}_t^{(i)} \backsim q(\tilde{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)}, \mathbf{z}_t)$$

    Compute weight:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(\mathbf{z}_t|\tilde{\mathbf{x}}_t^{(i)})p(\tilde{\mathbf{x}}_t^{(i)}|\tilde{\mathbf{x}}_{t-1}^{(i)})}{q(\tilde{\mathbf{x}}_t^{(i)}|\mathbf{x}_{t-1}^{(i)}, \mathbf{z}_t)}$$

  **end for**
  Normalise weights:

$$w_t^{(i)} = \tilde{w}_t^{(i)} / \sum_{j=1}^{N} \tilde{w}_t^{(j)}, i = 1, \ldots, N$$

  Compute filtering estimate:

$$\mathbb{E}[h_t] = \sum_{i=1}^{N} w_t^{(i)} h_t(\tilde{\mathbf{x}}_t^(i))$$

**end for**

given its neighbours. The pixel at this position is set to the center pixel intensity of the selected path and this is done for every pixel in the hole in an arbitrary visiting order. The particle filter is then updated and the ICM algorithm is repeated. This procedure is repeated until convergence is reached.

### A. A Patch Based Image Model

In this section we will introduce the notation used for the underlying image model used for global spatial consistency of the inpainting solution. The model used is a MRF (see Section II) defined through overlapping image patches.

Let $\mathbf{I} : \mathcal{D} \to \mathbb{R}^d$ denote an image on the domain $\mathcal{D} \subset \mathbb{Z}^2$ and let $\Omega$ be the region to be inpainted, where $d$ is the number of color channels. The known part of the picture is denoted $\mathcal{D}\backslash\Omega$ and will be used as data for learning the MRF model and generating a solution for the inpainting problem.

Let $\mathbf{X}_v \in \mathcal{P}$ be an image patch with center $v \in \mathcal{D}$ and let $\mathcal{P} = \mathbb{R}^{p \times d}$ denote the space of image patches consisting of $p$ pixels. Let $\mathcal{X}_v \subset \mathcal{D}$ denote the set of sites covered by the patch $\mathbf{X}_v$. These definitions are illustrated in Figure 1.

We define the *neighbourhood system* for image patches $\mathbf{X}_v$: two sites $v, w \in \mathcal{D}$ and $v \neq w$ are neighbours if and only if $w \in \mathcal{X}_v$ and $v \in \mathcal{X}_w$. The set of all *neighbours* of $v$ will be denoted by $\mathcal{N}(v) = \mathcal{X}_v \backslash v$. The collection of all neighbourhoods $\mathcal{N} = \{\mathcal{N}(v) : v \in \mathcal{D}\}$ is called the *neighbourhood system*.

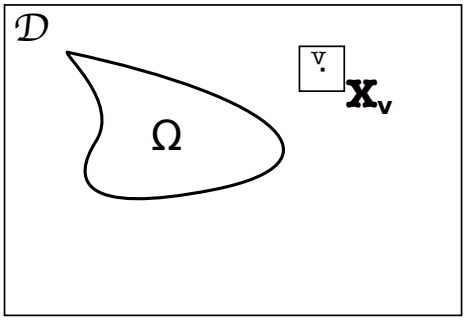



Figure 1. Illustration of the image domain $\mathcal{D}$, the region to be inpainted $\Omega$ and a patch $\mathbf{X}_v$ and its center point $v$

For simplicity of notation, $\mathbf{x}_v$ will be used for both the random variable $\mathbf{X}_v$ and its realisation throughout the report.

### B. Spatial Consistency

By spatial consistency we mean that neighbour patches are similiar and therefore we have to define what we mean by that. We define a similarity measure between overlapping frames given by a function $s : \mathcal{P} \times \mathcal{P} \to \mathbb{R}_+ \cup \{0\}$. A variety of choices are available and the simple sum of squared differences (SSD) similarity measure proposed by [1] is used:

$$s(\mathbf{x}_v, \mathbf{y}_w) = \sum_{i \in \mathcal{X}_v \cap \mathcal{Z}_w} (\mathbf{x}_{v,i} - \mathbf{z}_{w,i})^2 \tag{21}$$

We make a specific choice of the local potential energy $U$: Assume the two probability densities $p(\mathbf{x}_v)$ and $p(\mathbf{x}_w)$ are given and $p(\mathbf{x}_v) > 0$ for all $\mathbf{x}_v \in \mathcal{D}$, then

$$\begin{aligned} &U(\mathbf{x}_v, \mathbf{x}_1, \ldots, \mathbf{x}_k) \\ &= \frac{1}{K} \frac{1}{p(\mathbf{x}_v)} \sum_{w=1}^{K} \int p(\mathbf{x}_w) s(\mathbf{x}_v, \mathbf{x}_w) d\mathbf{x}_w \end{aligned} \tag{22}$$

where $\mathbf{x}_w \in \mathcal{N}(v)$. The potential energi is the average expected similarity between $\mathbf{x}_v$ and its neighbours $\mathbf{x}_1, \ldots, \mathbf{x}_K$ with respect to $p(\mathbf{x}_w)$. Futhermore, the potential is weighted as proposed by [1] by the probability density of $\mathbf{x}_v$. The interpretation of this potential is that patches $\mathbf{x}_v$ that on average are similar to its neighbours are given a high probability weighted by how likely this patch is, given by $1/p(\mathbf{x}_v)$.

The probability distributions $p(\mathbf{x}_v)$ and $p(\mathbf{x}_w)$ have to be learned from the known data in the image and for that we use particle filters. Ones these two distrubutions are known, we can sample from the MRF model using the ICM algorithm for a solution in the area to be reconstructed. This is explained in the following section.

### C. Sampling an Inpainting Solution from the MRF Image Model

We would like neighbour patches given by their corresponding sites to be consistent, i.e. neighbour sites should

not be very different. We sample from the MRF using the ICM method where each site $v$ is sampled such that $p(\mathbf{x}_v|\mathbf{x}_1,\ldots,\mathbf{x}_K)$ is maximised. Adjusting the equation in (2) to be conditioned on the neighbour patches, we get

$$p(\mathbf{x}_v|\mathbf{x}_1,\ldots,\mathbf{x}_K) = \frac{1}{Z}\exp(-U(\mathbf{x}_v|\mathbf{x}_1,\ldots,\mathbf{x}_K)) \quad (23)$$

Maximising $p(\mathbf{x}_v|\mathbf{x}_1,\ldots,\mathbf{x}_K)$ equals minimising the potential energy function $U(\mathbf{x}_v|\mathbf{x}_1,\ldots,\mathbf{x}_K)$.

For each patch, we choose the particle given by the particle filter, that minimise the energy with respect to the neighbour patches:

$$\hat{\mathbf{x}}_{v,t} = \arg\min_{x_{v,t}^{(1)},\ldots,\mathbf{x}_{(v,t)}^{(M)}} U(\mathbf{x}_{v,t}^{(i)},\mathbf{x}_1,\ldots,\mathbf{x}_K) \quad (24)$$

We assume that the model is defined at time $t$, that is the (previous) observed solutions $\mathbf{z}_{v,1},\ldots,\mathbf{z}_{v,t}$ are known. Then $p(\mathbf{x}_v)$ can replaced by the entirely defined approximation of the conditional $p(\mathbf{x}_{v,t}|\mathbf{z}_{v,1},\ldots,\mathbf{z}_{v,t})$ based on a Monte-Carlo approximation of the filtering density over $M$ particles:

$$p(\mathbf{x}_{v,t}|\mathbf{z}_{v,1},\ldots,\mathbf{z}_{v,t}) \approx \sum_{i=1}^{M}\omega_{v,t}^{(i)}\delta_{x_{v,t}^{(i)}}(\mathbf{x}_{v,t}) \quad (25)$$

The same approximation is valid for $p(\mathbf{x}_{w,t}|\mathbf{z}_{w,1},\ldots,\mathbf{z}_{w,t})$.

Substituting $p(\mathbf{x}_v)$ and $p(\mathbf{x}_w)$ in (22) by (25) we get

$$\begin{aligned} &U(\mathbf{x}_{v,t}^{(i)},\mathbf{x}_1,\ldots,\mathbf{x}_K) \\ &= \frac{1}{K\omega_{v,t}^{(i)}}\sum_{w=1}^{K}\sum_{j=1}^{M}\omega_{w,t}^{(j)}s(\mathbf{x}_{v,t}^{(i)},\mathbf{x}_{w,t}^{(j)}) \end{aligned} \quad (26)$$

where $\mathbf{x}_{w,t}^{(j)}$ is the $j$'th particle of neighbour $\mathbf{x}_{w,t}$ of patch $\mathbf{x}_{v,t}$.

By using this selection criterion for all patches using a random visiting order, we encourage a spatially consistent solution between all selected particles.

For the inpainting solution, only the center points from each found patch $\hat{\mathbf{x}}_{v,t}$ is used and the set of these points make up the new patches to fill the hole. The efford ensuring consistency is made at the expense of the structure of the picture. For recovering the structure, the patches made up from the center points are projected back onto the data $\mathcal{D}\backslash\Omega$ as proposed by [1]. This is done by finding the most similar patch in the data using the SSD measure for each of the learned patches and replacing the original center point by the center point from the found patch.

### *D. Updating the Image Model*

In this section we describe how to update the image model described in VI-A at time $t$, where it is assumed that the previous measurements $\mathbf{z}_1,\ldots,\mathbf{z}_t$ are known. The update step at time $t$ corresponds to estimate the conditional distribution $p(\mathbf{x}_{v,t}|\mathbf{z}_{v,1},\ldots,\mathbf{z}_{v,t})$ (as given in (23)) of image patches and is used to compute the local potential energy function defined in (22). For this to happen we need to estimate $\mathbf{x}_{v,t}$ and $\mathbf{x}_{w,t}$ which is done using particle filtering. Particle filtering is performed by two step, namely prediction and correction.

*1)* **Prediction Step:** The prediction step is described by the transition law $p(\mathbf{x}_{v,t}|\mathbf{x}_{v,t-1}^{(i)})$ for each new particle $\mathbf{x}_{v,t}^{(i)}$ and hence we do not need to know the exact from of $\mathbf{f}$ nor the noise term in the system model described in (4). We can write the transition law as a sum of Dirac distributions:

$$p(\mathbf{x}_{v,t}|\mathbf{x}_{v,t-1}^{(i)}) \propto \sum_{k=1}^{K}\delta_{\mathbf{x}_{v,t-1}^{(i),k}}(\mathbf{x}_{v,t}) \quad (27)$$

where $\{\mathbf{x}_{v,t-1}^{(i),k}\}_{k=1:K}$ are the $K$ most similar patches of the particle $\mathbf{x}_{v,t-1}^{(i)}$ in $\mathcal{D}\backslash\Omega$ where the SSD measure is used to compute similarity.

The formulation in (27) is saying, that each of the particles have the same probability and hence we pick on of the particles randomly.

In oppposite to exemplar-based approaches, we maintain a set of the $K$ most similar patches such that local uncertainty about the optimal patch is maintained due to the global consistency between patches.

*2)* **Correction Step:** Given the new measurement $\mathbf{z}_{v,t}$, the correction step is used for updating the weights $\tilde{w}_t^{(i)}$ for each particle $(i)$ in (19) and compute the conditional distributions $p(\mathbf{x}_{v,t}|\mathbf{z}_{v,1},\ldots,\mathbf{z}_{v,t})$. Here $\mathbf{z}_t$ is the inpainted (measured) solution after iteration $t-1$ constructed by the image model after projection on the data as described in Section VI-C and $\mathbf{x}_{v,t}$ is then the predicted patch after correction. The likelihood function corresponding to (5) is given by $p(\mathbf{z}_{v,t}|\mathbf{x}_{v,t}^{(i)})$ for each of the particles $i$ and assuming conditional independence of pixels in the patch $\mathbf{z}_{v,t}$ given $\mathbf{x}_{v,t}$, for one particle we can write

$$p(\mathbf{z}_{v,t}|\mathbf{x}_{v,t}^{(i)}) \propto \exp\left(-\sum_{j\in\mathcal{X}_v}\frac{(\mathbf{z}_{v,j,t}-\mathbf{x}_{v,j,t}^{(i)})^2}{2\sigma^2}\right) \quad (28)$$

where the subscript $j$ in $\mathbf{z}_{v,j,t}$ and $\mathbf{x}_{v,j,t}$ is indicating the intensity value of the $j$'th pixel in $\mathbf{z}_{v,t}$ and $\mathbf{x}_{v,t}$, respectively. This likelihood is then used to correct the weights in (20).

Having updated the weights, we can now compute the conditional probability density $p(\mathbf{x}_{v,t+1}|\mathbf{z}_{v,1},\ldots,\mathbf{z}_{v,t+1})$ by (25) and hence compute the local potential energy in (22) which corresponds to update equation (26):

$$\begin{aligned} &U(\mathbf{x}_{v,t+1}^{(i)},\mathbf{x}_{1,t+1},\ldots,\mathbf{x}_{K,t+1}) \\ &= \frac{1}{K\omega_{v,t+1}^{(i)}}\sum_{w=1}^{K}\sum_{j=1}^{M}\omega_{w,t+1}^{(j)}s(\mathbf{x}_{v,t+1},\mathbf{x}_{w,t+1}) \end{aligned}$$

Then a new inpainting solution can be sampled from the image model. The algorithm is then repeated until a satifying solution exists.

In general we should use a resampling step for preventing the weights to degenerate. In the inpainting algorithm described, no resampling step is performed and the reason for that is because of the great randomness for choosing particles in (27) and also that the found patch is projected back on the data resulting in a new site. This ensures that a small part of the weights will not attain to get all of the probability mass.

## VII. Implementation

The inpainting method has been implemented in Matlab and this section gives an overview of the implementation. In contrary to the somewhat abstract mathematical formulations in Section VI, we clarify the steps in a concrete manner and refer to the mathematical formulations when appropriate.

Furthermore, an optimisation is done when the similarity between patterns has to be computed and is done by K-means clustering.

### A. Optimisation using K-means

Searching for similar patterns in the data is very time consuming, therefore it is beneficial to make the search space smaller. For this purpose we have implemented a simpel method called K-means clustering[5]. K-means clustering is an iterative approach for partitioning an image into $K$ clusters. For our problem, we want to assign each patch $\mathbf{x}_v$ in the image to one of $K$ clusters and our implemented algorithm can be summarised as follows:

1) Pick $K$ cluster centers - given by patches - from $\mathcal{D}\backslash\Omega$ by hand.
2) Assign each patch in $\mathcal{D}\backslash\Omega$ to the cluster that minimise the SSD distance measure between the patch and the cluster center.
3) Re-compute the cluster center by averaging all of the patches in the cluster.
4) Repeat step 2 and 3 until convergence is attained, that is, no cluster centers changes value.

We have done the experiments both with and without K-means clustering.

### B. Overview of the implementation of the inpainting method

An overview of the steps in the implemented inpainting method is summarised in Tabel I. When appropriate, we refer to the more abstract mathematical formulation for the different steps.

**Initialisation:**

- Optional: perform K-means clustering.
- Fill $\Omega$ with uniformly distributed noise.
- Generate $M$ particles with uniformly distributed noise, i.e. generate the particles patches $\mathbf{x}_0^{(i)}$ for $i = 1 \dots M$.
- Set $\omega_0^{(i)} = \frac{1}{M}$ for $i = 1 \dots M$.

**for** $t = 1 \dots T$

**Prediction:** Find $K$ nearest neighbours $\{\mathbf{x}_{v,t-1}^{(i),k}\}_{k=1:K}$ for each particle patch $\mathbf{x}_{v,t-1}^{(i)}$ and choose one randomly as given by (27).

**Correction:**

- Correct the weights by $\omega_{v,t}^{(i)} = \frac{\omega_{v,t-1}^{(i)} p(\mathbf{z}_{v,t}|\mathbf{x}_{v,t}^{(i)})}{\sum_{i=1}^{M} \omega_{v,t-1}^{(i)} p(\mathbf{z}_{v,t}|\mathbf{x}_{v,t}^{(i)})}$ where $p(\mathbf{z}_{v,t}|\mathbf{x}_{v,t}^{(i)})$ is given by (28).

**Ensure structural consistency:**

- Perform ICM in MRF: Select the particles that minimise the potential energy $U(\mathbf{x}_{v,t}^{(i)}, \mathbf{x}_1, \dots, \mathbf{x}_K)$ given by (26).
- Project the patch consisting of the center points from all the selected optimal particles for each site $v$ onto the data and find the most similar patch. Use that center point as the solution for $\mathbf{z}_{v,t}$.

**endfor**

Table I
Overview of the implementation for the inpainting method.

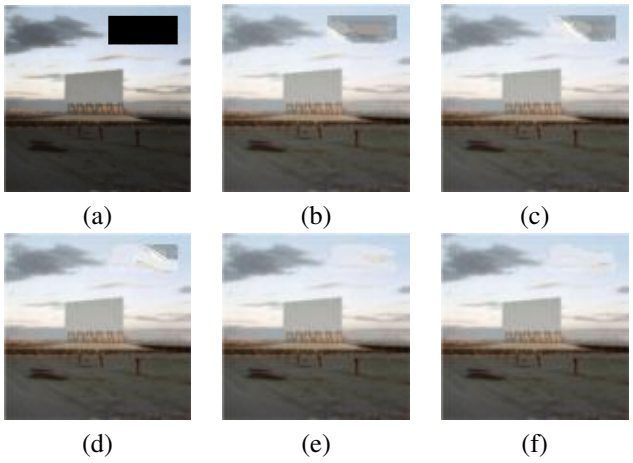


Figure 2. Inpainting for image "Mehldau cover" using K-means clustering with $K = 5$. The patch size is $13 \times 13$ and the number of particles is 10. (a) Original image with hole to be inpainted. (b) Inpainted image for $t = 3$. (c) Inpainted image for $t = 5$. (d) Inpainted image for $t = 7$. (e) Inpainted image for $t = 9$. (f) Final result for $t = 10$.

## VIII. Results

In this section we will apply the inpainting algorithm on three different images varying the number of particles, the patch size and using both the plain inpainting method without pre-processing the data using K-means clustering and the optimised method where K-means is used with 5 clusters and the initial cluster centers is hand picked.

### A. Image: Mehldau cover

In Figure 2 we see the results after timestep $t = 0, t = 3, t = 5, t = 7, t = 9$ and the final result after $t = 10$ using a patch size of $13 \times 13$ and 10 particles and K-means clustering. We clearly see the evolution from (b) to (d) where the sky is being inpainted. It is interesting to see that the right and upper areas are not changing, only the areas in the diagonal area is changing. This could be caused by the algorithm confusing the horisontal line in the middle of the picture with the upper line inpainted by the first iteration and is possibly due to the clustering of the patches meaning that a local minimum is found. For the area that is evolving, the found patches seems to come from the boundary of the clouds in the middle of the image since the inpainted colour is not blue but more pinkish. The final result is quite good but lacks consistency with regard to the blue colour of the sky.

In Figure 3 the convergence of the individual sites for the patches in the inpainted area is depicted with respect to the timestep $t$. At $t = 2$ the variance drops dramatically indicating that the differences between the solutions are stabilising. Indeed, this make senses because we have initialised the inpainting area with noise and hence the difference between $t = 0$ and $t = 1$ should be notable. First at $t = 9$ convergence is reached which is supported

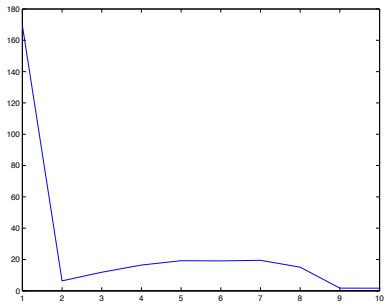



Figure 3. Convergence for the inpainted area. The x-axis is the time step and the y-axis is the difference measured by the squred differences for the individual sites in the patches.

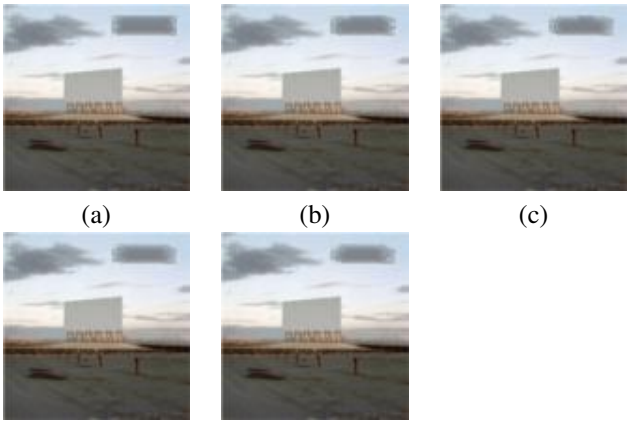



Figure 4. Inpainting for image "Mehldau cover" without using K-means clustering. The patch size is $9 \times 9$ and the number of particles is 3. (a) Inpainted image for $t = 1$. (b) Inpainted image for $t = 2$. (c) Inpainted image for $t = 3$. (d) Inpainted image for $t = 4$. (e) Final result for $t = 5$.

by (e) and (f) in Figure 2. In Figure 4 the inpainting solutions for 5 time steps is depicted. In contrary to the solutions using clustering, the solutions are evolving from all sides of the inpainting area. From the result it seems like the patches are taken from the clouds in the upper left area of the image. Notice also that the new colour in the boundary of the inpainting area is the same blue colour as the remaining sky, making the solution more consistent than the previous solution.

### B. Image: Corduroy

In Figure 5 we see the texture "Corduroy" where (a) and (b) is depicting the original picture and the area to be inpainted, respectively. In (c) the result after 5 iterations is shown and we see that the structure has been well captured, though the black vertical lines are a bit blurred. Also a vertical line is introduced in the right part of the image where the inpainting area ends, but could probably be resolved by doing more iterations.

### C. Image: Trash can

In Figure 6 the image "Trash can" is shown together with the inpainted solutions with and without using clustering. For the solution (c) using K-means, the structure of

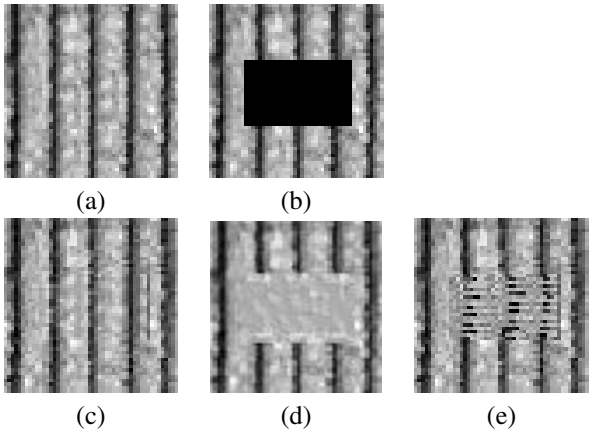



Figure 5. Inpainting for image "Corduroy". 5 iterations are used, $T = 5$. (a) Original image. (b) Hole to be inpainted. (c) Inpainted image without using K-means. The patch size is $9 \times 9$ and the number of particles is 3. (d) Inpainted image using K-means. Patch size is $9 \times 9$ and number of particles is 3. (e) Patch size is $13 \times 13$ and the number of particles is 6.

the picture are not taken into account giving a rectangular shape with no consistency with the horisontal lines of the sea, where only the sky and the ground is fits the solution. The reason for this could be, that the K-means algorithm does not cluster the patches around the horisontal line with a cluster center where the patch in the inpainting area around the same horisontal line has the smallest distance. This seems strange, but we have not persued the reason for that further due to time constraints.

For the solution in (b) no clustering algorithm is used but the result remains the same. Using a higher patch size could probably make the result better since more structure is captured.

### D. Comparison with GIMP using Resynthesize plug-in

In this section we will compare the solutions from the inpainting algorithm presented in this report with the solution from the open source program GIMP using the Resynthesize plug-in for inpainting. The comparison is shown in Figure 7. For the images in the first column, GIMP is doing a good job inpainting with a blue color consistent with the surrounding areas and it is hard to see the area that is inpainted. For our algorithm, the colours are not consistent to the same extend in (b). In (c) the colours are much more consistent and with more iterations the solution may be as good as the GIMP solution with respect to consistency. It looks like more details are captured in the sky, producing a more realistic result in the boundaries. Though, the overall result is not very consistent with the surroundings.

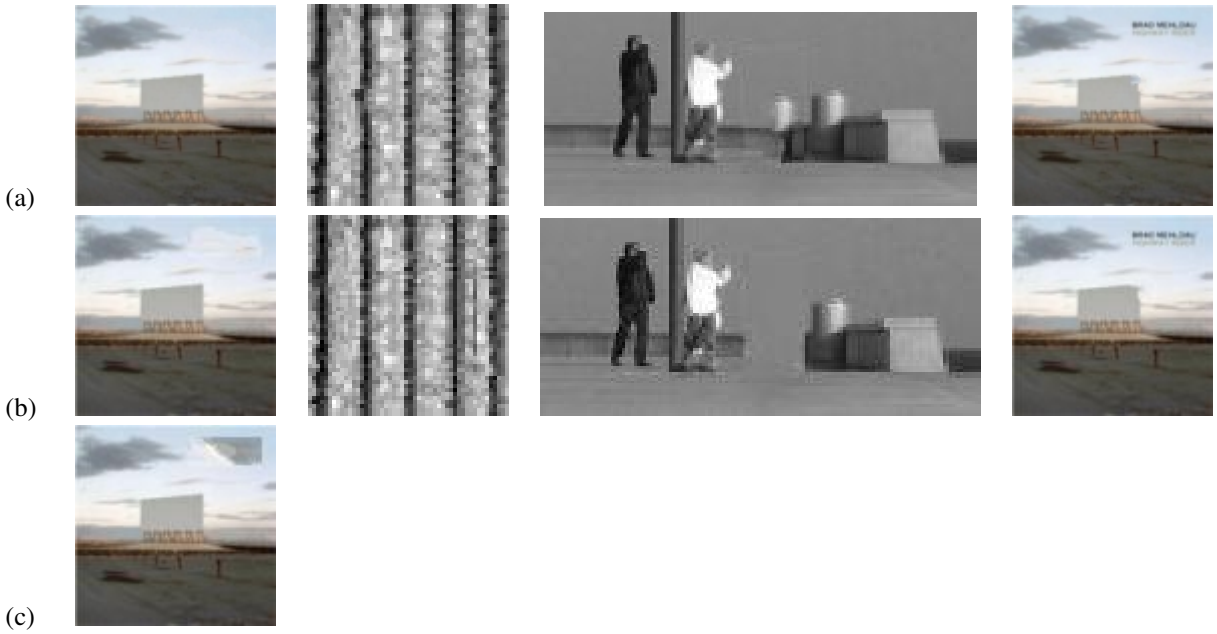
In the second column both solutions are quite good and is more or less in the same range.

In the third column the GIMP implementation captures the trash can and replicated half of it in the inpainted area, which is not a pleasant solution. Though the horisonal line is well captured. Our algorithm lacks the horisontal line, which is neither a satisfying result.

The fourth and last column is showing result for an additional image not discussed in the former section. The

(a) (c) (d)

Figure 6. Inpainting for image "Trash can". 5 iterations are used, $T = 5$. (a) Original image with hole to be inpainted. (b) Inpainted image without using K-means. The patch size is $9 \times 9$ and the number of particles is 3. (c) Inpainted image using K-means. Same results for patch size $9 \times 9$ and $13 \times 13$ and for the number of particles 3 and 6.



(a)

(b)

(c)

Figure 7. Comparison for the inpainting results using the method described in this report and the results from GIMP using the Resynthesize plug-in. (a) GIMP using Resynthesize. (b) + (c) Results from the inpainting algorithm described in the report.

inpainting area is from the upper right corner of the screen and here the GIMP implementation has captured a part of the cloud and used it for the solution. Our algorithm produces an obvious better result, though not perfect.

## IX. Conclusion

We have go through the theory of Markov Random Fields and particle filters and explained the inpainting algorithm in detail both in both a mathematically and practical way. The method has been implemented in Matlab and four experiment comprising divergent images was conducted. The experiments showed that the K-means clustering algorithm made the results remarkably poorer than without the clustering algorithm. For the results not using the K-means clustering algorithm, the inpainting solutions captured the structure well in the "Corduroy"texture image whereas the structure in the image "Trash can" was not captured at all. This could probably be resolved by changing the parameters for the patch size. For the "Mehldau cover", only 5 iterations were done making the final solution insufficient, but making more iterations could probably produce a consistent result with the surrounding area.

The overall conlusion is, that the results where not convincing.

Finally, a test between GIMP with the Resynthesis plug-in and the algorithm presented in the report was done. It proved that the results made by GIMP was notably better in most of the cases, though the presented algorithm was superior for one of the images.

## Acknowledgment

Thanks should go to Kim Steenstrup Pedersen from the Department of Computer Science, University of Copenhagen, for his great help.

## References

[1] A. Cuzol, K. S. Pedersen, and M. Nielsen, "Field of particle filters for image inpainting," *Springer Science, j Math imaging Vis 31:147-156*, 2008.

[2] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 13, NO. 9, pp. 1200-1212*, 2004.

[3] D. Gustavsson, K. S. Pedersen, and M. Nielsen, "Geometric and texture inpainting by gibbs sampling," *In: Swedish Symposium on Image Analysis (SSBA '07)*, 2007.

[4] ——, “Image inpainting by cooling and heating,” *In: Bjarne Ersbøll and Kim Steenstrup Pedersen, editors, Scandinavian Conference on Image Analysis (SCIA ’07), volume 4522 of Lecture Notes in Computer Science, pages 591-600. Springer Verlag*, 2007.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Science, 2006.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 2*, 2002.

[7] O. Cappé, S. J. Godsill, and E. Moulines, “An overview of existing methods and recent advances in sequential monte carlo,” *Proceedings of the IEEE VOL. 95, NO. 6, pp. 899 - 924*, 2007.