UCL Université catholique de Louvain

École polytechnique de Louvain (EPL)

epl

Denoising dose distributions from Monte Carlo simulations of proton therapy treatments.

Dissertation presented by
Marie VALENDUC

for obtaining the Master's degree in
Mathematical Engineering

Supervisor(s)
John A. LEE, Edmond STERPIN, Kevin SOURIS

Reader(s)
Rudi LABARBE, Nick REYNAERT

Academic year 2017-2018

Abstract

Radiation therapy is extensively used for cancer treatments. To compute radiation dose distributions in proton therapy, Monte Carlo dose calculation is known to provide better accuracy than analytical algorithms typically available in the treatment planning system. However, due to the stochastic nature of Monte Carlo methods, the resulting dose maps suffer from unavoidable statistical noise, which may affect clinical metrics such as the dose-volume histogram (DVH). A procedure to quantify this statistical uncertainty is described and discussed in this work. Noise can be reduced by increasing the number of simulated particles, which also increases the computation time. In order to keep the calculation time practical, several noise filtering techniques are studied here, including Gaussian and median filters, wavelet threshold denoising (WTD) and the non-local means (NLM) filter. Improvements regarding performance and speed of the NLM filter are proposed. The denoising performance of all considered methods are compared using various image quality criteria and dosimetric measures. Based on two test cases (i.e., lung and prostate tumor), it was found that the evaluated methods provided acceleration of MC results towards stastically more accurate results. The improved NLM filter produces the best results compared to the other methods.

Acknowledgements

The research included in this master thesis could not have been performed without the support and assistance of many persons. First of all, I would like to thank my three supervisors for mentoring me in the course of this year: Kevin Souris, for his great patience and availability; John A. Lee, for his enlightened advice; and Edmond Sterpin, for his unfailing enthusiasm and original insights. In particular I would like to express my gratitude for the many opportunities I had thanks to their trust.

I would additionally like to thank Loïc Van Hoorebeeck for believing in me and appeasing me when I had doubts. You make me see the life like a proton – always positive.

Finally I would like to extend my deepest gratitude to my family, and especially my mother, without whose love, support and care I could have never completed this work.

Contents

CHAPTER 1

Introduction

1.1 Context and motivations

Proton therapy

According to the World Health Organization, cancer is one of the leading causes of death in the world. A large portion of the cancer patients receive radiation therapy during their treatment. It is used either as the sole treatment or in combination with other therapies, such as surgery and chemotherapy. Photon or electron therapy is usually used. However improvements of the treatment can be expected by using proton beams instead of the conventional photon and electron beams; we can take advantage of the physical characteristic of proton beams, i.e., their finite range and Bragg peak, to increase the dose to the target volume or to decrease the dose to organs at risk [1]. Yet, to benefit from this increased precision in location, accurate dose calculation engines for radiation treatment planning must be used.

Monte Carlo dose calculation

Monte Carlo (MC) simulations are particularly suited for highly accurate dose (energy divided by the mass) predictions in proton therapy. MC dose calculation is a stochastic calculation method where individual protons are simulated as they traverse a numerical representation of the patient anatomy. The interactions of the protons with their environment (e.g., scattering, energy deposition) are simulated based on known probability distributions. This way, the energy deposited in each volume element, termed voxel in this work, can be computed, hence generating a two- or three-dimensional dose map. As MC methods readily allow the modeling of the proton physics through complex geometries, they are very reliable and accurate calculation techniques. In fact they are more accurate than the deterministic dose algorithms conventionally used in the current treatment planning systems. This explains the motivation for MC dose calculation engine in routine clinical practice.

However this high-accuracy method has two significant disadvantages: firstly, its computation time is much longer compared to conventional deterministic methods and secondly, statistical uncertainties (also called noise) are introduced in the calculated dose. Let us first discuss the simulation time issue.

Since the simulation time is proportional to the number of simulated particles, the MC calculation can be accelerated by simulating a smaller amount of protons. Nevertheless, as can be expected, the precision of the results strongly depends on the number of simulated particles. To illustrate this, examples of a two-dimensional dose map are shown in Figure 1.1: these dose maps correspond to the pencil beam

scanning (PBS) proton therapy treatment of a prostate tumor initially planned with the RayStation treatment planning system (TPS) and were computed using the fast Monte Carlo code MCsquare [2, 3]. The left one has been obtained with 10^9 protons in approximately 41min while the right one is the result of a simulation with 10^6 protons generated in approximately 4s. The price that is paid for this tremendous acceleration is observable by comparing the two dose maps: it is seen that the dose distribution presented on the right is less homogeneous, it is corrupted by statistical noise. We herewith come back to the second issue, i.e., the apparition of noise on the dose distribution.

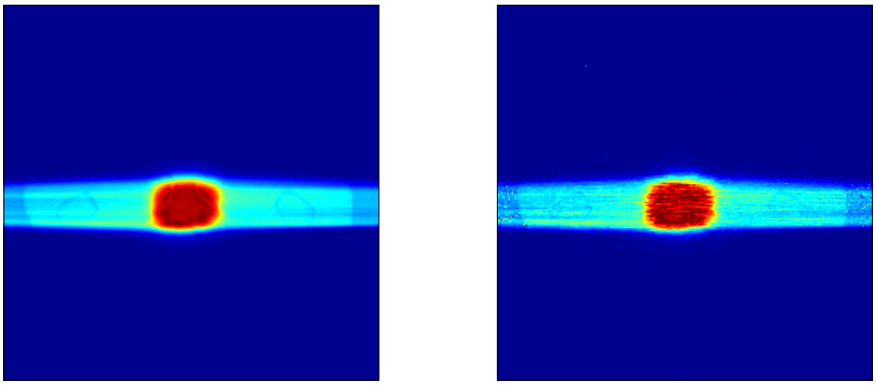


Figure 1.1: Reference and noisy dose maps obtained by simulating 10^9 and 10^6 protons respectively.

Each voxel of a MC computed dose distribution always contains some statistical uncertainty. This means that, for each voxel, the dose value X fluctuates around its mean value. This fluctuation is measured by the standard deviation of the means s_X; this quantity gives an estimate of the MC error. In the limit of a very large number of particles, it can be shown that this error is inversely proportional to the square root of the number of particles N,

$$s_X \propto \frac{1}{\sqrt{N}}. \tag{1.1}$$

Recalling that the simulation time is proportional to N, Equation 1.1 implies that the simulation time must be multiplied by 100 in order to reduce the error by 10. Naturally, the smaller s_X, the better the estimation of the dose value, however, an error-free dose map would take infinite time to calculate. This explains why MC simulations are slow.

Accelerating the MC dose calculation is of crucial importance for two main reasons. Firstly, dose distributions must often be computed almost in real-time. For instance, suppose that a patient is lying on the couch, expecting his proton therapy treatment, then letting him wait for more than a few minutes is not a possible scenario (because it increases the risk that the patient moves, because other patients need to be treated, etc.). Hence MC simulations need to be done very quickly in order to be used in routine clinical practice. Secondly, MC methods are also intended to be used as a tool in treatment optimization: this means that multiple simulations would need to be run for one treatment planning during the optimization algorithm. This underlines once again the importance of reducing the simulation time.

Before discussing solutions to accelerate Monte Carlo simulations, we study the impact of the noise on the dose-volume histogram (DVH).

Effect of statistical uncertainty

Treatment plans are generally assessed by evaluating the DVHs for the target and critical organs. It is thus natural to study how they are affected by the noise. As indicated by its name, the DVH is an histogram relating dose to tissue volume. It reads as follows: the height y corresponding to a dose x represents the fraction of volume receiving a dose greater than or equal to x. Some typical DVH-based metrics, such as the D_{95} or the D_{05}, are usually examined by the clinicians: the D_{95} (resp. D_{05}) is the minimal dose received by 95% (resp. 5%) of the volume.

Figure 1.2 gives an example of DVH for the target volume, where the D_{95} and D_{05} are highlighted. The red curve depicts an hypothetical situation where the DVH is a step function, i.e., the entire tumor receives a uniform dose [4]. In practice, however, it is impossible to deliver the exact same dose to the full target volume because of the trade-off between target coverage and organ at risk sparing; the blue curve represents a more realistic situation. This figure also points out the sensitivity of the D_{95} and D_{05} to the change of target DVH shape.

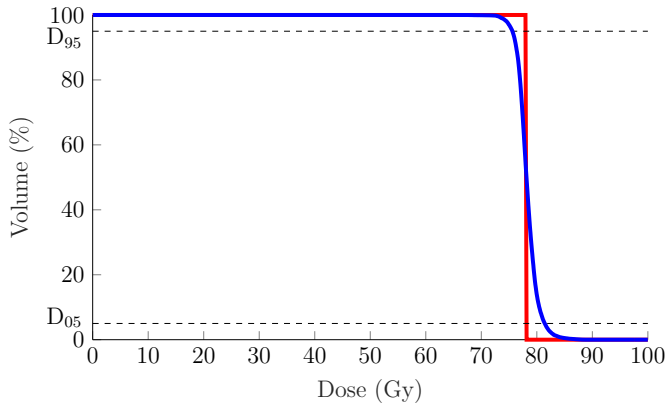


Figure 1.2: Illustration of cumulative DVHs: the red curve is an hypothetical case while the blue curve represents a more realistic situation. The broken lines indicate the D_{95} and D_{05}.

The effect of noise on the target DVH is shown in Figure 1.3: it displays DVHs when different numbers of protons are simulated. As can be seen, the histogram is blurred by the noise. Also, we observe that the D_{95} and D_{05} strongly vary, they are therefore good quantities to quantify the effects of the noise on the DVH. Since such a distorted DVH may mislead the clinician when evaluating the plan, the noise indirectly affects the patient.

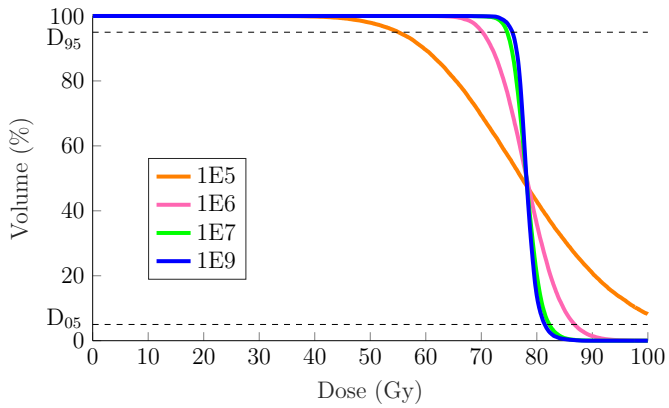


Figure 1.3: Illustration of cumulative DVHs when various numbers of particles are simulated. The broken lines indicate the D_{95} and D_{05}.

Remark that this blurring effect is negligible for critical organs DVHs, as pointed out by Keall *et al.* in [5]. Indeed it was found that the steeper the DVH, the more

significant the effect and, as the DVH associated with critical organs is a lot less steep than the target DVH, it is almost not affected by noise. This is the reason why this work focuses on target volume DVHs.

As can be seen, noise leads to the underestimation of the treatment quality. In the light of those explanations, it is clear that a dose map that is too noisy cannot be used for treatment planning assessment.

1.2 Problem statement

In the previous section, we have seen that the MC simulations can be greatly accelerated by simulating fewer particles at the cost of noise increase. One possible solution is to reduce noise from MC computed dose maps by means of post-processing denoising techniques. This solution is based on the idea that a MC dose map can be modeled as the combination of the true dose distribution and noise. Mathematically, this means that the dose in a voxel i, denoted as $d(i)$, can be written as,

$$d(i) = \mathcal{P}(s(i)), \tag{1.2}$$

where s is the dose value produced with an infinite number of simulated particles and $\mathcal{P}$ refers to the noise acting on the voxel value. As we are dealing with proton hits, the noise follows a Poisson distribution [1]. More precisely, it follows a *scaled* Poisson distribution: this notion will be further explained in the next chapter. This equation is valid for each voxel of the dose map.

The main goal of this master thesis is to obtain an estimate of the true dose map s based on the noisy one d, obtained by simulating fewer particles. This task is analogous to image restoration problems in the field of image processing. Note that denoising techniques must be used with caution as they can introduce a systematic bias into the calculation. The initial trade-off simulation time versus noise is now transformed into a trade-off between the noise and the systematic bias.

Based on Section 1.1, one might believe that reducing the noise present in a dose map is not required and that it suffices to denoise the DVH, since the latter is used by clinicians to assess the quality of a treatment plan. However it is not the only tool used to evaluate a plan; isodose lines for instance are frequently employed to represent dose distributions. Consequently we will focus on the denoising of the dose maps in order to remain general.

In the current state of the art, there is a wide agreement that a post-processing denoising step is a valuable technique (see [5, 6, 7]). Numerous techniques have already been investigated: a vast majority of these are directly inspired by image processing methods (e.g., convolution filters, wavelet denoising). However most of the used MC test cases involve electron or photon beams and few deal with proton therapy. Hence, in this work, we will review the performance of several proposed algorithms when applied to MC proton dose maps.

1.3 Plan of the master thesis

This work is organized as follows:

Chapter 1 : Introduction. This is the current chapter, it is intended to describe the context and the challenges in proton therapy. We define the studied problem in details and give the motivations and objectives of this master thesis.

Chapter 2 : Noise quantification. As the main purpose of this study is the noise reduction of MC computed dose maps, a reliable measure of the statistical uncertainty is required. This topic is addressed in the second chapter where two approaches for noise quantification are tested and discussed.

Chapter 3 : Local and wavelet filtering. This chapter is dedicated to the review of filtering techniques. In the first part, we introduce accuracy criteria that will be used to assess the performance of the methods. Then, filtering techniques such as Gaussian filter, median filter and wavelet threshold denoising are tested on MC datasets and performances are studied.

Chapter 4 : Non-local filtering. In the fourth chapter, we present a non-local filter which has already proved to be very efficient in the image processing field. Multiple improvements of the algorithm are proposed and analyzed on MC computed dose maps.

Chapter 5 : Conclusions. This final chapter provides a comparison between the different discussed filters and exposes some of the remaining issues.

In the next chapters, two distribution test cases are used. The first dose distribution corresponds to a PBS proton therapy treatment of a lung tumor, initially planned with the Eclipse TPS. The resulting dose map is a $274 \times 274 \times 162$ voxel grid. It will be referred as the lung patient. The second dose distribution is the one that has already been used earlier in this chapter: it is a patient with a tumor in the prostate treated with a PBS proton therapy treatment planned with the RayStation TPS. The dose map is a $256 \times 256 \times 71$ voxel grid. It will be referred as the prostate patient. Both dose distributions were recomputed using the fast Monte Carlo code MCsquare [2]. The time needed to obtain the dose maps for various numbers of particles is summarized in Table 1.1. Note that these timings confirm the linear relation between the number of particles and the simulation time.

	Number of simulated particles			
	10^6	10^7	10^8	10^9
lung patient	9	55	525	5213
prostate patient	4	25	250	2491

Table 1.1: Computation times with MCsquare (in seconds).

The configuration of the computer on which all the computation were performed is the following:

System: Linux
CPU: 2x Intel Xeon E5-2660 v4 (14 cores @ 2.0 GHz – 28 threads – 35 MB cache)
Memory: 128 GB

CHAPTER 2

Noise quantification

2.1 Introduction

Since the main goal of this master thesis is the reduction of the noise in MC computed dose distributions, it is fundamental to define a quantity that provides information about the amount of noise in such a dose map. This is a basic requirement to be able to compare different denoising techniques. In the existing research studies about denoising, diverse metrics are used but these are frequently not documented, meaning that it is not clear how the author quantifies the noise and, additionally, some of the chosen metrics are not well suited to describe the complete noise distribution.

Consequently the purpose of this chapter is to study various noise metrics in order to select a standard noise quantification procedure. Some papers already address this question in the case of electron and photon beams [5]; the question will be handled here in the specific case of protons. As protons have different physical properties than electrons and photons, distinct statistical uncertainty properties can be expected. The selected quantity will then be used in the rest of this work to compare the denoising techniques.

This chapter is organized as follows: in Section 2.2 we rigorously define the uncertainty we want to quantify. Two noise quantification procedures are thereafter proposed and discussed in Sections 2.3 and 2.4.

2.2 Uncertainty characterization

As mentioned in [8], in a vast amount of publications, the authors do not give a precise characterization of the measures of dispersion they are working with and, as a consequence, the reader fails to identify which measures are reported. For example, many authors express the uncertainty as a percentage without mentioning the reference value of which the percentage is taken. Also, many forget to report the volume on which the calculations are performed or which statistical measure is used, e.g., the mean, the maximum value. As this type of confusion definitely needs to be avoided, the uncertainties, to which it will be referred in the remaining parts of this chapter, are defined here.

First of all the variance will frequently be mentioned, classically denoted by σ_X^2: it refers to the variance of the dose X in a voxel and is estimated as

$$\sigma_X^2 = \sum_{i=1}^{n_s} \frac{(X_i - X_m)^2}{n_s - 1}, \tag{2.1}$$

where n_s is the sample size, X_i is the dose value in the selected voxel for simulation i and X_m is the mean dose value of the selected voxel. The sample size n_s is the number of simulations, also called the number of batches. If the variance is calculated for each voxel, we end up with a variance map, i.e., a 3D volume with, in each voxel, a value for the variance. The standard deviation, denoted by σ_X, is the square root of the variance.

Next to this, it will also be referred to the standard error of the sample mean, denoted by s_X: it is computed for each voxel as

$$s_X = \frac{\sigma_X}{\sqrt{n_s}}. \tag{2.2}$$

Clearly the standard error decreases as the number of batches n_s grows, while the standard deviation does not tend to change with increasing n_s. This is the reason why we are interested in the standard error: the more simulations are done, or equivalently the more particles are used, the smaller the uncertainty on the dose map. The standard error can also be reduced by simulating more particles per batch, hence decreasing the standard deviation σ_X. As before, if s_X is calculated for each voxel, a standard error map is obtained.

The computation of the variance with Formula 2.2 is fast but there are two drawbacks associated with it: firstly, using this approximation, the σ^2 values have their own statistical uncertainties. This means that the sample size n_s needs to be high enough in order to obtain a good estimate. In this study, 100 batches are used. Secondly, as pointed out in [9], with an arbitrary grouping of particles into batches, we ignore any correlations between incident particles. The history-by-history method, described in [10], eliminates these problems but also makes it more difficult to use parallel computing and hence indirectly leads to an increase of the simulation time. As one of the main concerns of this work is to keep the simulation time practical, the history-by-history approach will not be adopted here.

2.3 Approach 1: histogram-based measure

The first approach to quantify the noise present in a dose map is based on the distribution of the standard error s_X. After having computed the standard error map as explained in the previous section, an histogram is extracted from these data with, on the x-axis, the standard error and, on the y-axis, the number of voxels. Then, based on this histogram, the value that best represents the standard error distribution must be selected, e.g., the mean, the median, the maximum or the 95th percentile value. These different values are examined in this section.

Prior to this, recall that all the voxels do not provide relevant information from a clinical point of view. For instance, one is not interested in the uncertainty of the voxels that contain low or no dose since these regions have almost no impact on the patient. In fact a large fraction of the voxels contain low dose values: if these voxels are taken into account in the calculation, the obtained uncertainty value will be mainly based on irrelevant regions, hence leading to an irrelevant uncertainty estimation. Furthermore the uncertainty is higher in regions with low dose since there are fewer interactions. This confirms once again that low dose voxels would skew the noise quantification. Therefore it is not just recommended to remove these voxels, it is necessary. In this study voxels with values smaller than 10%

of the maximum dose are not considered. This threshold value has been selected experimentally, based on visual inspection of the dose map.

Additionally the dose delivered in the voxels located outside the patient is not important from a clinical point of view: the corresponding voxels are therefore not considered in the uncertainty computation.

Removing all the low dose voxels as well as the voxels outside the patient, the histogram displayed in Figure 2.1 is obtained for the lung patient with $n_s = 100$ and 10^6 protons per simulation. Note that this histogram is based on slightly more than 1% of the total number of voxels. Along with the histogram the following measures are shown:

1. the maximum standard error of the sample set;
2. the mean standard error of the sample set;
3. the median value, i.e., the value that separates the lower half values from the higher half;
4. the 95th percentile, i.e., 95% of the samples are below this value;

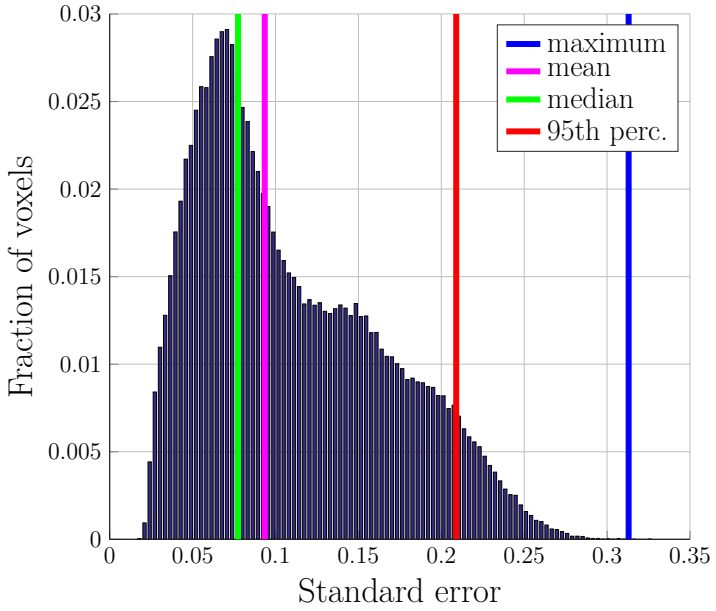


Figure 2.1: Normalized histogram of the standard error with $N = 10^6$ and $n_s = 100$.

Before analyzing the obtained results for various simulation parameters, let us make a few remarks about the above-mentioned measures. One commonly uses the mean value to describe the distribution of the standard error: however, without any information about the dispersion of the data around the mean, this measure is worthless. A similar comment can be made for the median value, in fact, the situation is even worse as it does not take into account the value of the extreme samples, meaning that we have no clue of how big the uncertainty is in the tail of the distribution. This issue is solved using the maximum value: with this measure, the magnitude of the largest standard error is known. In practice, the clinician is mainly interested in the maximum uncertainty value: suppose this uncertainty concerns an organ at risk, then this information is crucial. However, the maximum value tends to be an unstable measure, it strongly varies with the simulation outputs, and may not be adequate: this will be verified below. The 95th percentile is considered as an alternative solution to the maximum.

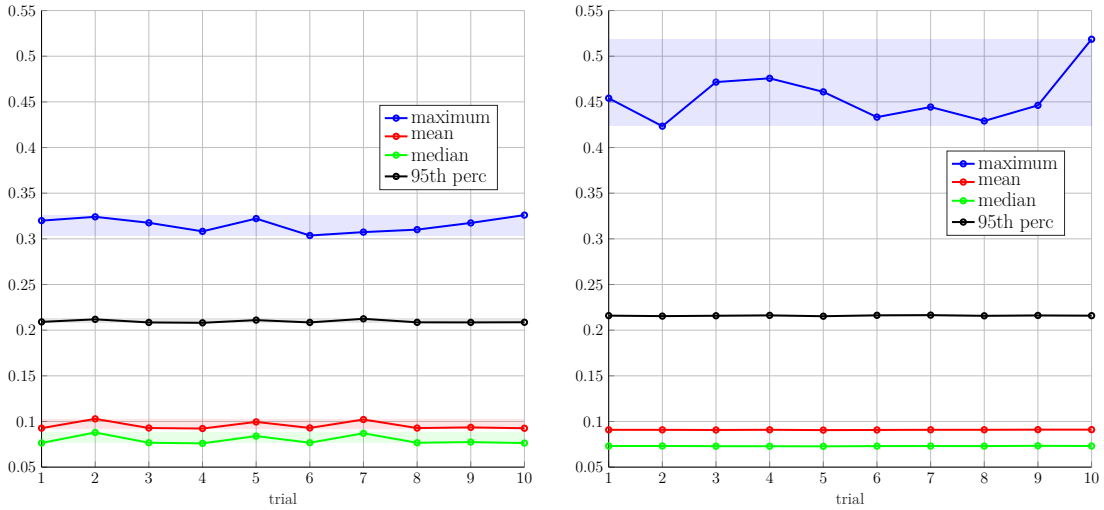
The four measures are calculated for multiple cases given in Table 2.1: the cases differ in the number of simulated protons N per simulation and/or the number of simulations n_s. Since the same total number of particles is simulated in cases 1 and

	Case 1 N = 1e+06, n_s = 100.	**Case 2** N = 1e+07, n_s = 10.	**Case 3** N = 1e+06, n_s = 200.	**Case 4** N = 1e+05, n_s = 100.
maximum	3.13	5.11	2.16	10.90
mean	0.94	0.91	0.65	3.85
median	0.77	0.73	0.54	3.61
95th perc	2.09	2.17	1.47	6.97

Table 2.1: Standard error of the sample mean ($\times 10^{-1}$) in Gray for various simulation parameters (lung patient).

2, it is expected to have a similar level of uncertainty for both situations. This is indeed approximately verified for all the measures, except for the maximum value which strongly changes. This confirms the instability of the maximum value.

Let us compare cases 1 and 3: for the latter, two times more batches are used, the uncertainty should be divided by $\sqrt{2}$ compared to the first case. This proportionality is respected for all the measures. Finally, comparing cases 1 and 4, there should be a proportionality coefficient of about $\sqrt{10}$ between the two columns, which is approximately true for all measures.



(a) $N = 10^6$, $n_s = 100$, 10 trials. **(b)** $N = 10^7$, $n_s = 10$, 10 trials.

Figure 2.2: Uncertainty levels for various simulation outputs (in Gray).

As said earlier, the standard error has its own statistical uncertainty: however, to assess the uncertainty level of a dose map, it is important to work with a measure that does not fluctuate – or barely fluctuates – with the simulations. This is not the case with the maximum value, it considerably changes depending on the simulation outputs. This is confirmed by the plots displayed in Figure 2.2: it shows the value taken by the uncertainty measures as a function of the trial (for each trial, the computation is based on a different group of simulation outputs). Clearly it is seen

on both graphs that the maximum value is unstable. Also, the fluctuation grows when the number of batches decreases, which was also awaited. From this figure it is also observed the 95th percentile is very stable. One can therefore consider to use a smaller number of batches.

Based on the above discussion we conclude that the 95th percentile is a well suited value as it gives us information on the almost worst case scenario and is stable. In the following, it will be referred to as $s_{X,95}$. Note that, since the information from a 3D volume is reduced to one unique number, our noise estimate will of course only give us a rough idea of the noise level.

2.4 Approach 2: proportionality coefficient

The second approach to quantify the uncertainty is based on the following assumption: for a Monte Carlo simulation, the relationship between the variance σ^2 and the dose D in each voxel is approximately linear, i.e.,

$$\sigma^2 = \alpha \mathrm{D}, \tag{2.3}$$

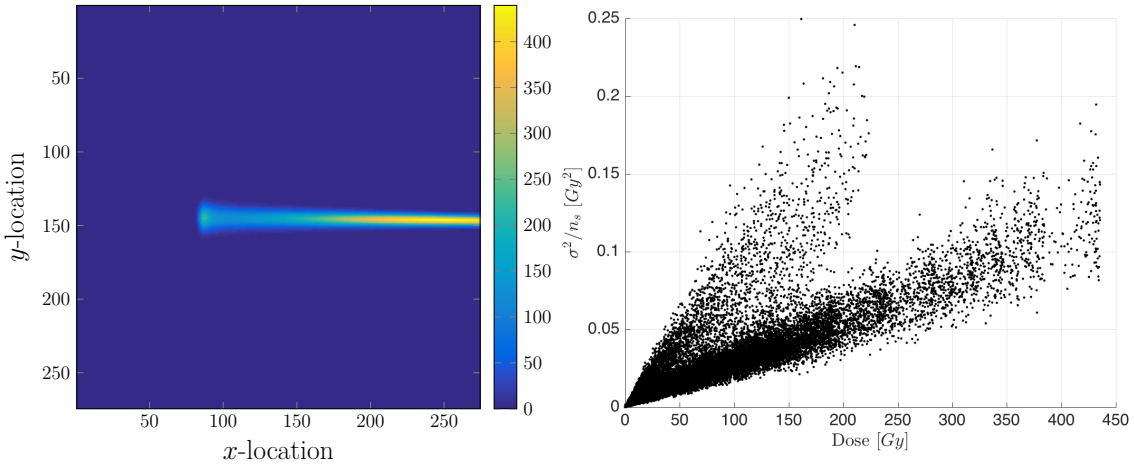
where α is the fitting parameter. Equation 2.3 amounts to say that the value in each voxel follows a scaled Poisson distribution: indeed it states that the variance is equal to the mean multiplied by a scale α. Equation 2.3 also means that the same α can be found for every voxel of the map. This hypothesis has been tested and validated in [5] for electron and photon beams. Moreover it has been found that lines of different slopes were observed for voxels of different densities.

The goal of this section is to establish whether this assumption is valid in proton therapy. If such a parameter α can be found, it could be used as an uncertainty measure afterwards. This factor could also be considered in the context of DVH deconvolution, as proposed in [4].

Assumption testing

To verify the validity of the hypothesis according to which there is a linear relationship between the variance and the dose, simulations in the simplest configuration possible are made, i.e., a beamlet is simulated in water. Beamlets can be seen as elementary functions such that, when summed according to some weights, they form the total dose distribution. Hence a beamlet is a small ray and corresponds physically to the dose delivered by one spot. Simulating one beamlet in water leads to the dose map given in Figure 2.3a; only a slice is shown here. Obviously, if the assumption is not verified for this simple case, it will also not be valid for other, more complex situations (multiple beamlets, inhomogeneous medium, etc.).

Figure 2.3b shows the squared standard error as a function of the dose. The variance σ^2 has been computed based on 100 simulations with each time 10^6 protons and the dose is collected from the mean dose map (i.e., averaged over the 100 computed dose maps). It is clear that a line is not obtained; instead of that, the data seem to be organized along two lines, giving rise to a v-shape. There is thus no linear relationship between the variance and the dose in the case of the protons. As the physics behind interactions of protons with matter is different from the one behind photons and electrons, this conclusion was to be expected from a physical standpoint.



(a) Slice 75 of the dose map.

(b) Squared standard error [Gy^2] as a function of the dose [Gy].

Figure 2.3: Beamlet in water ($N = 10^6$, $n_s = 100$).

To better grasp this point, let us briefly examine the photon physics. First of all the interaction between a photon and the matter is uncertain: the number of interactions follows a Poisson distribution of parameter λ. Moreover, after a photon interacts, it is often scattered and can be considered as *leaving* the photon beam. The mean deposited energy per interaction is approximately constant with depth. Based on this short reasoning, the mean dose in a voxel i, referred to as $\bar{\mathrm{D}}_i$, can be written as,

$$\bar{\mathrm{D}}_i = \beta\lambda_i, \tag{2.4}$$

where β represents the mean deposited dose due to an interaction, which is approximately constant in a homogeneous medium, and λ_i is mean interaction number in voxel i. Since the mean equals the variance for a Poisson process, it comes

$$\bar{\mathrm{D}}_i = \beta\sigma_i^2, \tag{2.5}$$

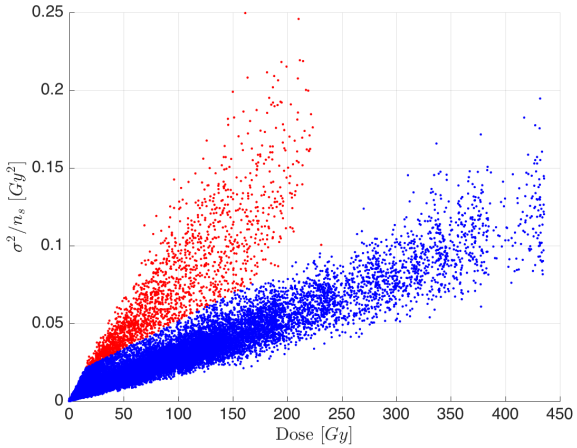
and the linear relationship between variance and dose is found. Note that this reasoning is based on a simplified view of the interactions between photons and matter and that other mechanisms should be taken into account to obtain a better estimate of the dose.

A similar reasoning cannot be used with protons in view of the two next remarks. Firstly the interaction of protons with matter is continuous and does not occur at some discrete time, as with photons. The delivered dose is not constant at all: it depends on the stopping power, which varies with the energy of the beam and with depth [1]. In other words, the delivered dose will be different every time a proton interacts with matter. Those two elements lead to a non linear relationship between dose and variance, which is observed in Figure 2.3b.

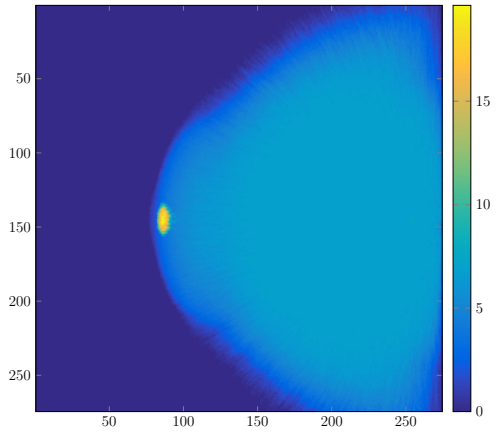
Uncertainty shape analysis

As mentioned earlier, the data seem to be organized along two lines of different slopes. This begs the following question: to which voxels in the original dose map corresponds the points along both lines?

To answer this question, a value equal to 1 is assigned to the voxels corresponding to the data points coloured in red in Figure 2.4a, while a very small value (i.e., 0.05) is assigned to the voxels corresponding to the blue points. The obtained map is plotted in Figure 2.4b: the voxels corresponding to the red (resp. blue) dots are given in yellow (resp. light blue). The zero dose voxels are given by the dark blue region. From this it appears that the red dots correspond to the voxels before the beam stops, where the Bragg peak occurs. Note that a similar pattern is obtained when the nuclear interactions are not simulated; the described phenomenon is hence not due to the nuclear interactions. The same procedure is repeated, but this time



(a) Squared standard error $[Gy^2]$ as a function of the dose $[Gy]$. Same data as in Figure 2.3b.



(b) Projected map.

Figure 2.4: Classification of the voxels for a beamlet in water ($N = 10^6$, $n_s = 100$).

with another simulation configuration: two beamlets are now simulated in water (see Figure 2.5).

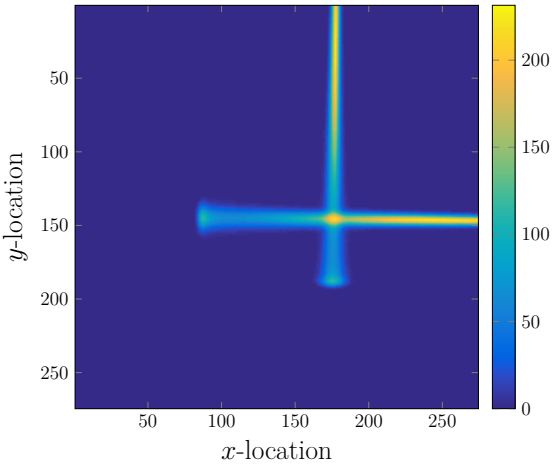
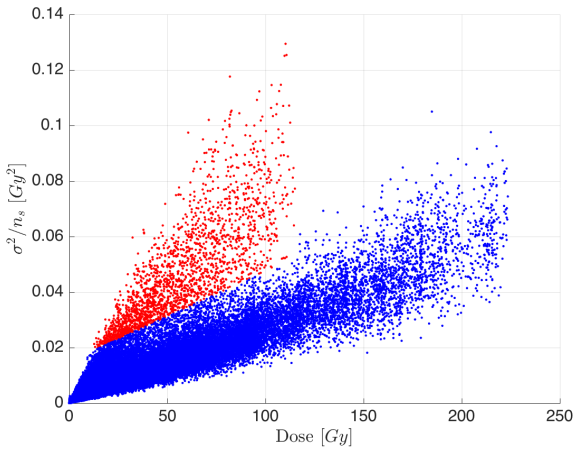


Figure 2.5: Two beamlets in water, slice 75 ($N = 10^6$, $n_s = 100$).

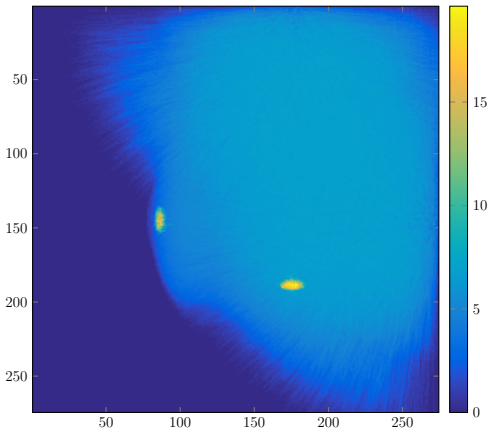
As can be seen in Figure 2.6a a similar pattern is obtained when plotting the dose versus the squared standard error, i.e., a v-shape. Classifying the voxels as done earlier, the same conclusion can be drawn: the red points correspond to the voxels where the Bragg peak occurs (see Figure 2.6b).

Based on these two experiments, the existence of two types of uncertainties is suspected in the case of protons. The first uncertainty type corresponds to low stopping power: an error in this range has small consequence for the final dose map, as low doses are delivered, hence leading to smaller uncertainties. This gives the line with the smallest slope. The second uncertainty type corresponds to high stopping power: in this case, small position differences between

two dose maps have huge impact, as the delivered dose is very large. This therefore gives the line with the largest slope.



(a) Squared standard error $[Gy^2]$ as a function of the dose $[Gy]$.



(b) Projected map.

Figure 2.6: Classification of the voxels for two beamlets in water ($N = 10^6$, $n_s = 100$).

Influence of an inhomogeneous medium

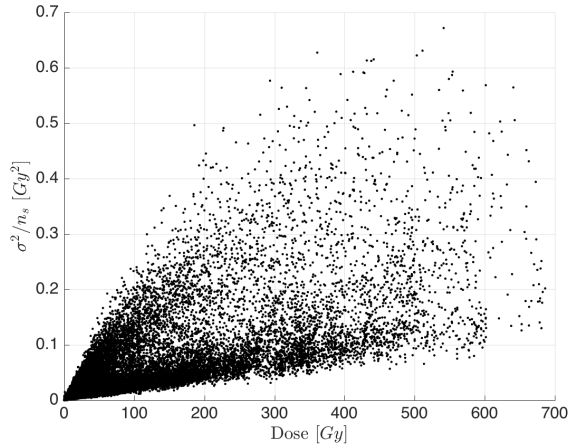
The purpose of this section is to examine the shape of the graph dose versus variance when the particles are travelling in a medium with different densities. As said earlier it has been noticed in [5] that lines of different slopes are observed for voxels of different densities in the case of photons and electrons; a similar effect is expected for the protons.

A third experiment where a single beamlet goes through the lung of patient is performed. The density map and the beamlet are displayed in Figure 2.7b. The resulting graph variance versus dose is given in Figure 2.7a. As expected, the data points are much spread. The above-mentioned v-shape can still be observed.

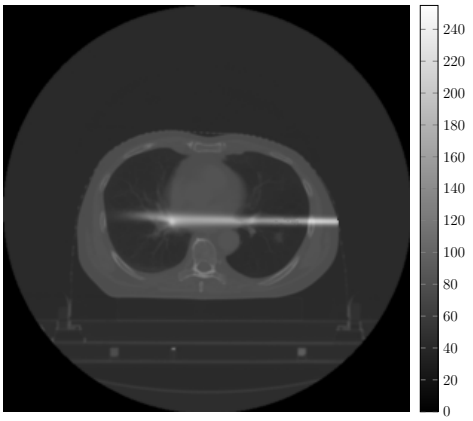
Multiple beams in an inhomogeneous medium

The most general situation is now studied, namely multiple beams are simulated into the human body. In the first place, the same density map as in the previous experiment is used; a treatment plan computed with Eclipse is simulated with MCsquare (see the axial view in Figure 2.8a). The resulting variance/dose graph is shown in Figure 2.8b. Secondly, a treatment plan computed with RayStation is simulated in the prostate patient (see Figure 2.9a). The variance/dose graph is in Figure 2.9b.

For both simulations, the two lines observed earlier are now difficult to see. Instead, a cloud of points is observed; this cloud seem to be delimited by two lines, as shown in red in Figure 2.9b. This is however very approximate.

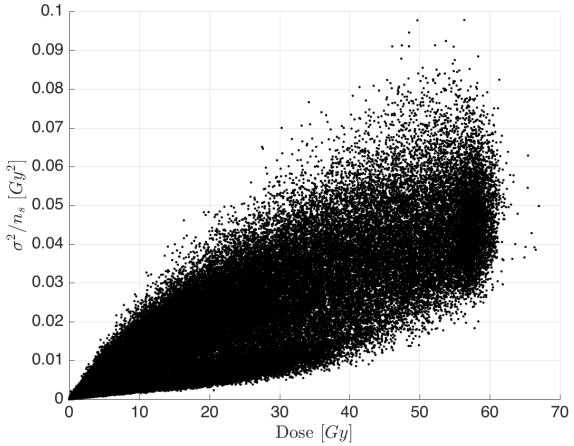


(a) Squared standard error $[Gy^2]$ as a function of the dose $[Gy]$.

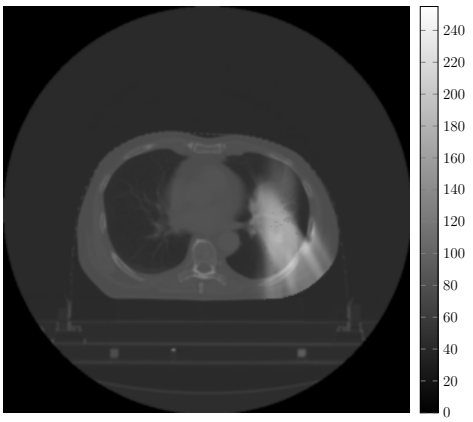


(b) Axial view (slice 75).

Figure 2.7: One beamlet in an inhomogeneous medium ($N = 10^6$, $n_s = 100$).



(a) Squared standard error $[Gy^2]$ as a function of the dose $[Gy]$.



(b) Axial view (slice 75).

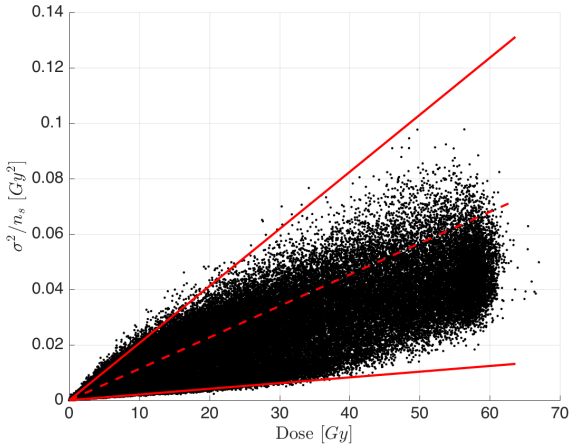
Figure 2.8: Lung plan ($N = 10^6$, $n_s = 100$).

Discussion

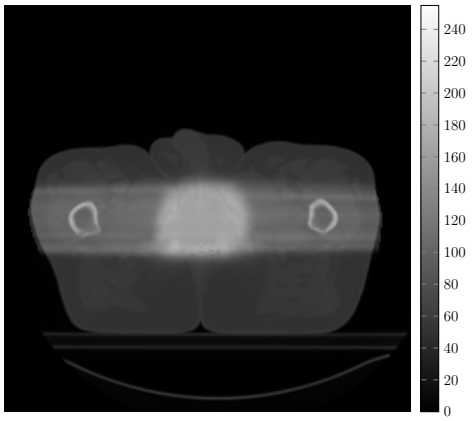
All the discussed examples point out the fact that there is no linear relationship between dose and variance in the case of protons. However, a particular pattern seems to be frequently observed, namely data points organized between two lines.

Hence, although no linear relationship is observed, this proportionality-based approach can potentially be used for noise quantification. One possibility would be to quantify the noise with the slope of the two lines that delimit the data points (for instance the full red lines in Figure 2.9a). Another approach would be to quantify the noise using the slope of a line in the middle (for instance the dotted red line in Figure 2.9a) and a measure of the dispersion along this line. Note that, with the two suggested methods, a high number of batches n_s needs to be used, otherwise, many outliers appear and the uncertainty is over-estimated.

If such methods work, it would allow the quantification of the noise using a measure representative of the entire distribution. This is an advantage compared



(a) Squared standard error $[Gy^2]$ as a function of the dose $[Gy]$.



(b) Axial view (slice 40).

Figure 2.9: Prostate plan ($N = 10^7$, $n_s = 100$).

to the previous approach, i.e., histogram-based measure. Hence, we emphasize that this method should be further investigated with other test cases.

2.5 Conclusions

Two approaches for quantifying the noise in MC computed dose maps have been tested in this chapter. Both approaches rely on the batch method to estimate the standard error in each voxel of the considered dose map. The challenge is to find the measure that best summarises the error in all voxels.

Regarding the histogram-based method, it has been concluded that the 95th percentile of the standard error is the most suited measure as it gives relevant information about the whole distribution, and more specifically about its tail.

The second approach was not conclusive: it was indeed found that the relation between variance and dose is not linear for proton beams. Consequently no proportionality coefficient can be used to quantify the noise. However a particular pattern seem to be frequently observed: the data points are spread between two lines. We make the hypothesis that these two lines are linked to different types of uncertainties (corresponding to high and low stopping power). Hence this method could potentially be used for noise quantification. It should be further investigated with other test cases.

As a final conclusion, the 95th percentile of the standard error will be used as standard measure to quantify the noise in the remaining of this work.

CHAPTER 3

Local and wavelet filtering

3.1 Introduction

In this chapter, we propose a solution to the problem described in the introduction, namely the reduction of the noise present in MC computed dose maps. Our solution relies on filtering methods: a filter is a post-processing step that is used to remove unwanted components. Of course, by removing image components, we also risk to remove useful information and hence to introduce bias in the data. The trade-off noise versus simulation time discussed earlier is now transformed into a trade-off noise versus bias.

More specifically, two local filters are studied in this chapter, i.e., the Gaussian and median filters. For both filters, the denoising process of a pixel is performed using its neighborhood, hence their categorization as local. A third filter, called the wavelet threshold denoising, is also studied; this technique cannot be categorized as a local filter, it belongs to the Wiener-like methods, which proceed by a threshold on space-frequency coefficients. All the mentioned methods are inspired by image processing methods.

The plan of this chapter is the following. Before describing the methods in details, we introduce in Section 3.2 the accuracy criteria used to evaluate the performances of the tested techniques. These criteria will also be used in the next chapter. Methods are then explained and tested in Sections 3.3 and 3.4. The goal here is not to examine all aspects of MC denoising, but rather to inspect the quality of the filtered dose maps in terms of specific denoising metrics. Discussion and conclusions can be found in Sections 3.5 and 3.6.

3.2 Accuracy criteria

To assess the performance of the studied filters, one needs to use multiple denoising metrics. These are described in this section, along with their purposes. Remark that, as we possess benchmark dose maps, which are simply obtained by simulating a high number of particles, full-reference metrics can be used. Direct comparison with the reference dose map makes the image quality assessment very reliable. Most of the following accuracy criteria have been introduced by Kawrakow in [6] and have become standards in this domain.

1. **Visual inspection** – Examining the filtered dose map with the human eye directly tells us if the used algorithm yields poor results, e.g., information loss, over-smoothing or presence of artifacts. Isodose lines or dose profiles may also

be inspected. Additionally, looking at the difference between the filtered and noisy dose maps (i.e., the data removed during the denoising process) may be interesting: ideally this substraction should yield an image of the noise only, it should not present any structure.

Visual inspection is, however, subjective and inaccurate, it is thus not sufficient to judge the performance of a filter.

2. **Mean squared error (MSE) comparison** – The mean squared error is the square of the Euclidean distance between the benchmark dose map x_r and the filtered dose map x_f, i.e.,

$$\mathrm{MSE}(x_r, x_f) = \frac{1}{N}\sum_{i=1}^{N}(x_{r,i} - x_{f,i})^2, \tag{3.1}$$

where the index i denotes the ith pixel and N is the number of pixels in the dose map. A smaller MSE indicates that the estimate is closer to the reference dose map. This numerical quality measurement is objective, since it does not rely on any visual interpretation, and is simple and inexpensive to compute, which makes it attractive.

With the MSE, it is assumed that all the pixels of an image are equally important. This assumption is false in our case since we have little interest in low or no dose samples. To address this issue, only the pixels whose dose value are above 10% of the maximum dose in the benchmark image will be taken into account for the MSE computation. Also, the MSE considers constant variance for all doses and, as shown in the previous chapter, this assumption is not verified here. Finally, as reported in [11], a small MSE does not assure a high visual quality. Consequently, one should not rely on the MSE only. It reflects however well the relative performances of the algorithms.

3. **Edge preservation index (EPI) comparison** – This is used to control whether the edges of the structures are preserved after the filtering process [12]. Let y_r and y_f be the edge maps of the benchmark and filtered dose maps respectively; such an edge map is obtained by using for instance a Laplacian filter. The edge preservation index is then computed as

$$\mathrm{EPI}(x_r, x_f) = \frac{\sum_{i=1}^{N}(y_{r,i} - \bar{y}_{r,i})(y_{f,i} - \bar{y}_{f,i})}{\sum_{i=1}^{N}(y_{r,i} - \bar{y}_{r,i})^2 \sum_{i=1}^{N}(y_{f,i} - \bar{y}_{f,i})^2}, \tag{3.2}$$

where $\bar{s}$ stands for the mean value of s. This equation is in fact the correlation between the signals y_r and y_f, it measures how similar two signals are. This index varies from 0 to 1: the closer to 1, the better the edges are preserved.

Edge preservation is important since biased knowledge about edges might impact the surrounding healthy tissues, which must be avoided as far as possible.

4. **Dose volume histograms (DVHs) comparison** – As explained in the first chapter, the DVH is one of the tools used by the clinician to evaluate a treatment plan. It is then only natural to assess the quality of a filter based on this metric.

More specifically, the L_2-norm of the difference between the reference and filtered DVHs, denoted by $\|\Delta\text{DVH}\|_2$, will be analysed. Also, we will compare the extreme values of the DVH, namely the D_{95} and the D_{05}. Doing so, information about the tails of the curve is taken into account.

5. **Energy conservation** – If a filter removes useful information, instead of only reducing the noise, then the data are biased. To verify that a filter does not yield biased dose maps, one possibility is to control that the total energy is conserved after filtering. Indeed, physically, the total dose must not be conserved, but the energy must. The energy map is easily computed based on the dose map: the latter must simply be multiplied by the density map. Then, we study the ratio between the total energy extracted from the filtered dose map and the total energy from the reference one. Ideally this ratio must be equal to 1.

6. **95th percentile uncertainty** – We will also use this measure, defined in Chapter 2, in order to quantify the standard error of the sample mean. Computation will be done based on 10 different dose maps. Recall that we refer to this measure by $s_{X,95}$.

3.3 Methods

The studied denoising methods are presented in this section. The two first methods, namely Gaussian and median filters, are basic filtering techniques that are widely used in image processing [13]. Their two- and three-dimensional versions are described. Afterwards we consider a more advanced technique, called the wavelet threshold denoising, investigated by Deasy *et al.* in the radiation therapy context in [7]. Finally we describe a variance-stabilizing transform which is used to deal with the Poisson nature of the noise.

3.3.1 Gaussian filtering

The Gaussian filter is a linear smoothing operator that can be used to reduce the noise of an image. The idea behind it is to replace every pixel of the dose map with a weighted average of its neighborhood.

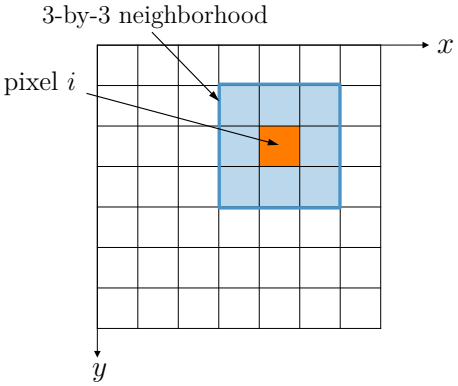


Figure 3.1: Pixel grid with pixel i and its 3-by-3 neighborhood.

For instance, in Figure 3.1, the 3-by-3 neighborhood of pixel i is represented in blue; with the Gaussian filter, the value of i is approximated by a weighted sum of the values in the blue window. The size of this neighborhood can be modified, this is the first parameter of the Gaussian filter: the larger the size, the more blurred the filtered image.

The weight associated with each neighbor is computed based on the Gaussian distribution, which has the following form in two dimensions:

$$G(x,y) = \frac{1}{2\pi\sigma^2} \exp\left(- \frac{x^2+y^2}{2\sigma^2} \right), \qquad (3.3)$$

where σ is the standard deviation of the distribution. This is the second parameter of the filter, it controls the spread of the distribution and hence the degree of smoothing of the filter: the larger σ, the more smoothed the image. The weights of the filter are computed by producing a discrete approximation of this distribution over the neighborhood. Given the shape of the distribution, it is clear that the weight corresponding to the central pixel (i.e., here pixel i) is the highest, relative to its neighborhood.

In fact, the Gaussian filter amounts to a convolution of the image with a Gaussian kernel.

The extension to a three-dimensional filter is direct: it suffices to take a 3D window and to compute the weights based on the 3D Gaussian distribution.

3.3.2 Median filtering

The median filter is another basic filter often used in image processing. Unlike Gaussian filtering, it is a nonlinear technique. It works similarly as the Gaussian filter, the only difference is that each pixel is replaced with the median value of its neighborhood, instead of a weighted average of the neighborhood. Recall that the median of a set is the value that separates the higher half of the set from the lower half. The only parameter of this filter is the size of the neighborhood. Again the extension to 3D is direct.

Under certain conditions, the median filter is known to better preserve the edges of the structures in the image than the Gaussian filter [14]. This claim will be verified in the next section.

3.3.3 Wavelet threshold denoising

Before detailing the wavelet threshold denoising (WTD) method itself, let us briefly explain what wavelets are and why they are useful in this context.

One must know that wavelets are basis functions, satisfying some mathematical properties, used to represent functions or data [15]. In the present work, wavelets are used to represent the dose distributions: for instance, for a $N \times N$ dose map, the value of the dose D in pixel $(i,\ j)$ is approximated by means of wavelets $\varphi_{m,n}$, i.e.,

$$D(i,j) = \sum_{m=1}^{N} \sum_{n=1}^{N} c_{mn} \varphi_{mn}(i,j), \tag{3.4}$$

where c_{mn} are the wavelet coefficients. This can be seen as the transformation of the dose distribution into the wavelet space: this first step is called the discrete wavelet transform (DWT). There exists various wavelet families, e.g., Haar, Legendre or CDF. In the following the 9,7-biorthogonal symmetric wavelets are used (shown in Figure 3.2): with this wavelet family, the discrete wavelet transform can be efficiently computed using the so-called lifting scheme [16].

The benefits of wavelets for denoising can now be explained: it appears that the wavelet transform concentrates the image features in a few large-amplitude wavelet coefficients, while the remaining small-amplitude wavelet coefficients carry mostly

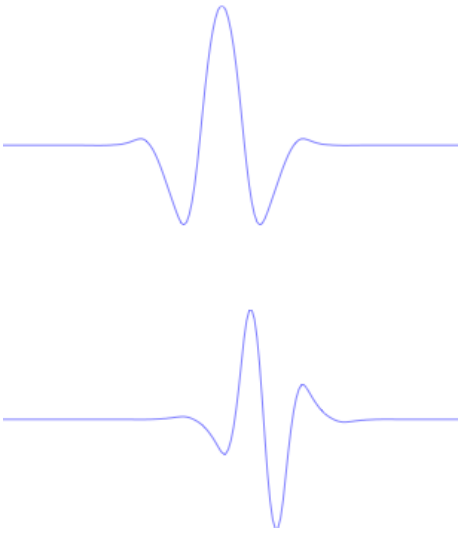


Figure 3.2: Examples of 9,7-biorthogonal symmetric wavelets.

noise. We say that the transform has a localizing property. Hence, noise can be reduced by shrinking or even removing the small-amplitude coefficients, without affecting the quality of the main image features. We will refer to this second step as coefficients thresholding.

Regarding the thresholding, we must define the threshold function and value. These will greatly impact the denoising process, especially the threshold value t: too small a value will result in poor denoising performances while too high a value might bias the data. Here we use hard thresholding [17], which is described by the following function:

$$T_h(x) = \begin{cases} x & \text{if } |x| \geq t, \\ 0 & \text{if } |x| < t. \end{cases} \tag{3.5}$$

The threshold value t is chosen as a percentage θ of the wavelet coefficient maximum $c_{\max}$, e.g., all coefficients under $\theta = 10\%$ of $c_{\max}$ are thresholded. Various percentage values will be tested during the experimentation part.

Also, we add without delving into the details that a multiresolution wavelet transform is done, meaning that the computed wavelet coefficients are associated with different scales of the image. In short and simplified, some wavelet coefficients correspond to the large scale features of the image while other correspond to the small scale features. As the largest scale coefficients contain the overall shape of the dose distribution (and hence no noise), it is kept untouched during the thresholding step [18].

After the thresholding step, the dose distribution can be reconstructed by simply performing an inverse discrete wavelet transform on the thresholded coefficients. This is the last step of the denoising process.

The complete procedure is summarized in three steps:

1. Computation of the discrete wavelet transform of the dose map.
2. Wavelet coefficients thresholding.
3. Computation of the inverse discrete wavelet transform to reconstruct the dose map.

Additionally, note that orthogonal wavelet transforms are not translation invariant; this means that the processing of an image and a translated version of the image give different results. To remedy this issue, *spin-cycling* is performed, i.e., we use the denoising technique on the original image and on the eight images shifted by one grid point and average the results. It has been shown that the use of spin-cycling reduces small artifacts introduced by the wavelet transform [7]. The extension of this method to three dimensions is not studied here. When the WTD is applied to a 3D dose map, it is done on a slice-by-slice basis.

3.3.4 Variance-stabilizing transform

As already mentioned in Chapter 2, dose distributions obtained with Monte Carlo simulation are corrupted by scaled Poisson noise with varying scale. Instead of dealing with the denoising problem by taking the noise statistics directly into account, it can also be approached indirectly through variance stabilization. This technique is explained in this section.

Let X be a random variable following a Poisson distribution. Hence the mean of X, denoted by $\mathbb{E}[X]$, is equal to its variance. This fact complicates the denoising process as there is a correlation between the variance of the noise and image data. To address this problem, the idea is to apply a transform A such that the variance of $A(X)$ becomes approximately constant. This transformation was proposed by Anscombe [19],

$$A : X \longrightarrow 2\sqrt{X}. \tag{3.6}$$

Under this transform, $A(X)$ is approximately a normal random variable with mean $2\sqrt{\mathbb{E}[X]}$ and variance 1. In a nutshell, we can say that this transform *gaussianizes* the noise. This is useful for denoising as an image with approximately Gaussian noise with known variance is easier to filter [20]; indeed, there is in principle no correlation between the variance of the noise and the image data anymore.

After the application of this transform, the denoising methods discussed above can be used. The data are then reconstructed with the inverse Anscombe transform. Hence, in the experimentation part, the next three-steps procedure is followed:

1. Application of the Anscombe transform to the noisy dose map.
2. Filtering using the methods described in the previous sections.
3. Application the inverse Anscombe transform to the filtered dose map.

3.4 Results

To assess the performances of the above described filters, wet set up a denoising experiment based on two test cases given in Chapter 1. Only the results obtained for the lung patient are shown here as both test cases lead to similar conclusions. In the first part, we study the two-dimensional version of the methods: each slice of the dose volume is treated independently for the other. Then, in the second part, the three-dimensional Gaussian and median methods are tested. Better performances are expected with 3D filters, as more information about the neighborhood is used. The

variance-stabilizing transform explained in Section 3.3.4 is applied in combination with each filter.

For the Gaussian and median filters, built-in MATLAB functions were used. We implemented the WTD method, using codes from [21] for the forward and backward discrete wavelet transforms.

3.4.1 Two-dimensional study

Figure 3.3 shows the isodose contours obtained with various filters. It is seen that the three algorithms produce smooth isodose lines, despite all the noise present in the original noisy dose map. This is particularly true for the WTD method, the contours even seem slightly over-smoothed. Noise is still present with the Gaussian and median methods; small speckles can be seen in the bottom right corner. Additionally, some edges are not recovered with these two methods, e.g., the transition between the large spot in the center and the right beam.

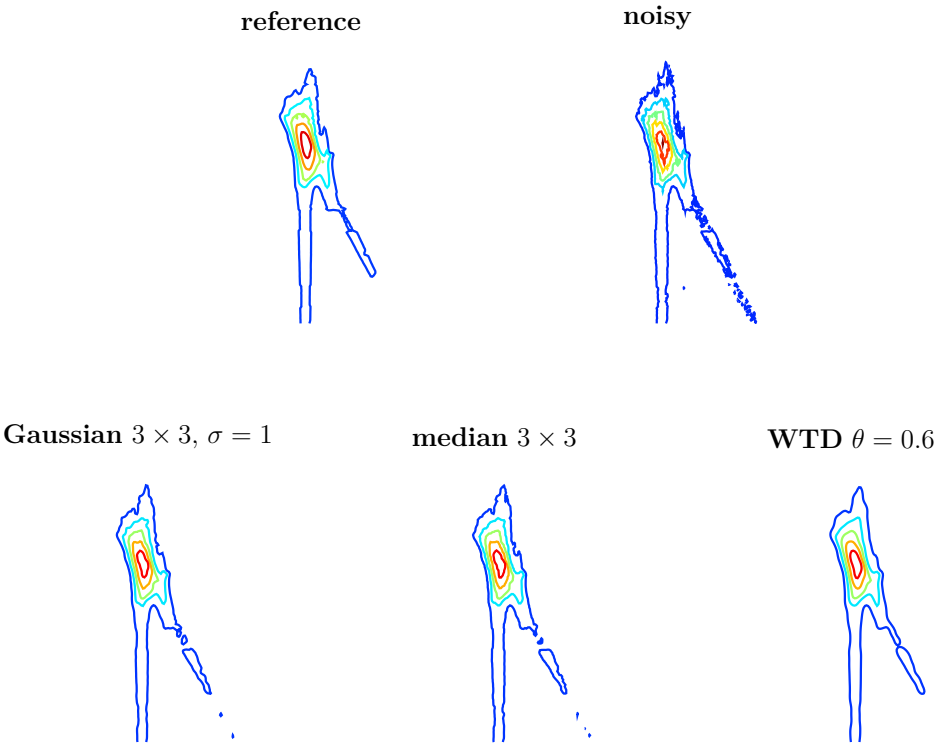


Figure 3.3: Isodose lines for various 2D filters (lung patient, slice 60).

For the WTD method, Figure 3.4 gives the value of accuracy measures with respect to the threshold, and more specifically with respect to θ, defined as the percentage of the wavelet coefficient maximum. We recall that, in the current implementation, the high-scale wavelet coefficients are not thresholded during the process. Hence, it makes sense to let θ vary from 0 to 1. Two observations are done based on this figure. Firstly, the optimal θ is approximately equal to 0.6 based on the mean squared error. Secondly, using a θ-value larger than 0.6 does not have a huge impact on the measures: the MSE slightly increases while the other measures are almost constant. Selecting the exact optimal value of θ is therefore not crucial here.

Table 3.1 gathers the accuracy values attained with the filters for various parameters. Note that the energy conservation ratio, mentioned in Section 3.2, is not given in the table as the ratio was found to be nearly equal to 1 for each test, hence indicating no bias.

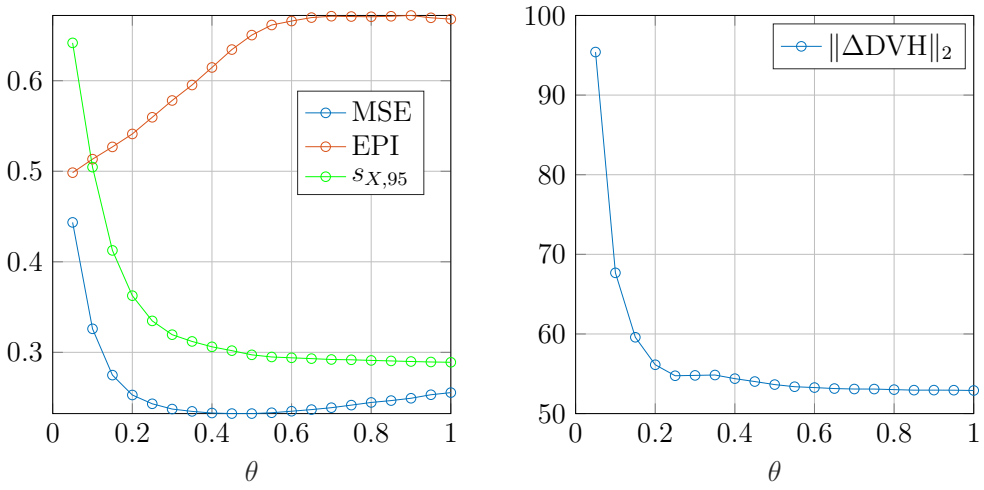


Figure 3.4: Wavelet threshold denoising – Resulting accuracy criteria values for different thresholds, computed as a percentage θ of the wavelet coefficient maximum.

Table 3.1 shows the results when the two parameters of the Gaussian filter, namely the filter size and the standard deviation σ, varies. The best scores are

	MSE	EPI	$\|\Delta\text{DVH}\|_2$	$s_{X,95}$
noisy	0.56	0.48	138	0.73
Gaussian filter (2D)				
3×3, $\sigma = 0.5$	0.29	0.61	101	0.53
3×3, $\sigma = 1$	0.20	0.74	94	0.38
3×3, $\sigma = 2$	0.20	0.70	98	0.37
5×5, $\sigma = 1$	0.22	0.76	113	0.33
Median filter (2D)				
3×3	0.26	0.56	94	0.46
5×5	0.37	0.50	134	0.38
WTD (2D)				
$\theta = 0.1$	0.33	0.51	68	0.50
$\theta = 0.6$	0.23	0.67	53	0.29
$\theta = 1$	0.26	0.67	53	0.29

Table 3.1: Comparative values for different two-dimensional filters with varying parameters ($274 \times 274 \times 162$ dose map, lung patient).

obtained for a 3×3 filter with $\sigma = 1$. For $\sigma < 1$, the outcome is noisy while for $\sigma > 1$, it is over-smoothed, as indicated by the EPI. A 5×5 filter size yield good MSE, EPI and uncertainty values but a high error in the DVH. Median filtering strikingly underperforms compared to the other methods. The WTD method yields the smallest error in DVH and 95th percentile uncertainty; resulting MSE and EPI are however slightly inferior than with Gaussian filtering. Remark that the experiments have been performed both with and without using the Anscombe transform:

no significant improvement is observed by using this transform, except for the MSE. However, as all the measures are slightly improved, we continue to use it.

Figure 3.5 displays the DVHs extracted from the reference, noisy and filtered dose maps. We naturally want the difference between the reference and filtered DVHs to be minimal. As can be seen, the WTD method yields the best performance. The deviation of the D_{95} and D_{05} from the reference value is the smallest. The Gaussian and median curves are almost exactly on top of each other.

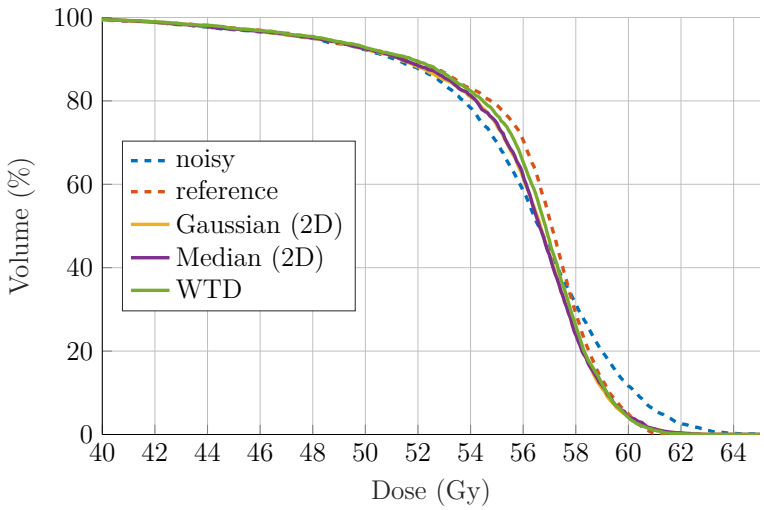


Figure 3.5: DVH corresponding to lung patient with noisy, reference, Gaussian (3×3, $\sigma = 1$), median (3×3) and WTD ($\theta = 0.6$).

Regarding the computation time, the Gaussian and median algorithms generate the filtered dose map instantaneously (less than 1s). The DWT method takes slightly more time (approximately 2.5s).

3.4.2 Three-dimensional study

The 3D Gaussian and median filters are investigated in this section. Table 3.2 gives the resulting accuracy measures for various filter parameters. Corresponding isodose lines are shown in Figure 3.6. For both methods, it is seen that the results are not strikingly better than in the two-dimensional case. In general smaller MSE, EPI and $s_{X,95}$ values are observed.

	MSE	EPI	$\|\Delta\text{DVH}\|_2$	$s_{X,95}$
Gaussian filter (3D)				
$3 \times 3 \times 3$, $\sigma = 0.4$	0.36	0.55	112	0.59
$3 \times 3 \times 3$, $\sigma = 0.5$	0.20	0.68	91	0.42
$3 \times 3 \times 3$, $\sigma = 1$	0.15	0.78	110	0.23
Median filter (3D)				
$3 \times 3 \times 3$	0.20	0.64	114	0.30
$5 \times 5 \times 5$	0.47	0.63	196	0.19

Table 3.2: Comparative values for different three-dimensional filters with varying parameters ($274 \times 274 \times 162$ dose map, lung patient).

A smaller window size must be taken, it leads otherwise to an over-smoothed dose map where some edges totally disappear; see for instance the isodose contours obtained for median filter with a $5 \times 5 \times 5$ filter in Figure 3.6. Remark as well that the edge linking the central dose spot and the right beam is not retrieved, with neither Gaussian nor median filter. Only the two-dimensional WTD succeeds in this task.

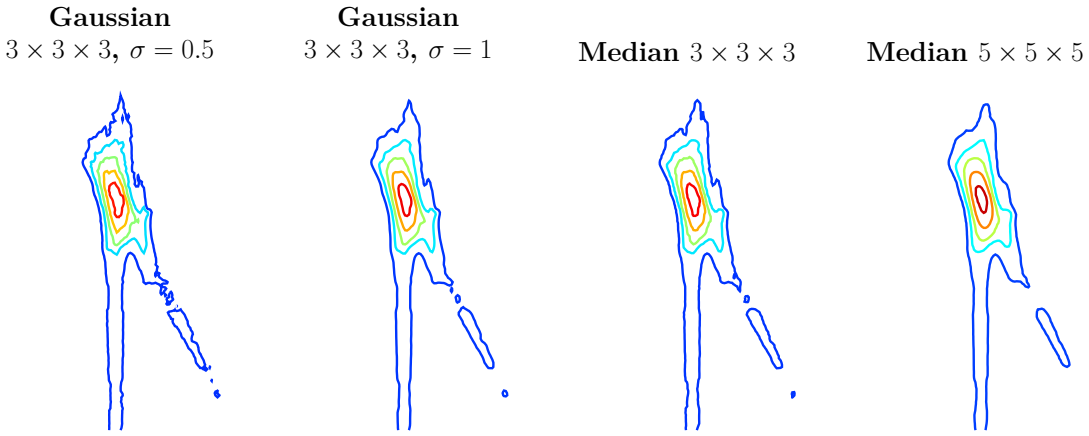


Figure 3.6: Isodose lines for various 3D filters (lung patient, slice 60).

Figure 3.7 shows the DVHs extracted from the Gaussian and median filtered dose maps. The curve extracted from the WTD map is also shown for comparison; note that the WTD is applied on a slice-by-slice basis. We see that the latter performs better than both 3-dimensional methods.

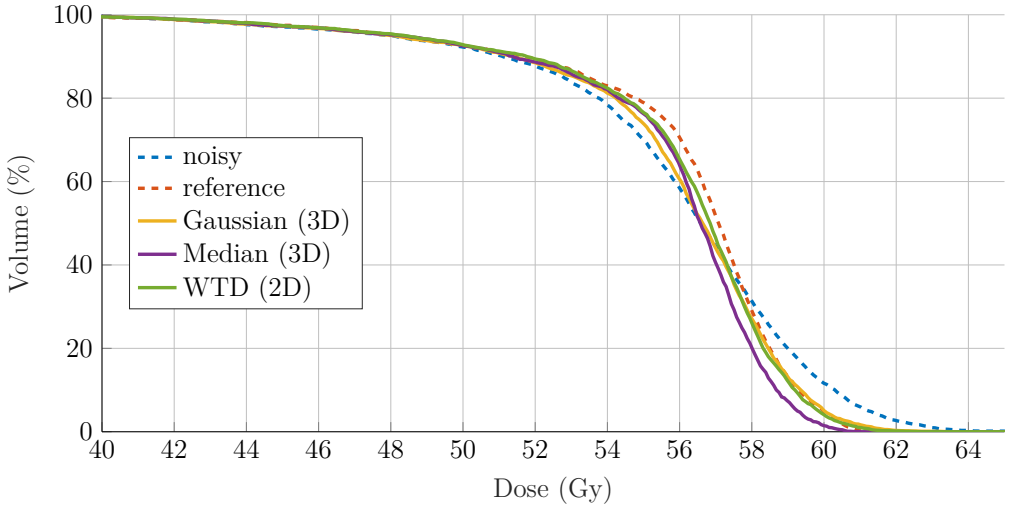


Figure 3.7: DVH corresponding to lung patient with noisy, reference, Gaussian ($3 \times 3 \times 3$, $\sigma = 0.5$), median ($3 \times 3 \times 3$) and WTD ($\theta = 0.6$).

The 3D Gaussian filter runs in less than 2s while the 3D median filter runs in approximately 23s (both for a $3 \times 3 \times 3$ filter size).

3.5 Discussion

Relative algorithm performances strongly depend on the ranking metrics. For instance, regarding the two-dimensional methods, the Gaussian filter yield the best scores based on the mean-squared error, the edge preserving index and the 95th percentile uncertainty. However, if we focus on visual inspection of the isodose contours and on the DVH curve, the wavelet threshold denoising method appears to be superior. It is thus difficult to declare a *winner* in absence of a more definite preference for one metric over another. Still, since the WTD algorithm also shows good scores for MSE, EPI and $s_{X,95}$ – these are slightly less good than the metrics obtained with the Gaussian filter, as seen earlier – we can say that it outperforms the other filters in general.

The threshold used for wavelet denoising has been selected in a rather heuristic manner. Other more sophisticated methods exist, such as SUREShrink, VisuShrink and BayeShrink [22]: they aim the minimization of a specific criterion. Such methods could be used to further improve the wavelet denoising. However, as described in the results, selecting the optimal or a near-optimal value has no huge impact on the accuracy metrics.

Computation time for 2D methods is negligible (especially compared to the Monte Carlo simulation time).

Finally, scores obtained with 3-dimensional methods are surprisingly not far better than with the 2-dimensional methods. The Gaussian filter outperforms once again the median filter based on every denoising metrics. Additionally, the Gaussian filter runs instantaneously while the median filter takes about 23s to produce an output.

The 2D WTD still surpasses the 3D Gaussian methods regarding isodose lines smoothness and shape and error on DVH. This supports the implementation of the 3D wavelet filter in order to further improve the performances of the algorithm.

3.6 Conclusions

The first – basic but important – conclusion is that filtering techniques indeed reduce the noise present in dose maps. The time needed to achieve this task is undoubtedly smaller than the time required by the MC simulation with higher number of particles. Filtering is thus a useful post-processing step which can be used to improve isodose line smoothness and dosimetric accuracy, at almost no cost. Hence MC denoising is a desirable component of the proton therapy treatment planning.

Relative algorithm performances strongly depend on the ranking metrics. All these metrics are necessary to evaluate the performance of denoising methods. If all the accuracy criteria are taken into account, the wavelet threshold denoising method yields the best results overall. As expected, three-dimensional methods yield (slightly) better results. This encourages us to test other 3D methods.

CHAPTER 4

Non-local filtering

4.1 Introduction

As the study of filtering techniques in the previous chapter was conclusive, we continue in the same direction in the present chapter. Our attention turns to a potentially more efficient filter type, namely non-local filtering. In particular, the use of the so-called non-local means filter is investigated for MC computed dose distributions. This method, as well as the proposed improvements, are described in Section 4.2. Results are reported and discussed in Section 4.3. A final comparison between this filter and the ones studied on the previous part is made in the last chapter.

4.2 Methods

4.2.1 Non-local means denoising

The non-local means (NL-means) filter has first been introduced by Buades *et al.* in 2005 [23]. It relies on a simple principle: replacing a pixel value with a weighted average of similar pixels. Two pixels are said to be similar if both pixels, along with their neighborhood, have a comparable dose level. As there is no reason to assume that similar pixels will be located close to each other, a search for the closest pixel values must be done across a vast portion of the pixel grid. Hence the NL-means filter tries to take advantage of the high degree of redundancy of a whole image.

This technique could be highly powerful in the present context since there exist a lot of analogous patterns in the dose maps. Think for example of a beam traveling through water, the resulting dose map shown in Figure 4.1 presents plenty of similar patterns, which are not exclusively located near each other.

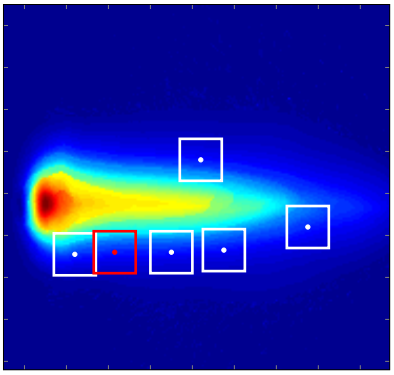


Figure 4.1: Example of similar patterns in a dose map (corresponding to one beam in water). The red box is the environment of a selected pixel i and the white boxes are the detected patches similar to i.

Algorithm

Let us now describe the method in details: given a discrete noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL[v](i)$, for a pixel i, is computed as a weighted average of all the pixels in the image,

$$NL[v](i) = \sum_{j \in I} w(i,j)v(j), \tag{4.1}$$

where the weights $w(i,j)$ depend on the similarity between the pixels i and j and are such that $0 \leq w(i,j) \leq 1$ and $\sum_j w(i,j) = 1$.

The similarity between pixels i and j is computed based, not only on the value in a single point, but on the whole neighborhood of i and j. Let $\mathcal{N}_k$ denotes the neighborhood (also called the similarity window) of pixel k: it is a square window of fixed size centered at k. Its half width is denoted by f, its full width is thus $2f+1$. The similarity between i and j depends then on the similarity between $v(\mathcal{N}_i)$ and $v(\mathcal{N}_j)$. Remark that comparing the value of a whole neighborhood instead of a single point makes the method more robust.

The similarity is measured using the weighted Euclidean distance, $\| v(\mathcal{N}_i) - v(\mathcal{N}_j) \|^2_{2,a}$, where $a > 0$ is the standard deviation of the Gaussian kernel. The weighted distance is needed as every pixel of the similarity window has not the same importance: the central pixel for example will have the highest weighting in the similarity measure relative to its neighbors.

The weights of Equation 4.1 are then defined as,

$$w(i,j) = \frac{1}{C(i)} \exp\Big(- \frac{\| v(\mathcal{N}_i) - v(\mathcal{N}_j) \|^2_{2,a}}{h^2} \Big), \tag{4.2}$$

where $C(i)$ is a normalizing constant. Pixels that are similar will consequently have larger weights in the average. The filtering parameter h controls the decay of the exponential and hence the decay of $w(i,j)$.

It follows from the above description that, for each pixel $i \in I$, we must compute its similarity with every other pixel $j \in I$. Suppose that we are dealing with a d-dimensional dose map, where $d = 2$ or 3 in this work, of size N^d. Then, this means that $N^d \times N^d \times (2f+1)^d$ operations are required for the weight computation. As this calculation is computationally intensive, the research region for similar pixels is restricted to a search window of size $(2s+1)^d$, which is much smaller than the whole image. The overall complexity of the algorithm is then $N^d \times (2s+1)^d \times (2f+1)^d$.

The different elements of the method are summarized in Figure 4.2.

Remarks and improvements

As reported in [23], the NL-means algorithm yields very good results. However the method is slow: even when the search area is limited to a 5×5 window, the method takes more than 30s to denoise a 512×512 image with a 3×3 similarity window. This is problematic in our situation as the goal is the denoising of a 3D volume within a limited time. Hence we obviously need to find a way to speed up the method.

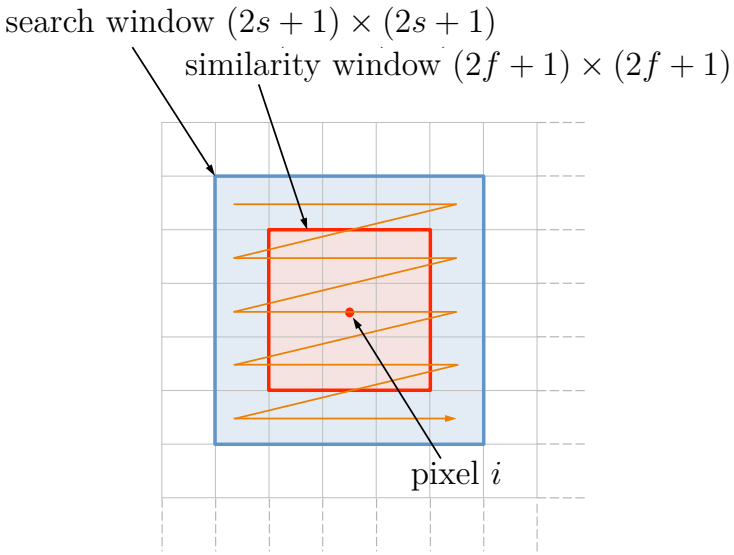


Figure 4.2: Schema of the NL-means method ($s = 2$, $f = 1$). The orange arrow gives the movement of the similarity window across the search window.

As a second comment, observe that the use of a search window in the algorithm amounts to searching similar pixels in a larger neighborhood, but still locally. This means that, despite its name, the non-local means algorithm is local. To benefit from the full power of the algorithm a search through the whole image should be performed.

We therefore face two challenges: make a search across a larger region while reducing the computation time of the process. Our solution is based on two ingredients: firstly we take advantage of the data we are using in order to reduce the number of pixels to filter. Secondly we use a data structure suited for nearest neighbor search in high dimensional spaces, namely the vantage point tree (vp-tree). Before further detailing the proposed algorithm, let us briefly examine the vp-trees and the motivation to use these.

4.2.2 Vantage point trees

Before considering the vantage point tree, let us look into the Nearest Neighbor (NN) problem. The latter can be formulated in the following way. Let $\mathcal{S}$ be a d-dimensional metric space and $\mathcal{S}_D \subset \mathcal{S}$ be a finite subset that represents the database against which NN queries are made. For a query $q \in \mathcal{S}$, the NN problem consists in finding a single minimally distant member of $\mathcal{S}_D$. This may be written as $\mathrm{NN}(q, \mathcal{S}_D)$ [24].

In the present context, $\mathcal{S}$ is the Euclidean space $\mathbb{R}^{(2f+1)^d}$, where $d = 2$ or 3, as will be shown in Section 4.2.3. The subset $\mathcal{S}_D$ will be referred to as the *similarity set*. The dimension of $\mathcal{S}$ increases rapidly with the size of the similarity window, e.g., for a $3 \times 3 \times 3$ similarity window, its dimension equals 27. Hence $\mathcal{S}$ is a high dimensional space.

In order to efficiently perform the NN search through the similarity set, the latter must be somehow organized. This is where the vantage point tree comes in. The vantage point tree is a data structure used to organize general metric spaces such that the K nearest neighbors (K-NN search) of a query can be efficiently found.

The structure is especially valuable when working in high dimensional spaces.

To build the vp-tree, the first step is to select a data point (i.e., the vantage point) in the space. This point can be randomly chosen. Then the data points are partitioned into two parts: half of the data points are enclosed in a hypersphere with the vantage point as center and radius μ, this creates the inside subtree, while the other half of the data points are outside this hypersphere, forming the outside subtree. The radius μ is the distance from the vantage point to the median value of the data set. In Figure 4.3, this first partitioning corresponds to the salmon pink curve and hence to the root in the associated tree.

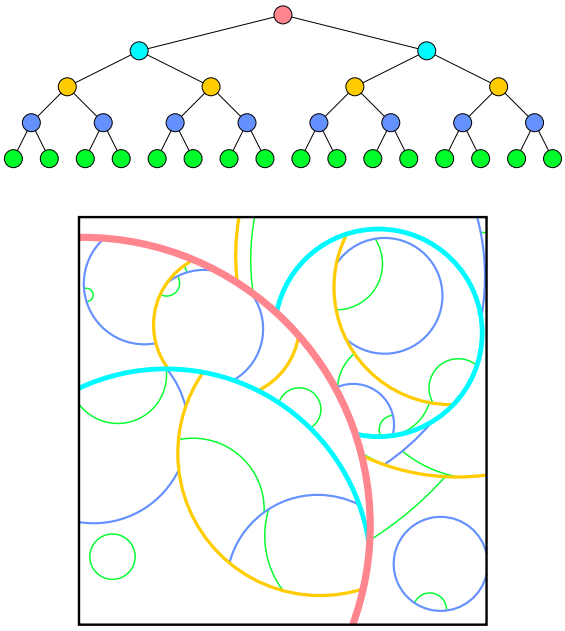


Figure 4.3: Vantage point tree. Tree data structure and corresponding space partioning. Source: [25].

The inside and outside subtrees are then further divided by recursively applying this procedure: in Figure 4.3 this leads to the cyan curves, then to the yellow one, etc. Doing this, we obtain a structure where neighbors in the tree are likely to be neighbors in the space.

The K-NN search through the vp-tree is a recursive branch-and-bound tree search in which a variable keeps track of the closest neighbors yet encountered. For more details, we refer the reader to [24].

4.2.3 VP-tree based non-local means denoising

As pointed out in Section 4.2.1 improvements of the NL-means filter are necessary in order to take fully advantage of its potential while keeping the computation time acceptable in real applications. In this section a solution is proposed to tackle this challenge. We first describe the main two additional elements on which our method is based, namely pixel selection and vp-tree search. The complete algorithm is given afterwards. In the following, the description is first provided for two-dimensional filtering in the interest of simplicity. The generalization to 3D is given at the end of this section.

Pixel selection

Firstly we exploit the data type we are working with, i.e., dose maps. As already mentioned earlier, there are plenty of low or null dose regions in a dose map. Since we have little interest in these regions from a clinical point of view we decide to only filter the pixels whose dose value is above a selected threshold. Doing this, the number of pixels to process is drastically reduced.

In practice the pixel selection step is done by sliding a $(2f+1)\times(2f+1)$ window across the pixel grid; if the sum of the dose values inside the window is above a specified threshold, the dose values inside the window (also called a patch) are stored together with the central pixel location. If not, the patch is not saved. Remark that the selection window has the same size as the similarity window mentioned in Section 4.2.1. An example of this step is displayed in Figure 4.4.

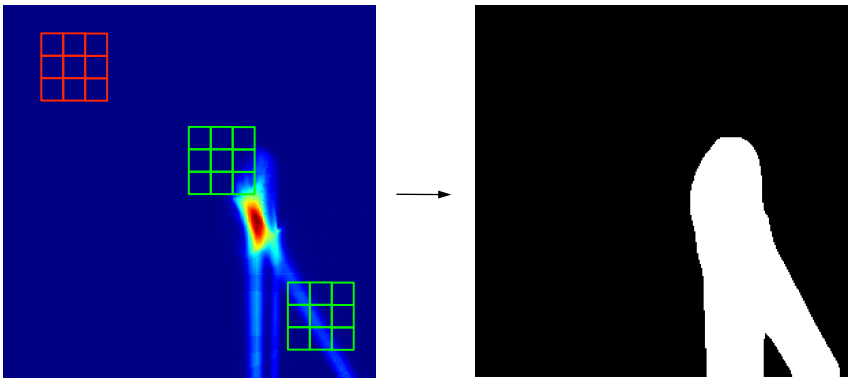


Figure 4.4: Pixel selection step. (Left) A selection window is slid across the pixel grid. If there is enough dose, the pixel patch is stored (e.g., green windows). If not, the pixel patch is not saved (e.g., red window). (Right) Resulting pixel selection. White (resp. black) pixels are (resp. are not) selected.

Each selected $(2f+1)\times(2f+1)$ patch is stored as a $(2f+1)^2\times 1$ vector in a matrix (as shown in the explanatory Figure 4.5). So, suppose that n patches are stored after this step, each one corresponding to one pixel, then we end up with a matrix $\boldsymbol{P}$ of size $(2f+1)^2\times n$.

VP-tree search

After the pixel selection step we need to find, for each selected pixel, the K most similar pixels (among the selected one). This problem can be reformulated as: for each column $\boldsymbol{p}$ of $\boldsymbol{P}$, find the K closest neighbors of $\boldsymbol{p}$ among $\boldsymbol{P}$. Using the notation introduced in Section 4.2.2, we need to perform a K-NN($\boldsymbol{p}$,$\boldsymbol{P}$) search for each $\boldsymbol{p}$ of $\boldsymbol{P}$. Note that the search is made in a space of dimension $(2f+1)^2$, i.e., the size of the similarity window.

As explained earlier, the vp-tree is a data structure that is suited for nearest neighbor search in high dimensional spaces. It corresponds thus well to our needs. In this work, we used an implementation of the tree construction and NN search initially written as part of the construction of K-NN graphs for clustering in [26]. It takes as input the matrix $\boldsymbol{P}$ and returns a matrix $\boldsymbol{M}$ of size $K\times n$ such that the i-th column of $\boldsymbol{M}$ contains the index of the K closest patches of patch i (see Figure 4.5).

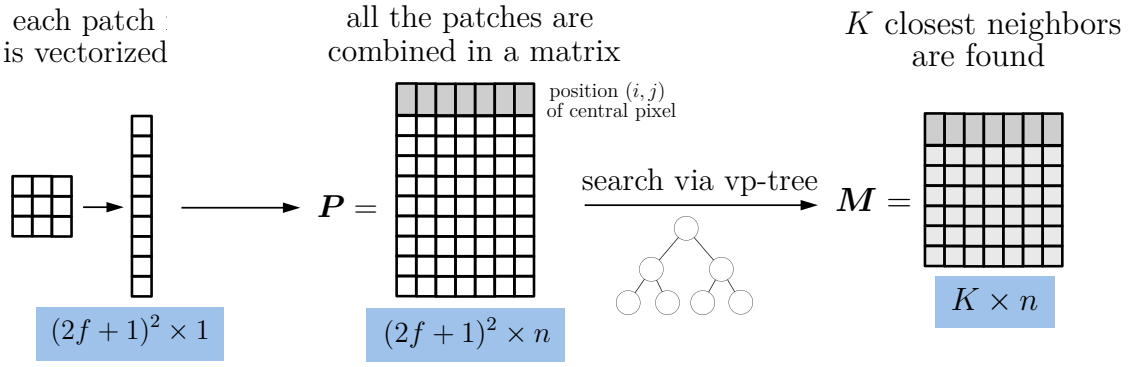


Figure 4.5: Visualization of the steps and corresponding matrices. At the end of the pixel selection step, each selected patch is vectorized leading to a $(2f+1)^2 \times n$ matrix $\boldsymbol{P}$. The indexes of the K closest neighbors of each column of $\boldsymbol{P}$ are then found using the vp-tree structure, leading to a $K \times n$ matrix $\boldsymbol{M}$.

During the search, every query is treated independently from the other, meaning that the knowledge of the neighbors of a pixel is not used to find the neighbors of another pixel. This allows the parallelization of the code with Open Multi-Processing (OpenMP). We will come back to this comment in Section 4.2.4.

VP-tree based NL-means algorithm

After the pixel selection and the nearest neighbors search steps, the classical NL-means algorithm, as given in Section 4.2.1, is performed. It only remains to measure the similarity of each patch with its neighboring patches in order to compute the weights. Afterwards the central pixel value of each patch is replaced with the weighted average of its neighbors, leading to an image where the noise is reduced. The complete algorithm is given by Algorithm 1 below. In the remainder of this text, this method will be referred to as VP-NLM/1.

Let us get back to the input arguments of the algorithm. Naturally, the input image must be given. The size of the similarity window (and hence the selection window) is also specified: intuitively, the larger the window size, the more information about the neighborhood is used and hence the more accurate the method is. However a too large window size might bias the filtered image. The filtering parameter h can be adapted too: the higher its value, the smoother the image appears.

Finally the number of closest neighbors to be found K is specified: the more neighbors are used, the better the denoising process. Note that, if a high number of neighbors is searched, we risk finding neighbors that are not especially similar anymore. Yet this is not a problem as the similarity of the neighbors is afterwards calculated during the weight computation: hence an unsimilar neighbor will receive a small weight and will not affect the quality of the image.

In order to study the complexity of the algorithm, suppose as before that you have a $N \times N$ image. The pixel selection step makes it difficult to express the complexity in terms of N, since the number of selected pixels n depends on the image. One can only say that $n = \alpha N^2$, where $0 \leq \alpha \leq 1$. In practice, with our data, we have $n \ll N^2$, as will be shown later.

Tree construction executes in $\mathcal{O}(n \log(n))$ time, and search is under some circum-

Algorithm 1: VP-tree based non-local means denoising

Input : image I to be denoised, half width of the similarity window f, filtering parameter h, number of neighbors K.
Output : denoised image $\tilde{I}$.

Boundary handling.
Pixel selection: for *pixel* i **do**
  Take the neighborhood $\boldsymbol{p_i}$ of pixel i
  if *sum* $\boldsymbol{p_i} \geq$ *threshold* **then**
    Store vectorized $\boldsymbol{p_i}$ in matrix $\boldsymbol{P}$ together with pixel index i.
end
VP-tree construction for database $\boldsymbol{P}$.
K-NN($\boldsymbol{p_i}$,$\boldsymbol{P}$) search for each $\boldsymbol{p_i}$ in $\boldsymbol{P}$.
Classical NL-means algorithm: for $\boldsymbol{p_i}$ *in* **$\boldsymbol{P}$ do**
  Compute the similiarity of $\boldsymbol{p_i}$ with closest neighbors.
  Compute the weights $\boldsymbol{w_i}$.
  Replace value of the pixel i with weighted average of neighbors.
end

stances and in the limit, $\mathcal{O}(\log(n))$ expected time for each query[1] [24]. Hence, the search for similar patterns through the whole dose map requires $\mathcal{O}(n \log(n))$ time, whereas the search required $\mathcal{O}(N^2)$ time for the initial version of the algorithm. The weight computation requires afterwards $\mathcal{O}(n)$ operations.

VP-tree based NL-means algorithm (3D)

Note that we have described the improved algorithm in the two-dimensional case for the sake of simplicity. However it can easily be generalized to 3D by using 3D windows for pixel selection and similarity computation. Let $f_t = 2f+1$ be the width of a window. Then we must use $f_t \times f_t \times f_t$ selection and similarity windows.

For computational purposes, we restrict the search area along the third dimension, i.e., smaller 3D volumes are treated independently from each other. Indeed treating the entire 3D volume all at once is computationally intensive and leads to a lot of useless work in our case: as can be seen in Figure 4.6, one slice exhibits plenty of redundancies, there is thus no need to explore the whole 3D volume to find similar patterns. Also, the most similar patterns are most often located in the same region. So, when searching for similar patterns of a pixel in slice 1, there is little to be gained from scanning the last slice. In fact it appears from the experimentation that the results are better when restricting the search to smaller volume.

As a consequence a new parameter f_z is added for the 3D implementation in order to give the size of the search window along the third dimension: for example $f_z = 3$ means that volumes of 3 slices are handled one after the other.

[1]Note that the complexity is given here in the asymptotic case, i.e., as $n \to \infty$. This justifies why it is independent of the dimension of the query and the number of neighbors to be found.

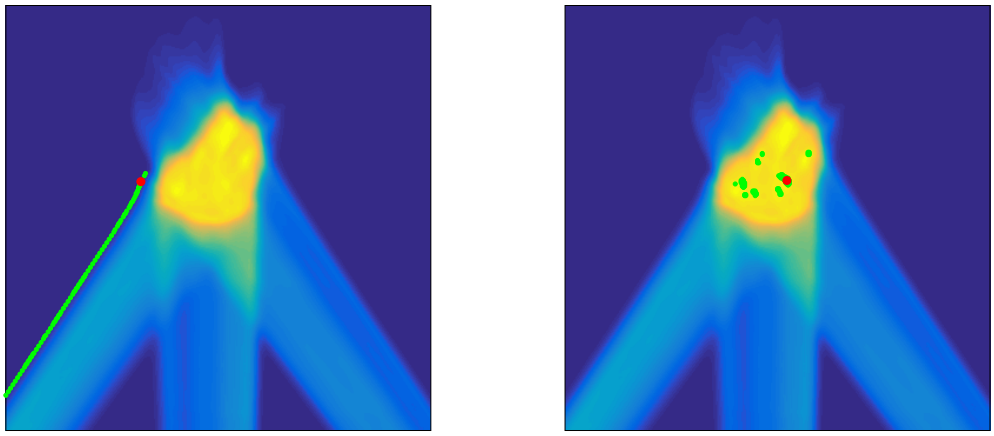


Figure 4.6: Example of similar pixels selected using the vp-tree. The green dots are the 100 closest neighbors of the red dot.

4.2.4 Accelerated VP-tree based NL-means algorithm

When analysing the time taken by each step of the previous algorithm we have observed that the main part of the computation time is used to perform the nearest neighbors search through the vp-tree. For this reason, we focus on the acceleration of this step in order to further speed up the NL-means algorithm. We hope to have a speedup large enough to allow the use of even bigger windows and hence to improve the reduction of the noise in a limited time.

Let us come back to the following remark concerning the vp-tree search: during the search, every query is treated independently from the other, meaning that every search is a completely new search. Doing so, we do not exploit the symmetry of the relation « is close to » (if i is close to j, it follows that j is close to i). Our idea is to take advantage of this symmetry to accelerate the search through the vp-tree. This is done as follows: after having found the nearest neighbors of pixel i, we attribute the exact same neighbors list to the identified neighbors. Let us clarify this with the example based on Table 4.1: in this example, the nearest neighbors of pixel a are $\{b, c, d\}$. The same neighbor list is then attributed to b, c and d (except at the place of the query itself, which is replaced by a). The work of finding the closest neighbors in the tree for b, c and d is thus avoided. Note that parallelization during the search is more complex in this case, as the queries are dependent from each other.

$\boldsymbol{a}$
b
c
d

$\longrightarrow$

$\boldsymbol{b}$	$\boldsymbol{c}$	$\boldsymbol{d}$
a	b	b
c	a	c
d	d	a

Table 4.1

With this algorithm, it is recommended to use a relatively small number of neighbors. Indeed the more neighbors are used, the faster the method goes but the more approximate it becomes. Note once again that a bad neighbors choice does not bias the dose approximation since a weight based on the similarity is associated with each neighboring values; it *only* reduces the denoising power of the method.

It is worth emphasizing that j being the closest neighbor of i does not imply that i is the closest neighbor of j or, in other words, the relation « is the closest neighbor of » is not symmetric. This means that, by applying the above rule, a list of (hopefully close) neighbors is attributed to some pixels, instead of the list of the exact closest neighbors (as with the previous algorithm). Hence this method – that we will call VP-NLM/2 – is a sub-optimal version of VP-NLM/1.

Regarding the complexity of this accelerated algorithm, a short reasoning suggests that the number of searches to be performed is approximately divided by the number of neighbors.

4.3 Results and discussion

In this section, we test the proposed denoising algorithms based on two different experiments : the first one consists in reducing the noise present in a 3D dose map, it is thus similar to what has been done in the previous chapter. Results are reported and discussed in Section 4.3.1. In the second experiment the aim is to reduce the noise of a collection of beamlets; this will be explained and discussed in details in Section 4.3.2.

For the original version of the non-local means filter, a MATLAB implementation written by Buades *et al.* is used. We improved it by implementing the two proposed algorithms, namely VP-NLM/1 and 2, with MATLAB, based on the vp-tree code given in [26].

4.3.1 Filtering a dose map

To assess the performance of the two described filters and evaluate the speedup gained by using these filters instead of the standard NL-means filter, we set up a denoising experiment with the two test cases given in Chapter 1. The results are assessed using the same quality metrics as in the previous chapter (see Section 3.2).

In the following, when a 2D filtering method is used to filter a 3D dose map, it means that each 2D slice is filtered separately by the 2D method and then recombined into a 3D volume. When a 3D method is used, we set $f_z = 3$. The filtering parameter h has been determined experimentally and set to $h = 2.5$ in both cases.

Results for the lung patient

Let us first report the results for the dose map corresponding to the lung patient. The reference (resp. noisy) dose map is obtained by simulating 10^8 (resp. 10^6) particles using the fast Monte Carlo code MCsquare. The visual performance of the methods is compared in Figure 4.7. The two-dimensional standard NL-means filter is referred to as Buades. The 3D dose map has been filtered using Buades method with a 3×3 similarity and a 5×5 search window in approximately 18 minutes. With our new algorithms, we can afford to use a larger similarity window, here a $5 \times 5 \times 5$ window is chosen. The filtered dose map is obtained in 10 minutes with VP-NLM/1 and 3 minutes with VP-NLM/2. The original method is thus significantly speeded up.

It is clear that the VP-NLM algorithms yield improved dose maps: for example, the red oval shape is not totally recovered with Buades, while it is perfectly recovered with the two other methods. Also, the dots present at the right bottom corner in the noisy image are almost completely removed by the VP-NLM methods. Hence, as expected, using larger search windows can favor the reconstruction of an image in some cases.

Regarding the differences between the outputs of VP-NLM/1 and VP-NLM/2, we remark that the latter is noisier. This is especially apparent in Figure 4.8, where the dose profiles along $x = 180$ are displayed: the last subplot corresponds to the VP-NLM/2 filtered output which is less smooth than the other, especially towards the end. We see here the impact of attributing the same neighbors list to multiple

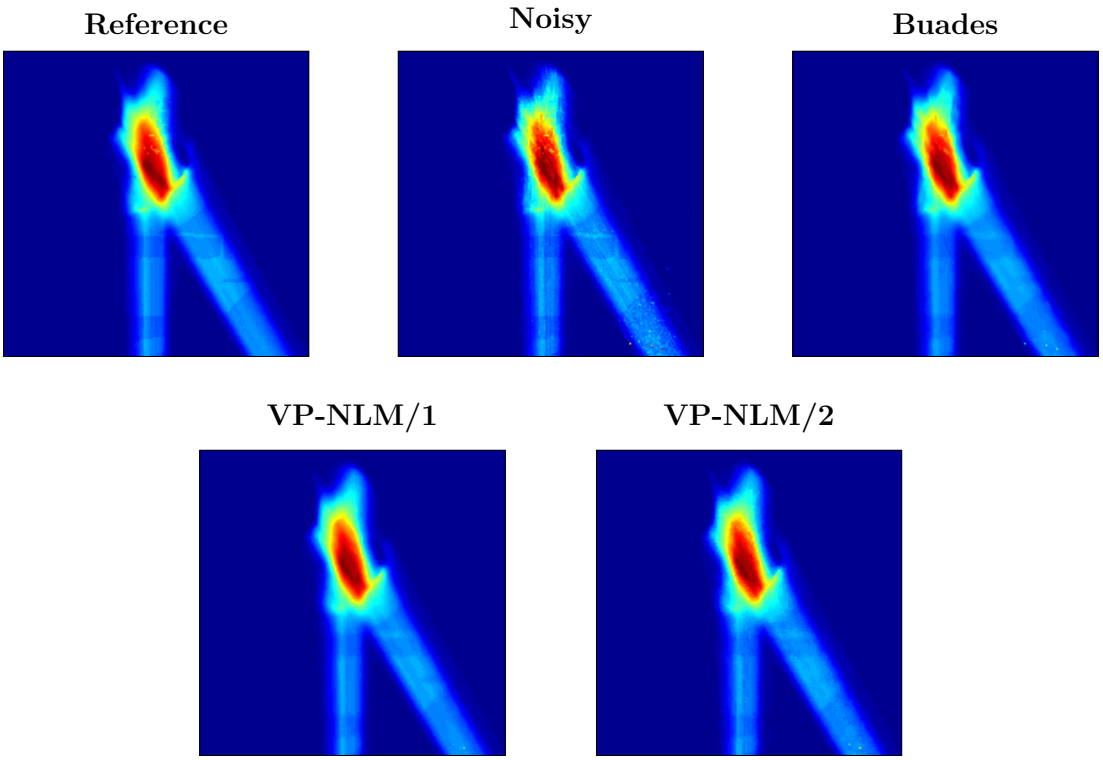


Figure 4.7: Visual inspection of a slice of the dose map (lung data): (i) reference, (ii) noisy, (iii) filtered dose map obtained with Buades with a 3×3 window using a 5×5 search zone, (iv) filtered dose map obtained using VP-NLM/1 with a $5 \times 5 \times 5$ window, $K = 100$ and (v) filtered dose map obtained with VP-NLM/2 with a $5 \times 5 \times 5$ window, $K = 30$.

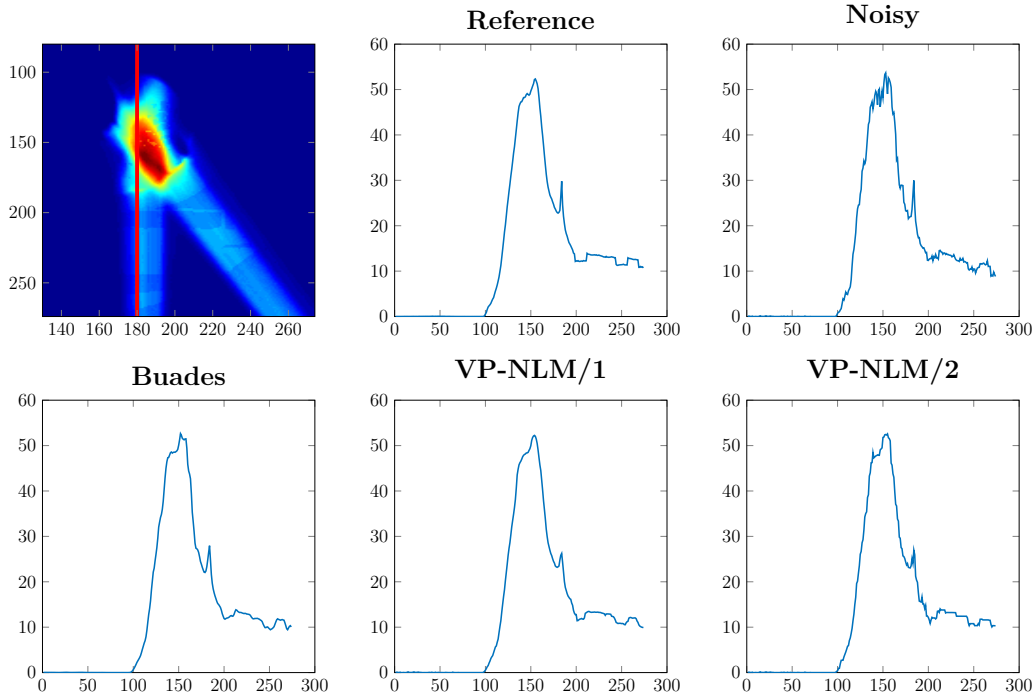


Figure 4.8: Dose profile along $x = 180$ corresponding to the same methods and parameters as in Figure 4.7. The red line in the first subplot represents the cut.

pixels. This figure shows once again the superior results obtained with VP-NLM/1: observe that, with this method, the reference dose profile is almost fully retrieved.

Table 4.2 displays the accuracy criteria for the same dose map when diverse parameter values are selected, i.e., different window sizes and numbers of neighbors K. The two- and tree-dimensional versions of the methods are studied. As a reminder, the used accuracy criteria are described in Section 3.2. The computation time is given here as an indication.

	Timing $[s]$	MSE	EPI	$\|\Delta\text{DVH}\|_2$	$s_{X,95}$
noisy	/	0.56	0.48	138	0.73
Buades (2D), $h = 2.5$					
3×3 5×5	1110	0.23	0.67	62	0.47
VP-NLM/1 (3D), $h = 2.5$					
$3 \times 3 \times 3$ $f_z = 3$ $K = 30$	140	0.20	0.60	58	0.34
$3 \times 3 \times 3$ $f_z = 3$ $K = 50$	144	0.18	0.63	53	0.36
$3 \times 3 \times 3$ $f_z = 3$ $K = 100$	188	0.17	0.65	39	0.28
$3 \times 3 \times 3$ $f_z = 3$ $K = 200$	265	0.16	0.66	36	0.26
$5 \times 5 \times 5$ $f_z = 3$ $K = 100$	612	0.16	0.72	54	0.23
VP-NLM/1 (2D), $h = 2.5$					
3×3 $K = 100$	94	0.28	0.55	63	0.42
5×5 $K = 100$	136	0.26	0.57	48	0.36
VP-NLM/2 (3D), $h = 2.5$					
$3 \times 3 \times 3$ $f_z = 3$ $K = 30$	73	0.27	0.51	71	0.40
$5 \times 5 \times 5$ $f_z = 3$ $K = 30$	183	0.24	0.55	44	0.37
$7 \times 7 \times 7$ $f_z = 3$ $K = 30$	479	0.27	0.55	43	0.38
VP-NLM/2 (2D), $h = 2.5$					
5×5 $K = 25$	85	0.34	0.49	91	0.45
7×7 $K = 25$	68	0.34	0.51	75	0.45

Table 4.2: Comparative values for different non-local means algorithms with varying parameters ($274 \times 274 \times 162$ dose map, lung patient). Similarity and search window sizes are indicated for Buades filter.

The first striking observation is that the computation time of the initial method (Buades) is drastically reduced. Comparing it with the two proposed methods makes little sense since a full search is performed with the new algorithms, i.e., there is no restriction in the search area anymore. However, it is clear that the computation time of the proposed methods is always way smaller than the initial one.

We also compare the speedup achieved with VP-NLM/2 with respect to VP-NLM/1: for the same window size, an acceleration of a factor 2 to 3.5 can be observed. Nevertheless, when examining the accuracy criteria, it clearly appears that the accelerated VP-NLM/2 yields poorer results, as already noticed based on the dose map and profile. We note that the 3D methods perform better than their corresponding 2D methods.

Concerning the impact of the parameters value for VP-NLM/1, we unsurprisingly see that the more neighbors are sought, the better the quality measures are (but the higher the calculation time is). For VP-NLM/2, it has been experimentally observed that using too high a number of neighbors degrades the results (not shown

here), as already suggested before. Then, regarding the impact of the window size, increasing the size of the similarity window does not always result in better outcomes.

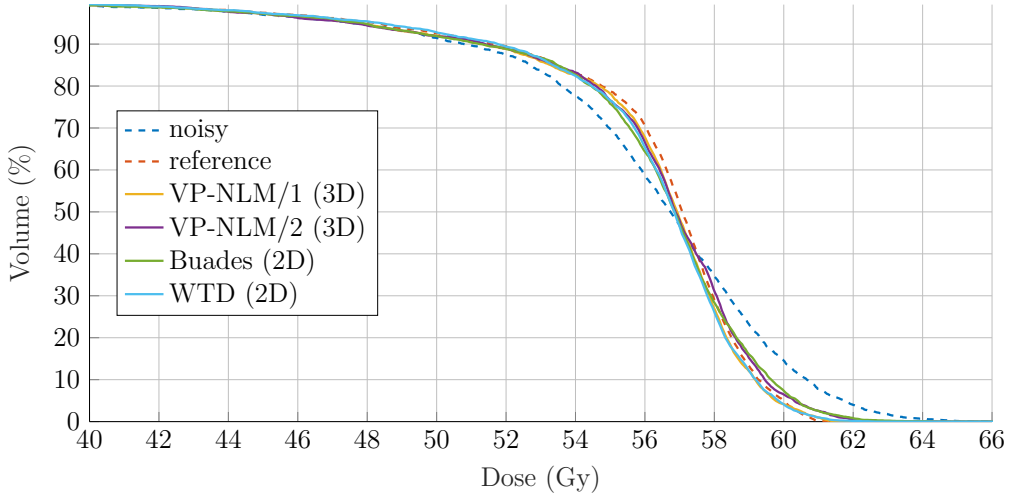


Figure 4.9: DVH corresponding to the lung patient with (i) noisy, (ii) reference, (iii) VP-NLM/1 ($3 \times 3 \times 3$ window, $K = 100$) (iv) VP-NLM/2 ($5 \times 5 \times 5$ window, $K = 30$) (iv) Buades filter (3×3 similarity window, 5×5 search zone) and (v) WTD (2D) with $\theta = 0.6$.

The DVHs are shown in Figure 4.9. The curve associated with VP-NLM/1 is the closest to the reference curve. It confirms again the good results obtained with this method.

Results for the prostate patient

We now turn on to the reporting and analysis of the second data set: a $256 \times 256 \times 71$ dose map for a patient with a tumor in the prostate. This dose map contains fewer slices than the previous one: the computation time needed to filter the dose map will therefore be shorter. The reference (resp. noisy) dose map is obtained by simulating 10^9 (resp. 10^7) particles using the fast Monte Carlo code MCsquare.

The visual performance of the methods is compared in Figure 4.10. As above, a small window size is used for Buades algorithm for computational purposes and a larger window size is taken for the two other methods. Note that the filtering with Buades method is realized in 6.3 minutes. The dose map obtained with VP-NLM/1 (in 2.5 minutes) seems visually close to the reference dose map and is comparable to the Buades dose map. VP-NLM/2 (obtained in 2.5 minutes) yields once again a noisy image. The impact of the pixel selection step is visible for the VP-NLM methods: indeed, as explained earlier, only pixels whose dose value is above a defined threshold are processed. Hence noise is still present in the low dose regions.

A comparative table, similar to the one displayed for the previous data set, is given in Table 4.3. From this table, it appears that a large window size leads to less good results for the tree-dimensional methods in this case.

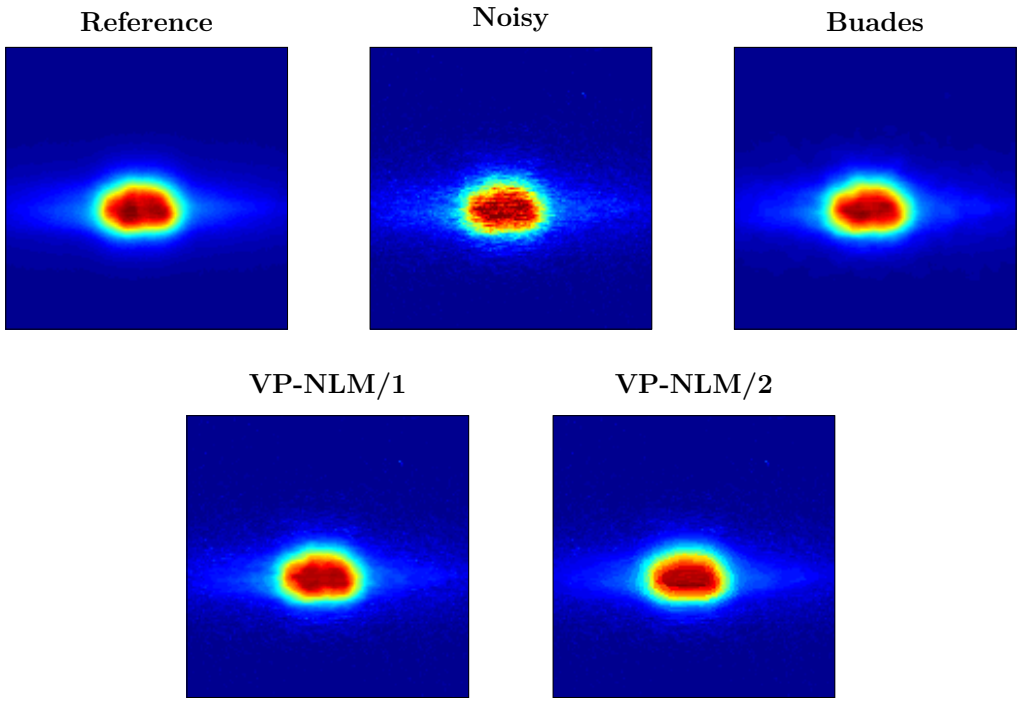


Figure 4.10: Visual inspection of a slice of the dose map (prostate data): (i) reference, (ii) noisy, (iii) filtered dose map obtained with Buades with a 3×3 window using a 5×5 search zone, (iv) filtered dose map obtained using VP-NLM/1 with a $3 \times 3 \times 3$ window, $K = 200$ and (v) filtered dose map obtained with VP-NLM/2 with a $7 \times 7 \times 7$ window, $K = 30$.

The resulting DVHs are given in Figure 4.11: the curves extracted from the VP-NLM filtered dose maps are on top of each other and are the closest to the reference curves.

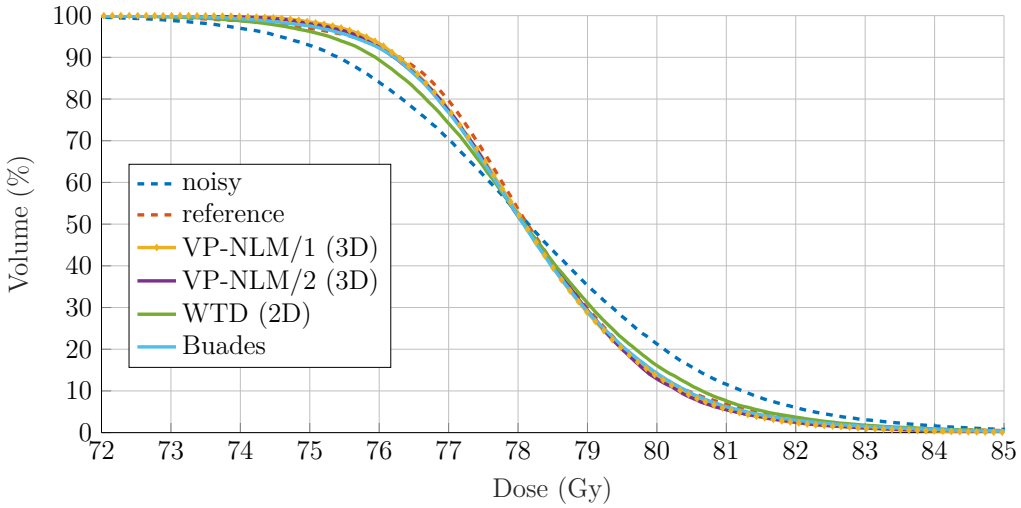


Figure 4.11: DVH corresponding to the prostate patient with (i) noisy, (ii) reference, (iii) VP-NLM/1 ($3 \times 3 \times 3$ window, $K = 100$) (iii) VP-NLM/2 ($5 \times 5 \times 5$ window, $K = 30$) (iv) WTD with $\theta = 0.6$ and (v) Buades filter (3×3 similarity window, 5×5 search zone).

	Timing $[s]$	MSE	EPI	$\|\Delta\text{DVH}\|_2$	$s_{X,95}$
noisy	/	0.85	0.64	105	0.59
Buades (2D), $h = 2.5$					
3×3 5×5	380	0.36	0.86	22	0.34
VP-NLM/1 (3D), $h = 2.5$					
$3 \times 3 \times 3$ $f_z = 3$ $K = 30$	68	0.30	0.83	36	0.30
$3 \times 3 \times 3$ $f_z = 3$ $K = 50$	69	0.27	0.85	34	0.29
$3 \times 3 \times 3$ $f_z = 3$ $K = 100$	95	0.26	0.86	26	0.26
$3 \times 3 \times 3$ $f_z = 3$ $K = 200$	145	0.26	0.87	24	0.24
$5 \times 5 \times 5$ $f_z = 3$ $K = 100$	255	0.31	0.87	40	0.20
VP-NLM/1 (2D), $h = 2.5$					
3×3 $K = 100$	100	0.38	0.83	47	0.36
5×5 $K = 100$	173	0.37	0.85	29	0.29
VP-NLM/2 (3D), $h = 2.5$					
$3 \times 3 \times 3$ $f_z = 3$ $K = 30$	37	0.42	0.76	44	0.36
$5 \times 5 \times 5$ $f_z = 3$ $K = 30$	67	0.46	0.76	19	0.33
$7 \times 7 \times 7$ $f_z = 3$ $K = 30$	148	0.53	0.75	24	0.33
VP-NLM/2 (2D), $h = 2.5$					
3×3 $K = 30$	39	0.55	0.73	74	0.45
5×5 $K = 30$	50	0.52	0.74	42	0.39
7×7 $K = 30$	75	0.55	0.75	24	0.37

Table 4.3: Comparative values for different non-local means algorithms with varying parameters ($256 \times 256 \times 71$ dose map, prostate patient).

Discussion

The reported results confirm the good performances of VP-NLM/1: the denoising process is well performed and the computation time is drastically reduced (compared to Buades method). We emphasize that the standard NL-means filter has not been improved strictly speaking, it has only been accelerated. As it allows the use larger similarity windows and the full search, the denoising process can in some situations be improved.

Now that the calculation time is smaller, the use of this algorithm can be considered for real-life applications. For example, we remember from Table 1.1 that the noisy and reference dose maps for the lung patient are obtained in 9s and 525s respectively. As good results are obtained in approximately 200s with VP-NLM/1 (according to Table 4.2), a reliable dose map can be obtained in one half on the time previously required. For the prostate patient, the speedup is even more considerable as the noisy and reference maps are obtained in approximately 25s and 2491s respectively. Since the filtering process takes about 150s with VP-NLM/1, the speedup factor is close to 15.

For the second discussed method, namely VP-NLM/2, the computation time is clearly accelerated but the price to pay for this acceleration is that the denoising process is less efficient. As often we face a trade-off between speed and accuracy. Additionally, this method is strongly dependent on the order of the list of the pixels

that must be filtered. Indeed, for the first pixel in the list, we will always search the closest neighbors while for the last one, there are a lot of chances that we will not have to. Other approaches could therefore be tested. For example, we could imagine a random shuffle of the list before the search; this would prevent from placing the performance of the method in the first pixels only. Another possibility would be to select the pixels that already possess a neighbors list and, for each of these pixels, perform a nearest neighbors search with a probability p. These two probabilistic approaches are meant to redistribute the chances between all the pixels of having a list of nearest neighbors and hence having a better dose approximation.

Regarding the impact of the parameters, we note that, in general, increasing the size of the similarity window does not always give better results. The optimal window size is dependent on the data set that must be filtered. It appears that a large window may lead to the loss of details. The effect of the filtering parameter h has not been shown here as it has not a huge impact on the results, as studied experimentally. Too high h-values (i.e., $h > 3$) tend to produce slightly over-smoothed dose maps. Finally, concerning the number of neighbors, the more neighbors are taken, the better the outcome is for VP-NLM/1. This is not the case for VP-NLM/2: taking too high a number of neighbors leads to a rough approximation of the dose value for some pixels and is thus not recommended.

4.3.2 Filtering beamlets

In this section, we set up a new denoising experiment where the purpose is the reduction of the noise in a collection of beamlets. Before examining and discussing the results obtained with the proposed algorithm, let us describe in details the goal and challenges of the experiment.

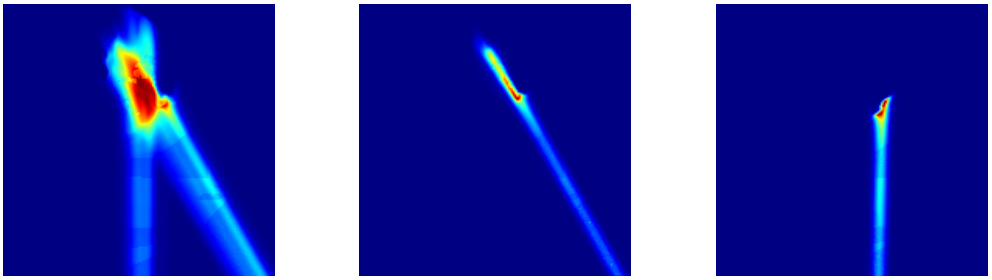


Figure 4.12: Example of beamlets (lung patient): the left image is the total dose map and the two rights images are examples of beamlets that are summed to generate this dose map.

As explained in Section 2.4, a proton beam can be decomposed in so-called beamlets, i.e., smaller rays. Indeed the dose maps that have been examined throughout this work can be seen as a weighted sum of beamlets. For instance the dose map associated with the patient with a tumor in the lung is made of 243 beamlets; examples of such beamlets are shown in Figure 4.12. Beamlets are very useful (e.g., for dose calculations treatment planning, for intensity distribution optimization) but, the computation of such a family of beamlets is extremely expensive in time. For information purposes only, the computation time needed per beamlet is given

in Table 4.4 when different particle numbers are used per beamlet; this table is of course specific for the lung patient. Based on this table, we see that more than 1 day is required to obtain the 243 beamlets with 10^6 particles per beamlet. Hence, to reduce the calculation time, fewer proton are simulated per beamlet; however, the resulting family of beamlets is very noisy.

	Number of simulated particles per beamlet		
	5E4	1E5	1E6
Timing $[s]$	23.5	40.9	358.5

Table 4.4: Mean time needed for the computation of a beamlet with fast Monte Carlo code MCsquare (in seconds).

The challenge here is thus the reduction of the noise present on beamlets with the proposed algorithm. In order to make the denoising process even more efficient, we will take advantage of the large amount of beamlets that we have. Indeed, as pointed out in [23], the non-local means algorithm is particularly valuable when there are plenty of redundancies in the image, which is logical in view of the nature of the algorithm. Hence, combining multiple beamlets would add similar patterns and therefore improve the denoising process. So, in a nutshell, the idea here is to reduce the noise present in a beamlet by using dose values coming from other beamlets. This technique is comparable to the method proposed in [27] where patch dictionaries are build before effectively denoise the images.

In practice, this is done by simply giving a subgroup of beamlets as input to the algorithm, this way the vp-tree search is performed through the whole subgroup of beamlets. The VP-NLM/2 method is not tested here as we have seen in the previous section that its performances were inferior. Only the VP-NLM/1 algorithm is tested.

Table 4.5 gives the comparative results for two sets of noisy beamlets when different sizes of subgroup are used. For instance, when the subgroup size is equal to 5, this means that each beamlet is filtered using pixels contained in itself and in 4 other beamlet dose maps. A subgroup size equal to 1 means that each beamlet is treated independently. Once again, the timing per beamlet is provided for information purposes. For each beamlet, the mean squared error between the reference (obtained with 10^6 particles/beamlet) and filtered dose map is computed based on the non-zero dose pixels of the reference only. The mean squared error for each beamlet is displayed in Figure 4.13. In Table 4.5, we show the mean and total MSE (total refers to the sum of the 243 MSE-values). The total MSE between the reference and noisy collection of beamlets with 5E4 particles (resp. 1E5 particles) is 0.77 (resp. 0.41). Remark that the impact of the parameters of the VP-NLM/1 is not studied here, we only observe the effect of the subgroup size.

We have two main observations: firstly, it is clear that the MSE decreases with the subgroup size. In other words, the more beamlets are combined, the better the performances. Hence we could logically combine the whole set of beamlets with the hope of having the smallest MSE possible. However the gain obtained by combining steadily more beamlets decreases. We can thus suppose that there is no need to combine all the beamlets together, as there are already lots of redundancies in smaller groups. Additionally, using such a large subgroup size would lead to a

computation time potentially higher than the time needed to compute the reference beamlets.

	Subgroup size	Timing [s]	mean MSE (10^{-3})	total MSE (10^{-1})
5E4 particles /beamlet	1	17	1.61	3.91
	2	20	1.39	3.38
	5	35	1.27	3.10
	10	55	1.25	3.05
1E5 particles /beamlet	1	15	1.41	3.43
	2	18	1.19	2.89
	5	29	1.04	2.54
	10	53	0.98	2.40
	15	67	0.97	2.36

Table 4.5: Comparative values for different subgroup sizes for two noisy collections of beamlets. The VP-NLM/1 (3D) method is used with a $3 \times 3 \times 3$ window, $h = 2.5$ and $K = 100$.

Secondly, note that the time needed for the filtering of a 3D beamlet is way smaller than the time required to filter a 3D dose distribution (as given in the previous section). It is due to the fact that the beamlets are very sparse matrices. Consequently the number of pixels that must be processed per beamlet is small (as only nonzero pixels are filtered).

4.4 Conclusions

In this chapter, a new filter type has been tested, namely a non-local filter. As its name suggests it, the non-local means filter uses the information of the whole image and not only the direct neighbors of a pixel; this is what differentiates this filter from the ones covered in the previous chapter. The NLM filter is especially powerful for reducing the noise of images with lots of redundant patterns. Since dose maps present plenty of redundancies, the NLM filter suits the situation.

However, the NLM filter, as initially proposed by Buades, is still local in a certain sense and is extremely slow. To remedy theses issues, we have proposed an algorithm which is truly non-local and performs the filtering much faster. This algorithm, called VP-NLM/1, is based on the vp-tree data structure to perform the nearest neighbor search. It has shown very good performances regarding the timing and the denoising process. For the same window size (e.g., 3-by-3) a speedup factor close to 11 has been observed. This method has also been tested in the case of beamlets filtering: when combining the denoising of multiple beamlets, the total mean squared error between the reference and filtering dose maps is divided up to a factor 2.

A second method, called VP-NLM/2, has been built with the aim of further accelerating the filtering process. It takes advantage of the possible symmetry of the

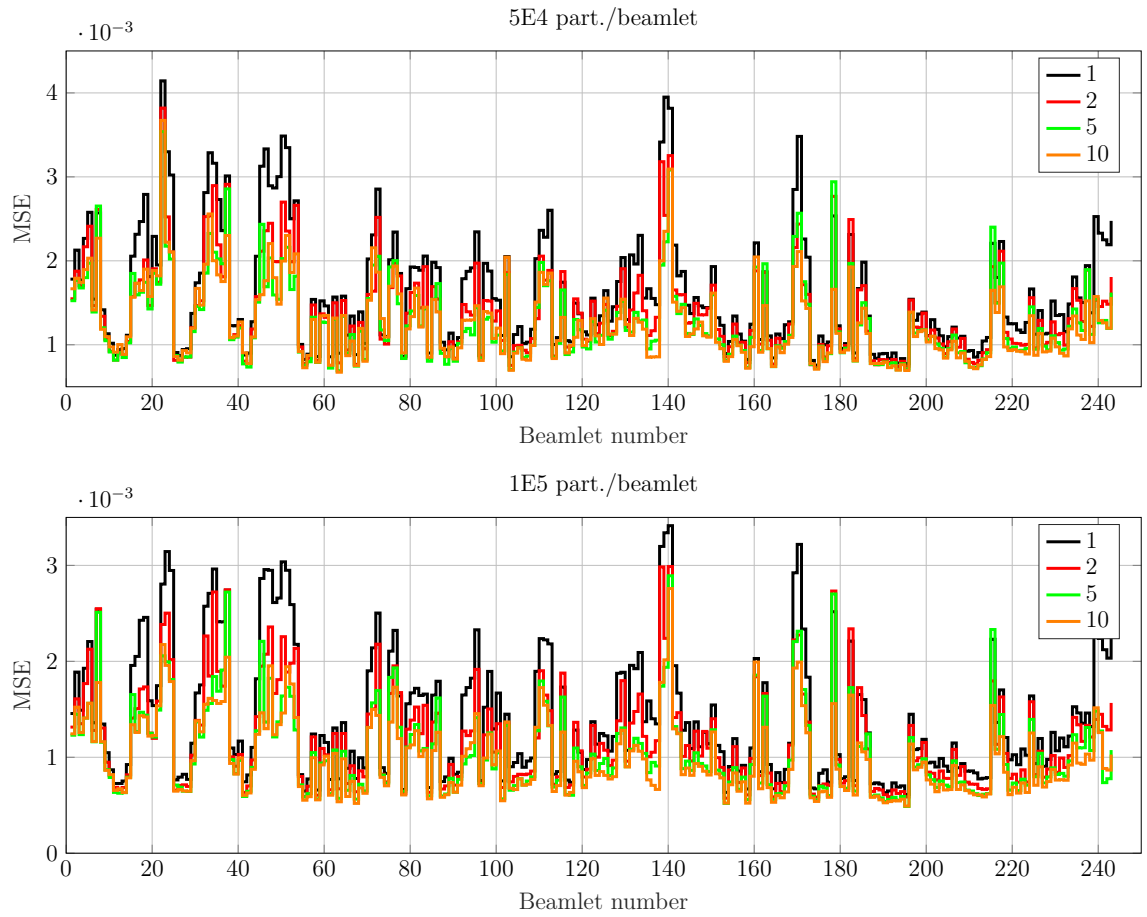


Figure 4.13: Mean squared error between reference (i.e., 10^6 part./beamlet) and noisy collection of beamlets for different subgroup sizes. The subgroup size is indicated in the legend.

relation « is the neighbor of » in order to perform less searches trough the vp-tree, which is the most time-consuming part of the process. VP-NLM/2 is faster but less efficient. Yet this method is promising, areas for improvement are given earlier in the chapter.

CHAPTER 5

Conclusions

Denoising as post-processing step

The present work demonstrates that denoising methods considerably reduce the statistical fluctuations in Monte Carlo computed dose distributions for proton therapy while decreasing the error with respect to the reference dose map. For both studied test cases, the statistical uncertainty, measured by the 95th percentile of the standard error of the mean, is divided up to a factor 3. Additionally the mean squared error between the reference and filtered dose distribution is reduced up to a factor 3.5. Consequently, by using a denoising method on a dose map computed with a reasonable number of protons, the overall computation time needed to obtain a reliable dose map is significantly reduced. We therefore emphasize that this post-processing step should be added in the MC based treatment planning.

Two issues encountered during the work should be investigated. Firstly one of the main challenges with denoising methods is to decrease statistical errors as much as possible without introducing systematic errors. It is even more important in the present context as systematic error may have a direct impact on the patient treatment. In the current study the potential bias introduced in the filtered dose maps is measured using the energy conservation ratio (see Section 3.2) ; it has been defined based on a physical reasoning. However it appears that this ratio is close to one for all the tested techniques, hence leading to the conclusion that no bias was introduced by any of the filtering techniques globally. Yet it provides no information locally, e.g., edge blurring, introduction of large errors in a few pixels. We believe that the energy conservation ratio is not suited to accurately measure the bias. Hence, we judge that other metrics should be investigated in order to detect systematic errors more accurately. Note that the edge preservation index provides a partial solution to this issue regarding the edge blurring.

Secondly, we were only able to make a comparative study here ; this means that the performances of the methods are compared with each other but it cannot be stated whether the noise is *enough* reduced. Indeed, as far as we know, there is for the moment no defined acceptable noise level for proton therapy dose maps. This medical decision must be taken based on the judgment of clinicians. We hence point out the need to establish which uncertainty level is acceptable for proton therapy dose evaluation.

Filtering techniques

Filtering techniques were selected as denoising methods in this work. Various filters have been tested on two dose distributions. All the denoising metrics indicate the superiority of the non-local means filter over the techniques tested in Chapter 3, i.e., Gaussian filter, median filter and wavelet threshold denoising. This superiority is also confirmed by Figure 5.1: it shows the difference between the filtered and the reference dose map, scaled by the square root of their sum,

$$\mathcal{F}(x, y) = \frac{y - x}{\sqrt{y + x}}, \tag{5.1}$$

where x and y are the reference and filtered dose distributions. Equation 5.1 is known as the Fisz transformation [28]. It yields the remaining noise of the filtered image and it approximately follows a Gaussian distribution.

Hence, as one expects from a good denoising method that it removes noise only, this substraction should ideally yield an image where the features of the initial map are not visible. Clearly, this is not the case for subplots (a), (b) and (c): the geometrical features can be easily distinguished and the noise has a specific structure, similar to the shape of the beams.

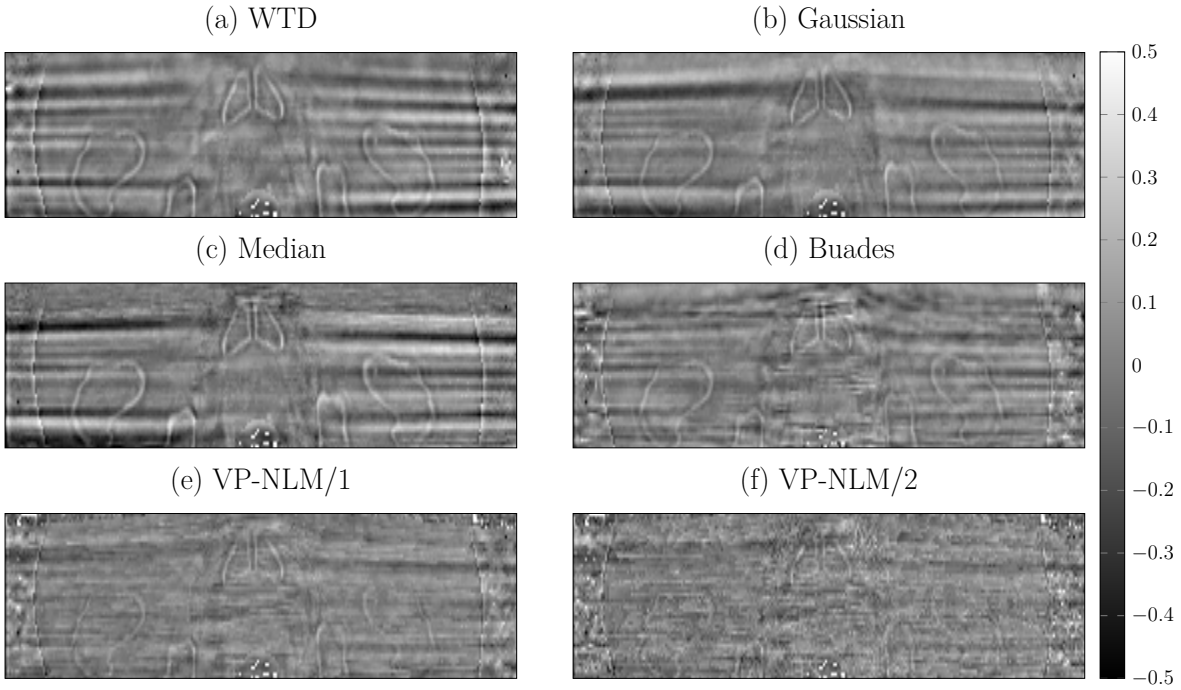


Figure 5.1: Noise study for various filters (prostate patient): (a) WTD with $\theta = 0.6$, (b) 3D Gaussian filter with a $3 \times 3 \times 3$ window and $\sigma = 1$, (c) 3D median filter with a $3 \times 3 \times 3$ window, (d) non-local means filter with $h = 2.5$ and 3×3 similarity and 5×5 search windows, (e) VP-NLM/1 method with $h = 2.5$, a $3 \times 3 \times 3$ window and 100 neighbors and (f) VP-NLM/2 method with $h = 2.5$, a $3 \times 3 \times 3$ window and 30 neighbors.

Similar observations can be made for the non-local filters, but it is way less striking, especially for the VP-NLM methods. It therefore confirms that the latter better preserves the features of the dose distribution. Note that contours are less noticeable

with VP-NLM/2 than with VP-NLM/1: the reason for this is not that the former method better preserves the edges, it is due to the fact VP-NLM/2 produces noisier dose maps compared to VP-NLM/1, as explained earlier.

A final comparison between all studied filters is provided in Figure 5.2. It shows the isodose lines in a similar fashion to Figure 3.3 and highlights once again the superior performances of the VP-NLM method, e.g., the method succeeds in perfectly recovering the transition between the large spot in the center and the right beam, as opposed to the other filters. Small speckles are however still noticeable in the bottom right corner.

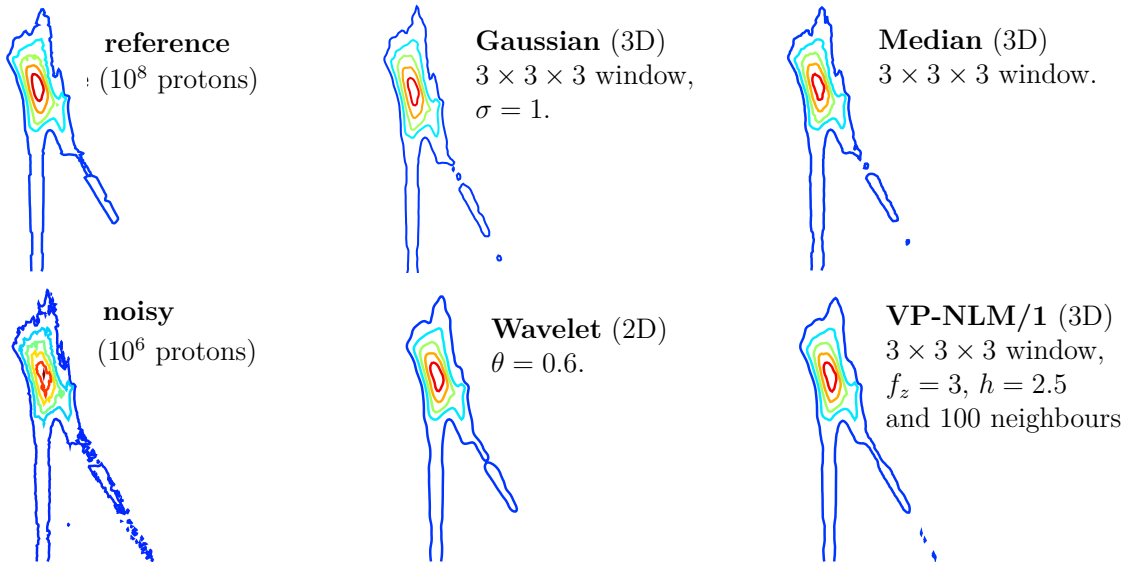


Figure 5.2: Isodose lines for all studied filters (lung patient, slice 60).

Deep learning algorithms

Deep learning is part of the broader family of machine learning methods. These algorithms rely on neural networks which are able to recognize patterns, after having been trained with a large amount of data. These are widely used in image processing. For more details, we refer the reader to [29]. Deep learning is currently investigated in the context of proton therapy for dose map denoising.

The aim of this section is to compare the performances of a deep learning method with the VP-NLM/1 filter. The deep learning algorithm (DLA) considered here has been implemented by J. Asensi. Approximately 2.5 hours were needed to train the algorithm; this step must however be performed once and for all. Both denoising methods are tested on a dose distribution different from the ones previously used in this work. The test case corresponds to a PBS proton therapy treatment of a lung tumor, initially planned with RayStation. The resulting dose map, obtained with MCsquare, is a $512 \times 512 \times 148$ voxel grid. This sudden test case change is dictated by the DLA implementation: for the moment the latter requires a specific data format. In addition, the network has currently been trained to reduce the noise present

in dose maps obtained by simulating 10^7 particles. Hence the dose map obtained with 10^7 protons is considered here as the noisy image. The reference dose map is produced by simulating 10^9 protons.

Resulting dose distributions are shown in Figure 5.3. Remark that the dose map considered as noisy seems very similar to the reference one: the denoising process is therefore less challenging than in the previous chapters. Indeed, earlier in this work, the dose maps considered as noisy were computed based on 10^6 protons.

No striking difference between the outputs of the denoising methods can be observed with the naked eye. Both methods seem to succeed in recovering the reference dose map. Small speckles are however visible in the corner of the dose distribution produced by the VP-NLM filter.

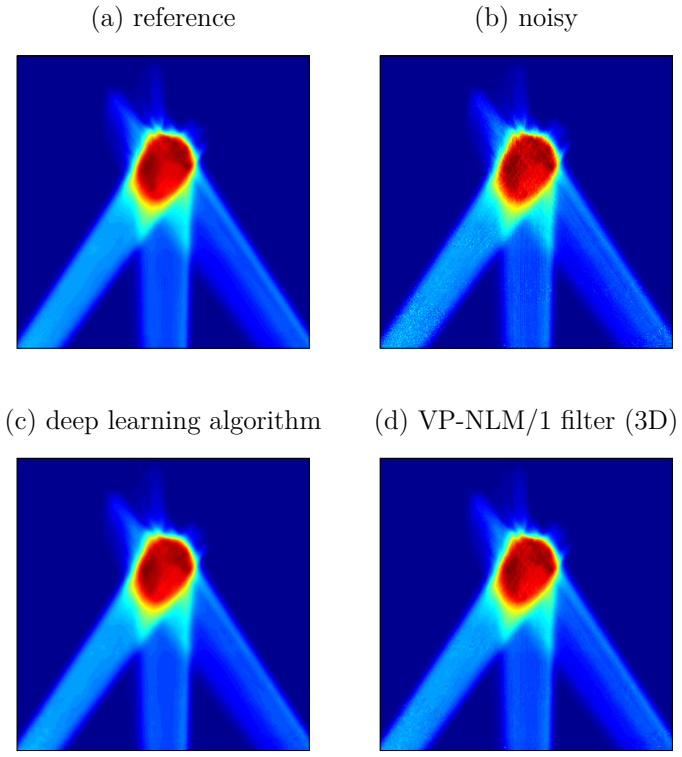


Figure 5.3: Visual inspection of a slice of the dose map: (a) reference obtained with 10^9 particles, (b) noisy dose map obtained with 10^7 particles, (c) filtered dose map obtained the deep learning method and (d) filtered dose map obtained using VP-NLM/1 (3D) with a $3 \times 3 \times 3$ window, $nK = 100$ and $h = 2.5$.

Accuracy criteria introduced in Section 3.2 are presented in Table 5.1. The VP-NLM filter has been tested with two different window sizes: in case (i), a $3 \times 3 \times 3$ window is used, while in case (ii) we use a $5 \times 5 \times 5$ window. Case (ii) yields obviously better results, but also requires a much larger computation time. Hence the user itself must choose between speed and precision.

Based on Table 5.1, we see that DLA outperforms the VP-NLM filter: the computation time with DLA is significantly smaller than with VP-NLM. The MSE associated with DLA is smaller than the one associated with VP-NLM. However better results are obtained with VP-NLM regarding the norm of the DVH difference.

It is worth noting that the deep learning tool used here is part of a library highly optimized to reduce the computation time. Additionally the deep learning denoising process has been performed on a very powerful graphics processing unit (GPU), i.e., the Geforce Titan X. On the other hand, the tested VP-NLM filter is a prototype currently implemented in MATLAB; it could however be easily rewritten

	Timing $[s]$	MSE	EPI	$\|\Delta\text{DVH}\|_2$
noisy	/	1.161	0.243	73
DLA	43	0.088	0.881	30
VP-NLM (i)	870	0.371	0.409	17
VP-NLM (ii)	1818	0.149	0.615	12

Table 5.1: Comparative values for DLA and VP-NLM filters with (i) a $3 \times 3 \times 3$ window, $nK = 100$ and $h = 2.5$ and (ii) a $5 \times 5 \times 5$ window, $nK = 100$ and $h = 2.5$ ($512 \times 512 \times 148$ dose map, lung patient).

and optimized for GPU computing. Hence a comparison between the computation times makes little sense here and should not be taken into account to assess the performances of the methods. Timing is presented for information purposes only.

This brief analysis shows that deep learning algorithms are very promising to reduce the noise in dose distributions. However this method presents some disadvantages, compared to the filters discussed in this work. Firstly, as mentioned earlier, the current implementation requires specific data format, which is constraining.

Secondly a very large amount of data is needed to train the algorithm: the more data we have, the more efficient the denoising process will be. One might notice that this reasoning also applies to the VP-NLM filter: the more similar patterns there are in an image, the more efficient the denoising process is. The difference is that, with DLA, the data are provided in advance in order to train the network and they come from plenty of different patients, hence making the dictionary of similar patterns larger. This also implies that if an unusual pattern is observed in the dose distribution, the DLA will fail to denoise this pattern.

Perspectives

We have seen that the VP-NLM method produces the best results, compared to the filters studied in Chapter 3. However the computation time associated with this filter is still large; further work regarding the acceleration of the method may hence be interesting. The method might be accelerated by further parallelizing the algorithm and/or by using GPU computing. Another idea would be to pursue research regarding the VP-NLM/2 method, which tries to use the likely symmetry of the relation « is a neighbor of », and to make the method less sensitive to the initial pixel ordering.

Finally, it has been shown in the previous section that deep learning algorithms have a great potential for denoising. These algorithms should surely be further investigated and tested on a larger set of test cases.

List of Acronyms

DLA	deep learning algorithm
DVH	dose-volume histogram
DWT	discrete wavelet transform
EPI	edge preservation index
MC	Monte Carlo
MSE	mean squared error
NN	Nearest Neighbor
PBS	pencil beam scanning
VP-NLM	vantage point tree based non-local means
WTD	wavelet threshold denoising

List of Symbols

D_{05}	refers to the minimal dose value that 5% of the volume receives.
D_{95}	refers to the minimal dose value that 95% of the volume receives.
K	is the number of neighbors, used in the context of non-local means algorithm.
N	is the number of protons simulated with the MC method (per batch).
$\|\Delta\mathbf{DVH}\|_2$	is the L_2-norm of the difference between the reference and filtered DVHs.
σ_X^2	is the variance of the dose X in a voxel.
θ	is the percentage of the wavelet coefficient maximum, used in the context of wavelet threshold denoising.
f	is the half width of the similarity (and selection) window, used in the context of non-local means algorithm.
h	is a filtering parameter, used in the context of non-local means algorithm.
n_s	is the number of simulations or batches.
s_X	is the standard error of the sample mean.
$s_{X,95}$	is the 95th percentile of the standard error of the sample mean.

List of Figures

List of Tables

Bibliography

[1] H. Paganetti. *Proton Therapy Physics*. CRC Press, 2013. 704 pp. ISBN: 978-1-4398-3645-3.

[2] K. Souris. *MCsquare: fast monte carlo simulations for proton therapy*. URL: http://www.openmcsquare.org/ (visited on 05/06/2018).

[3] K. Souris, J. A. Lee, and E. Sterpin. "Fast multipurpose Monte Carlo simulation for proton therapy using multi- and many-core CPU architectures". In: *Medical Physics* 43.4 (2016), p. 1700. DOI: 10.1118/1.4943377.

[4] S. B. Jiang, T. Pawlicki, and C.-M. Ma. "Removing the effect of statistical uncertainty on dose-volume histograms from Monte Carlo dose calculations". In: *Physics in Medicine & Biology* 45.8 (2000), p. 2151.

[5] P. J. Keall et al. "The effect of dose calculation uncertainty on the evaluation of radiotherapy plans". In: *Medical Physics* 27.3 (2000), pp. 478–484. DOI: 10.1118/1.598916.

[6] I. Kawrakow. "On the de-noising of Monte Carlo calculated dose distributions". In: *Physics in Medicine & Biology* 47.17 (2002), p. 3087. DOI: 10.1088/0031-9155/47/17/304.

[7] J. O. Deasy, M. V. Wickerhauser, and M. Picard. "Accelerating Monte Carlo simulations of radiation therapy dose distributions using wavelet threshold denoising". In: *Medical Physics* 29.10 (2002), pp. 2366–2373. DOI: 10.1118/1.1508112.

[8] D. G. Altman and J. M. Bland. "Standard deviations and standard errors". In: *BMJ : British Medical Journal* 331.7521 (2005), p. 903. ISSN: 0959-8138. DOI: 10.1136/bmj.331.7521.903.

[9] J. Seco and F. Verhaegen. *Monte Carlo Techniques in Radiation Therapy*. Taylor & Francis, 2016. 334 pp. ISBN: 978-1-4665-0794-4.

[10] F. Salvat, J. M. Fernández-Varea, and J. Sempau. "PENELOPE-2008: A code system for Monte Carlo simulation of electron and photon transport". In: *Workshop proceedings*. Vol. 4. 2009.

[11] Wang Z. and A.C. Bovik. "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures". In: *IEEE Signal Processing Magazine* 26.1 (2009), pp. 98–117. DOI: 10.1109/MSP.2008.930649.

[12] F. Sattar et al. "Image enhancement based on a nonlinear multiscale method". In: *IEEE transactions on image processing* 6.6 (1997), pp. 888–895.

[13] A. Marion. *Introduction to Image Processing*. Springer US, 1991. DOI: 10.1007/978-1-4899-3186-3.

[14] E. Arias-Castro and D. L. Donoho. "Does median filtering truly preserve edges better than linear filtering?" In: *The Annals of Statistics* 37.3 (2006), pp. 1172–1206. DOI: 10.1214/08-AOS604.

[15] A. Graps. “An Introduction to Wavelets”. In: *IEEE Computational Science and Engineering* 2.2 (1995), pp. 50–61. DOI: 10.1109/99.388960.

[16] I. Daubechies and W. Sweldens. “Factoring wavelet transforms into lifting steps”. In: *Journal of Fourier Analysis and Applications* 4.3 (1998), pp. 247–269. DOI: 10.1007/BF02476026.

[17] R. Bouchouareb and D. Benatia. “Comparative Study between Wavelet Thresholding Techniques (Hard, Soft and Invariant-translation) in Ultrasound Images”. In: *International Journal of Bio-Science and Bio-Technology* 6.6 (2014), p. 11. DOI: 10.14257/ijbsbt.2014.6.6.04.

[18] J. Suri, D. Wilson, and S. Laxminarayan, eds. *Handbook of Biomedical Image Analysis : Segmentation Models, Partie 1*. Springer Science & Business Media. Vol. 1. 2006. 648 pp.

[19] F. J. Anscombe. “The transformation of poisson, binomial and negative-binomial data”. In: *Biometrika* 35 (1948), pp. 246–254.

[20] H. Talbot et al. “Efficient Poisson denoising for photography”. In: IEEE, 2009, pp. 3881–3884. DOI: 10.1109/ICIP.2009.5414042.

[21] G. Peyré. “The Numerical Tours of Signal Processing”. In: *Advanced Computational Signal and Image Processing IEEE Computing in Science and Engineering* 13.4 (2011), pp. 94–97.

[22] D. L Donoho and I. M Johnstone. “Adapting to Unknown Smoothness via Wavelet Shrinkage”. In: *Journal of the American Statistical Association* 90.132 (1995), pp. 1200–1224. DOI: 10.2307/2291512.

[23] A. Buades, B. Coll, and J.-M. Morel. “Non-Local Means Denoising”. In: *Image Processing On Line* 1 (2011), pp. 208–212. DOI: 10.5201/ipol.2011.bcm_nlm.

[24] P. N. Yianilos. “Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces”. In: *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms* (1993), pp. 311–321. DOI: 10.1145/313559.313789.

[25] N. McNutt. *NickMcNutt.com*. 2017. URL: https://nickmcnutt.com (visited on 04/04/2018).

[26] M. D. Collins et al. “Spectral clustering with a convex regularizer on millions of images”. In: *European Conference on Computer Vision*. Springer, 2014, pp. 282–298.

[27] H. Bhujle and S. Chaudhuri. “Novel Speed-Up Strategies for Non-Local Means Denoising With Patch and Edge Patch Based Dictionaries”. In: *IEEE Transactions on Image Processing* 23.1 (2014), pp. 356–365. DOI: 10.1109/TIP.2013.2290871.

[28] M. Fisz. “The limiting distribution of a function of two independent random variables and its statistical application”. In: *Colloquium Mathematicae* 3.2 (1955), pp. 138–146. URL: http://eudml.org/doc/209997.

[29] J. Xie, L. Xu, and E. Chen. “Image Denoising and Inpainting with Deep Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Nevada: Curran Associates Inc., 2012, pp. 341–349. URL: http://dl.acm.org/citation.cfm?id=2999134.2999173.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**