

École polytechnique de Louvain

Anatomy Tracking by Fluoroscopic Image Processing combined with 3D Finite Element Motion Model

Authors: **Aurore CAPPELLARI, Brieuc EVERARD**

Supervisor: **Benoît MACQ**

Readers: **Damien DASNOY-SUMELL, Vincent LEGAT**

Academic year 2020–2021

Master [120] in Electrical Engineering

Master [120] in Mechanical Engineering

Acknowledgements

We would first like to thank our supervisor Benoit Macq, who offered us the opportunity to work on an interesting and innovative topic. A subject whose approach and results could contribute to the improvement of the treatment of mobile tumors by proton therapy.

Secondly, we express our gratitude to Damien-Dasnoy Summel, who has been available throughout the year. His regular follow-up of our work, as well as his precious advice, were essential in the realization of this master thesis.

We would also like to thank Estelle Löyen, who kindly accepted to read some passages of this work in the most crucial times, and Michel Henry, who also helped us to reread the theory of the finite elements.

I, Aurore, would like to personally thank my parents who allowed me to get here and who supported me until the end despite the difficult times. I also thank Nicolas, who was able to give me back courage in certain more complicated moments and who always managed to make me let go when necessary. I am also grateful for the support and good humor that my roommates and friends, Noémie, Julie, Milèna and Gladys have given me throughout this work. Finally, I would particularly like to thank Brieuc, my friend, for his support, his work and the good times we have spent in the context of this thesis.

I, Brieuc, would also like to thank my parents for their constant support and for offering their help on many occasions. I would like to give a special thanks to Simon who has always been there to encourage me and who makes sure that I feel at my best every day. And finally, I would like to thank my friend Aurore, with whom I did this work. We supported and encouraged each other. Throughout the year, we collaborated fruitfully to produce this work of which we are very proud.

Abstract

In 2020, nearly 3 million deaths were caused by lung or liver cancer [1]. These cancers are particularly difficult to treat with X-ray treatment, also called radiotherapy. First because the tumor is positioned near vital organs, and secondly because the position of the tumor and its shape may vary during treatment due to chest movements, i.e. breathing. This type of tumor is called a mobile tumor. To cope with this, safety margins, that is a volume irradiated greater than the tumor volume, are considered. This ensure that the tumor will always be reached. However, traditional radiotherapy does not make it possible to target a precise 3D volume, the X-rays will irradiate everything in their path, before and after the tumor. Through proton therapy, volume can be targeted more precisely. Nevertheless, this type of beam involves more energy and, in the case of a mobile tumor, can lead to significant injuries if the radiating beam is not perfectly synchronized with the movement of the tumor. This results in futile irradiation of the surrounding tissue, in this cases lungs or liver, which can be quite dangerous. Thus, it appears to be crucial to have precise information concerning the exact position of the tumor in real-time during the treatment. In this work, we tested a new method that could potentially meet this need. This method combines two main parts: the determination of a metric from fluoroscopic images which gives information on the patient's respiratory state, and the construction of a patient-specific finite element model representing a tumor in a liver for several positions. The idea is as follows: based on the metric calculated on the images available in real time (fluoroscopy), the closest corresponding 3D model is chosen to represent the current respiratory state of the patient. This latter gives precise information on the position and shape of the tumor. The obtained results are encouraging since they show the possibility of reducing the tumor margins, which are currently necessary to be sure of irradiating the mobile tumor whatever the respiratory state, and thus save more healthy tissues. To verify the accuracy of the method we used DICE coefficients indicating a level of region similarity between the ground truth tumor volume, and the simulated tumor volume chosen based on metric. The average DICE values calculated for the tumor region were: 0.8176 ± 0.0547 , 0.8755 ± 0.0333 and 0.7906 ± 0.0446 respectively for patients 1, 2 and 3, over about 200 iterations for each of them.

Contents

1	Context	6
1.1	Cancer in General	6
1.2	Radiotherapy	7
1.2.1	Photon vs Proton Therapy	7
1.2.2	Proton Therapy : Methods	8
1.3	Challenges in Protontherapy	10
1.3.1	Mobile tumors	10
1.3.2	Imaging	11
1.4	Objectives of this Work	12
2	State of the Art	13
2.1	Tracking Tumor Position in X-ray Images	13
2.2	Finite Elements for Mobile Tumors	14
3	Background Knowledge	15
3.1	Medical Imaging	15
3.1.1	X-rays and Digital Radiography	15
3.1.2	Computed Tomography	17
3.1.3	Fluoroscopy	18
3.1.4	Noise in X-ray Imaging	19
3.2	Finite Element Method	21
3.2.1	General Principle	21
3.2.2	Shape Functions	22
3.2.3	Strong and Weak Formulations	22
3.2.4	Construction of the Numerical System	23
3.3	Biomechanical Model of the Liver	23
4	Materials	27
4.1	Data and Respiratory Cycle	27
4.2	Conventions	28
4.3	Projections	29

5	Method	31
5.1	Prepare Reference Metrics	32
5.1.1	Data Preprocessing	32
5.1.2	Segmentation	35
5.1.3	Pixels Count (metric)	36
5.2	Prepare 3D Reconstructed Images	37
5.2.1	Mesh Construction	37
5.2.2	Deformation Fields Extraction	42
5.2.3	Finite Elements Simulation	42
5.2.4	3D Images Reconstruction	51
5.3	Process Aiming to be Used in Real-Time	53
5.3.1	From Fluoroscopic Images to the Phase Determination	53
5.3.2	3D Image Selection Strategy	54
6	Validation Results and Discussion	57
6.1	Tracking the Respiratory State	57
6.2	Noise Robustness	61
6.3	Finite Element Model	64
6.4	Interpolation and Extrapolation Method	70
6.5	Complete Method	72
6.6	Real-Time Validation	80
7	Limitations and Improvements	81
8	Conclusion	83
	Appendices	92
A	Shape Functions	92
B	Trilinear Interpolation	95

Acronyms

- BC : Boundary Conditions
- COM : Center of Mass
- COME : Center of Mass Error
- CS : Continuous Sequence
- CT : Computed Tomography
- DF : Deformation Field
- FE : Finite Element
- FEA : Finite Element Analysis
- GT : Ground Truth
- GTV : Gross Target Volume
- HU : Hounsfield Unit
- LE : Linear Elastic
- MC : Marching Cubes
- MidPCT : Mid-Position Computed Tomography
- MR : Mooney-Rivlin
- NH : Neo-Hookean
- ROI : Region of Interest

1 Context

1.1 Cancer in General

Cancer is now one of the leading causes of death in the world: it is estimated that around 10 million people died from cancer in 2018. This is particularly true in Western Europe and North America. In countries with a high HDI (Human Development Index)¹, with the evolution of medicine, the number of deaths for vascular diseases as well as for infectious diseases (vaccines, antibiotics, ...) has been drastically reduced. This, together with the improvement in the standard of living, has led to an increase of life expectancy while the birth rate has fallen. The proportion of older people who are more likely to develop cancer has therefore increased [2]. On the following figures 1.1 and 1.2, a map of the world HDI and a map of cancer ranking as premature death (30-69 y.o.). One can see a clear link between these two maps.

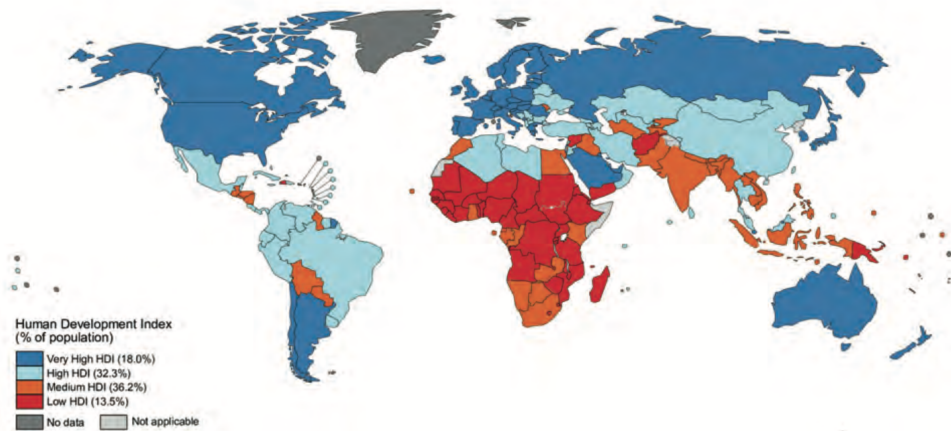


Figure 1.1: World HDI [2]

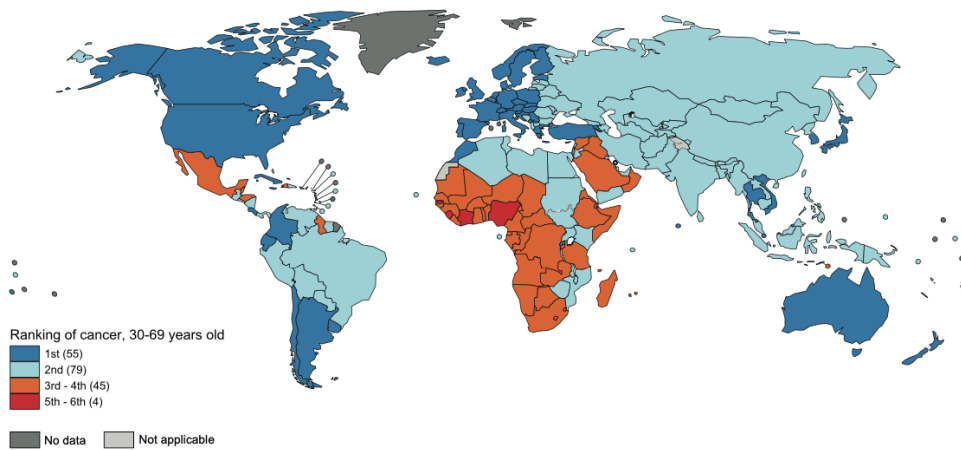


Figure 1.2: World Cancer Ranking for premature death cause [2]

¹This index aims to quantify the relative development of populations. It is based on three criteria : life expectancy, standard of living and average years of education

In Belgium, cancer represents 25% of the total deaths and is also the second cause of death following vascular diseases (25.9%). Lung cancer is the second most common cancer for both men (15.2%) and women (9.4%).

The proportion of lung cancer is therefore high. Together with liver cancer, they are the most affected by the phenomenon of mobile tumors (section 1.3.1) and represent around 14% of cancers in Belgium. It is therefore imperative to find a way to treat them as accurately as possible.

1.2 Radiotherapy

Radiotherapy is considered as being one of the most effective ways to treat cancer, right after the surgery method [3]. In the case of non-metastatic tumors, radiations appear to be a good solution since it allows to target a specific region. Thanks to their radioactive characteristic the particles damage cancer cells by breaking their DNA chains. This damage prevents the tumor from reproducing and spreading. Once they are damaged, cells die and are finally removed by the body. This method does not give results right away; it takes days or sometimes weeks of treatment before cancer cells die [4]. The drawback of it is that surrounding healthy tissue can also be attacked.

1.2.1 Photon vs Proton Therapy

In conventional therapy, photon beams are mostly used to treat cancer, but it actually exists another type of radiation: proton. Protons travel through body quickly and stop abruptly when reaching tissues at a very specific depth where they will deposit a high dose of radiation, unlike photons, which deposit their radiation doses close to their entrance into the body. Thereby, protons are more likely to preserve healthy tissues surrounding a localized tumor. This aspect can be explained with the Bragg curve. This curve plots the energy loss of ionizing radiation during its travel through matter and is illustrated in figure 1.3. One can notice that the energy of protons remains quite stable until a point where the peak occurs. After this peak, the energy drops steeply which means that radiation does not reach tissues behind this peak. It is this aspect which can be useful to spare healthy tissues. Indeed the penetration depth of a proton beam, where the Bragg peak occurs, can be adjusted with the speed at which the particle is accelerated and then depends on the cyclotron (or synchrotron) [5]. Protons are much more sensitive to tissue density than photons are. That is why we can control the penetration depth. Unfortunately, it also leads to important issues if the user does not know precisely the tumor position, and so densities the beam will pass through. Indeed any change in tissue composition, such as organ motion, lung expansion, or alteration in bone position during the treatment can lead to bad target coverage. In this case, a region which does not include the tumor would be irradiated (healthy tissues), and the tumor would not receive any radiations [6].

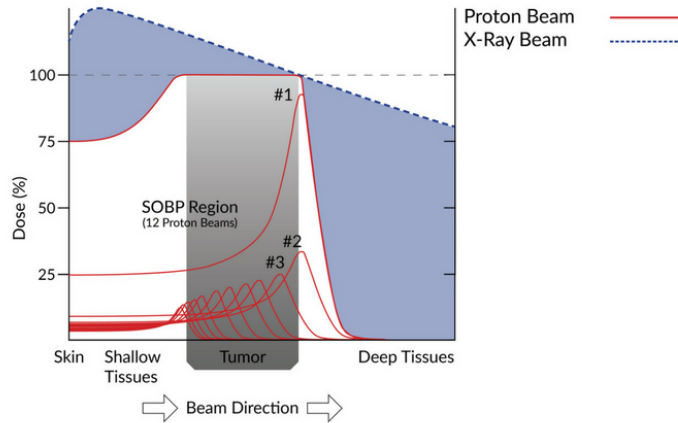


Figure 1.3: Dose produced by a native proton beam, by a modified proton beam passing through tissue, compared to the absorption of a photon or X-rays beam. Credit : [7]

Another issue concerning proton therapy is its cost. The cost of proton therapy can be significantly higher than that of traditional radiotherapy. If the side effects of proton radiation therapy are significantly less, then the increased cost of proton radiation therapy could be worth the benefit of decreased complications and organ injury. Unfortunately, there are not enough clinical trials to prove these benefits. Then it becomes difficult to convince hospitals to invest in this technology, or insurance to reimburse for that. Some researches have been undergone to compare side effects [8] caused by proton and photon therapies, it appeared that patients treated with proton therapy were much less likely to experience severe side effects than patients treated with radiotherapy [9]. Nevertheless, as it is detailed in [9] those experiments are limited and results cannot be generalized for now.

1.2.2 Proton Therapy : Methods

There actually exist two main methods in proton therapy: passive scattering (PS) which is the oldest one, and pencil beam scanning (PBS) or active scanning.

Passive Scattering

This technique aims to generate a broad beam by using passive scatterers to increase the lateral spread of a narrow proton beam. After that, the enlarged beam is passing through a collimator and a compensator in order to shape as close as possible the target volume. In order to achieve a distribution that is a spread out Bragg peak, a spinning range modulator wheel is added in order to tune the beam energy. The different energies are not all delivered simultaneously, but all of the energies are delivered in a single cycle of the range modulator wheel [10]. This technique is illustrated in figure 1.4.

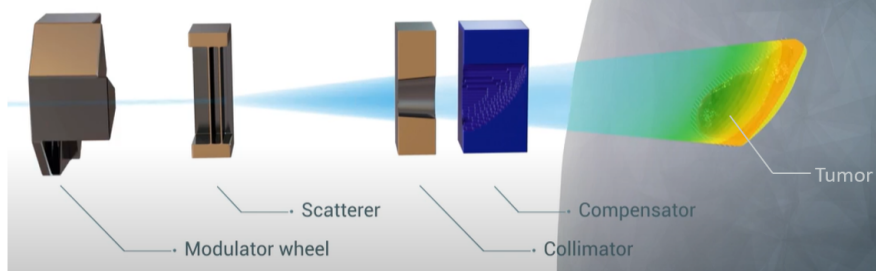


Figure 1.4: Illustration of Passive Scattering technique, capture from [11]

Pencil Beam Scanning

Pencil beam scanning, or active scanning, is a much more recent method which is now widely used. In this case, the dose can be more precisely directed to the target volume. Protons being charged particles with a significant mass their direction can be controlled by the use of two magnetic fields. If we add to this the ability to vary the protons acceleration as explained in section 1.2.1, thus the penetration range of protons, we are then able to position the pencil beam at any spots in the target-volume (tumor). This results in a fully three-dimensional control for the dose deposition inside the patient, as illustrated in figure 1.5. This method helps sparing healthy tissue surrounding the tumor, even more than passive scattering, moreover it also reduces the need for hardware, for these reasons the active scanning is usually preferred to passive scattering. More details about this method can be found in [12].

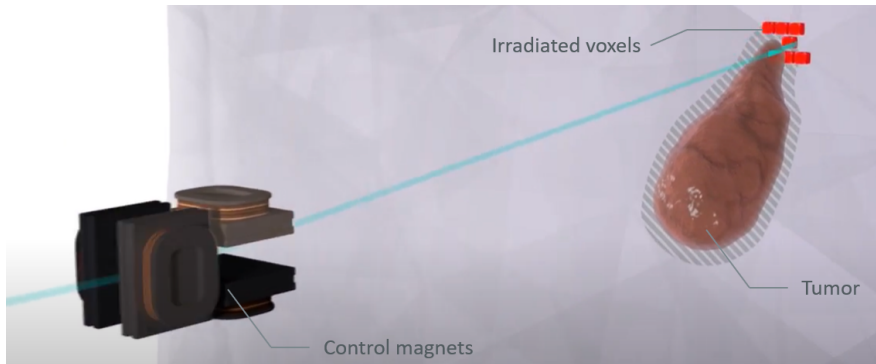


Figure 1.5: Illustration of Active Scanning technique (or PBS), capture from [11]

1.3 Challenges in Protontherapy

Now that we understand the protons behavior, their benefits, and how it is used for proton therapy (passive scattering or PBS) it is important to analyze different challenges linked to this technique.

1.3.1 Mobile tumors

The proton penetration depth can be adjusted so that it will irradiate the tumor region. We already mentioned the importance to know precisely the tumor position before the treatment in section 1.2.1. The main issue concerns what is called mobile tumors. Indeed mobile tumors are susceptible to move during the treatment and then leave the irradiated region which is quite narrow in proton therapy. Most of the time, they are localized in the cardio-thoracic region. In this region, organs are vital (heart, lungs) it is then fundamental to not irradiate them. Moreover, these organs imply important movements, mostly due to respiration, which makes the tumors moving. One can observe the three-dimensional motion (in cm) of a lung tumor in figure 1.6. To ensure the total coverage of the tumor, user should add important margins (a maximum which is more than 2 cm for the vertical movement). This would lead to too important damages for organs like lungs or heart situated in the cardio-thoracic region.

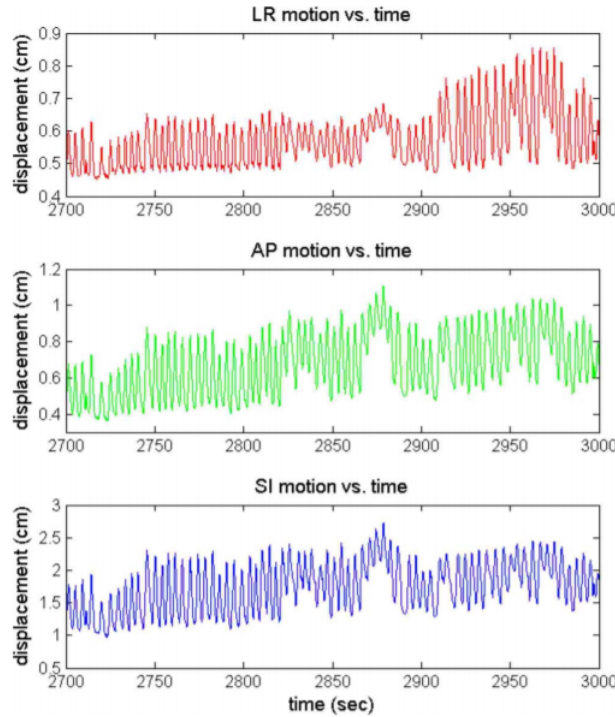


Figure 1.6: Observed variation in lung tumor motion with time in three dimensions (LR: left-right, AP: anterior-posterior, SI: superior-inferior) from Cyberknife Synchrony data (source data : [13]). Notable is that there is significant motion in three dimensions, there are variations in the baseline position of the tumor in all three dimensions and variations in the cycle-to-cycle period, respiratory cycle shape, and range of motion (source graph : [14]).

These movements occur during the treatment, some techniques exist to minimize the organs movements, such as abdominal compression or respiratory gating [15], but those cannot be totally avoided. Then it becomes important to be able to anticipate or detect those movements in real-time if we want to adjust the beam as to spare vital organs. This issue constitutes the first challenge of proton therapy : tumor tracking in real-time.

1.3.2 Imaging

Now that we understand the importance to know precisely the position of the tumor during the treatment, it is interesting to discuss the information that is currently available in real-time. What we would like to achieve is a motion-adaptive proton therapy system. To do so some characteristics are needed: the system should have real-time performance sufficient to detect any movements (eg. respiration) and it would have a large field of view in order to localize the region of interest surrounding body structures. Sonja Dieterich et al. in [14] built a table comparing some imaging methods for real-time tumor localization in the scope of radiotherapy (see table 1). Some of these techniques are unfortunately not available with proton therapy machines, like MRI (Magnetic Resonance Imaging) for example. What can be used together with proton therapy is fluoroscopy. Proton beam therapy systems can be installed in conjunction with X-ray systems for patient positioning. These systems are usually operated in fluoroscopy mode. In fluoroscopy mode, X-ray images of the patient are obtained with a low dosage and high acquisition rates, allowing for analysis of organ motion and accurate identification of the breathing phase. In this mode, tumors can not be directly detected due to the lack of contrast [16]. The images we use in our study are based on fluoroscopy projections (see section 4.3).

Localization method	Invasive?	Availability	Imaging Frequency
Electromagnetic [17, 18, 19]	Yes	Commercially available for prostate	10 Hz
Fluoroscopic [20, 21, 22, 23, 24]	Yes	Translated to patient treatments	30 Hz
Fluoroscopic/optical [25, 26, 27, 28]	No ^a and yes	Commercially available	Before every beam/30 Hz
Monoscopic/optical [29, 30]	Yes	Under development	30 Hz (theoretical)
MRI [31]	No	Under development	N/A
Ultrasound [32]	No	Under development	25 Hz

Table 1: Different Imaging Methods of real-time localization

^a Lung tumor projection method (Xsight Lung) available from Accuray

Another imaging method playing an important role in our study is computed tomography (CT) imaging, detailed in section 3.1.2. CT can give very precise information concerning the tumor shape and position by taking multiple 2D slices and construct a 3D volume based on these (3DCT); unfortunately, it does not exist a possibility to use it in real-time due to its computation time. Still, CT images are necessary to plan a treatment strategy; a sequence of 3DCTs (4DCT) is constructed in order to have an idea of the tumor movements during an entire cycle of respiration. Each frame of this planning sequence is associated with a phase of respiration.

1.4 Objectives of this Work

The work presented in this paper aims to develop a method that can combine two aspects. First of all, the possibility of following mobile tumors on low contrast images such as obtained with fluoroscopy. This would eventually make it possible to use the method in real-time since fluoroscopy can be used in parallel with treatment by proton therapy. As we will see in a next section (see section 2.1), the tracking of mobile tumors by X-ray image processing presents an important drawback between the use in real time and the precision of the tracking. Secondly, it is fundamental to have as much information as possible about the position of the tumor as well as the shape it may have (indeed it may vary due to respiratory movements).

What we propose in this work is to separate the tracking into two parts: first, the analysis of 2D radiographic images with low contrast on which we would extract information on the patient's respiratory state. This analysis can be simpler than looking for the tumor directly since it can be based on more visible elements of the image (such as lung or diaphragm). So this task can be performed without requiring too many computational resources. The second part would make it possible to return a more precise 3D image based on this deduced information. To do this, several precalculations with a finite elements model must be carried out upstream, by anticipating the different possible respiratory phases of the patient. These would be based on more precise images that can be acquired before treatment. These collected images form what is called the planning 4DCT (see section 4.1). Note that CT scans, as was previously mentioned (section 1.3.2), are not available during a proton therapy treatment. The complete method will be described in more detail in section 5.1.

2 State of the Art

We have already mentioned the importance of knowing precisely where the tumor is located during treatment with proton therapy because of the movements induced by breathing. This section aims to resume which techniques have already been investigated concerning the two main domains of our method. On one hand the tracking of tumor in X-ray images, or more precisely in fluoroscopy. And on the other hand, what has already been found concerning the finite element techniques that could serve for tumor tracking.

2.1 Tracking Tumor Position in X-ray Images

X-ray images can be obtained in real-time through the use of fluoroscopy. However, these do not offer enough contrast to clearly distinguish the soft tissues from each other. Therefore, a tumor located in the lungs or liver will not be visible in such an image. This induces problems when tumor monitoring is necessary as is the case for proton therapy applied to tumors located in the thoracic cage, also called mobile tumors (see section 1.3.1).

This task of mobile tumor tracking on fluoroscopic images is widely studied. One of the most widely used techniques concerns the implantation of radiopaque fiducial markers. Indeed, the implantation of such markers near the tumor can help to better visualize its position because they are more absorbent and then present a much better contrast than the tumor itself made up of soft tissues. Nevertheless, the implantation procedure is invasive and therefore can present several risks such as pneumothorax complication which is the most common one [33, 34], or any other infection. Moreover, van der van Zyp et Al. [35] showed the possibility of such markers to present intrafractional displacements because of the instability of those when they are placed in the lungs. Even if these techniques have been much investigated and showed potential improvements during those last years, both for lung tumors [36] and liver tumors [37], it still represents an additional burden in the clinical routine [38]. For those different reasons, it is still desired to find marker-less methods to track tumors on X-ray images.

One can find various marker-less methods in the literature. Among those one can find several based on key points technique. This latter assumes that some regions in the image, linked to the tumor position, are more apparent or distinguishable from others. These regions are detected and described as *key points*. Once they are acquired they can be easily tracked in X-ray images. The movement of these points will then give information about local movements of the tumor. There exist several papers about this technique : Rottman et al. developed a soft tissue localization (STiL) which is used to calculate soft tissue displacement in 2D radiographs in real-time [39]. In this paper, the key points detection in the sequence is made by using local variance filter as a texture measure and select local maxima of this quantity. They tested their method both on a dynamic motion phantom and then on lung SBRT (Stereotactic body radiotherapy) images. They were able to show a mean track error being $\leq 1.0\text{mm}$ (they also assessed their technique in more practical situations in [40]). Nevertheless, it happens that the landmarks, or key points, are attached to a structure that is not sufficiently unique throughout the entire breathing cycle because of deformation caused by the respiration. Moreover, landmark candidates are chosen based on the texture measure but it does not consider temporal gray value gradient. Due to this, there may be candidates attached to static structures rather than the desired target. These two potential issues can lead to the degradation of tracking accuracy. Xie et al.

in [41] have developed a more robust method also based on key points for the tracking of soft tissues for X-ray imaging. In their method, they use a scale-invariant characteristic transformation (SIFT). This makes it possible to describe the key points in a feature space which is invariant to several transformations such as scaling, translation, rotation, illuminations among others. This leads to more precise tracking, unfortunately, it is quite computationally expensive. Therefore, the method cannot be used in real-time. Most recent studies still present potential improvements for the key-points based method [42]. The latter combines both Gaussian filtering and optical filter and is available in real-time. Nevertheless, it still requires some improvements to take any deformation into account.

All of these methods focus on lung tumors. In the lung, tumors have a slightly better contrast to tumors located in the liver which are completely invisible. In our case, we would like to be able to find a method that could be used on all kinds of mobile tumors, including liver tumors. Moreover, we look for a method which is non-invasive, usable in real-time and which provides precise information on the position and possible deformation of the tumor.

2.2 Finite Elements for Mobile Tumors

The purpose of this section is to present a quick state of the art on the approaches that have already been investigated concerning tumor tracking using finite element modeling. Efforts have long been made to try to manage tumor movement in the chest and abdomen.

The tracking of tumor motion by finite elements has developed a lot with the improvement of computer performances. The finite element method allows to take into account physical constraints in the motion tracking problem. A model is chosen, which characterizes the relationship between deformations and constraints, and boundary conditions are defined, i.e. the displacements that we impose on the boundaries of our organ. The finite elements will calculate in a discrete way the displacements inside the organ taking into account these biomechanical properties and these boundary conditions.

Much research has been done on the lung. It is indeed the lung that is responsible for the movement of the thoracic organs through the mechanism of the respiratory cycle. Brock et al. (2005) [43] have developed a linear elastic model with a Young's modulus $E = 5KPa$ and a Poisson ratio $\mu = 0.45$. Eom et al. (2010)[44] chose a hyperelastic model with $E = 6kPa$ and $\mu = 0.4$. Werner et al (2009)[45] studied lung tumors in 12 patients. They showed that the size and location of the tumor played a role in lung displacement. Tehrani et al. (2015)[46] compared 3 biomechanical models for lung displacement between phase 0 and phase 50 (end of expiration to end of inspiration). They determined the optimal biomechanical parameters of the tumor and lung that minimized the error between the centers of mass of ground truth and simulated tumors. They concluded that these optimal parameters varied widely between patients.

On the other hand, very little research on liver tumor tracking using finite elements has been done. Marchesseau et al. (2015)[47] reviewed the literature on the different biomechanical models of liver tissue. Ghahfarokhi et al (2019)[48] analyzed the deformations undergone by the tumor for different positions of the tumor inside the liver. For this purpose they modeled a liver with finite elements, subdividing it into several parts. They added a spherical tumor and compared the deformation of this tumor according to the part of the liver in which it was located.

3 Background Knowledge

3.1 Medical Imaging

Medical imaging ([49, 50]) refers to techniques or processes used to image various parts of the human body. This is an essential point in the medical world since it contributes to more accurate diagnoses from the outset. Some of those techniques are² :

- Traditional X-rays radiography
- Fluoroscopy
- Magnetic Resonance Imaging (MRI)
- Computed Tomography (CT)
- Endoscopy

X-rays are the oldest and most popular imaging technique. It was discovered in 1895 and used for the first time on a human body in 1896. Now it exists several different techniques based on X-rays (like fluoroscopy and computed tomography). It is also important to mention a non-negligible issue of those techniques being its **radioactive effect**. Indeed radiations can destroy healthy cells present in the human body. For this reason, the duration of radiation exposition is restricted as most as possible. The reader can find more information about these side effects in [51].

More generally, the term *medical imaging* also involves measurements and recording techniques that do not produce images but data generally represented as graphs or maps. This section is mainly based on reference [52].

3.1.1 X-rays and Digital Radiography

Digital radiography is generally used to observe: skeletal system, oral cavity (bones and teeth), any ingested objects, lungs, etc. This imaging method is based on **ionizing radiations**. Ionizing radiations can be seen as a particle, or electromagnetic wave, carrying sufficient energy to detach electrons from atoms or molecules, thereby ionizing an atom or a molecule. In radiography, those ionizing radiations are used to image the human body being a non-uniformly composed material. An illustration of this imaging technique is given in figure 3.1. X-rays travel in straight lines at the speed of light, they cannot be deviated by magnetic fields. Once they pass through the material they are attenuated. The attenuation is a combination of two important effects: **scattering** and **absorption**. The degree of penetration depends on the kind of matter and ray energy. This property explains why we can distinguish different parts of the body: some of them are more absorbing than others. The attenuation rate, or more precisely the radiodensity [54], can be quantified according to the **Hounsfield** scale. The Hounsfield coefficient is defined as follow :

$$HU = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} \quad (1)$$

²this list is not exhaustive

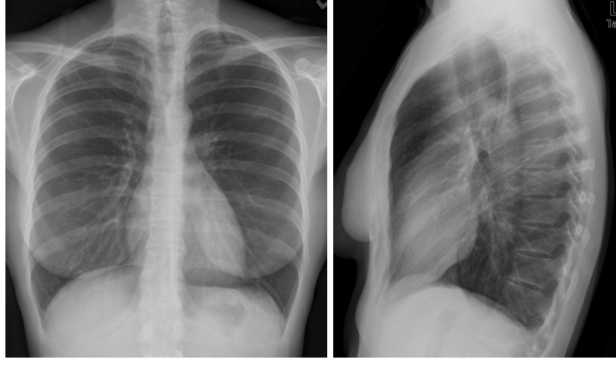


Figure 3.1: Standard frontal (left) and lateral (right) chest radiograph (credits : [53])

with μ the linear attenuation coefficient of the material, μ_{water} and μ_{air} respectively the linear attenuation coefficient of water and air. The higher the value of HU , the more absorbing is the material. Some interesting values are given in table 2.

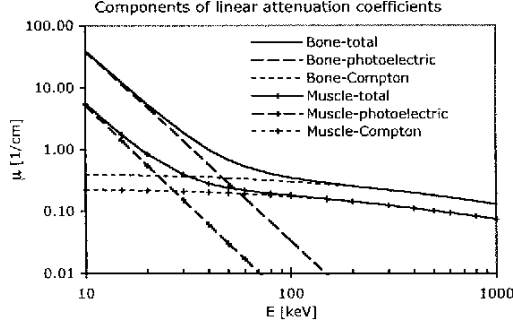
Material	HU
Air	-1000
Lung	-700 to -600
Fat	-120 to -90
Water	0
Cerebrospinal fluid	+15
Blood	+35 to +45
Muscle	+35 to +55
Liver	+60 $\pm$ 6
Soft tissue	+100 to +300
Cortical bone	+500 to +1900

Table 2: HU value for different materials [55]

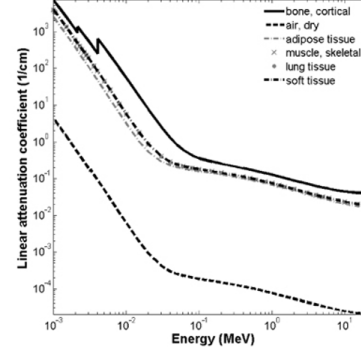
The attenuation coefficient μ can be described as a function of energy for different materials. As mentioned previously the attenuation is due to the interaction between penetrating radiations and matter, it actually corresponds to the summation of different types of interaction :

- Compton Scattering (C),
- Rayleigh Scattering (R),
- photoelectric Effect (PE),
- Pair Production (PP).

Their dominance depends both on the matter atomic number and also on the ray energy level. Some of them can cause an issue in the image called **noise** which will be discussed in a further section (section 3.1.4). Their effects are illustrated in figure 3.2.



(a) Contribution of different interaction on bones [56]



(b) attenuation coefficient for different matters [57]

Figure 3.2: Linear attenuation coefficient

The attenuation function can also be described by the **Beer-Lambert law**, also called **Beer's law**. This latter relates the attenuation of rays to the properties of the material through which the beam is travelling. In the case where the attenuation coefficient μ does not vary with the matter depth x it can be defined as follow³ :

$$I = I_0 \times e^{-\mu x} \quad (2)$$

with I_0 the initial X-ray intensity and the attenuation coefficient μ depending on other parameters :

$$\mu = 0.0006 Z^3 \lambda^3 \rho \quad (3)$$

with Z the material atomic number, λ the ray wavelength and ρ the material density.

3.1.2 Computed Tomography

Computed Tomography, also called CT scan, is an imaging technique that measures X-rays attenuation by different tissues in the body. These measurements are taken for different angles and then processed through the use of reconstruction algorithms. This produces detailed cross-sectional internal images. The first CT scanner for medical use dates from 1972. This technique is mostly used to look for tumors, pulmonary embolism, abdominal aortic aneurysms, etc. [49]. CT scan is mostly used to produce 3D images (3DCTs) based on successive 2D cross-sectional images. Those 3DCTs are 3D tables made of HU values varying depending on the material(s). Different views of CT slices and a 3D rendering are illustrated on figure 3.4.

The machine used for this imaging technique, the CT scanner, has evolved a lot during past years. Today we can count 8 different generations, the third one being the most commonly used. This latter is composed of a wide fan beam and a large array of detectors as illustrated in figure 3.3c. The beam and detectors array execute a 360° rotation around the target. This gives the cross-sectional images, or slices.

³for monochromatic source

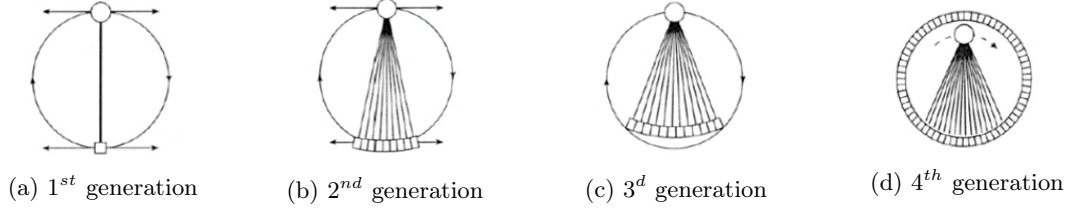


Figure 3.3: First generations of CT scanner

More specific and evolved machines have been built since the 4th generation⁴ :

- Electron Beam CT (or ultra fast CT) [58]
- Helical CT [59]
- Multidetector CT (MDCT) [60]
- Combination of MDCT and Helical CT
- Cone Beam CT (mostly used for dental application) [61]

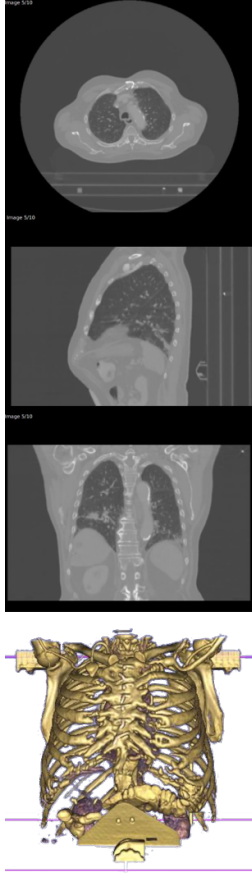


Figure 3.4: Respectively from top to bottom : 3 different views of CT slices and a 3D rendering

Computed tomography provides very accurate information but is time consuming and requires a fairly high dose of radiation compared to conventional radiography or fluoroscopy. This is why it cannot be used in real-time during a proton therapy treatment. In the case of this work, 3DCT sequences are used to validate our results as ground truth data because they contain more information than fluoroscopic projections (section 3.1.3): like tumor contours. However, since fluoroscopy is a technique available in real time during proton therapy treatment, it is on this type of data that our model will have to operate. The images used as input data for our model must therefore be fluoroscopic projections. These will be simulated on the basis of ground truth 3DCTs (see section 4.3) to be sure to compare two identical respiratory states when validating our results.

3.1.3 Fluoroscopy

Fluoroscopy uses X-rays to visualize the internal structure in real-time. The provided images are similar to those produced by conventional/digital radiography (see figure 3.1) but there are still two main differences between those two techniques [62] :

⁴This is not exhaustive

1. Conventional/Digital radiography provides static images of the inside of the body in still form. Fluoroscopy imaging provides images in a live video format, which will show movement in the body or any object. The different things that these imaging technologies can accomplish are mainly related to this difference.
2. Conventional/Digital radiography carry a smaller risk of radiation-related risks than fluoroscopy does. This is due to the fact that fluoroscope must irradiate a part of the body over the whole treatment or diagnosis. Sometimes this is a brief moment like for diagnosis. In other cases, the patient is exposed to radiation for the duration of surgery, or treatment (eg. proton therapy irradiation).

The first point just described is the main reason why this technology will be fundamental in this work context. Indeed, the tumor tracking has to be realized in real-time. Fluoroscopy can produce continuous series of images at a rate of 25-30 complete images per second (eg. like conventional television or video) [63]. For this reason, fluoroscopic images will serve as input to our model to simulate a realistic context.

3.1.4 Noise in X-ray Imaging

The noise aspect is fundamental when considering building an image processing algorithm. Indeed an algorithm could show very good results on noiseless images but totally fail when facing noise impact.

Noise can be seen as random variation of brightness in images. This has generally no particular pattern. In most of cases, it reduces image quality, especially for smaller or low-contrasted objects, resulting in textured or grainy images. All medical images contain noise. In X-ray imaging, one can find different types of noise like Gaussian noise, Salt and Pepper noise and Poisson noise. The reader can find illustrations in figure 3.5 and brief descriptions in table 3. Although noise gives an image a generally undesirable appearance, the most important issue is that noise can cover and reduce the visibility of certain features within the image, or make an image processing algorithm (eg. segmentation) fails, and then lead to bad diagnosis in medical context.

Type of noise	Brief description
Gaussian noise	Additive normal noise
Salt & Pepper	Corrupted black or white pixels
Poisson noise	Poisson transformation of pixels
Gaussian & Poisson	Poisson followed by Gaussian

Table 3: Brief description of popular noise types

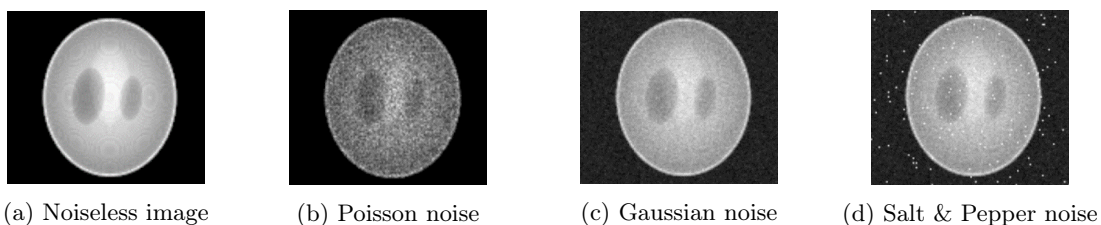


Figure 3.5: Different type of noise simulated with Tomopy library

According to [64], [65], [66] and [67] Poisson noise, or quantum noise, appears to be the most predominant

type of noise in X-ray imaging. This latter is due to the non-uniform distribution of photons. Indeed the number of photons reaching the detectors can fluctuate from point to point, this fluctuation is characterized by a random process similar to the Poisson distribution. This means that even if detectors are exposed in the absence of objects on the path, this would results in a grainy image rather than a uniform grey image. This effect is even more present in fluoroscopy due to the use of low-radiation doses as illustrated in figure 3.6. An underexposed image will present more quantum noise but with less radiation dose. On the contrary, overexposed image results in a good quality image but high radiation dose. It is then fundamental to find the balance leading to an acceptable quantity of noise (enough to be able to establish a correct diagnosis), and an acceptable radiation dose (see [68]).

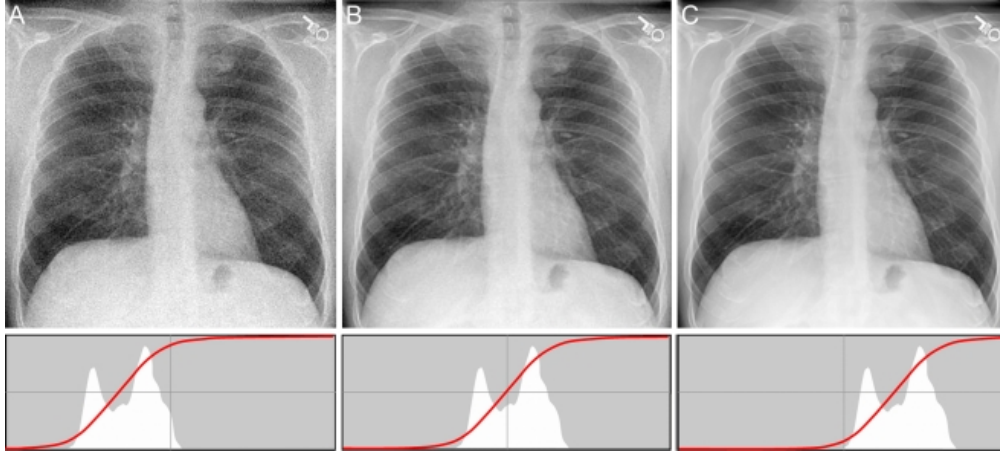


Figure 3.6: Response of a digital detector to exposure intensity variations. A) Underexposure. B) Correct exposure. C) Overexposure. Underneath each image is a histogram representing the grey level distribution of the image (directly related to intensity of X-rays transmitted through the patient). The x-axis represents digital value and the y-axis represents the number of occurrences. (Credits to [68])

Computed tomography (CT), radiography and fluoroscopy imaging systems all use X radiations to capture images, that is why Poisson noise appears to be the most occurring noise in resulting images. For this reason, many papers about its removal exist (see section *Denoising methods on CT and X-Ray images* in [66]).

Articles [69] and [70] mention the possibility to face both Poisson and Gaussian noises. Gaussian noise is also called electronic noise because it occurs in amplifiers, detectors, or generally speaking in the electronic circuit. The mixture of first Poisson noise and then Gaussian noise during the process of recording and writing the image on the device can be described as follow [69] :

$$P = Poisson(\Upsilon) + \eta \quad (4)$$

with P the observed signal, Υ the noiseless signal, the additive term η representing the Gaussian noise, and the Poisson noise represented as a function $Poisson$. The bit-wise $Poisson$ function actually represents the application of Poisson distribution with a mean value equal to the pixel value. This is also called multiplicative noise, in contrast with additive noise like Gaussian type is. As a reminder the Poisson distribution is given by :

$$F(x) = \frac{e^{-\lambda} \lambda^x}{x!} \quad (5)$$

with λ being the mean which in this context corresponds to the pixel value and then is equal to x in the above equation. Several papers like [71] and [72] present some techniques to model possible noise or combination of multiple noises with more precision.

3.2 Finite Element Method

This section attempts to explain the finite element method in general terms. The keys are given to understand the chosen approach of this master thesis. This section is mainly based on [73] and [74].

The finite element method is a technique for solving partial differential equations (PDE), which models numerous physical problems such as heat transfer, material resistance, acoustics, etc..

To solve the PDE, the complex domain is divided into smaller and simpler pieces called finite elements, on which we seek to solve the weak formulation of the PDE. This process is called discretization and gives a mesh, which is a set of finite elements, connected to each other, approximating the geometry of the domain. The mesh allows to interpolating the solution, using shape functions (section 3.2.2) which leads to a problem simplification.

Since we seek to solve physical problems, boundary conditions are prescribed. In order to take them into account, they can be strongly imposed while building the linear system, in the form of values on the nodes of the mesh boundary. The finite element method simplifies the physical problem into a linear system to solve. A balance between the accuracy and the computational cost is essential to keep the method efficient.

3.2.1 General Principle

Let us consider a complex domain Ω with boundary $\delta\Omega$. We want to find the function u defined on Ω which is a solution of the PDE with defined boundary conditions. For example, here our PDE is a material's resistance problem: we have an elasticity model with defined material properties. We want to find the displacement of the liver (u) on the entire domain, knowing the displacement of the outside wall, which represents our boundary condition.

As we do not know the exact solution of our elasticity model with boundary conditions on such a complex domain as a liver, we will discretize the domain in order to separate it into several small simple elements. This discretization will allow to interpolate the function u on the elements knowing the values at the vertices or nodes. In our case, the elements are tetrahedrons, the simplest polyhedron. The interpolation is obtained by using **shape functions** (section 3.2.2). The interpolation function u^h is expressed as follows :

$$u^h(X_i) = \sum_{i=1}^n U_i \tau_i(X_i) \quad (6)$$

where X_i are the coordinates of the point, U_i are the nodal values, τ_i are the global shapes functions. If

X_i is a node then :

$$u^h(X_i) = U_i \quad (7)$$

3.2.2 Shape Functions

The shape function is the function which interpolates the solution function u between the discrete values obtained at the mesh vertices or nodes. The difference between a node and a vertex is the following: vertices are the geometrical vertices of the element at the intersection of two elements while nodes are points where a nodal value U_i is defined, inside the element.

Shape functions are functions associated to each node, which have the following property: it is 1 at the associated node and 0 at the other nodes. The expression of the functions will depend on the number of nodes and their coordinates. Examples of shape functions are developed in Appendix A for dimension one, to better understand the concept and for dimension 3 on a tetrahedron for our benefit ⁵.

Due to the complex mesh geometry, the elements have different dimensions and volumes. The shape functions will therefore be developed on a reference template called, parent element $\hat{\Omega}$. And any element Ω_e can be found using an isomorphism, a mapping. An example is provided in the Appendix A.

3.2.3 Strong and Weak Formulations

Now that we know how to approximate the solution function u with nodal values, let us introduce another concept: the formulation. Physical problems can be expressed in different forms: the strong formulation, and the weak formulation.⁶

The **strong formulation** is just the expression of the physical law in local form.

The **weak formulation** is used to relax the *strong* conditions of the PDE solution. The weak formulation is an integral formulation. It is derived from the conservation laws applied to the control volumes. It is a formulation widely applied in physics. It is logical to solve this formulation. Weak form is used in the finite element method because we can derive the discrete formulation: it allows to solve the PDE at the nodes of the discretized domain by the shape functions interpolation property

To switch from strong to weak form, let us consider a differentiable operator R and f a function defined on a domain Ω . We want to find u defined on Ω that verifies :

$$R(u) = f \quad (8)$$

⁵It should be noted that the shape functions of the same problem may differ depending on the user. The restriction of being 1 at one node and 0 at the other nodes can give a wide range of possibilities, and this is especially true for dimensions greater than 1. Some examples of shape functions for 1, 2 and 3 dimensional elements are shown here : [75]

⁶Here we will give intuitive definitions of strong and weak formulations rather than more rigorous mathematical explanations that are much less understandable without the necessary notions. If you are interested in this subject and want to go deeper into it, you can consult : [73]

anywhere in Ω . We know that u , solution of (8) is also a solution of the following problem :

$$\int_{\Omega} R(u)v = \int_{\Omega} f v \quad \forall v \in V \quad (9)$$

Where v is a test function. Then, taking into account boundary conditions, we perform operations on this last expression (such as integration by parts, green's theorem, ...) to obtain a scalar product.

3.2.4 Construction of the Numerical System

Then by using the shape functions, i.e. by using the approximation of equation (6), we can obtain the linear system :

$$\sum_{j=1}^n A_{ij} U_j - B_i = 0 \quad (10)$$

Where U_j are the nodal values of the solution function u . The A_{ij} matrix is determined with the integration of the known shape functions, according to the expression of the weak formulation. The B_i matrix is obtained also with the shape functions and the expression of the function f . And boundary conditions are applied by assigning to the nodes of the boundary U_j the values defined in the problem description. Thus what we will obtain is a discrete solution, a set of values U_j , local solutions of the weak formulation at the nodes of the finite element mesh.

3.3 Biomechanical Model of the Liver

Now that the theoretical basis of the finite elements has been given, let us look at the equations that govern the mechanical behavior of the liver. Liver tissue, like all other tissues, is considered as a mechanical material with its own mechanical properties. To understand correctly the behavior of liver tissue, it is necessary to present an essential tool in mechanics of materials : the stress-strain curve.

This curve is used to characterize the deformation of a material undergoing forces. It relates stress σ and strain ϵ , with :

$$\sigma = \frac{F}{A_0} [Pa] \quad and \quad \epsilon = \frac{\Delta L}{L_0} [-] \quad (11)$$

where F is the force applied to the material $[N]$, A_0 is the initial cross section area $[m^2]$, ΔL is the change in material length $[m]$, and L_0 , the initial length of the material $[m]$.

Here an example of the shape of the stress strain curves for different class of materials. In this case, the curves have the shape of a typical uniaxial tensile test.

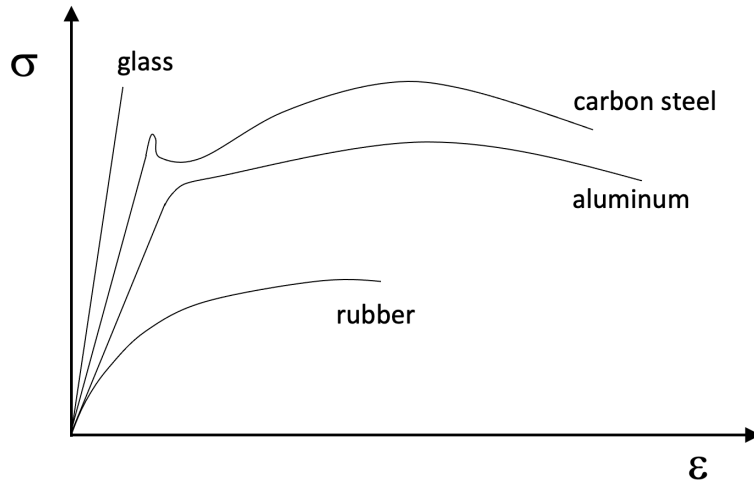


Figure 3.7: Shape of Stress-Strain curves for different class of material in the case of a uniaxial tensile test

Most of the materials have what is called elastic stage. It means that for a range of stress applied (from 0 to σ_0) the material will recover its original form without any residual strain. If the stress exceeds σ_0 , permanent deformation will remain, it is the plastic stage. But this does not concern us since the deformations applied during the respiratory movement are far too small to generate plasticity, so we will remain in the case of elasticity.

We can see that materials such as metals, alloys or glasses have a linear part in the stress-strain curve. This is the linear elastic stage: the stress is proportional to the strain. Materials such as rubber does not have a linear elastic stage, they are hyperelastic materials: the stress-strain relation derives from a function called the strain energy density function W . Liver tissue is also characterized by a hyperelastic behavior.

There is no clear consensus on the biomechanical model that characterizes the liver. However, the community agrees that it is an (nearly) incompressible hyperelastic material [76], [77]. We chose to study two distinct elasticity models: the linear elasticity model and the nearly incompressible Mooney-Rivlin hyperelastic model [77]. We made this because we wanted to see if a significant difference would appear between the linear model and the hyperelastic model that is supposed to be more accurate. Also, the equations of the two models were implemented in SFEPLY, the python finite element module that we used.

In the following part, a lot of tensors commonly used in solid mechanics will appear (deformation tensors, stress tensors, jacobian, ...). Detailed definitions and explanations of these tensors can be found in [78], or good introduction on the *Wikipedia* page of Finite Strain Theory [79].

Linear Elasticity

The linear elasticity problem is defined by the following PDE equation :

$$\nabla \cdot \sigma + f = 0 \quad (12)$$

where σ is the stress tensor, $\nabla \cdot \sigma$ is a notation of a divergence vector field, and f are the volume forces (weight). We have :

$$\begin{aligned} \nabla \cdot \sigma &= \frac{\partial \sigma(u)}{\partial x} + \frac{\partial \sigma(u)}{\partial y} + \frac{\partial \sigma(u)}{\partial z} \\ \sigma &= \lambda \text{tr}(\mathbf{E})\mathbf{I} + 2\mu\mathbf{E} \\ \mathbf{E} &= \frac{1}{2}(\nabla u + \nabla u^T) \end{aligned} \quad (13)$$

u is the displacement vector field, $\mathbf{E}$ is the Green-strain tensor, $\mathbf{I}$ is the identity matrix and λ and μ are Lamé coefficients (materials parameters). We need to find u that satisfies the weak form :

$$\begin{aligned} \int_{\Omega} (\sigma(u) + f)v d\Omega &= 0 \quad \text{with } u = t \text{ on } \Gamma \\ \int_{\Omega} \sigma(u) v d\Omega &= \int_{\Omega} f v d\Omega + \int_{\partial\Omega} t v d\Gamma \end{aligned} \quad (14)$$

where v is the test function and t is a fixed and known value of the displacement field, a boundary condition applied on the boundary of the volume Γ .

Mooney-Rivlin

Mooney-Rivlin Model follow the strain-energy density function :

$$W = c_1(\bar{I}_1 - 3) + c_2(\bar{I}_2 - 3) + \frac{K}{2}(J - 1)^2 \quad (15)$$

In this equation, c_1 and c_2 and K are material parameters, $\bar{I}_1$ and $\bar{I}_2$ are the first and second Invariant of $\mathbf{B}$ the left Green deformation tensor and $\mathbf{J}$ the determinant of $\mathbf{F}$, the deformation gradient tensor.

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \quad \mathbf{J} = \det(\mathbf{F}) \quad \mathbf{B} = \mathbf{F}\mathbf{F}^T$$

where $\mathbf{X}$ represent the coordinates of the system in the undeformed position and $\mathbf{x}$ in the deformed position.

If the material is homogeneous (which we will consider in our case), then the second Piola Kirchhoff stress tensor (PK2) S_{eff} is ⁷:

$$S^{eff} = \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}} = 2 \frac{\partial W(\mathbf{C})}{\partial \mathbf{C}} \quad (16)$$

After complex manipulations of equations [80], this leads to find u such that [81]:

$$\begin{aligned} \int_{\Omega} (S^{eff}(u) + K(\mathbf{J} - 1)\mathbf{J}\mathbf{C}^{-1}) : \delta \mathbf{E}(v) \, dV &= 0, \quad \forall v \\ S^{eff}(u) &= \mu \mathbf{J}^{-\frac{2}{3}} (\mathbf{I} - \frac{1}{3} tr(\mathbf{C}) \mathbf{C}^{-1}) + \kappa \mathbf{J}^{-\frac{4}{3}} (tr(\mathbf{C}\mathbf{I} - \mathbf{C} - \frac{1}{3}((tr(\mathbf{C}))^2 - tr(\mathbf{C}^2)))\mathbf{C}^{-1}) \end{aligned} \quad (17)$$

where $\mathbf{C}$ is the right Green deformation tensor; $\mathbf{C} = \mathbf{F}^T \mathbf{F}$

⁷The PK2 is an alternative stress tensor, again for more information [78]

4 Materials

4.1 Data and Respiratory Cycle

This work was performed out on 3 patients. For each of them, we have access to a planning sequence containing 10 3D images, a 3D image being the average over those planning sequence phases: the Mid-Position Computed Tomography (MidPCT or simply midp), and a continuous sequence represented by more than 200 3D images.

The 10 images from the planning sequence are acquired during a short time while the patient breaths normally. The patient performs a few breathing cycles. Each of these 10 images is reconstructed by averaging all the images corresponding at the same moments of these cycles. Each 3D image is composed of the assembly of many CT scan slices, which are 2D transverse (see section 4.2) images. This is used to plan the treatment. Each of these ten 3D images is called a 3DCT and represents a precise moment in the respiratory cycle, called **a phase**. Thus, the cycle is divided into 10 phases, from phase 0 to phase 90 and loops back. Phase 0 corresponds to the end inspiration while phase 50 corresponds to the end expiration. During inspiration, the lung inflates, the pressure rises, it fills the space and pushes on the liver. The liver goes down. And during expiration, it is the opposite movement, the lung deflates, that frees space and the depression allows the liver to go up. So phase 0 is low and phase 50 is high according to the z-direction, the direction going from the patient's feet to his head). From phase 60 to 0 the patient inhales, and from phase 10 to 50, the patient exhales. In Figure 4.1, we can see the evolution of the z position of the liver center of mass (obtained in section 5.2.3) for the 10 phases.

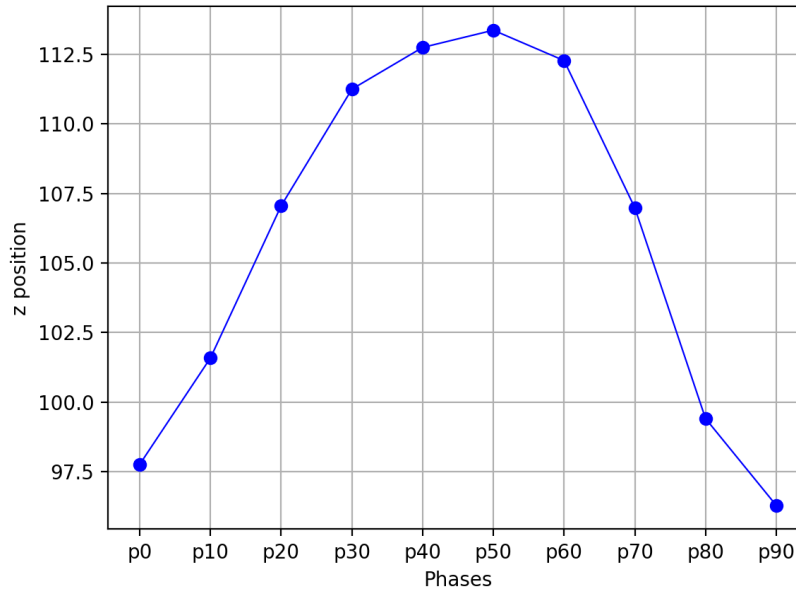


Figure 4.1: Z position of liver center of mass for patient 1

The average phase image MidPCT is generated by using the median of each voxel of all phases deformed

on the average. It reduces noise and artifacts of the MidPCT compared to the other phases. Thus, contours of region of interest (ROI) have been drawn by a doctor on this image.

The continuous sequence is a longer sequence of 3DCT images. Each iteration (or frame) of this sequence is in fact a 3D image distorted from the midp phase with the method previously developed by the laboratory [82]. This method can be summarized as follows. The goal was to recreate a 4DCT sequence that follows the respiratory movement from a 2D+t MRI sequence (2 dimensions + time). They took a 2D+t MRI sequence that has captured a patient's breathing for a given period of time. On these MRI images, they placed trackers at some strategic anatomical interfaces, visible on the MRI and on the MidpCT, therefore not the interface with the tumor, this one not being contrasted enough on the CT. The trackers will allow determining locally the respiratory phase (of the planning) of these MRI images. On the basis of these trackers, each new deformation field is reconstructed with an interpolation of the local deformation fields of the planning phases detected by the trackers. With these new deformation fields, the midp phase of the 4DCT planning is deformed. This results in an artificial 4DCT that does not only model one cycle, but several different cycles of variable intensity over a longer period of time. Its role is to recreate the possible behavior of the patient in the treatment room.

Another type of accessible data is masks. Masks are data associated with medical images (here 3DCT) and allow to define a specific region of these images (organs, tumor, i.e.). A mask could be represented as follows: a 3D array that for each voxel, the mask indicates "true" if it lies in the region (i.e. region of liver), and false otherwise. The masks we have access are the so-called *ground truth* (GT) masks of the tumor and liver of the 4DCT planning and continuous sequence. These masks are obtained by deforming the masks drawn on the MidpCT by a doctor with deformation fields. *Ground truth* masks of the planning are obtained with the deformation fields from midp to the corresponding phase. These DF are obtained with a non-rigid registration algorithm (section 5.2.2). In the case of the *ground truth* mask of the continuous sequence, the deformation fields are the ones obtained with trackers in the article [82]. For this work, we will consider these *ground truth* masks as our exact reference, i.e. that give exact information of the liver and tumor position for the 3DCT planning and continuous sequence images. This is debatable, but it is outside the scope of this work.

4.2 Conventions

Now it can be interesting to present the conventions that will be used further. First, is presented the axis reference system that will be used in this thesis. The origin of this reference system is set at the lowest, leftmost and closest voxel to us when looking at the human body from the front (see figure 4.2). The x axis is parallel to the ground and is oriented from the right hand to the left hand of the body. The y-axis is also parallel to the ground and runs from the front of the body to the back of the body. The z-axis is perpendicular to the ground and runs from the feet to the head.

In the medical field, anatomical planes are used, which define orientations for making anatomical cross-sections, i.e. interior views of the human body in different directions. These conventions are defined as follows: the sagittal planes are vertical planes that cut the body lengthwise, separating the right and left parts of the body. They are perpendicular to the x axis. The coronal planes are also vertical planes, perpendicular to the coronal planes, which separate the body into a ventral and dorsal part. They are perpendicular to the y

axis. And the transverse planes are horizontal planes, parallel to the ground, which separate the body into an upper and lower part. They are parallel to the z axis. This is illustrated on figure 4.3.

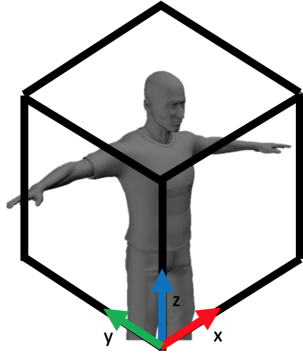


Figure 4.2: Reference axis system

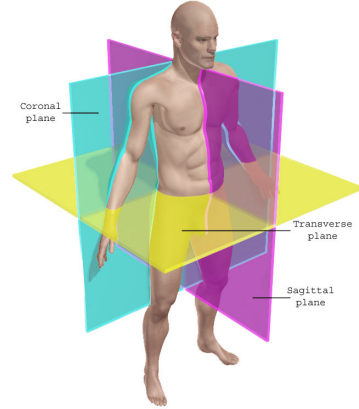


Figure 4.3: Conventional anatomic planes

4.3 Projections

As mentioned in section 1.3.2, the imaging technique available in combination with proton therapy is fluoroscopy. So to test our method we need fluoroscopy projections in order to reproduce the best matching with the real situation. Nevertheless, 3DCTs will be very helpful to validate our method since it gives more information concerning the tumor position. We can produce projections from our 3DCTs dataset thanks to a simulation function from Tomopy [83]. This Python library allows simulating a projection from a 3D object (eg. 3D array) similar to a fluoroscopy projection. You can find an illustration of the simulation result in figure 4.4.

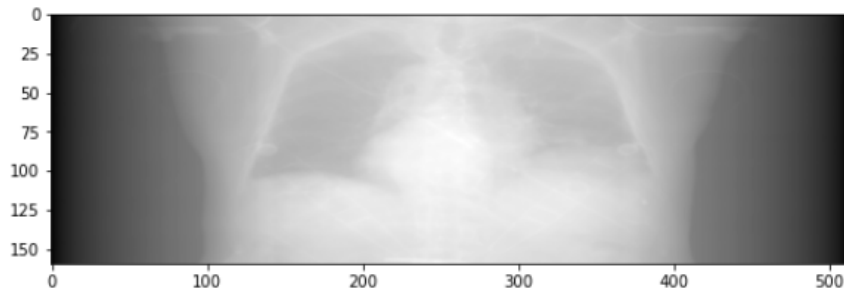


Figure 4.4: Tomopy simulation result (image from the planning sequence : coronal view)

Several projection angles have been tested (see figure 4.5) in order to find the one that offers the best information : coronal, sagittal, and transversal views have been tried, but also non-conventional angles. Those have been used at different steps of the process and finally it appeared that coronal view was the best choice.

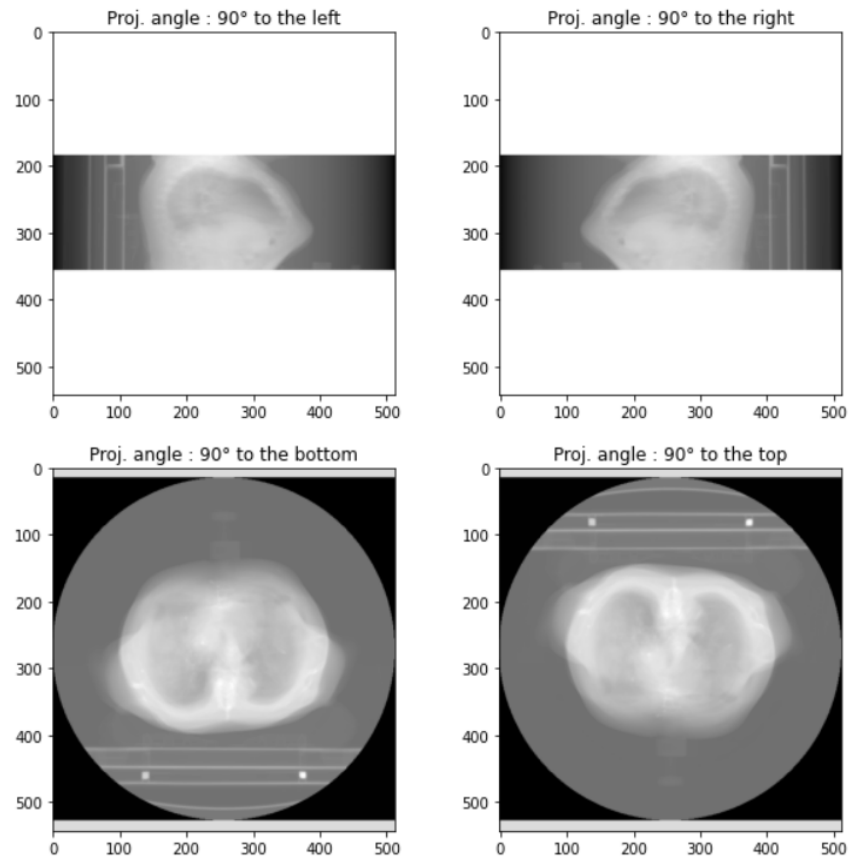


Figure 4.5: Tomopy simulation results with different projection angles

5 Method

In this section, we will detail the whole process of this master thesis. This general approach, which includes all the intermediate steps of our method, is visible in the figure 5.1 below.

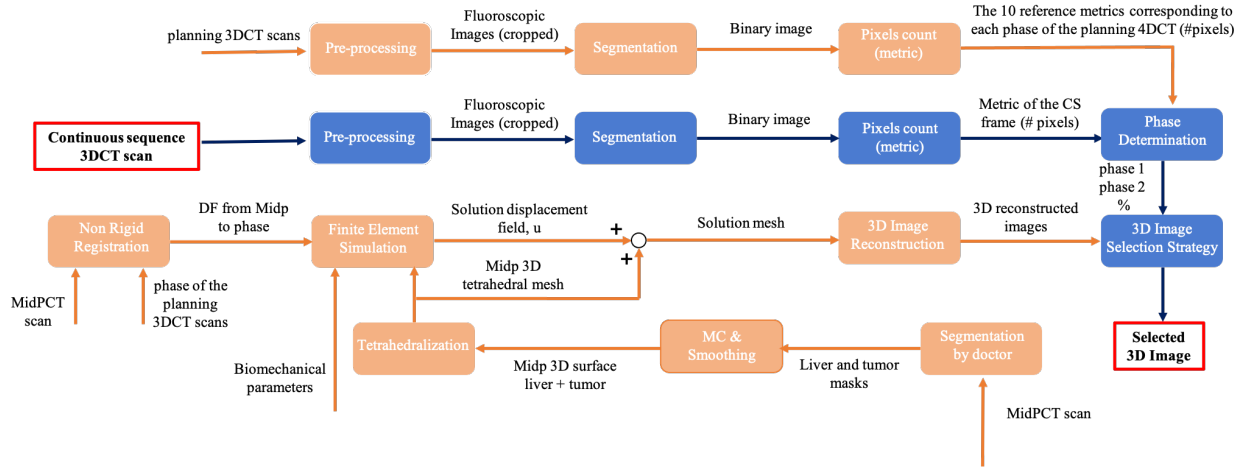


Figure 5.1: Global Method Process

The final objective is to be able to recognize in real-time the current respiratory state of the patient and to return a 3D information which approximates the real position of the tumor at this precise moment. For this, we will take our continuous 4DCT sequence on which we will make a 2D projection, simulating fluoroscopy. We will consider this new fluoroscopy sequence as our real time sequence on which we will determine the current respiratory state of the patient. With this recognized respiratory state, we will reconstitute the correct 3D image that corresponds the most to this respiratory state. This is the chain to be used in real-time (in blue on the figure 5.1). But for this chain to work, 2 other chains must be prepared beforehand (in orange in figure 5.1), so not in real-time. These 2 chains are built on the basis of the planning 4DCT, performed on the patient precisely to plan the treatment. The first chain (in orange above) allows running the method established to recognize the respiratory state (called the metric) on the planning 4DCT to obtain reference data that will be compared with the respiratory state during the real-time process. This will allow situating this current real-time respiratory state in relation to the pre-recorded states of the planning respiratory cycle. The other chain (in orange below) consists in precalculating the 3D images that will be returned once the current respiratory state is correctly identified and compared with the planning cycle. This chain allows anticipating, using finite elements, as many different 3D configurations of the liver and the tumor as possible in order to render the 3D image closest to the estimation of the current respiratory state.

To realize this work, we have chosen to work with Python, more precisely, using Jupyter [84], developed on Anaconda [85], for its ease of use, for its large panel of opensource software and for its interactive interface making it easy to merge the different parts of this work.

5.1 Prepare Reference Metrics

The goal of this step is as follows: calculate a metric for all respiratory phases (from p0 to p90). These so-called reference metrics will be reused to compare that obtained on a new input datum (a fluoroscopic image). With this comparison, it will be possible to assign a phase interval in which the patient's current respiratory state is located, and thus to locate where he is in relation to his respiratory cycle from the planning sequence.

To reach this goal, these three techniques will be applied to the simulated projections obtained based on CT scans: the pre-processing of images (section 5.1.1), the segmentation (section 5.1.2) and the metric computation consisting in a pixels count (section 5.1.3). Those different steps, as well as those input/output, are represented on the diagram given in figure 5.2.

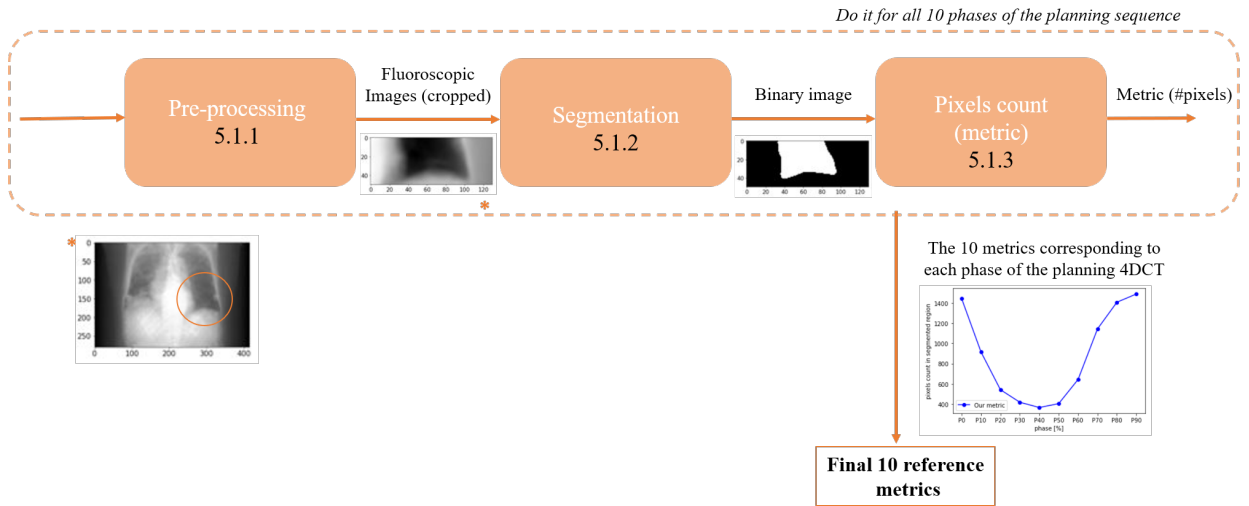


Figure 5.2: Diagram presenting the different steps used to prepare the reference metrics

5.1.1 Data Preprocessing

Before giving data as input to the algorithm used to determine the phase of a sequence image, the simulated projections have to be preprocessed. The different processes are explained below.

Resizing

As one can notice on figure 4.4 lungs appear to be flattened. This can be explained by the fact that the detector slice thickness is twice the pixel spacing. To make the image more realistic a resizing is needed. This one is kind of global and consists in resampling and doubling the number of pixels in the Z direction (head to foot direction). The resizing result can be found on figure 5.3a.

Normalization and Contrast Enhancement

In order to work on the same intensity scale (0 to 255) for both planning and continuous resized sequences normalization is required. This can be achieved by applying this simple equation to the entire image (pixel-

by-pixel) :

$$x_{new}[i, j] = 255 \times \frac{x[i, j] - x_{min}}{x_{max} - x_{min}} \quad (18)$$

with $x[i, j]$ the pixel at position i, j in the image. This normalization also allows to use an **histogram equalization** to enhance the contrast of projections which are quite low for the moment. You can find the non-contrasted and contrasted projections comparison on figure 5.3.

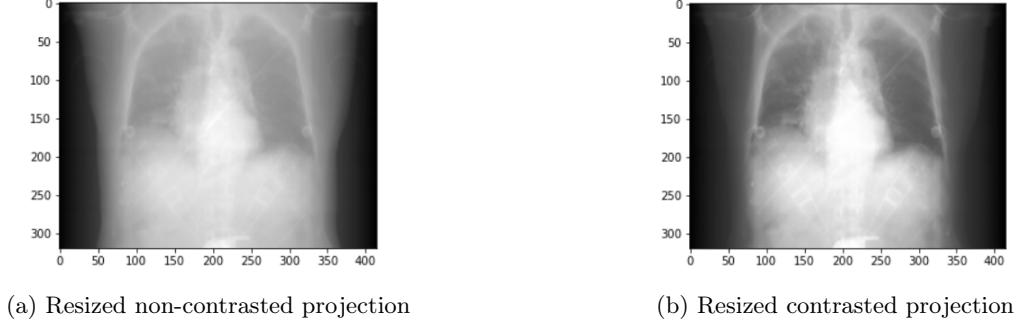


Figure 5.3: Contrast enhancement result

Histogram equalization is an image processing technique allowing image contrast improvement by manipulating the intensity histogram. In intensity histogram the horizontal axis represents the pixel intensity in a certain range (in our case 0 to 255), while the vertical axis corresponds to the number of pixels having a specific intensity value. As illustration you can find the histogram of an image from the continuous sequence (a) before and (b) after an histogram equalization 5.4. Histogram equalization corresponds to a spread out of the most frequent intensity values, in other words it stretches out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast [86].

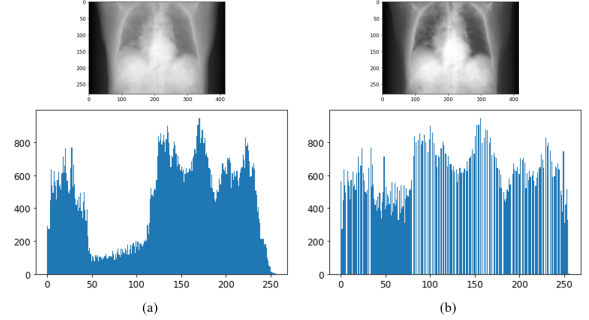


Figure 5.4: Histogram (a) before and (b) after the histogram equalization applied to the frame 0 from the continuous sequence of patient 1

Image Cropping

To analyze the image it is easier to crop it in order to focus on a particular region presenting the biggest changes from frame to frame in the sequence. This region is situated in the lungs lower part, where we better see the diaphragm going up and down. Depending on how the diaphragm compresses the lung we will be able to estimate the corresponding phase. The image is then cropped in the lower part of the lung : either the left one or the right one, depending on present artefacts. The size of the section also depends on those artefacts. For these reasons, the latter has to be chosen manually. The cropping applied to the images present in the planning of patient 1 is illustrated in figure 5.6b.

Combination for Planning Sequence

As visible on figure 5.6b the right borders of the region is not so contrasted. This is important to delimit the lung region as this will play an important role in the lung segmentation (see section 5.1.2) which will be used to deduce the phase in the continuous sequence in a further step.

To remedy this issue we will use another available dataset containing the contours defined by a doctor (precise enough). These contours are contained in 3D arrays : one 3D array for each image of the planning sequence. These arrays are made of 1's and 0's. The positions that belong to the lung region are represented by a 1 and others by a 0.

The process used to make the lung borders more visible is divided into two stages. First, we will create a new data set made up of the multiplications between each 3DCT planning and its corresponding contours array. By doing this, we will keep only the HU values of the lungs, that are negative, into a 3D array. All other values that do not correspond to the lung region are replaced by 0's. After that, we can make a projection, as explained in section 4.3, over these 3D arrays containing lungs HU values. The result of this projection is presented in figure 5.5a.

Finally, after the normalization, contrast enhancement and resizing of both datasets, by simply adding this result to the projections made on the planning 3DCTs one can add contrast to the borders, as illustrated in figure 5.5 below. The results on a cropped section of the image is also given in figure 5.6c. Note that the region growing algorithm applied to the projection containing only lungs HU values does not lead to good results compared those obtained with the combination.

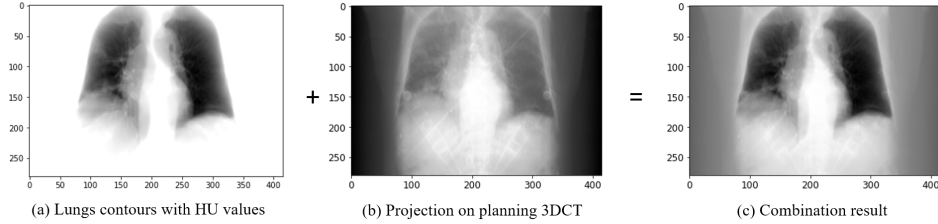


Figure 5.5: Combination with lungs contours

Image Smoothing

It also appeared that a smoothed image results in a better segmentation for the *region growing algorithm* described in section 5.1.2. To do so a convolution on the 2D projections has been applied thanks to the openCV python library containing a filter function [87]. The result of this convolution can be seen on figure 5.6d. This ends the data preprocessing. Now data can be given as input to the region growing algorithm in order to compute how many pixels aim to the *lung region*.

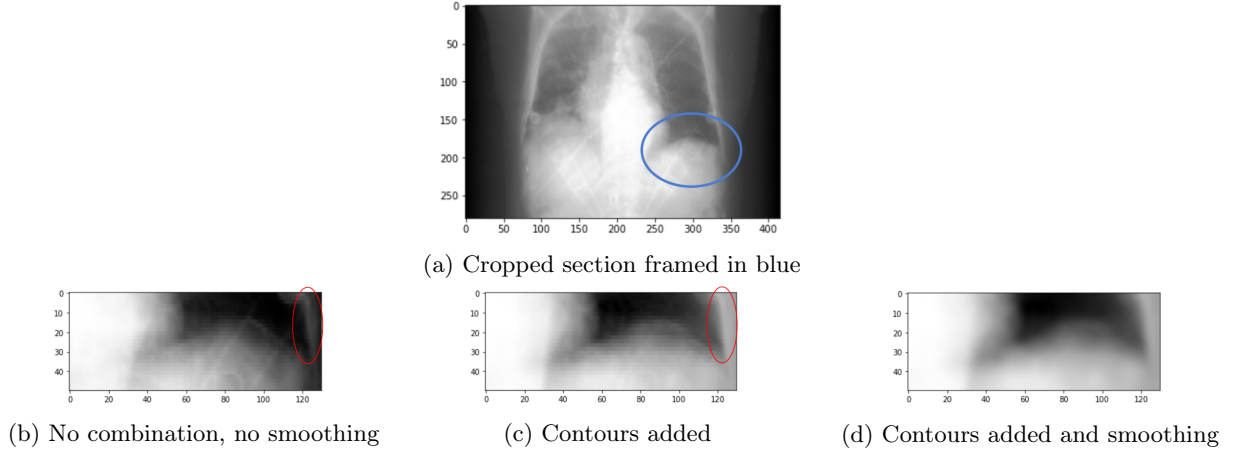


Figure 5.6: Planning cropped image of patient 1

5.1.2 Segmentation

The first step is the segmentation of the region of interest, i.e. the lower part of the lung. To segment the image, a **region growing** algorithm is used. The latter, starting from an initial point, also called seed, will determine whether or not the neighboring points belong to the desired region. To do so, the algorithm is based on one or several criteria that will determine pixels belonging to the region. In our case, there is a single criterion which is based on the average computed over the values of the pixels residing in the region (initially a single point, being the seed, then a single value and not an average). In order to be chosen to be part of the region, a given pixel has to respect the following equation:

$$\begin{aligned}
 x_i &= 1 & \text{if} & & |x_i - \text{average}(X)| &\leq \text{threshold} \\
 x_i &= 0 & \text{otherwise} & & &
 \end{aligned}$$

with x_i being the intensity value of the i^{th} pixel, and X the set of pixels belonging to the region. The *threshold* is manually fixed to 40 for planning images, this value appeared to be the most appropriate showing rather good results for the 3 different patients. We end up with a so-called binary image made of 0's and 1's : 1's correspond to the lung region and 0's to the "not-lung" region. This input/output can be seen on figure 5.7. The reason we have chosen a region growing algorithm rather than a simple binary threshold applied pixel by pixel is that it prevents selecting an outlier. Region growing only checks **neighboring pixels**, once a pixel is not selected its neighbors will not be checked (unless they are neighbors of another selected pixel). This means that once we get to the borders of the region we want to segment, and no more pixels will be added, any other pixels outside those borders will not be selected either. Then, no outliers will be accepted in the region. The algorithm is illustrated in figure 5.8. For this algorithm to be functional, we need the seed to be for sure in the desired region. At this stage, it is chosen manually so that it lies more or less at the middle of the lung region.



Figure 5.7: Segmentation bloc

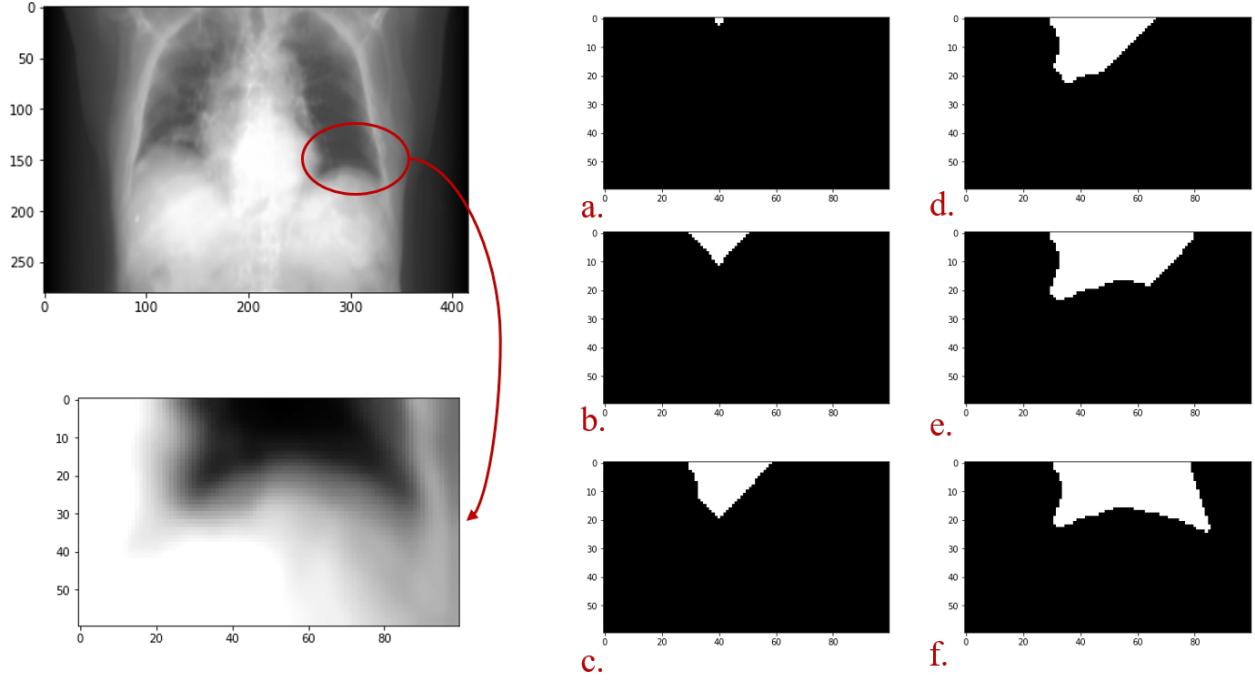


Figure 5.8: Illustration of some steps from the region growing algorithm applied to a continuous sequence frame of patient 1

5.1.3 Pixels Count (metric)

This step is the simplest one. Indeed the latter consists in a count over the 1's in the binary image just obtained. This gives us information about *how the lung is inflated* and then should evolve, from one phase to the next, in the same way as the patient's respiratory cycle. This will be verified in section 6.1. This bloc is reminded in figure 5.9.

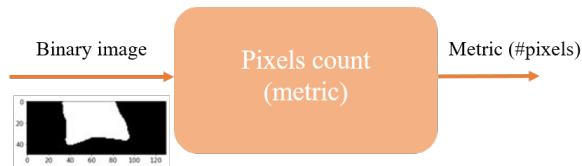


Figure 5.9: Metric computation bloc (pixels count)

5.2 Prepare 3D Reconstructed Images

In this section, we will detail the method that allows us to obtain the different 3D images among which the metric will choose the image closest to the patient's respiratory state. These 3D images are reconstructed images based on the results of the finite element model. At the end of the preparation process, we will obtain a total of 28 3D images for each patient. These images can be of 3 different types. Either images representing one of the 10 phases of the planning cycle, or interpolations of these phases (for a respiratory state that is between 2 phases in the scope of the planning cycle) or extrapolation (for a respiratory state outside the scope of the planning cycle).

The global process of construction of these 3D images is illustrated in figure 5.10 below. It can be divided into several steps; building the mesh (section 5.2.1): extraction of surfaces, tetrahedralization; getting deformation fields from a phase to another (section 5.2.2); apply the boundary conditions, define mechanical parameters and run the simulation (section 5.2.3).

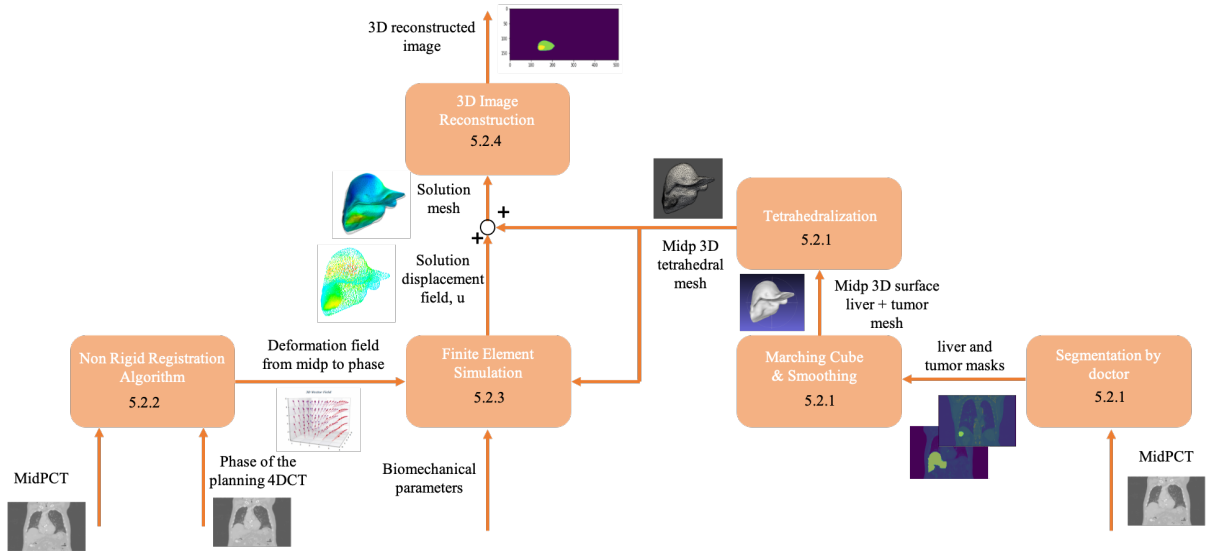


Figure 5.10: Global 3D images Reconstruction Process using Finite Element Model

5.2.1 Mesh Construction

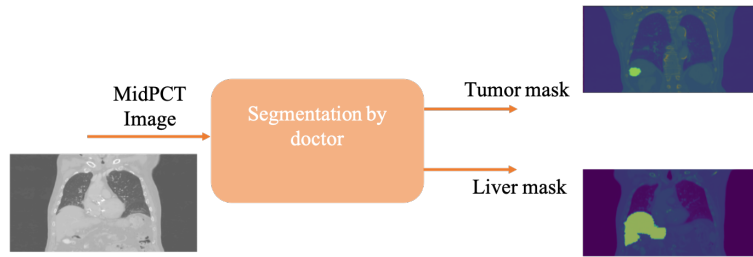


Figure 5.11: Segmentation Block

The first step is the construction of the mesh. The mesh is, in our case, a discrete approximation in 3 dimensions of the volume of interest (i.e. the liver). It is composed of a multitude of small simple entities (the tetrahedra), connected together. To build it, we need to extract the Region of Interest (ROI) from the Four-Dimensional Computed Tomography (4DCT) files. We had to choose a phase for which we will extract these masks. The idea is that we build the mesh only once and deform it to all other phases. The MidPCT (Mid-Position Computed Tomography), seemed to be the most suitable since being the average phase and as explained in section 4.1, noise and artifacts are reduced for this phase. The distance to the other phases is smaller and therefore the errors during the deformation are likely to be less important. Also, contours of ROI that we needed (liver, tumor...) were already defined by a doctor for the MidPCT unlike for the other phases. Figure 5.11 below illustrates this first step.

The ROIs are included in a separate file, linked to a 3D image (here the MidPCT phase) and are saved as a mask. We could extract the ROIs of liver and GTV (GTV for Gross Target Volume which is the closest thing of the exact tumor volume) in the form of binary numpy arrays (3D Matrix filled with 1 or 0), thanks to a code provided by Damien Dasnoy-Summel⁸. In the figure 5.12, the GTV ROI (red mask) is superimposed on a slice of the MidPCT (in coronal view).

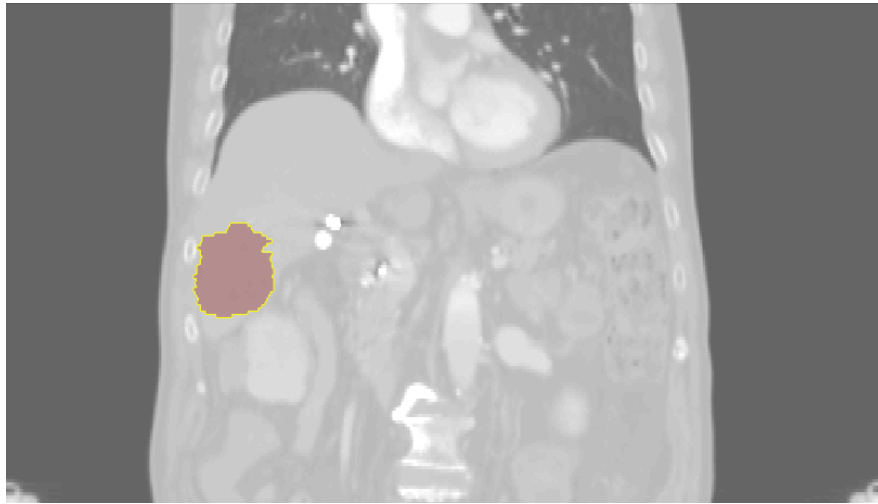


Figure 5.12: ROI of tumor (GTV) of patient 2

Then these numpy arrays were converted into the external surface of volumes, using two Python modules : *scikit-image* [88] to obtain the vertices and faces of the external surface of liver with the **marching cubes (MC) algorithm** and *numpy-stl* [89] to transform vertices and faces into a surface saved in a STL file ⁹. The **marching cube algorithm** is used to draw polygons of the surface that passes through cubes with respect to the nodal values of the cubes. In our case, nodal values are defined from the mask (1 if the voxel lies in the mask, 0 if not), and the cube grid links center of voxels to which binary values are assign (figure 5.13). Depending on the number and position of 1's and 0's at the vertices of the marching cubes, different polygons are drawn (figure 5.14). This only allows obtaining the external surface of the target volume since the cubes completely inside or completely outside have their 8 vertices being equal respectively to 1 and 0

⁸Doctor and assistant at UCLouvain in Louvain-la-Neuve

⁹STL, is for STereoLithography, is a file format created by 3D systems for its CAD software. Widely used for 3D design (3D impress i.e.), it allows to represent surfaces with small triangles.

and thus no polygon will be drawn for these combinations.

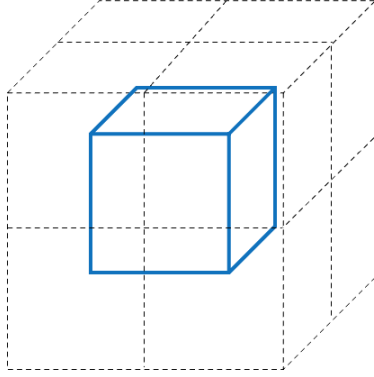


Figure 5.13: The marching cube grid. Dotted cubes are voxels. Blue cube is the cube used for MC algorithm formed by linking the 8 centers of voxels

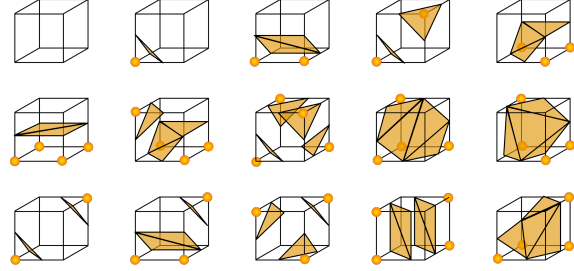


Figure 5.14: Configuration of MC polygons with respect to the nodal values. [90]

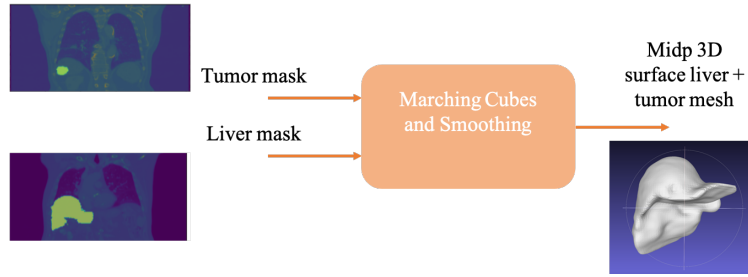


Figure 5.15: MC and Smoothing Block

After extracting the external surface of our liver and tumor, we needed to smooth these surfaces since the marching cubes algorithm provides rather angular surfaces, which are not sufficiently representative of the liver tissue and tumor surface for the finite element model. The tool that was used is a filter, available in the module *pymeshlab* [91] which is a python library that interfaces to MeshLab [92], an opensource software for processing and editing 3D triangular meshes. The filter that was used is called: *Surface Reconstruction VCG*. It is a filter which comes from the *Visualization and Computer Graphics Library*, a C++ library that also does mesh processing and editing [93]. Without going into technical details, the VCG filter smoothing the surface by keeping the connectivity of the points of the surface and by filling the small holes. The VCG filter is based on the Curless et al. algorithm, whose complete method is detailed in [94] for in-depth explanations. In figure 5.16 we see the surface extracted with the marching cubes algorithm and in figure 5.17 the smoothed surface with the Surface Reconstruction VCG filter. The steps of MC and smoothing are shown on the figure 5.15.

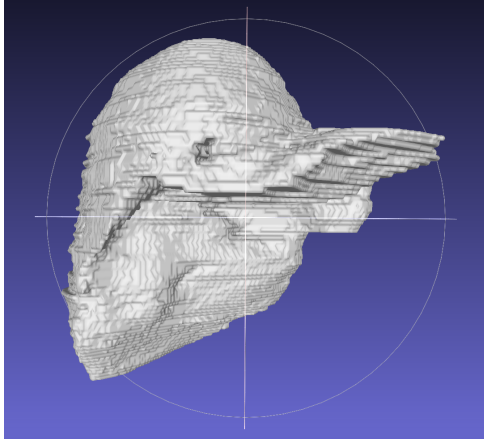


Figure 5.16: Liver of patient P2 from MC

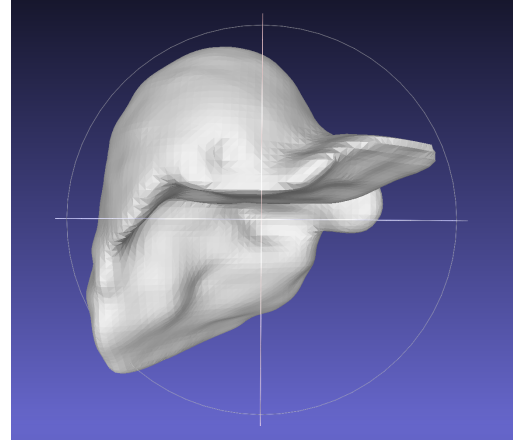


Figure 5.17: Liver of patient P2 smoothed with VCG

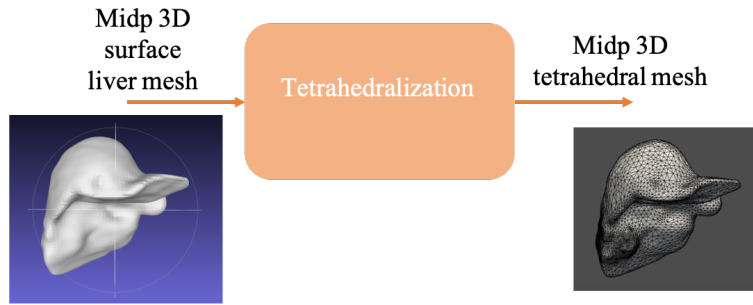


Figure 5.18: Tetrahedralization Block

The next step was to obtain a volume mesh, until now we only had extracted the external surface. It is possible to transform a closed outer shell into a 3D mesh with some modules. However, we have 2 distinct external surfaces saved in 2 different .STL files, the liver and the tumor, the second one being inside the first one. We could have simply taken the liver surface and filled it with tetrahedrons, but we would not have had any control over the random arrangement of the tetrahedrons inside, i.e. the tumor surface would not have been taken into account. However, we obviously needed a precise mesh at the level of the tumor to be able to follow it better. So we had to extract the vertices and faces of the 2 files (with *IGL* [95]) to write them in a single .STL file. Then, we could proceed to the tetrahedralization with the *wildmeshing* module [96] and save the final mesh in a .vtk file¹⁰ with *meshio* [97]. We can see in figure 5.20 the layout of the 3D mesh, the tetrahedrons inside the volume for a cut made along the y axis. We can see that the mesh is finer where the external surface of the tumor is located. This step of the process is represented in figure 5.18.

¹⁰Visualization ToolKit : developed by KitWare, initially used by Paraview to save and visualize simulation results. It also allows to save 2D and 3D meshes.

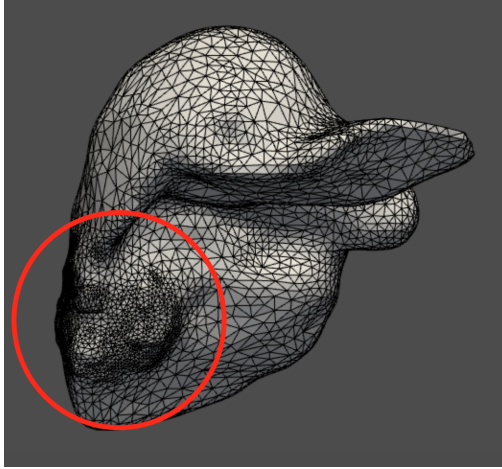


Figure 5.19: Liver 3D mesh of patient P2 : full view. Red circle shows tumor location.

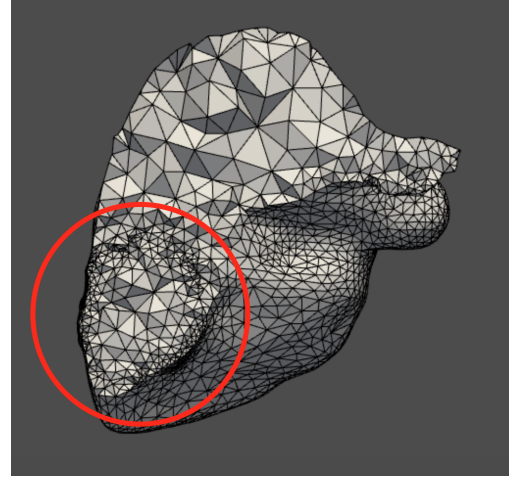


Figure 5.20: Liver 3D mesh of patient P2 : cut view. Red circle shows tumor location.

The meshes of the three patients are shown here under and their characteristics are listed in table 4 below. In view of the very different shapes of the patients' livers, one realizes the need for a specific model for each patient.

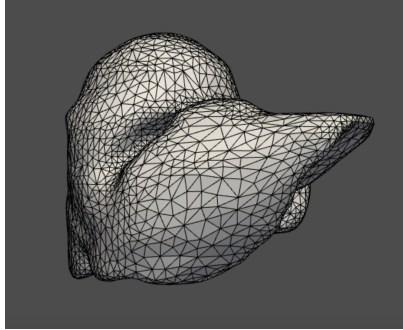


Figure 5.21: Mesh of Patient 1

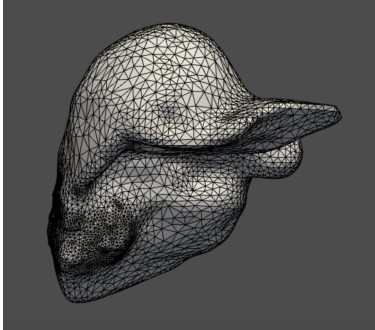


Figure 5.22: Mesh of Patient 2

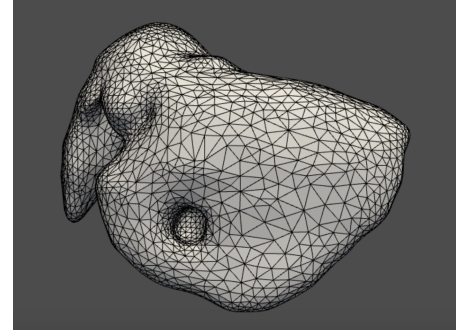


Figure 5.23: Mesh of Patient 3

	Patient 1	Patient 2	Patient 3
Number of vertices	9035	9614	7056
Number of elements	38 760	42 868	29 225

Table 4: Mesh Characteristics of Patients

5.2.2 Deformation Fields Extraction

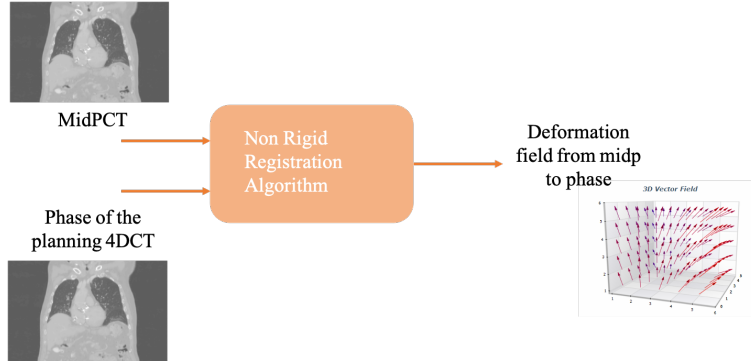


Figure 5.24: Non Rigid Registration Algorithm Block

Once the mesh is obtained, we needed to know how it moved. Therefore, we extracted the deformation fields (DF) from the 4DCT images. A deformation field is a tool that allows quantifying the deformation between 2 images: each point in the original image (in this case the voxels) is assigned a displacement vector that connects the position of this point in the original image to its position in the deformed image. In other words, if we apply the deformation field to the original image, we obtain the deformed image. We thus want to obtain the DF from the MidPCT to other phases, since we built the mesh for the MidPCT. For this, we used a **non-rigid registration algorithm**, implemented by REGGUI [98], which is a medical image processing software. This algorithm allows precisely to extract the deformation fields between two 3D images, and in our case from the MidPCT phase to the other phases. The underlying mathematical equations of this algorithm will not be detailed here but complete explanations are given in this article [99].

5.2.3 Finite Elements Simulation

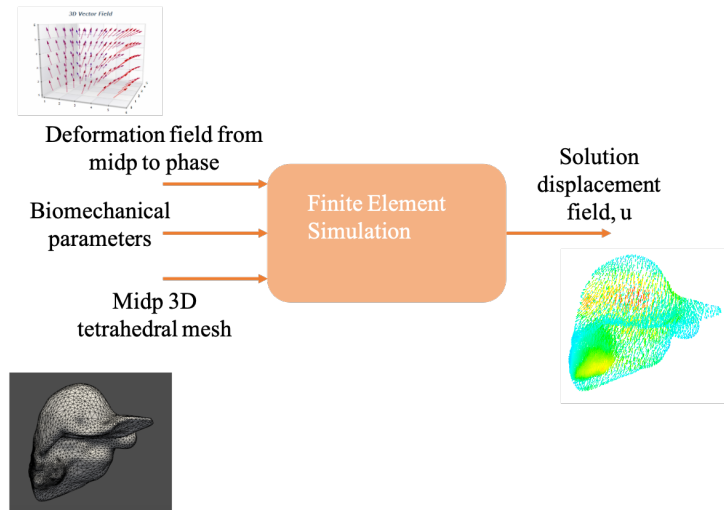


Figure 5.25: Finite Element Block

Then, the biomechanical parameters had to be properly chosen to implement the partial differential equations governing the behavior of the liver. As described in the section 3.3, we chose two elasticity models for the behavior of the liver. One elastic: the linear-elastic model, and the other hyperelastic: the nearly incompressible Mooney-Rivlin hyperelastic model. The module SfePy used for the simulation supports these two models. It requires the values of some biomechanical parameters of the liver tissue to build the numerical system described in section 3.2.4. As a reminder, here are the general equations of both models :

Linear elasticity :

$$\nabla \cdot \sigma + f = 0 \quad (19)$$

Nearly Incompressible Mooney-Rivlin, strain energy density function :

$$W = c_1(\bar{I}_1 - 3) + c_2(\bar{I}_2 - 3) + \frac{K}{2}(J - 1)^2 \quad (20)$$

Mechanical parameters required by SfePy are listed in the following array.

Linear Elasticity		
Symbol	Name	Description
E	Young Modulus	Quantify the stiffness of materials, links stress and strain
ν	Poisson Ratio	Deformation of material perpendicular to the specific direction of loading
Mooney-Rivlin		
Symbol	Name	Description
K	Bulk Modulus	Links stress and strain for isostatic compression
κ	Shear Modulus of MR term	Quantify the shear stiffness, ratio of shear stress to shear strain
μ	Shear Modulus of NH term	Quantify the shear stiffness, ratio of shear stress to shear strain

with :

$$\begin{aligned}
 E &= \frac{\sigma}{\epsilon} & \nu &= \frac{-d\epsilon_{trans}}{d\epsilon_{axial}} & K &= \frac{E}{3(1-2\nu)} \\
 \kappa &= 2(c_1 + c_2) & \mu &= \frac{E}{2(1+\nu)}
 \end{aligned} \quad (21)$$

The Young Modulus E can be obtained with a Fibroscan [100]. The FibroScan is a type of non-invasive examination, performed for the diagnosis of the stage of fibrosis of the liver tissue. The FibroScan works as follows: a sound wave is propagated through the liver tissue and its travel speed is measured. Knowing the density of the liver tissue, we can calculate the elasticity of the tissue with the following relationship:

$$v = \frac{E}{\rho} \quad (22)$$

When the liver is injured, it produces scar tissue in an attempt to replace the dying cells. Fibrosis is the accumulation of scar tissue in the liver, which is made up of collagen, a protein that provides resistance to tissue stretching. This explains the increase in elasticity with the state of fibrosis, and a direct link could be established. Knowing that hepatic tumors develop 4 times out of 5 in a hepatic tissue affected by cirrhosis (last stage F4 of fibrosis) [101], for linear elasticity model, we have chosen as a measure: 20 kPa for hepatic tissue and 40 kPa for the tumor tissue [102]. The Poisson ratio ν was also chosen from [102] being equal to 0.4. Concerning the Mooney-Rivlin model, shear modulus of Neo-Hookean term μ and Bulk Modulus K were calculated from (21) with Young and Poisson Ratio from Linear Elasticity model. Shear Modulus of Mooney-Rivlin term was computed from (21) with $c_1 = c_2 = 0.135$ kPa [77].

We then had to define the regions of our mesh to assign them their own biomechanical parameters. We had to determine which tetrahedra were in the tumor region and which were in the liver region. To do this, we superimposed the tumor mask of the midp with the mesh. We looked for each tetrahedron if the vertices were inside the mask. If the 4 vertices of the tetrahedron are inside the tumor mask, then the tetrahedron belongs to the tumor region. Otherwise, it belongs to the liver region. We could obtain a list of all the elements that form the tumor region. This process is illustrated with an example in 2 dimensions in figure 5.26 and 5.27 below. In figure 5.26, in red, the mesh. The tumor mask is shown in gray. In figure 5.27, the region of the mesh selected as the tumor region.

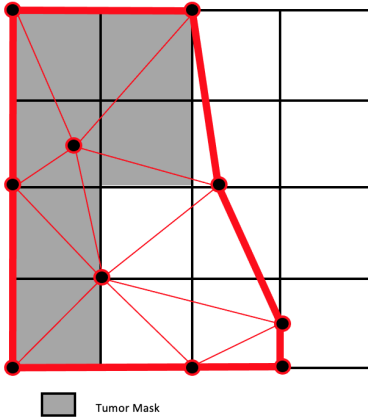


Figure 5.26: Mesh and tumor mask

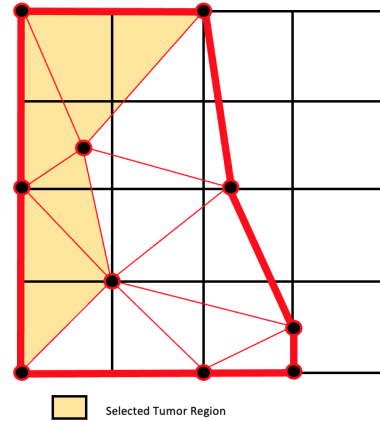


Figure 5.27: Selected tumor region

After that, we defined the boundary conditions. In our case, it is about values of displacement that we impose to the nodes of the mesh on its border. These values are taken from the displacement fields extracted with the registration algorithm. Of course, these displacement fields are different according to the phase towards which we will deform the liver mesh. The displacement fields are available in the form of a 3D matrix, with a vector associated to each voxel. It was necessary to apply these homogeneous and regular deformation fields to the nodes of the mesh surface which occupy the space in a disordered way. To do so, we used the **trilinear interpolation**. The trilinear interpolation is a simple method to interpolate a function established on a regular grid (the voxel grid) at intermediate points (nodes of the mesh). See Appendix B.

Now that we have computed the mesh, extracted the deformation fields, applied boundary conditions on the external surface, chose appropriate mechanical parameters for both models, we could run the SfePy simulation. We chose quadratic shape functions for the linear elastic model and linear for the Mooney-Rivlin model. This is due to the fact that the Mooney-Rivlin model requires a much longer execution time than the linear model because it requires a resolution in several iterations. The linear shape functions allow reducing the number of nodes per element, thus the size of the local matrices and the size of the global matrix, which largely reduces the execution time. The reason why the Mooney-Rivlin requires several iterations comes from the nearly incompressible nature of the model: if the initial deformation is too large, elements are warped and no automatic recovery is possible. The deformation must therefore be divided into several steps and gradually increase until the final deformation is reached. The execution time of each phase is about 900 seconds for linear elasticity and 300 seconds for Mooney-Rivlin. Both models are solved with a non linear solver using the Newton method. And the non-linear solver takes a linear solver: the SciPy direct solver.

And thus finally, we were able to reproduce the 10 different phases of the respiratory cycle with this process.

The solution of the FE simulation, which is the solution of the PDE (Eq.(14), (17)), is the displacement field u . This displacement field u is therefore the difference between the deformed mesh of a phase obtained after the simulation and the mesh of the midp. The finite element simulation generates an array, containing u , that associates to each vertex the corresponding displacement vector (with components x, y and z). This array is saved in a .vtk file and can be seen with Mayavi [103] as in the figure 6.9. The mesh is deformed so that the vertices reach the solution positions by adding the displacement u to their position in the midp phase. A color code is provided representing the relative displacement u (in mm) of the deformed mesh with respect to the black frame which corresponds to the mesh of the midp phase. The following figure 5.28 represents the liver for the patient 2 in the y-view for the phase p0.

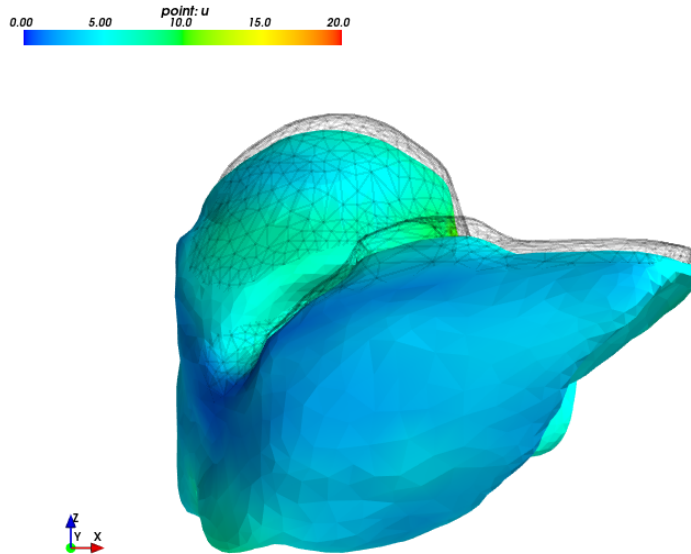


Figure 5.28: Results of FE simulation for patient 2, phase p0. A color to quantify the relative displacement u with respect to the midp mesh (in [mm])

Interpolation

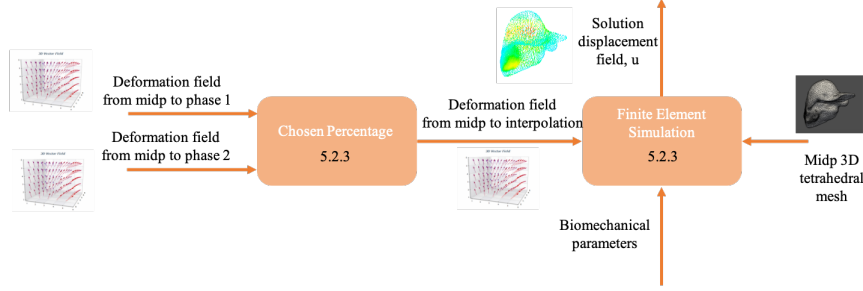


Figure 5.29: Global Interpolation Process

Of course, during the treatment, the tumor will not always be in the exact positions of the planning phases. It is therefore important to be able to create intermediate phases to these 10 phases. This could be useful if the tumor is detected at an intermediate position between two phases by the metric. This can be achieved simply by taking two successive phases and by doing a mean of the two fields.

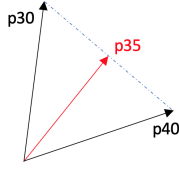


Figure 5.30: Mean DF of two phases

Here a representation of one of the vectors of the DF assigns to the same voxel for different phases : p30 and p40. The two vectors in black are the vectors of their respecting phases, and the red vector is the mean vector of these two vectors and can be obtained with :

$$DF_{35}[x, y, z] = \left[\frac{x_{p30} + x_{p40}}{2}, \frac{y_{p30} + y_{p40}}{2}, \frac{z_{p30} + z_{p40}}{2} \right] \quad (23)$$

This process is applied to every single vector of the DFs.

The method can be extend to any intermediate position, not only at the middle of two phases. We defined the relative position (in percent) of the phase we want to construct between the two known successive phases. Then :

$$DF_{30+r}[x, y, z] = [(1 - r) x_{p30} + r x_{p40}, (1 - r) y_{p30} + r y_{p40}, (1 - r) z_{p30} + r z_{p40}] \quad (24)$$

This simple process is shown on the figure 5.29. It was decided to precalculate 10 interpolation phases for the global method. The 10 phases are located in the intermediate positions between the phases of the planning cycle. I.e. p05, p15, p25 ... until p95.

Extrapolation

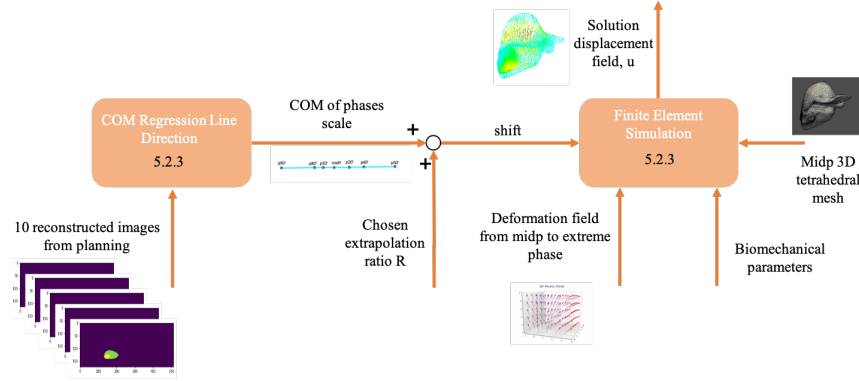


Figure 5.31: Global Extrapolation Process

It may also happen that the patient inhales more or less air than in the planing sequences, which will cause more displacement of the lungs and consequently of the liver and the tumor. Thus, it is also useful to construct phases that represent the amplified movement of the liver.

To do this, one can be tempted to just multiply the DF by a factor which reflects the state of respiration relative to the cycle modeled in the different phases of 4DCT. For example, if the metric detects the position of the liver to be 2 times lower due to a heavy breathing, the factor would be 2. But the problem with this method is that it will also multiply deformations: as the displacement vectors do not have the same direction, simply increasing their length will accentuate gaps and striction between vertices of the mesh resulting in incoherent forms of the liver. This is not realistic because of a certain rigidity of the liver tissue. A greater displacement does not mean bigger deformations. This is illustrated in the figures below. For the sake of clarity, we have shown an example for a 2D mesh, but the principle is exactly the same for a 3D mesh. The grey frame represents the initial mesh on which we applied a DF (blue arrows) to each vertices of the mesh. The red frame is the deformed mesh. In figure 5.32 we applied a 'normal' DF. In figure 5.33, we applied the same DF multiplied by a factor 2. We can see that it gives a warped frame. The chosen method is illustrated in figure 5.34. We first performed a translation (green arrows) before applied the DF to the frame. In this way, we obtained what we were looking for: our deformed liver but further.

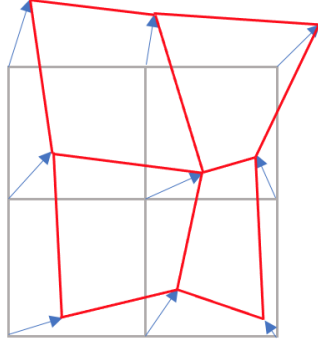


Figure 5.32: Normal deformation

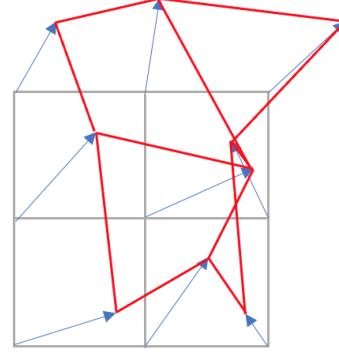


Figure 5.33: Wrong Extrapolation

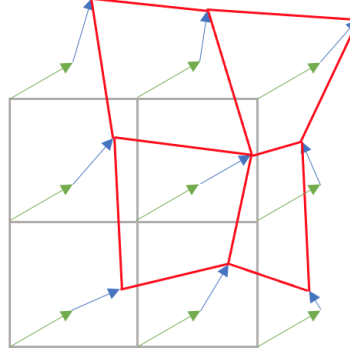


Figure 5.34: Chosen Extrapolation

To be able to do the translation, we needed to determine the direction of movement of the liver. To decide which direction to choose, we analyzed the overall motion of the liver. Even though the deformation fields have vectors pointing in different directions, there is still a general trend. During inspiration and expiration, the liver, whose movement depends on the lung, moves rather vertically in the rib cage (Section 4.1).

For this, we have computed the position of the center of the liver for the 10 phases. For that, we took the liver mask of the midp phase and we computed the geometric center by taking the mean point of voxel positions:

$$c_{x,y,z} = \left[\frac{\sum_{i=1}^N x_i}{N}, \frac{\sum_{i=1}^N y_i}{N}, \frac{\sum_{i=1}^N z_i}{N} \right] \quad (25)$$

where x_i , y_i and z_i are the coordinates of the center of voxels and N is the voxel number belonging to the liver mask.

Then we looked for the vertex of the 3D tetrahedral midp mesh that was closest to this point. With the results of the finite element model that we constructed for the normal cycle, we recorded the position of this

same vertex for each of the 10 phases. After that, we computed the regression line of those 10 positions, i.e. the line that best fits the points in the least square sense¹¹. The figure 5.35 represents the regression line obtained with the 10 phases of the 4DCT for patient N°1.

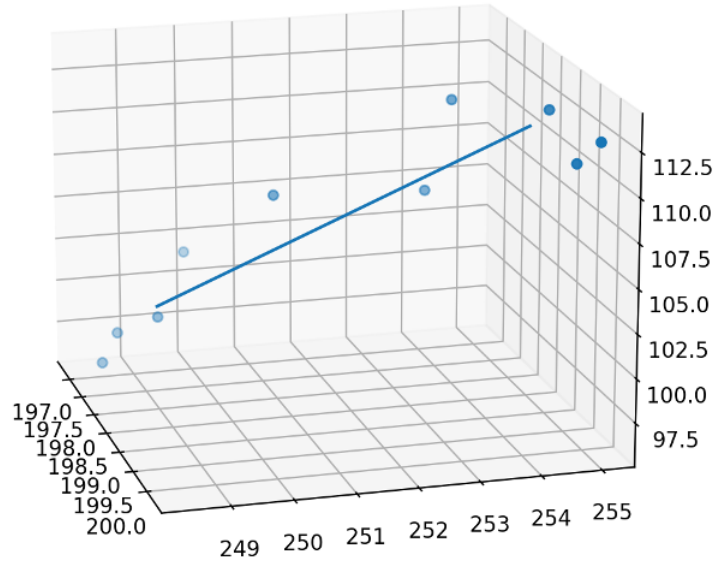


Figure 5.35: Regression line for the 10 phases of patient 1

Patients	x component	y component	z component
1	0.1521	0.3677	0.9173
2	0.1098	0.6611	0.7421
3	0.2427	0.3218	0.9151

Table 5: Orthogonal directional vectors of the regression line

With the movement direction, we just needed to determine the distance the liver has to travel along that line. To do so, we made the line pass through the point of the midp (which is still the position of the center of the liver) and we computed the distance between the midp point and the points that were the orthogonal projections of the 10 other phases on this regression line. This is represented in the figure 5.36.

¹¹Method to approximate a set of point by a line: the line is constructed such that it minimizes the sum of squared residuals (orthogonal distance between points and line)

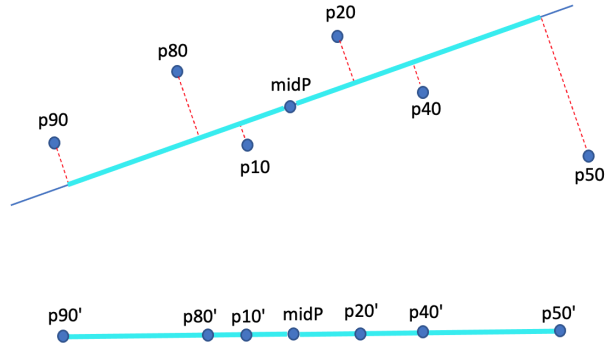


Figure 5.36: Orthogonal Projection of center of liver for all phases on the regression line

Patients	p0	p10	p20	p30	p40	p50	p60	p70	p80	p90
1	-8.69	-4.75	1.73	6.38	7.88	8.20	6.68	0.73	-6.93	-10.14
2	-7.50	-4.66	0.32	3.56	4.88	5.62	5.05	1.89	-2.95	-7.02
3	-4.95	-3.15	-0.51	1.74	3.23	4.25	3.51	1.23	-1.54	-4.01

Table 6: Relative distances between the midPCT liver center and all phases liver center projections on the regression line (in [mm])

The objective of this method is to keep the relative proportions between phases. If a more important breathing is detected (1.5 times more important for example), we can simply multiply the distances between projections and midP on the line by a coefficient R (1.5 here) and we obtain the new positions of the liver center that we have to reach. But as explained above, we cannot multiply the DF by the ratio, we have to make a translation, the shift. The shift is the difference between the distances from the extrapolation phase projection $R * P'$ to the midp and the distance from the reference phase projection P' to the midp. Then the DF is applied to allow to reach the target point $R * P'$. This is illustrated in equation (26) and in Figure 5.37.

$$R = \frac{|MidP, R * P'|}{|MidP, P'|} \quad shift = |MidP, R * P'| - |MidP, P'| \quad (26)$$

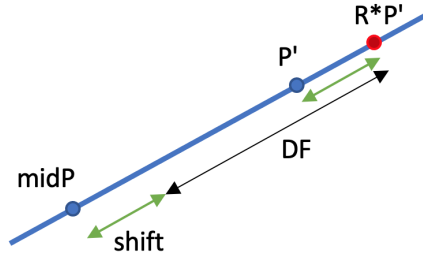


Figure 5.37: Extrapolation of a further phase.

This method thus allows to model extrapolations for any phase. But in practice, we will only use it for the extreme phases. We have indeed considered that the liver moved on a straight line. So if we want to model the liver further, on this line, we will calculate the appropriate shift, but we will use the DF of the extreme phase on the same side.

So if the metric detects a respiratory state that is within the planning range, we will choose either the closest phase of the planning cycle, or an interpolation of 2 planning phases. On the other hand, if the metric detects a respiratory state outside the range of extreme phases of the planning. We will extrapolate these extreme phases to reach the breathing state.

This extrapolation process can be seen in figure 5.31. We decided to create 4 extrapolated phases on each side (so 8 in all) with ratios of 1.25, 1.5, 1.75 and 2. We will call them phases 125%, 150%, 175% and 200%. These phases are thus obtained by calculating and applying the respective shift to the mesh, then by using the same deformation field as for the extrapolated extreme phase.

5.2.4 3D Images Reconstruction

The finite element simulation does not give a 3D image, but a mesh, i.e. a list of vertices and their positions, and a list of elements to which they are connected. The reconstruction process of a 3D image from a mesh is explained just below. We needed a process to reconstruct a 3D image, i.e. an image containing voxels, of the same dimension as the 3DCT.

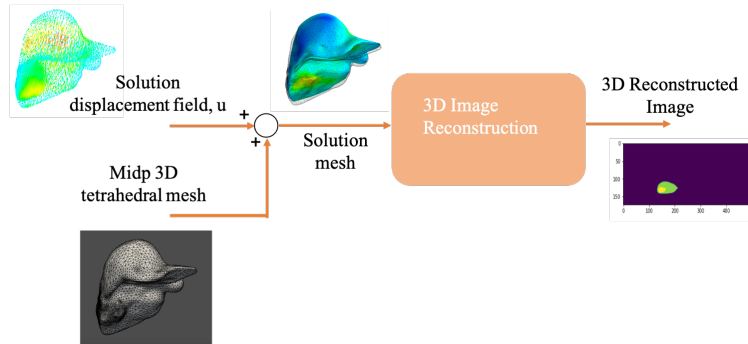


Figure 5.38: 3D Image Reconstruction Block

The reconstruction process is as follows: we took an empty numpy array (3D matrix) of the same dimension as the 4DCT planning images (i.e. $512 \times 512 \times 173$ voxels for patient 1) where each cell represents a voxel. We first filled this numpy array with HU corresponding to air (-2000). Then, knowing the size of the voxels for a 3DCT image ($1.074 \text{ mm} \times 1.074 \text{ mm} \times 2 \text{ mm}$) we checked, for each voxel if it was inside a tetrahedron of the mesh (more precisely if its center was inside). If yes, then we filled the numpy array cell with a value of 60 HU (corresponding to the liver). Knowing also the region of the tumor in the mesh, i.e. the list of tetrahedrons that form the tumor, we could check which voxels, inside the liver, were also inside the tumor and assign them another HU value of 500 (corresponding to the bone but it is to have a better contrast and to be able to actually see the tumor). After this process, we had a 3D numpy voxel array, filled with HU values, where the liver mask is the set of voxels for which the HU are 60 or 500 and the tumor mask is the set of voxels for which the HU are strictly 500. From this 3D image, it was easy to extract binary masks of liver and tumor. In the figure below, an 2D example of the reconstruction process of a mask from a mesh. In figure 5.39, we have our empty numpy array in black, representing the voxel grid, and in red our mesh with boundaries represented by bold red line. In Figure 5.39, the yellow triangles represent the tumor region while the red ones represent the liver. The reconstructed masks of the corresponding color are represented in Figure 5.40.

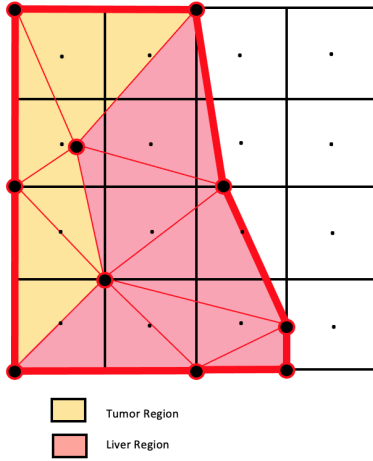


Figure 5.39: Mesh and voxel grid superimposed

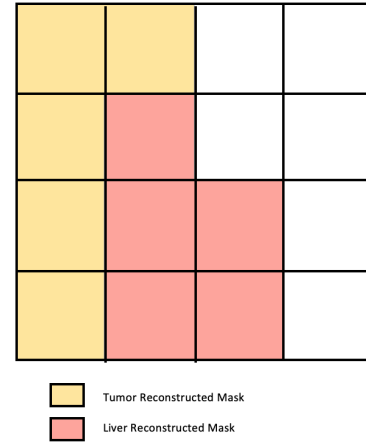


Figure 5.40: Reconstructed masks

This example illustrates the fact that we have to approximate the results obtained with the simulation and that it is not possible, since the voxels are parallelepipeds and the mesh is a set of tetrahedrons, to obtain masks exactly equal to the mesh. This can be a small source of error at the edges of the mesh. This process was used to reconstruct all the 3D images corresponding to the results of all the planning cycle simulations, interpolations and extrapolations.

5.3 Process Aiming to be Used in Real-Time

Now that we have gathered all the necessary pre-computed data : reference metrics and 3D reconstructed images, it is interesting to see how they will be used in the *real* context. In other words, how the new input data will be processed in order to infer which phase corresponds to the patient's current respiratory state.

5.3.1 From Fluoroscopic Images to the Phase Determination

In a real situation, the input corresponds to a fluoroscopic image. In our case the input is a projection trying to imitate a fluoroscopic image from a CT scan. The corresponding metric is computed in the same way than for the planning images (see section 5.1.1, section 5.1.2 and section 5.1.3). Once the metric is obtained, a number of pixels, it is necessary to know which reference metric corresponds to it best. To do this, we have calculated a difference between this newly obtained metric and each of the reference metrics. The phase with the metric showing the smallest difference is chosen as the **closest matching phase**, P_{sel} . For sure, patient's respiratory state rarely exactly corresponds to a reference phase, it will lies between two successive phases most of the time. This is illustrated in figure 5.41 where $\mathbf{A}$ corresponds to the metric obtained for the new input data, P_{sel} the closest matching phase and $P_{sel} - 1$, $P_{sel} + 1$ respectively the previous and next phase of P_{sel} . $\mathbf{A}$ can be either a little lower than the closest matching phase (subfigure 5.41a), or a little higher (subfigure 5.41b). Depending on this condition the returned interval can be either

$$[P_{sel} - 1, P_{sel}] \quad \text{if} \quad \mathbf{A} < P_{sel}$$

or

$$[P_{sel}, P_{sel} + 1] \quad \text{if} \quad \mathbf{A} > P_{sel}$$

In addition to the interval, an information about where $\mathbf{A}$ stands regarding P_{sel} is also returned in percentage (corresponding to *Perc* on figure 5.41). The latter will allow us to determine whether one must choose an intermediate phase, obtained by interpolation (see section 5.2.3), or rather an original phase coming from the 10 planning references. This value *Perc* will also be used in the case of extrapolation. Indeed $\mathbf{A}$ could also be positioned higher than the maximal reference phase, or lower than the minimal one. In this case we will no longer speak of interpolation but of extrapolation since it is situated between a reference phase and a non-existent respiratory state. To deal with this we are going to imagine 2 additional phases: one would be under the minimum phase, and the other would be greater than the maximum phase. This is illustrated in figure 5.42. The additional phase is defined as being the metric obtained for P_{min} (respectively P_{max}) subtracted by (respectively added by) 100 pixels and is called P_{-100} (P_{+100}). This amount of 100 was chosen to be acceptable according to the metrics obtained for the planning sequence images. The choice of the amount is not so critical since the percentage will be adapted further to chose the correct extrapolated model (see section 5.2.3).

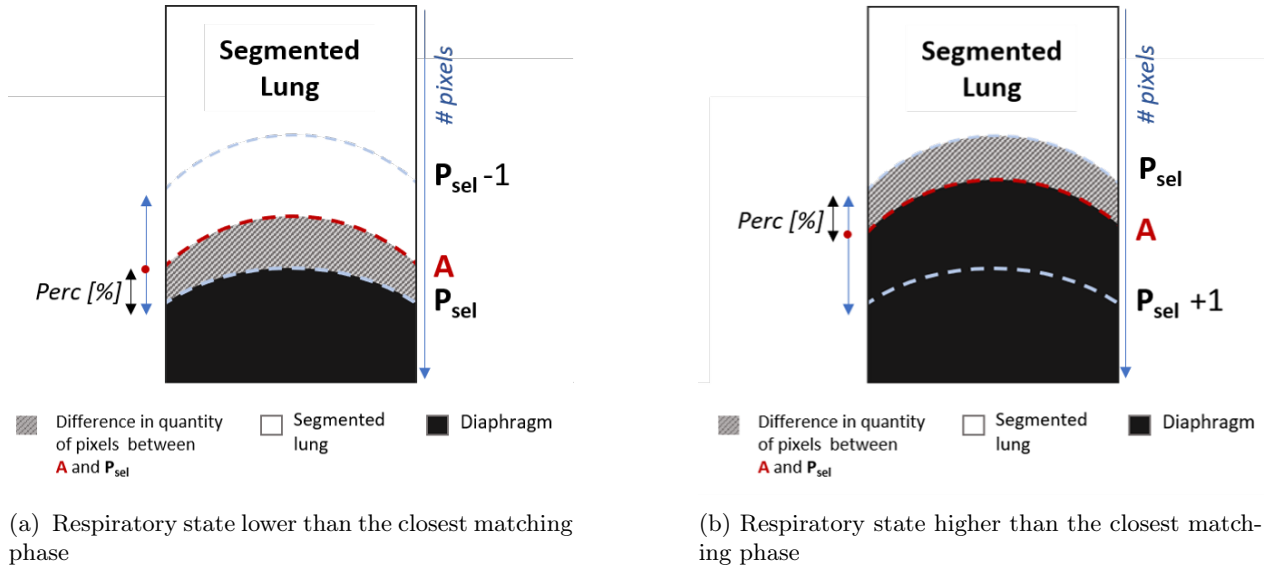


Figure 5.41: Example of a respiratory state positioned between two reference phases

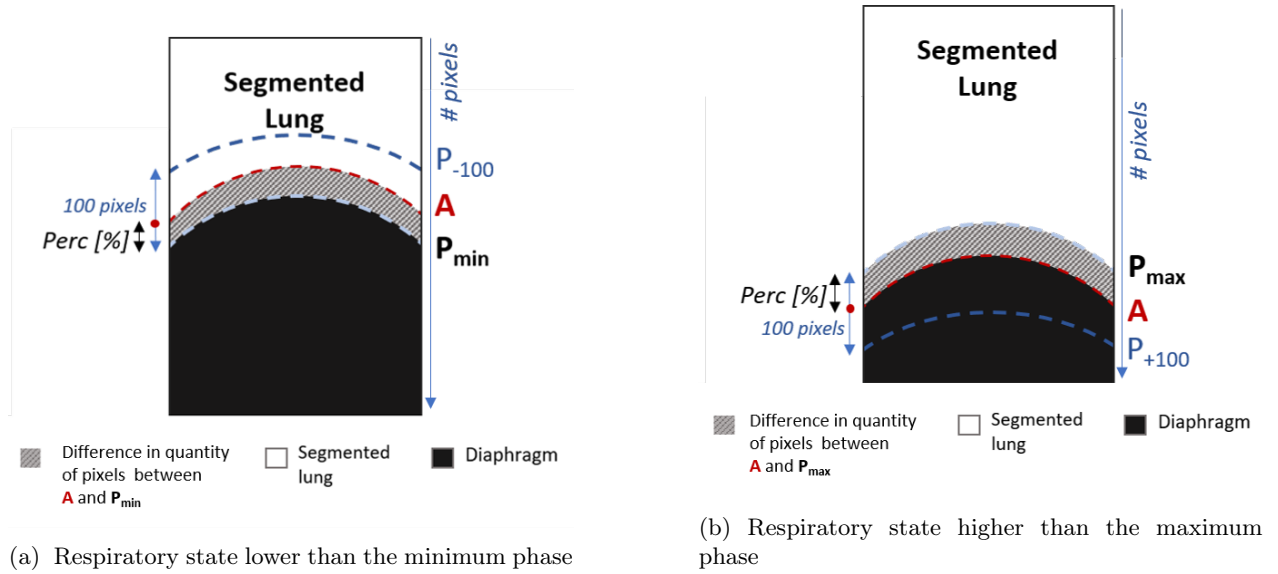


Figure 5.42: Example of a respiratory state positioned out of the planning phases

At this point, a phase interval as well as a percentage indicating how far the patient's respiratory state is within that interval is returned. These different pieces of information will help determine which reconstructed 3D image should be chosen to best represent the patient's current respiratory state.

5.3.2 3D Image Selection Strategy

Based on the results just obtained regarding the current respiratory state of the patient, a strategy will be determined to select the best matching reconstructed 3D image. As explained in the previous section,

the metric returns, for each iteration, two phases and a percentage. The 3D image selection strategy will be divided into two parts. On the one hand, if in the phases 1 (P_{sel}) and 2 ($P_{sel\pm1}$) returned by the metric there are no artificial phases of $P_{\pm100}$ then the patient is breathing in the planning range and we are in interpolation. On the contrary, if phase 2 ($P_{sel\pm1}$) is one of the two artificial phases $P_{\pm100}$, we are necessarily outside the planning range and we are in extrapolation. As the percentage will then be used a little differently in these two cases, we will treat these parts independently. The 3D image selection strategy process is illustrated below (figure 5.43).

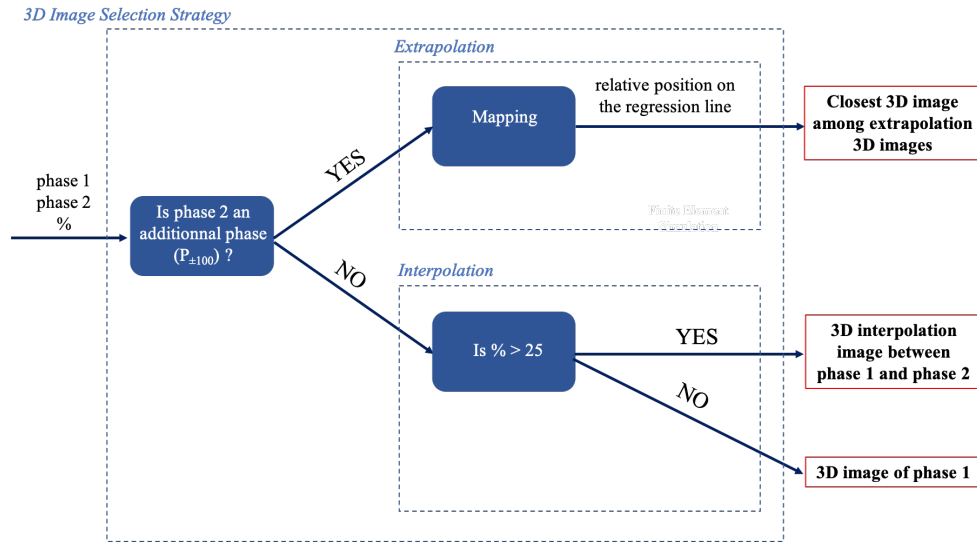


Figure 5.43: 3D Image Selection Strategy

Interpolation

The interpolation selection process is very simple. The 10 intermediate phases have been prepared for each phase of the planning cycle. These phases are located halfway between 2 successive phases of the cycle. The percentage returned by the metric is the difference (#pixels) between the current state and phase 1 compared to the difference between phase 1 and phase 2. As phase 1 is always the closest to the current state, the percentage will never exceed 50% since otherwise it would be closer to phase 2 than to phase 1 and therefore phase 2 would become phase 1 and percentage would become $100 - perc$. All this to say that we will have only 2 cases to deal with. If the percentage is higher than 25%, the current state is closer to the middle of phase 1 and 2 than phase 1. We return the intermediate 3D image of the two phases ($\frac{phase1+phase2}{2}$). If the percentage is less than 25%, we are closer to phase 1. We return the 3D image of this one.

Extrapolation

The metric gives as output for the extrapolations the phase to be extrapolated and a percentage. This percentage represents the relative position (in number of pixels) of the detected extrapolation between the extreme planning phase in the same direction and an artificial extrapolation phase of 100 additional pixels. We had to find a mapping between the percentage returned by the metric and the ratio to be used to

choose the right extrapolation, among the 4 available on the same side. The mapping is done by doing a normalization, i.e. by matching the extreme phases of the planning scales of the model and the metric. The scale of the model is therefore the scale obtained in the section 5.2.3 above. It is the scale of the projections of the liver center of mass positions for the different planning phases and for the midP. The scale of the metric is the scale of the number of pixels corresponding to each planning phase. This is illustrated in Figure 5.44 below. In green, the scale of the metric where the n_p represent the pixel counts corresponding to each planning phase. In blue, the scale of the model where the x_p are the relative projection positions of the liver COM for each planning phase. The red point corresponds to any extrapolation detected by the metric. Knowing the percentage, we can know its corresponding number of pixels n_{p50ext} (Eq. (27), left). Then, with this number of pixels and the normalization relation (Eq. (27), right), we can know the position $x_{p50'*R}$ of this extrapolation on the model scale. It is enough to choose the extrapolated phase that is closest to this position. In the example shown, the chosen extrapolation will be the $x_{p50'125\%}$, since it is the closest to $x_{p50'*R}$.

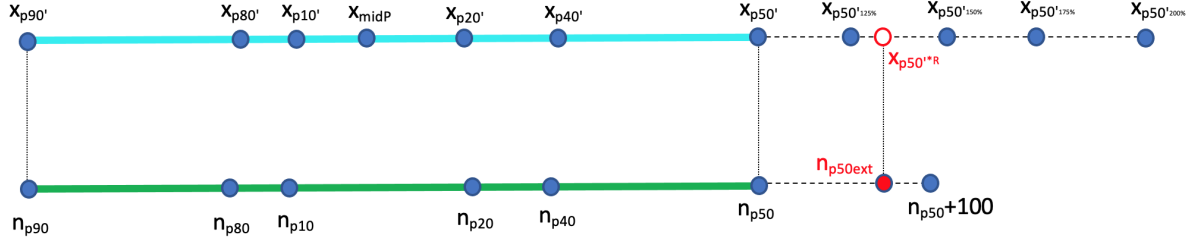


Figure 5.44: Mapping between the planning metric scale (green) and the finite element model scale (blue), i.e. positions on the regression line of planning liver COM.

$$perc = \frac{n_{p50ext} - n_{p50}}{n_{p50+100} - n_{p50}} \quad \frac{n_{p50ext} - n_{p90}}{n_{p50} - n_{p90}} = \frac{x_{p50'*R} - x_{p90'}}{x_{p50'} - x_{p90'}} \quad (27)$$

6 Validation Results and Discussion

Next, we sought to validate the methods developed in the previous section. We will present the different validation methods developed for each part individually (finite element model and metric) followed by an interpretation of their results. Finally, the validation method and results of the global tumor tracking process, which combines the metric with the finite element model simulations, will be presented and analyzed.

6.1 Tracking the Respiratory State

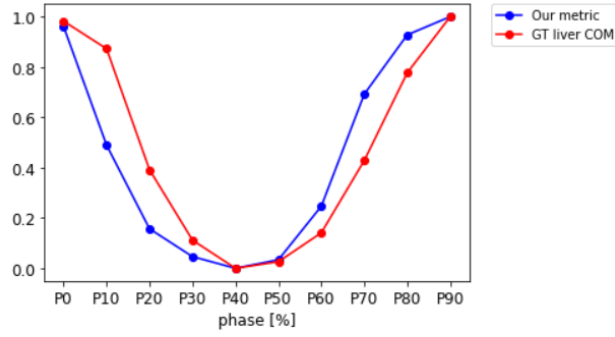
This section aims to verify the accuracy of the technique we used to determine the patient's respiratory state, which is the first step of our method. To do so, one can compare, first for the **planning data** :

- The 10 reference metrics obtained with our method and forming the **metric curve for the planning**
- The 10 positions of the liver center of mass (COM, see section 6.3) on z axis (according to axis described in section 4.2) for each image in the planning 4DCT, forming what we will call the **ground truth (GT) curve for the planning**

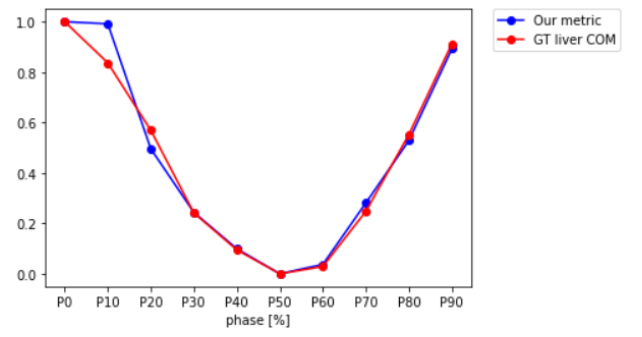
Indeed, the movement of the lung should be synchronized to the liver movement. When the lung inflates our metric increases, since there are more pixels in the lung region, and the liver is pushed down so its z -position decreases. That being said, we know that our metric curve should be opposite to GT curve for planning but should still show the same kind of movements. To make it easier to interpret, the GT curve for planning will be inverted ($-z$ direction rather than z direction), and both the metric curve for planning and GT curve for planning will be normalized between 0 and 1. These curves are represented in figure 6.1 for 3 different patients. Thus, the phase presenting the minimum value for one curve must correspond to that presenting a minimum on the other curve (same for the phase presenting the maximum value).

Regarding the results for the planning sequences of patient 2 and patient 3, it clearly appears that the two curves behave in a fairly similar manner. For all patients minimum and maximum of both GT curve and metric curve occurs at the same phase. Nevertheless, we can observe a significant difference in the case of patient 1.

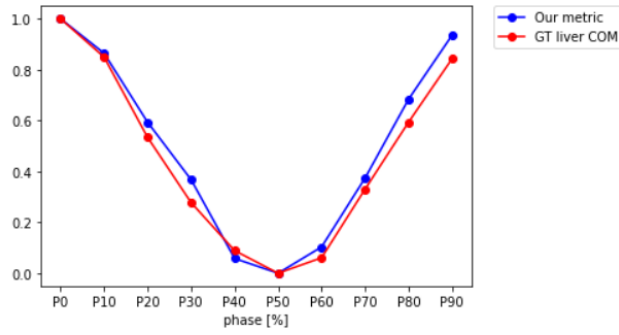
Indeed we can see in figure 6.1a that the blue curve (metric curve for planning) decreases faster at the beginning, for phases between phase 0 and phase 40, than the red curve (GT curve for planning). From the minimum point, the blue curve increases also faster than the red one. One thing to keep in mind is that the liver stands under the right lung, since our metric is computed based on the lower part of the left lung. Then it can be interesting to visually verify what are the main differences existing between the left and the right lung. Figures 6.2 and 6.3 below show 3DCT coronal views corresponding to phases 0 and 10 for the patient 1. On the latter, we can see the vertical difference and horizontal difference induced by respiratory movement from phase p0, to phase p10 respectively in figure 6.2 and in figure 6.3. The same observation can be made for these two figures. The observed difference, both vertical and horizontal, is smaller in the case of the right lung (related to the liver) compared to the left lung (the bottom of which is used to calculate our metric). The vertical difference from p0 to p10 indicates an upward movement of the lung (an exhalation,



(a) Planning sequence - Patient 1



(b) Planning sequence - Patient 2



(c) Planning sequence - Patient 3

Figure 6.1: Comparison between our metric and GT liver COM for the planning sequence and for both 3 patients

the lung deflates). In the case of the left lung, this movement is greater than in the case of the right lung. This implies a greater reduction in the metric since it is calculated on the basis of the pixels included in the region of the bottom of this lung. The horizontal movement from p0 to p10 shown in figure 6.3 is due to the crushing of the heart against the diaphragm during expiration. This is almost zero in the case of the right lung while it is not negligible in the case of the left lung. This also implies a greater narrowing of the lower section of the left lung. These two observations could explain why the decrease from phase 0 until the end of expiration (phase 40) is greater than that of the position of the liver COM observed on the red curve which is linked to the bottom of the right lung.

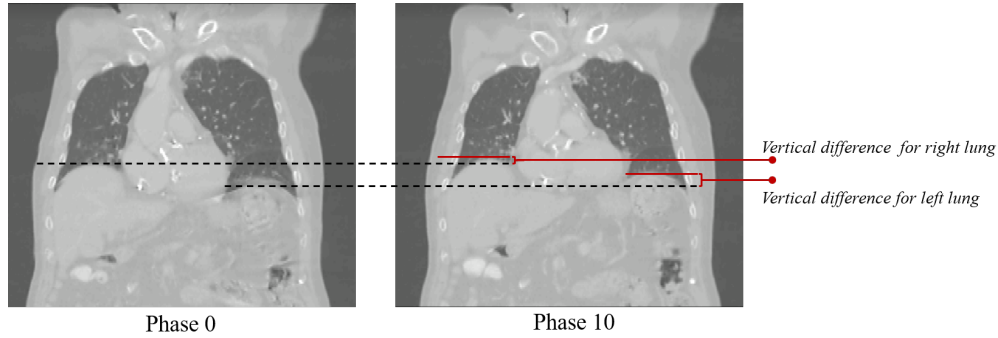


Figure 6.2: 3DCT coronal view of phase 0 (top) and phase 10 (bottom) from the planning sequence of patient 1 - vertical movement of the lungs

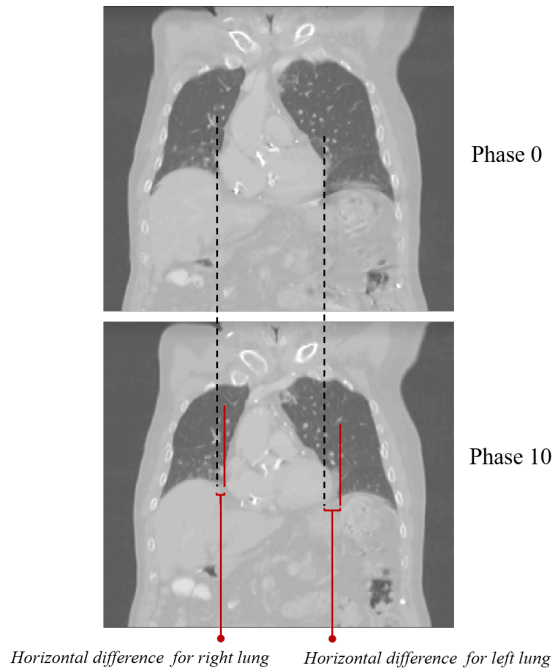


Figure 6.3: 3DCT coronal view of phase 0 (top) and phase 10 (bottom) from the planning sequence of patient 1 - horizontal movement of the lungs

The reverse reasoning may explain why the curve of the metric, from phase 50 to phase 90 and therefore during inspiration, increases faster than the red curve related to the center of mass of the liver.

Now that we have explored the behavior of our metric for the planning sequence we can do the same kind of validation test on the **continuous sequence**. This gives the results depicted in the figure 6.4 below.

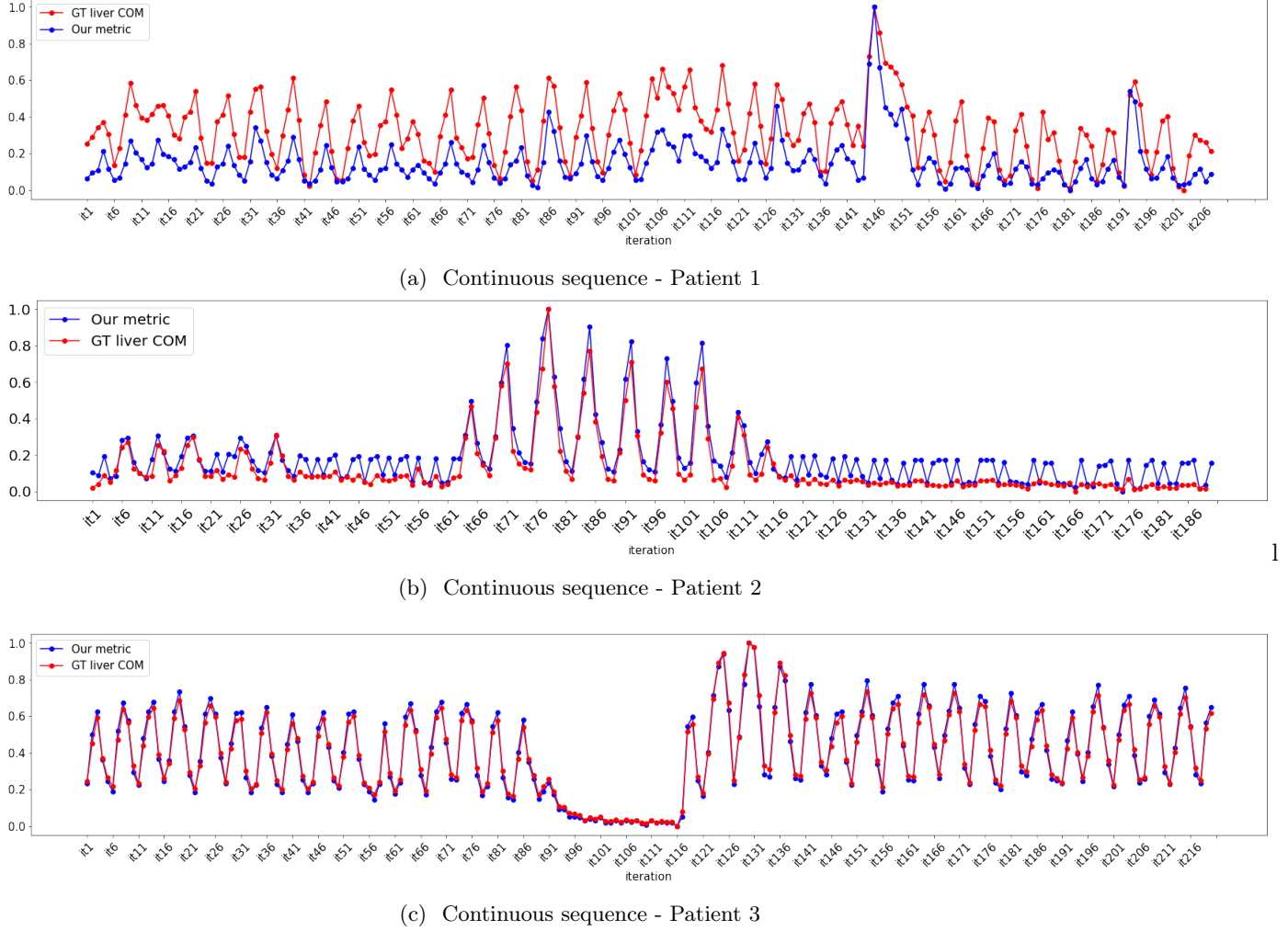


Figure 6.4: Comparison between our metric and GT liver COM for the continuous sequence and for both 3 patients

The results seem excellent for patient 3, indeed the two curves overlap almost perfectly. On the other hand, we can observe quite significant differences in the case of patients 1 and 2.

Concerning the second patient (figure 6.4b), we can observe that the liver does not always follow the pulmonary movement described by our metric : the metric indicates a slight movement of the lung while the liver does not seem to move (for example for the last iterations). There are several hypotheses to explain it. The first reason could be that the patient is not breathing deeply. The short breaths would only involve small movements and the diaphragm would remain in a rather high position. This would mean that it does not even touch the liver, or just a little, and therefore does not make it move. Another possibility would be that the force induced by small lung movements on the liver is not sufficient to counteract the static frictional forces. Finally, since the cropped cut for patient 2 is made on the left lung, the same reasoning we did to explain the differences between our metric curve and the GT curve for planning can also be done here.

For the first patient (figure 6.4a), it seems to be the opposite. The liver indicates higher movements than the lung. However, this could be due to the normalization which forces the higher points of the two curves to match. This would mean that the peaks do not currently match. One possible explanation would be that a peak in breathing, which induces significant lung movement, can push the liver to a maximum point described by the patient’s anatomy. Once this point has been reached, it is no longer possible to go any further, the liver is therefore crushed and not displaced. This assumption can be verified by normalizing the curve to a smaller peak rather than the highest. The results of this verification can be found in the figure 6.5. Indeed it seems to improve the correspondence between the curves quite well.

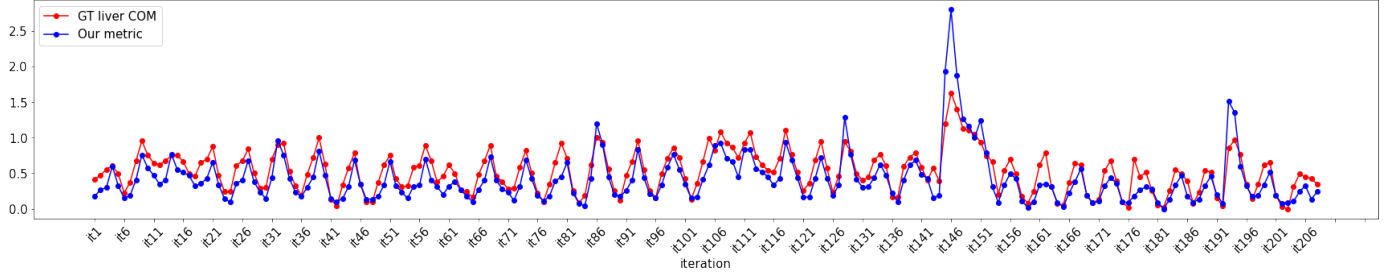


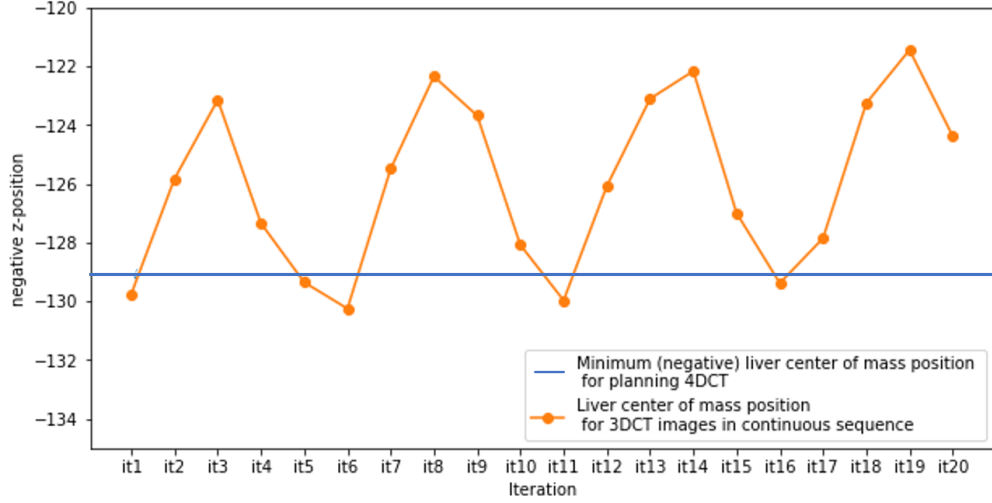
Figure 6.5: Hypothesis validation for patient 1

Finally a hypothesis that would explain why the results for patient 3 are much better could be that this is the only patient for which the cropped cut is made on the right lung, which is directly linked to the liver located below.

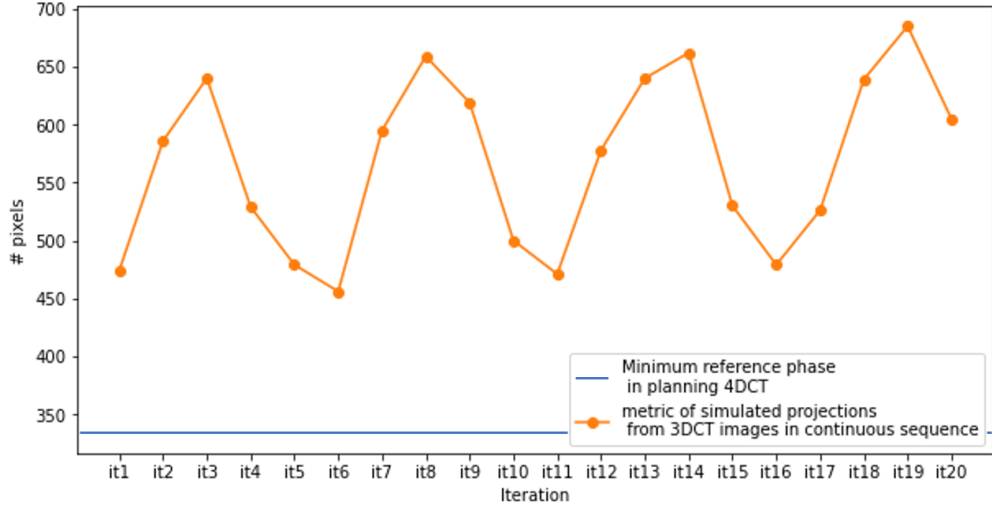
While executing these validation tests, it appeared to us that there could be a significant difference between the relation linking the GT curve for planning and the GT continuous curve (based on the GT liver COM) and that linking our two curves (based on our metrics). Our planning curve was sometimes too high, or too low, compared to the continuous curve. This is illustrated in figure 6.6. Since the graph drawn in figure 6.6a is considered to be the ground truth (obtained with reconstructed 3DCTs being more detailed), our metric must lead to obtaining the same relationship between planning and continuous results as what is obtained in the case of liver COM. To achieve this objective, an offset will be applied for subsequent results (For example in the case of patient 1 visible in figure 6.6 we will apply an upward offset to the planning sequence). At this point, this offset is calculated based on a point in the continuous sequence. However, this data will not be available in the real context, to deal with it another idea will be presented in section 7.

6.2 Noise Robustness

As mentioned in section 3.1.4, noise is omnipresent in X-ray imaging, especially in the case of fluoroscopy due to the lower energy used. Poisson noise appears to be the most predominant and difficult to manage. This is why it can be interesting to test the Poisson noise robustness of our method. To do so, Poisson noise can be added to our projections by using a function from the Tomopy library [83]. The latter applies a Poisson distribution function (from *numpy* package [104]) over the entire image given in argument, pixel by pixel. The value used as the interval expectation of the distribution corresponds to the value of the pixel on which it is calculated. This means that a Poisson distribution is calculated for each of the pixels. For a given pixel, the distribution is calculated and then one of the values from it is chosen randomly to replace



(a) Bottom position of the liver center of mass during the planning in relation with the liver center of mass for the 20 first images in the continuous sequence



(b) Bottom position of the planning curve in relation with the 20 first metrics computed on the continuous sequence

Figure 6.6: Offset between our metric and the liver center of mass

the initial value of that pixel. The result of this manipulation is given on figure 6.7.

We can verify the impact of Poisson noise by comparing the obtained metric (pixels count) for the continuous sequence of each patient, this is illustrated in figure 6.8. On each of these 3 sub-figures we distinguish 3 different curves: one corresponds to the metric obtained without noise (in blue), and the two others correspond to the noisy image with (in green) and without (in yellow) the use of Gaussian filter. Although this Gaussian filter is initially used in our method in the pre-processing part, it was interesting to see how it could help in the case of noisy images by computing the metric without it. The impact of noise differs from one patient to another, it depends mainly on the chosen crop. Nevertheless, it clearly appears that the noise can be sufficiently corrected to obtain results similar to those of a noiseless image with a simple Gaussian filter.

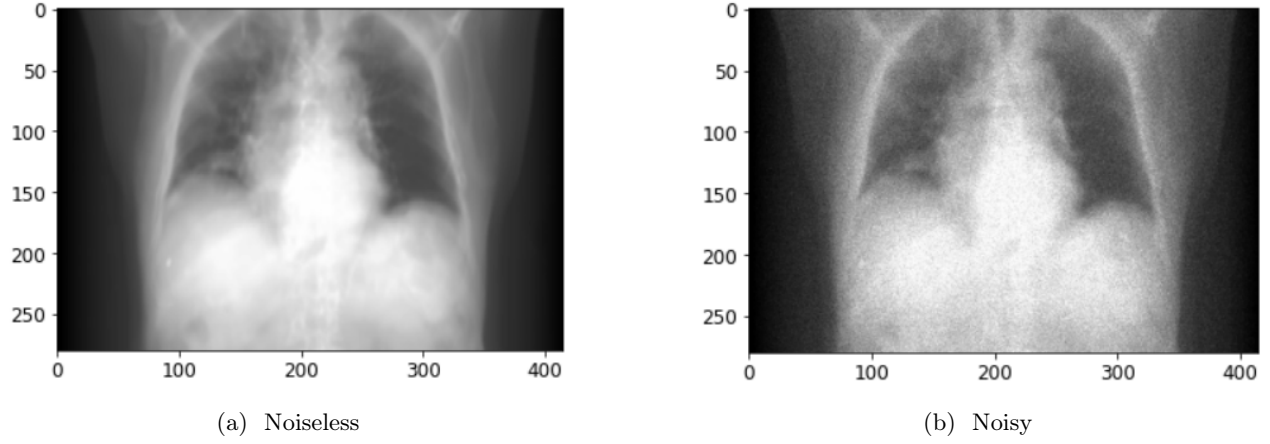


Figure 6.7: Iteration 0 from the continuous sequence of patient 1 : (a) without and (b) with noise from left to right respectively.

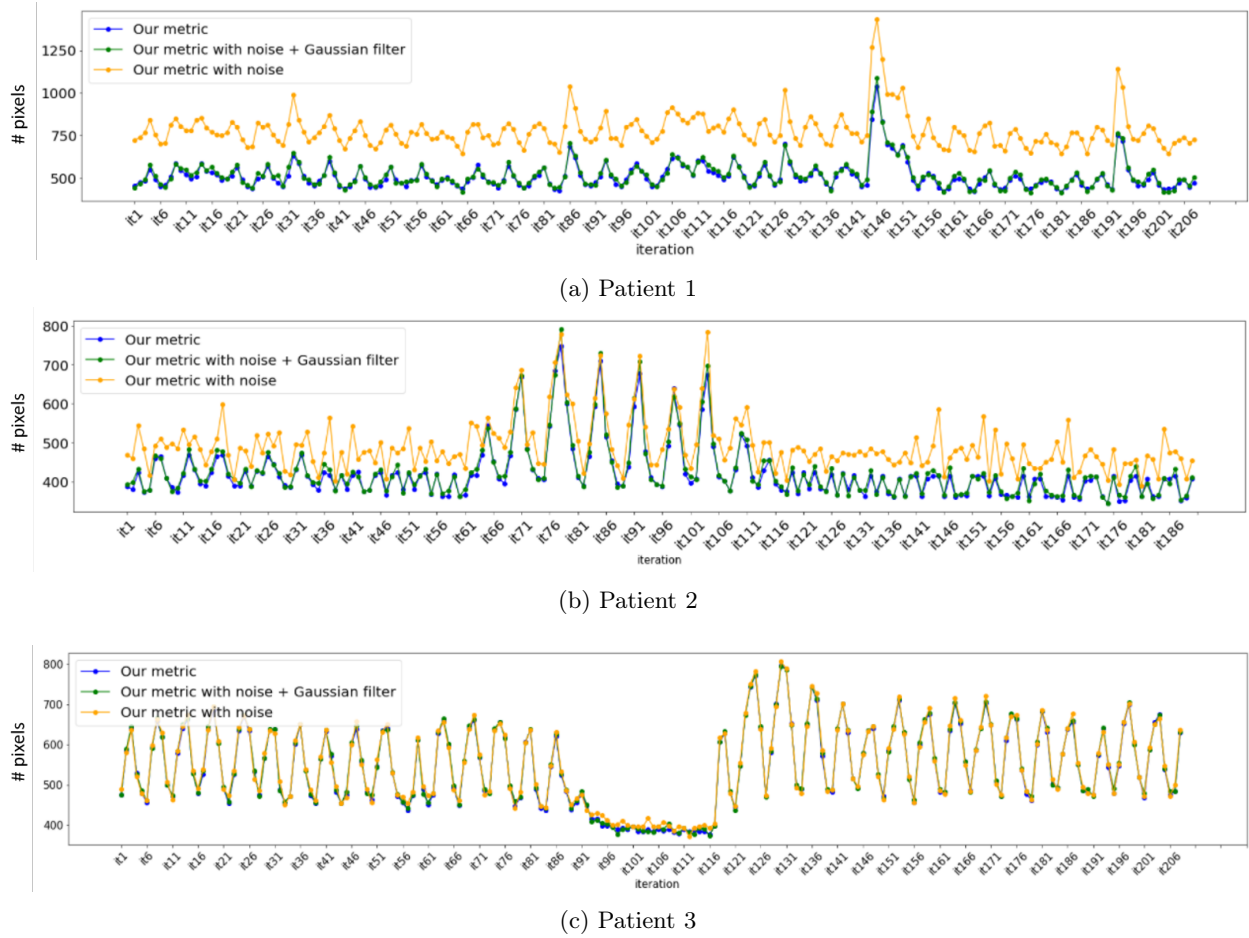


Figure 6.8: Noise impact on metric computation for the 3 patients continuous sequence

6.3 Finite Element Model

In this section, we tried to quantify the accuracy of the finite element model, here in particular, for the modeling of the 10 phases of the planning 4DCT. We also compared the results of the 2 biomechanical models presented in section 3.3. Then, we gave an interpretation and a discussion of those results. Therefore, we ran the finite element model for patients 1, 2, and 3 for each phase of the planning 4DCT and for each biomechanical model. As an example, the results of the simulations obtained for the phases 0 and phase 50 of the three patients for the Linear Elastic model can be seen in figures 6.9, 6.10 and 6.11.

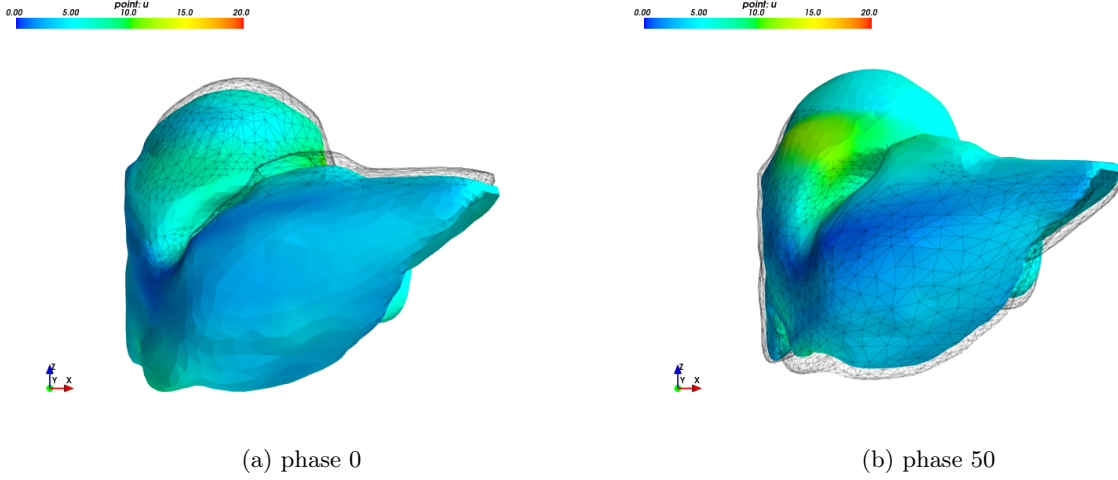


Figure 6.9: Simulation Results for opposite phases of patient 1. Color displacement scale with respect to the midp mesh in [mm].

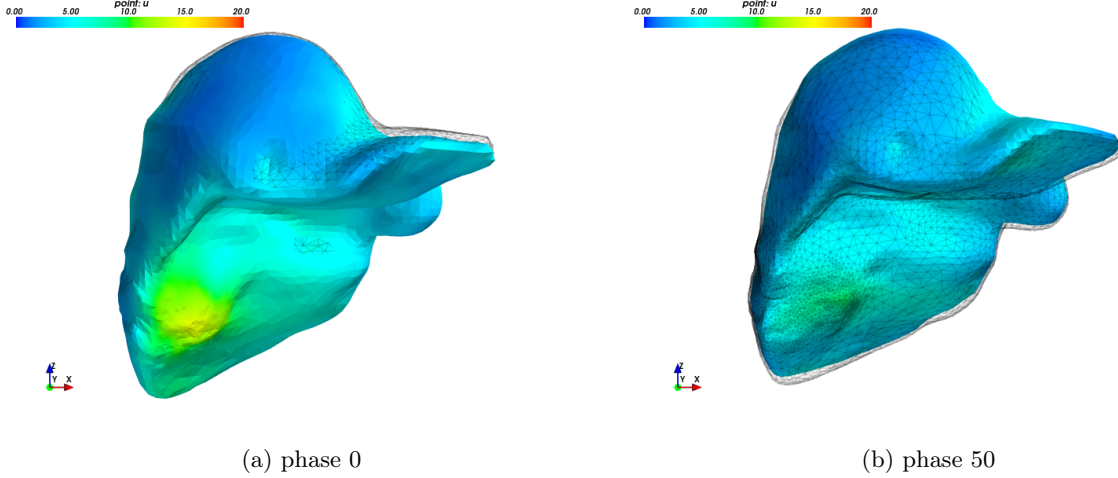


Figure 6.10: Simulation Results for opposite phases of patient 2. Color displacement scale with respect to the midp mesh in [mm].

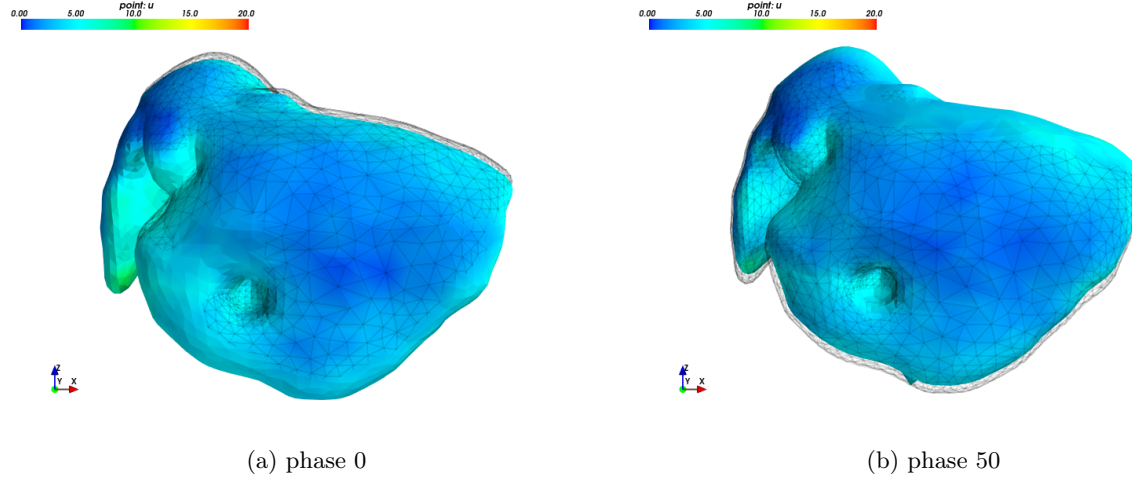
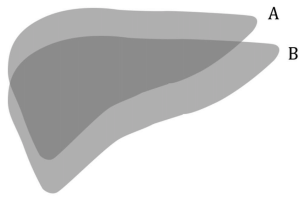


Figure 6.11: Simulation Results for opposite phases of patient 3. Color displacement scale with respect to the midp mesh in [mm].

From all these results we have reconstructed 3D images (section 5.2.4), and from these 3D images, which are 3D numpy arrays filled with HU values, we could easily extract tumor and liver masks. We then needed a method to evaluate the accuracy of this finite element model. Two approaches were chosen: calculate the **DICE coefficient** and compute the **center of mass error (COME)**.

The DICE coefficient is a statistical number between 0 and 1 that quantifies the similarity between two samples. The samples we compared were the *ground truth* liver and tumor volumes (in the form of masks) of the planning, and the liver and tumor *simulated* volumes of the corresponding phase. So here the DICE coefficient aims to verify that the *simulated* liver (or tumor) position and shape is close enough of the *ground truth* liver (or tumor). Concretely, we took the *ground truth* 3D mask from the 4DCT of planning, and we compared it with the 3D mask reconstructed from the simulation, using the DICE formula (28).



The DICE coefficient measures the superposition of volumes. The more they are superimposed, the closer the simulation is to reality and the closer the index will be to 1. On the contrary, if the volumes have no intersection, the DICE coefficient will be 0.

$$DICE = \frac{2|A \cap B|}{|A| + |B|} \quad (28)$$

Figure 6.12: Liver volumes to compare [105]

It is important to recall how these masks are obtained. The so-called *ground truth* masks are obtained by deforming the masks of the midp, using the deformation fields of the midp towards the 10 phases (section 4.1). These are our 'exact' reference masks. But the masks of the simulations have been extracted from a 3D image reconstructed from the mesh. These are the masks produced by the method.

Below we can find an example of the masks we were able to extract for the phase 0, for patient 1 and for the Linear Elastic Model. In figure 6.13, we see the mask of the ground truth liver (deformed from the midp image), in figure 6.14 it is the mask of the liver reconstructed from the simulation mesh. And in the figure

6.15 we have superimposed these masks to show the kind of matches we have obtained. Attention, these are 3D masks that we are comparing, we have put here 2D slices of the results for more clarity.

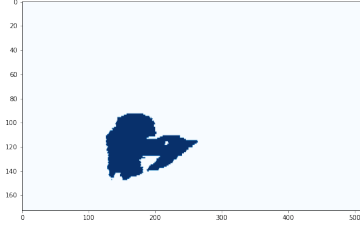


Figure 6.13: Patient 1, p0, GT liver mask, LE

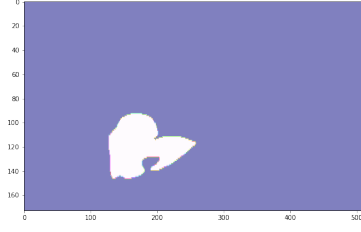


Figure 6.14: Patient 1, p0, FE liver mask, LE

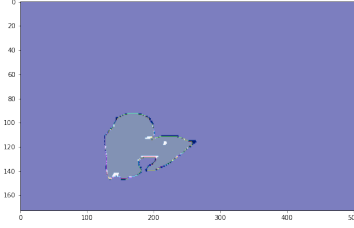


Figure 6.15: Patient 1, p0, GT+FE liver mask, LE Model

The other comparison tool that is used in this work is the center of mass error (COME). It is the absolute distance between the *ground truth* center of mass and the *simulated* center of mass of regions (again liver or tumor). The reason why we wanted to compute this distance in addition to the DICE coefficient is that it allows checking alignment of masks.

To obtain the center of mass, we have made the assumption that the densities of the masks were constant (which is plausible since it is composed of the same biological tissue). Then, knowing that masks are composed of an integer and finite number of voxels, which all have the same size, we could approximate the mass center of masks by their geometric center, which is simply the average point of positions of voxels. We applied the following formula :

$$COM = \left[\frac{\sum_{i=0}^N x_i}{N}; \frac{\sum_{i=0}^N y_i}{N}; \frac{\sum_{i=0}^N z_i}{N} \right] \quad (29)$$

Where x_i , y_i and z_i are the coordinates of the mask voxels, N is the number of voxels belonging to the mask.

To compare the results of our 2 biomechanical models, we first compared the DICE coefficients for the liver and tumor mask and for the two models. In the tables below, are the results of the DICE coefficients for the three patients, for the tumor and liver masks and for the Linear Elastic and Mooney-Rivlin models.

phase	Linear Elastic		Mooney-Rivlin	
	DICE liver	DICE tumor	DICE liver	DICE tumor
p0	0.9631	0.8476	0.9631	0.8461
p10	0.9628	0.8587	0.9628	0.8584
p20	0.9631	0.8522	0.9631	0.8520
p30	0.9641	0.8659	0.9641	0.8638
p40	0.9640	0.8595	0.9640	0.8574
p50	0.9625	0.8564	0.9625	0.8536
p60	0.9639	0.8520	0.9639	0.8509
p70	0.9633	0.8355	0.9633	0.8352
p80	0.9629	0.8378	0.9629	0.8375
p90	0.9621	0.8390	0.9621	0.8390

Table 7: DICE coefficients for patient 1

phase	Linear Elastic		Mooney-Rivlin	
	DICE liver	DICE tumor	DICE liver	DICE tumor
p0	0.9549	0.8405	0.9549	0.8375
p10	0.9546	0.8386	0.9546	0.8384
p20	0.9552	0.8216	0.9552	0.8222
p30	0.9562	0.8361	0.9562	0.8354
p40	0.9587	0.8370	0.9587	0.8368
p50	0.9585	0.8384	0.9585	0.8367
p60	0.9562	0.8339	0.9562	0.8332
p70	0.9575	0.8319	0.9575	0.8315
p80	0.9559	0.8360	0.9559	0.8356
p90	0.9555	0.8452	0.9555	0.8435

Table 8: DICE coefficients for patient 2

phase	Linear Elastic		Mooney-Rivlin	
	DICE liver	DICE tumor	DICE liver	DICE tumor
p0	0.9587	0.7131	0.9587	0.7081
p10	0.9582	0.7128	0.9582	0.7100
p20	0.9578	0.7296	0.9578	0.7286
p30	0.9588	0.7496	0.9588	0.7464
p40	0.9596	0.7547	0.9596	0.7520
p50	0.9596	0.7543	0.9596	0.7523
p60	0.9590	0.7514	0.9590	0.7501
p70	0.9591	0.7211	0.9591	0.7202
p80	0.9581	0.6973	0.9581	0.6963
p90	0.9581	0.7013	0.9581	0.6947

Table 9: DICE coefficients for patient 3

And then we also compared the absolute distance between the mass centers of the *ground truth* and *simulated* masks (COME) obtained with formula in section 6.3.

	Linear Elastic		Mooney-Rivlin	
phase	COME [mm]	COME [mm]	COME [mm]	COME [mm]
p0	1.24	2.01	1.24	2.01
p10	1.30	1.06	1.30	1.06
p20	1.21	0.79	1.21	0.77
p30	0.92	0.86	0.92	0.86
p40	0.90	0.73	0.90	0.73
p50	1.01	0.60	1.10	0.56
p60	1.14	0.43	1.14	0.39
p70	1.14	1.20	1.14	1.18
p80	1.13	1.78	1.13	1.76
p90	1.27	2.25	1.27	2.22

Table 10: Absolute distance between mass centers of masks for Patient 1

	Linear Elastic		Mooney-Rivlin	
phase	COME [mm]	COME [mm]	COME [mm]	COME [mm]
p0	1.50	0.33	1.50	0.39
p10	1.55	0.17	1.55	0.18
p20	1.54	0.52	1.54	0.52
p30	1.40	0.96	1.40	0.95
p40	1.31	0.73	1.31	0.72
p50	1.29	0.80	1.29	0.73
p60	1.31	0.67	1.31	0.61
p70	1.28	0.53	1.28	0.50
p80	1.38	0.46	1.38	0.47
p90	1.47	0.46	1.47	0.51

Table 11: Absolute distance between mass centers of masks for Patient 2

	Linear Elastic		Mooney-Rivlin	
phase	COME [mm]	COME [mm]	COME [mm]	COME [mm]
p0	0.88	0.65	0.88	0.70
p10	0.93	0.84	0.93	0.87
p20	1.03	0.45	1.03	0.45
p30	0.98	0.96	0.98	0.90
p40	0.94	0.88	0.94	0.83
p50	0.95	1.06	0.95	1.02
p60	0.94	0.50	0.94	0.49
p70	1.05	1.10	1.05	1.09
p80	0.97	2.04	0.97	1.96
p90	0.95	1.43	0.95	1.49

Table 12: Absolute distance between mass centers of masks for Patient 3

As one can observe in the tables above, the method obtained excellent DICE coefficients for the liver and for all 3 patients (greater than 0.95). This seems perfectly normal when one remembers the masks used for

comparison. On the one hand, the *ground truth* mask is deformed from the midp mask using the deformation field obtained with the registration algorithm (section 5.2.2). And on the other hand, there is the *simulation* mask, reconstructed from the mesh, mesh which is also built from the midp mask. And this mesh is deformed thanks to its boundary condition which is the same deformation field as for obtaining the *ground truth* mask. The difference is the following: in the case of the *ground truth*, the deformation field is applied on the whole mask while for the *simulated* mask, the deformation field is applied only on the contours. But this difference is not seen for the liver since the internal arrangement of the nodes, which is governed by biomechanical equations, is not taken into account in the case of a mask. We thus obtain exactly the same masks for the 2 models, since they have the same boundary conditions (but not the same biomechanical equations). And these masks are also the same as the *ground truth* masks, with a very small difference due to smoothing (explained later). But the position of the simulated tumor inside the liver will depend on the model since it is inside the liver and therefore no boundary condition is directly applied to it. Its displacement will be due to the movement of the liver and to the solutions of the biomechanical equations.

Thus what is interesting here are the tumor DICE coefficients, because it will be representative of the accuracy of our model. We obtain fairly good DICE for patients 1 and 2 (around 0.83) and correct DICE for patient 3 (between 0.7 and 0.75). In general, we obtain slightly smaller DICE than for liver.

It is interesting to look at the results of COME. We see that this distance is in most cases less than 2 mm, which corresponds to the height of a voxel. It is therefore excellent COME results. There is no significant difference between the liver and the tumor. This means that in both cases, the simulated masks are well aligned with the ground truth masks. The model succeeded in positioning the tumor correctly with respect to the phase. The difference between the tumor and the liver can be explained as follows: when smoothing the meshes (section 5.2.1), the angularities are flattened, which can be seen as removing a thin layer all around the mask. The same layer thickness was removed from the liver mask as for the tumor mask. But the tumor is smaller than the liver. The proportion of tumor removed was larger, giving a lower DICE (Figure 6.16).

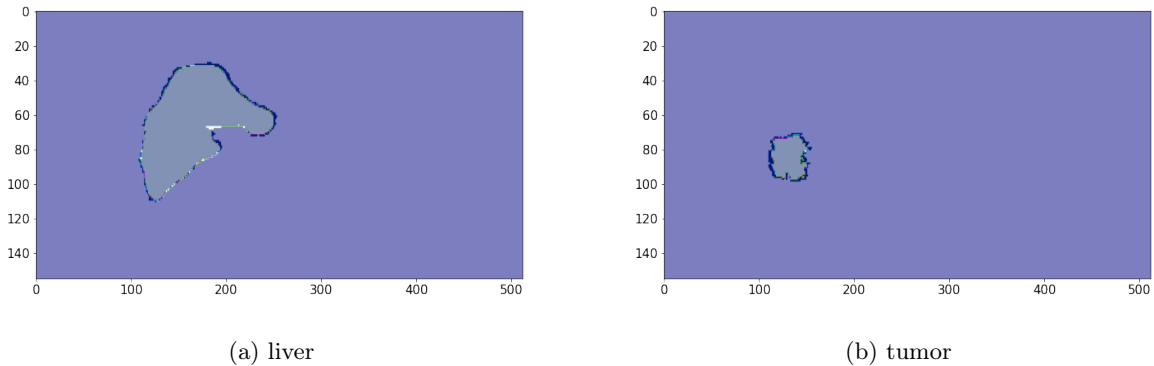


Figure 6.16: Masks superposition of ground truth (blue) and simulation (white) for liver and tumor of patient 2

It is also important to remember that the reference mask, i.e. the *ground truth* mask, is a mask distorted from the midp mask, drawn by the doctor. The reason of that is because it is difficult to automatically segment the tumor mask of the phases since it is difficult to locate it in the liver tissue. Indeed its contrast is very close to that of the liver in a 4DCT. Therefore, one cannot be sure that the mask really targets the

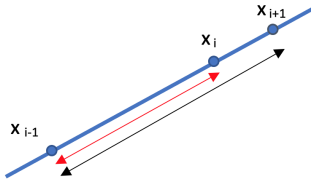
tumor. So, although they have been taken as a reference for comparison, the *ground truth* tumor mask does not necessarily represent reality since it is derived from a deformation and this deformation does not take into account the intrinsic characteristics of the tissue materials.

There is no significant difference between the two models. They have the exact same DICE coefficients for liver masks since it is the same mesh with the same boundary conditions, meaning the same liver mask extracted. And they have almost the same DICE for tumor masks, sometimes the elastic model shows a little better results. Regarding the alignment, it is good in all cases, proving the efficiency of both models. As we could not prove a better efficiency of one of the two models, we considered from this point only the Linear Elastic model for its better speed of execution.

6.4 Interpolation and Extrapolation Method

After that, we wanted to verify the accuracy of the interpolation and extrapolation methods. Obviously, since these methods create phases that do not yet exist, it is not possible to verify if they are correct using the ground truth. A simple way to get around the problem is to delete one phase of the cycle and try to get it back with interpolation or extrapolation. We can then compare this recreated phase with the corresponding phase of the 4DCT. If the deleted phase is one of the extreme phases (typically phase 0 and phase 50) we will use extrapolation, while if another phase is removed then we will use interpolation. The lowest phase is p90 for patient 1 and p0 for patients 2 and 3, while the highest phase is p50 for all patients. This is based on the results of the liver center projections on the regression line (section 5.2.3).

In the case of interpolation, the metric returns two phases and a percentage that quantifies the relative position of the current respiratory state between these two phases. From this percentage, it is possible to know the position x_{pi} of the current respiratory state on the regression line (Eq. 30). For this validation we did not compute the metric, we simply assumed that the percentage sent by the metric would have allowed us to obtain the position x_{pi} on the regression line (which we already know). Since the phase p_i has been removed, we must approximate the deformation field to be applied. The percentage allows us to interpolate the deformation fields of the adjacent phases. In this case of validation, we calculated the percentage ($p\%$) from the position x_{pi} on the regression line (Eq. 30), but normally this percentage is provided by the metric and allows to find the x_{pi} position.



$$p\% = \frac{x_{pi} - x_{p_{i-1}}}{x_{p_{i+1}} - x_{p_{i-1}}} \quad (30)$$

Where x are the relative positions of the deleted phase (i) and the adjacent phases ($i - 1, i + 1$).

Figure 6.17: Relative positions on regression line

Therefore, the new deformation field is obtained with :

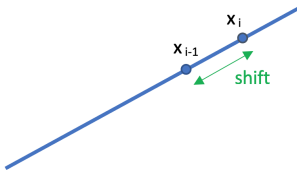
$$DF_{p_i} = (1 - p\%) * DF_{p_{i-1}} + p\% * DF_{p_{i+1}} \quad (31)$$

Then we could run the simulation with the new deformation field. We chose to retrieve two different phases with the interpolation method. The phase p10 which is close to an extreme phase (p0) and the p70 phase which is rather in the middle of the planning range (thus close to the midPCT). The masks extracted from these new phases p10 and p70 were compared with the GT masks of these phases. We then obtained the coefficients of DICE and COME. Below, the results for all patients of these phases we try to retrieve (table 13).

Patient	phase	DICE liver	DICE tumor	COME [mm]	COME [mm]
1	p10	0.9523	0.7859	2.21	3.73
1	p70	0.9584	0.8274	1.12	2.17
2	p10	0.9563	0.8238	1.43	2.72
2	p70	0.9520	0.8322	1.84	1.15
3	p10	0.9571	0.6947	1.66	2.92
3	p70	0.9575	0.7191	1.33	1.05

Table 13: Results for the validation of the interpolation method

For the extrapolation, the metric also returns a percentage that quantifies the relative position of the current breathing state. Then we perform a mapping (section 5.2.3) to transform this percentage into a position on the regression line. Again, we made the assumption that the metric would have given us a percentage that, after mapping, would have allowed us to find the x_{p_i} position. Since we did not run the metric in this validation, we skip this step and take directly the known position (from table 6) of the phase on the regression line. Thus, we must first calculate the shift (section 5.2.3). Since we have removed the extreme phase, the new extreme phase of the cycle is the second farthest from the midPCT point in the same direction. It is necessarily one of the phases adjacent to the removed phase. The shift is the difference between the new extreme phase and the phase we want to recover (Figure 6.18). The deformation field is simply the mean of the two adjacent phases deformation fields.



$$\begin{aligned} shift &= x_{p_i} - x_{p_{i-1} \text{ or } i+1} \\ DF_{p_i} &= \frac{DF_{p_{i-1}} + DF_{p_{i+1}}}{2} \end{aligned} \quad (32)$$

Where x are the relative positions of the deleted phase (i) and the adjacent phases ($i - 1, i + 1$).

Figure 6.18: Shift measurement

And here under the results for the extrapolation (table 14).

Patient	phase	DICE liver	DICE tumor	d liver [mm]	d tumor [mm]
1	p90	0.9579	0.8235	1.28	2.60
1	p50	0.9595	0.8566	1.28	0.79
2	p0	0.9583	0.8363	1.06	1.16
2	p50	0.9557	0.8374	1.48	0.97
3	p0	0.9580	0.7023	0.71	3.00
3	p50	0.9589	0.7572	1.05	0.96

Table 14: Results for the validation of the extrapolation method

For liver DICE, very good results are observed, close to those obtained with the normal cycle. For the DICE of the tumor, we also obtain good results overall, with always less good DICE for patient 3 (section 6.3). One small exception can be noted. The DICE of the interpolated phase p10, for the tumor of patient 1 is lower than the average of the normal cycle. When we look at the distance between the centers of mass or display the mask overlay, we can see that the tumor is slightly above where it should be (figure 6.19). And it is the same if you look at the liver mask. The

error could come from the deformation field. As we have removed the one of the p10 phase, we have made a weighted average with a percentage of the adjacent phases (p0 and p20). However, the deformation field obviously does not evolve in a linear or uniform manner between 2 phases of the respiratory cycle. Maybe that the non-linearity, or non-uniformity, occurring between p0 and p10 is particularly predominant.



Figure 6.19: Tumor masks of patient 1, interpolated p10

6.5 Complete Method

In this subsection, we will validate and present the results of the complete method that is intended to be used in real-time, which is the following: a radio projection is made on the patient using real-time fluoroscopy. The metric is computed on this image after a pre-processing (section 5.1.1). The latter is then used to estimate the respiratory state of the patient in real-time with the method described in the section REF. This estimation is given by an interval of two successive phases in which the respiratory state of the patient resides, and a percentage determining where the state lies exactly in this interval. In the next block, depending on the percentage obtained, a pre-calculated 3D model (section 5.2.4) will be chosen. This model can either correspond to a phase of the planning sequence, or to an inter/extrapolation of these.

First, the CT images of the 3 continuous sequences available had to be transformed into fluoroscopic images in order to reproduce the real situation. After that, the entire process can be performed on the continuous sequence for each of the 3 patients. For each image in the sequence, a reconstructed 3D image containing liver and tumor mask is returned. Then we compared the masks of these reconstructed images with the ground truth masks of the frame. The ground truth masks of the continuous sequence are simply obtained by deforming the masks of midp. The deformation fields used for this operation are the same deformation fields from the trackers to obtain each 3DCT image of the continuous sequence [82](see section 4.1). We were able to calculate the DICE (mean $\pm$ standard deviation) and COMEs (Center of Mass Errors, mean $\pm$ standard deviation) coefficients of the tumor and the liver for all frames of the three continuous sequences.

First, general results will be presented and discussed and then each patient will be examined individually to point out certain irregularities.

In the table 15 below are the results obtained for the 3 patients, with the continuous sequence. The continuous sequences consist of approximately 200 3DCT images (208, 190 and 220 for patients 1, 2 and 3 respectively).

Patient	frames	DICE liver	DICE tumor	COME [mm]	COME [mm]
1	208	0.9427 ± 0.0126	0.8176 ± 0.0547	1.67 ± 1.14	3.48 ± 2.11
2	190	0.9560 ± 0.0177	0.8755 ± 0.0333	2.17 ± 1.11	1.86 ± 1.48
3	220	0.9670 ± 0.0083	0.7906 ± 0.0446	1.29 ± 0.53	2.01 ± 1.12

Table 15: Summary of the results obtained for the continuous sequence

The three figures composing figure 6.20 show the spread of the DICE coefficients obtained for the continuous sequence comparing the tumor masks.

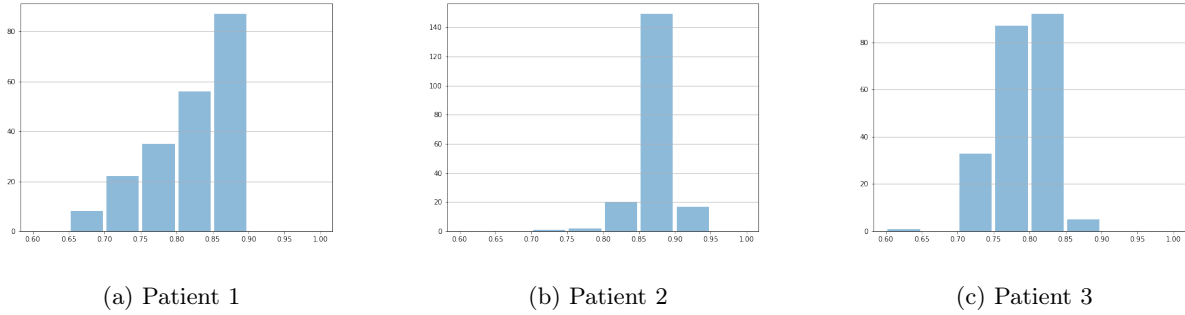
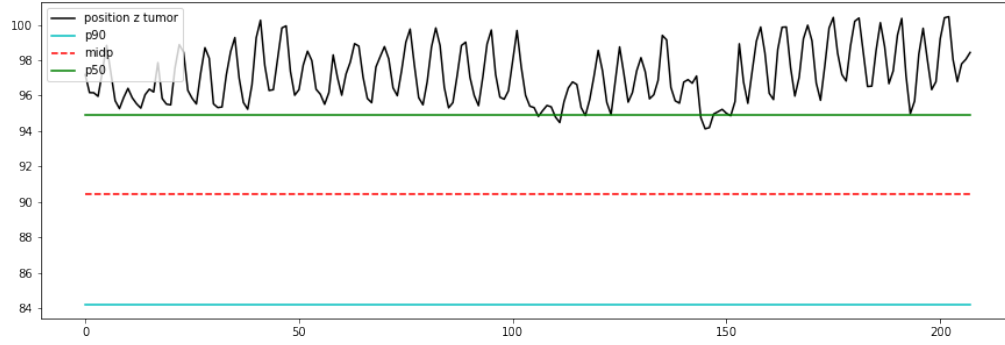


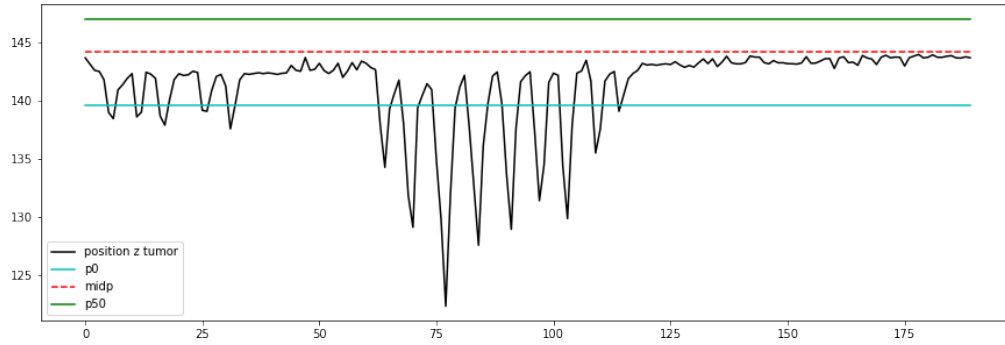
Figure 6.20: Histogram of the tumor DICE

In order to perform a complete analysis of the global method, we have generated different types of graphs: *breathing graphs*, *DICE graphs* and *absolute position error graphs*. These different graphs allow us to better understand the results obtained

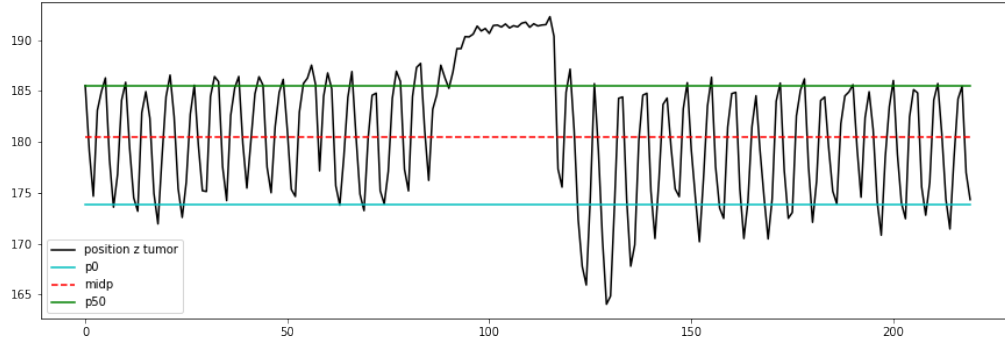
First, we find the graphs that we will call the *breathing graphs*. They represent the z-position of the tumor COM for the continuous sequence. On these graphs, the z-positions of the tumor COM for the extreme phases of the planning and for the midp are also shown. This allows to visually see how the patient breathes with respect to the continuous sequence. On the x-axis, the iteration number of the continuous sequence. On the ordinate, the z-position of the COM (in mm).



(a) Patient 1



(b) Patient 2

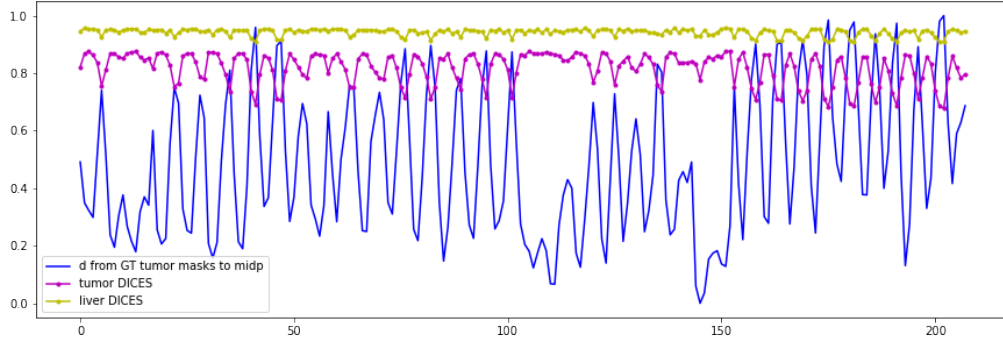


(c) Patient 3

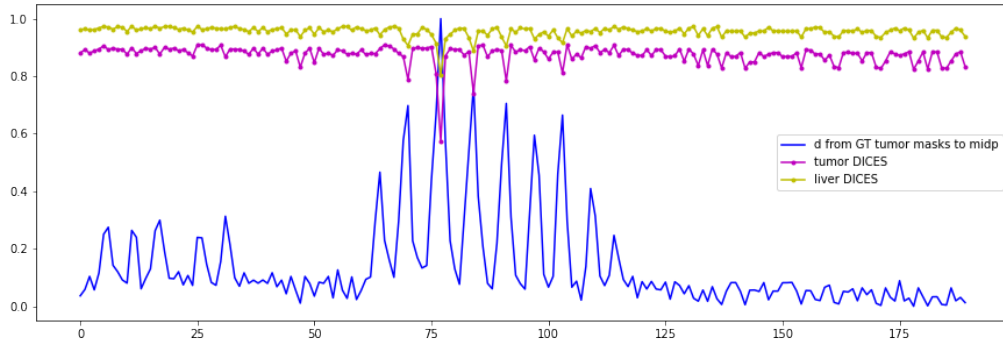
Figure 6.21: Respiration graphs of all patients. Black line: z-position of the GT tumor COM for the continuous sequence. In dotted red: z-position of GT tumor COM of midp. In green: z-position of GT tumor COM of highest phase. In blue: z-position of GT tumor COM of lowest phase.

The second type of graphs are the *DICE graphs*. They show, for each iteration of the continuous sequence, the DICE coefficients obtained (in yellow for the liver, in purple for the tumor). They are displayed parallel to the absolute distance of the GT tumor from the midp (normalized between 0 and 1). Thanks to them, we can see for which iterations the method predicts a bad position or shape of the tumor. We make a link with

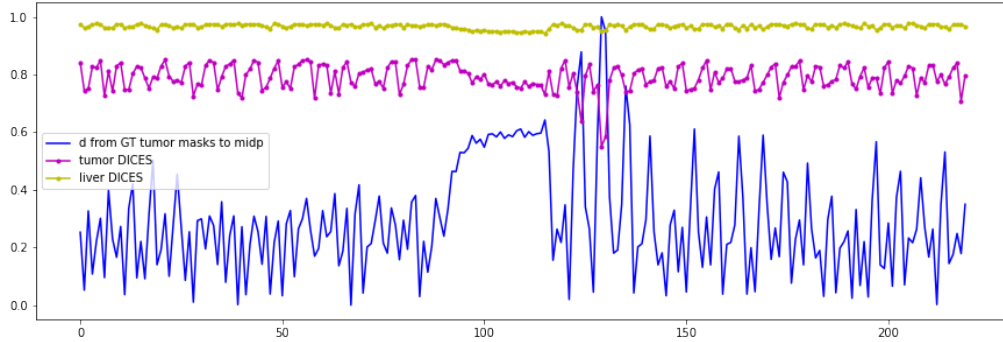
the absolute distance of the tumor from the midp.



(a) Patient 1



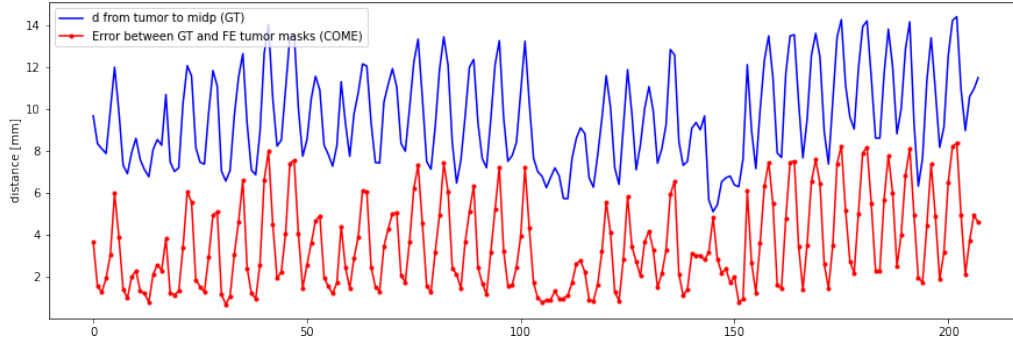
(b) Patient 2



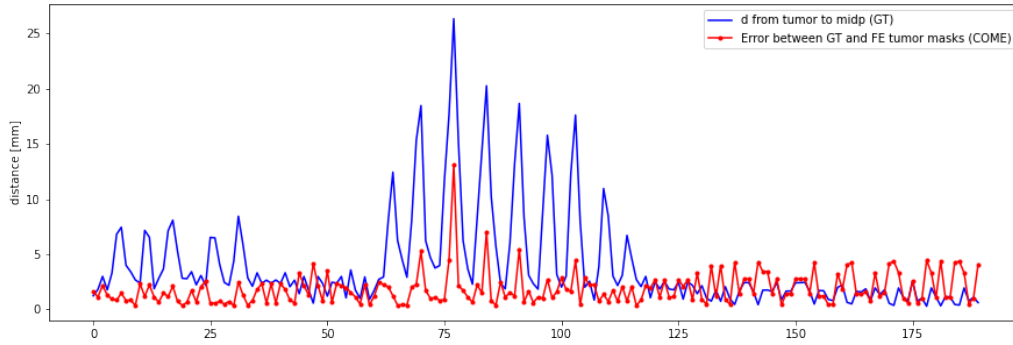
(c) Patient 3

Figure 6.22: DICE graphs. In yellow (liver) and purple (tumor): the DICE coefficients for the continuous sequence. In blue: the distance between the GT tumor COM and the midp.

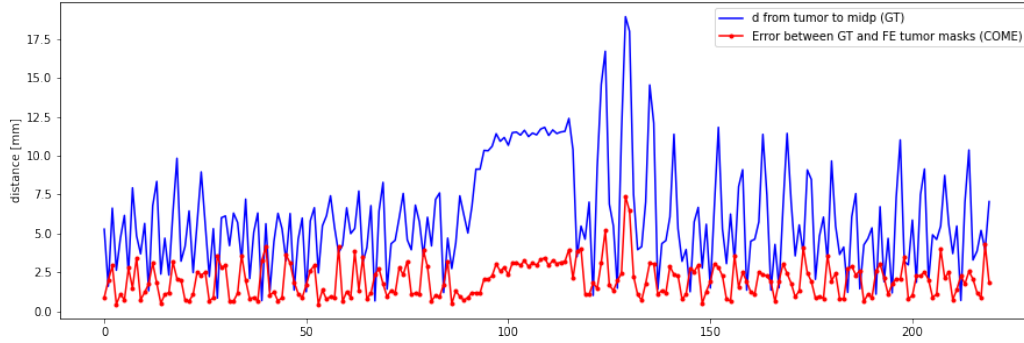
The third type of graph we computed are the graphs of the *absolute position error*. We display simultaneously the distance of the *GT* tumor from the midp, and the absolute position error between the centers of mass of the *GT* tumor and the *simulated* tumor, so-called the COME.



(a) Patient 1



(b) Patient 2



(c) Patient 3

Figure 6.23: Absolute position error graphs. In blue: the distance between the GT tumor COM and the midp. In red: the absolute position error between GT tumor COM and FE tumor COM.

Let's now try to analyze and interpret these results. Overall, we get good average DICE for the continuous sequence. The DICE comparing the liver masks for the CS are very close to the liver DICE obtained for planning. For the tumor DICE, they are better than the planning for patients 2 and 3 and worse for patient 1. It seems unlikely that we obtain better results for the continuous sequence than for the planning. For the

planning DICE, both masks that we compared were planning masks, they were just obtained in different ways (ground truth and simulation). Here we compared the 3D images of the CS with the images chosen by the metric among those of the planning (including interpolations and extrapolations) which are the closest to the CS iteration. The images are therefore less likely to match, especially since there are extrapolations, which have not been constructed from reference images but from the method (5.2.3). To find an explanation, we analyzed the volume of the tumor masks that are comparing for the planning and the continuous sequence. The volume was computed with the masks (as numpy arrays) by knowing the voxel size (1.074 x 1.074 x 2 mm). We obtained the results shown in the following table 16:

Patient	V planning [cm^3]	V CS [cm^3]
1	27.06	24.19
2	85.68	74.76
3	24.36	19.64

Table 16: Average volume of GT tumor masks for planning and for continuous sequence [cm^3]

The average volume of the GT tumor masks is larger for the planning than for the continuous sequence. A plausible explanation is the presence of artifacts on the planning images. These artifacts can be caused by two things. First, the 10 3DCT planning images, the phases, are averages of several 3DCT images corresponding to the same phase of the respiratory cycle. Second, the 3DCT images themselves used to reconstruct the planning images are also reconstructions since they are a stack of 2D slices, processed one after the other. This means that there is an offset in acquisition time between 2 slices of the same 3DCT image. Therefore there are parts of the same 3DCT image that correspond to different moments. These two phenomena lead to the presence of artifacts for the planning phases. Less artifacts are observed for the continuous sequence because it is reconstructed by distorting the midp phase. The midp phase is, as a reminder, a cleaner average phase of the planning, with less impurity and artifacts. The link with volume is as follows. The most obvious artifact is the one we are interested in, visible just above the liver in the planning 4DCT. We observe a duplicated liver above the real liver (Figure 6.24b). This duplicated liver is taken into account in the liver mask. So we have a liver slightly larger in height than it really is. However, the GT masks of the tumor are distorted from the midp with the method [82] based on trackers. The trackers will detect a stretched liver, when it is not. The deformation field will be recalculated on the basis of these trackers and the tumor will tend to be a little bit overstretched too for the planning phases. On the other hand, the continuous sequence images are reconstructed from the midp, where there is less artifact. This explains a less stretched tumor for the continuous sequence. In Figure 6.24 below, we see the artifact of patient 1 for phase 0 (Figure 6.24a) and in Figure 6.24b we see the artifact attenuated on an iteration of the continuous sequence where the metric detects phase 0. The tumor is schematically represented in blue.

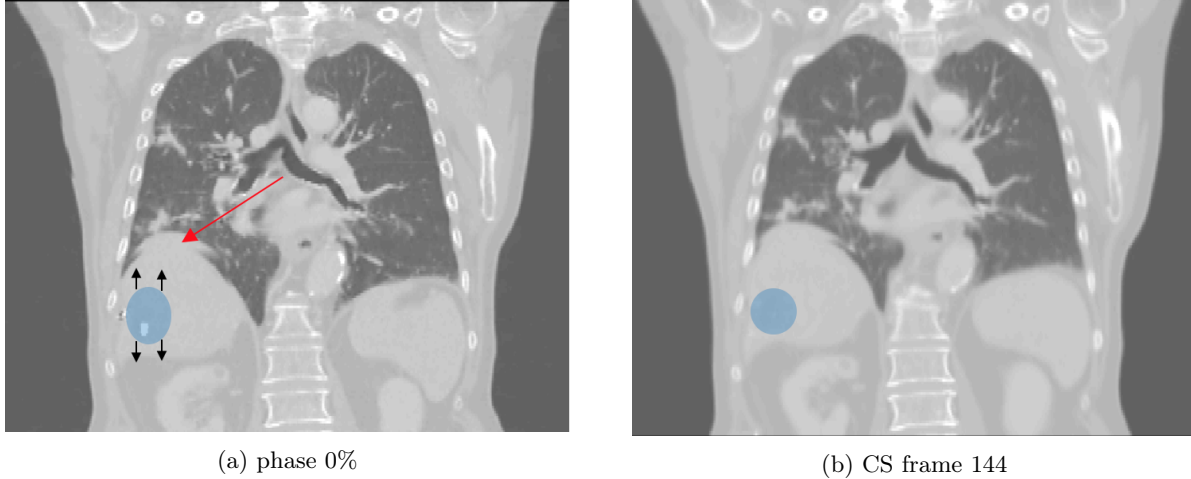


Figure 6.24: Comparison between coronal view of patient 1. Figure 6.24a shows the phase 0 with an artifact (red arrow). Tumor mask is represented in blue. Figure 6.24b shows frame of the CS close to p0 according to metric, with less important artifact.

This leads us to believe that the tumor DICE obtained for the planning are probably underestimated, given that the COME are excellent for the planning, which is a sign of a good alignment. The case of patient 1 (who has worse DICES for the continuous sequence) will be treated individually later.

When we look at the shape of the DICE plot, we see that the liver DICE vary very little with breathing (except for patient 2 in the largest extrapolations). This means that the movement was fairly well followed. The tumor itself is more sensitive to variations in breathing. A likely explanation is that the liver, being much larger than the tumor, will observe a smaller proportional volume difference for the same offset than the tumor. And so the difference in DICE will be less noticeable. We note that the tumor DICE are somewhat worse for the larger extrapolations (especially for patient 2 and 3).

If we look at the COME, we observe worse results for the continuous sequence, both for the liver and for the tumor. This seems more logical to us. If we look at the graphs of the absolute position error, two things can be noticed. First, the error between the COM of the GT and simulated tumor masks is on average smaller than the absolute distance of the GT tumor and the midp. Thus, we give an estimate of the tumor position that is close to the GT tumor position with respect to the distance to the midp. So we can follow the displacement of the GT tumor, to a certain extent, with always a margin of error. And second, more important, we see that the error is not coupled to breathing (for the vast majority of iterations in patients 2 and 3). It is most often below 5 mm. This independence between COME and the absolute position of the tumor shows once again that we have succeeded in following the displacement of the tumor. The case of patient 1 and the large extrapolations of patients 2 and 3 are analyzed below.

Let's now look at each patient individually to try to understand the different sources of error. Concerning patient 1, we see that he almost always breathes outside the planning cycle, according to the z-position of the tumor COM. He breathes higher, which means that his lungs are less full, on average. We observe bad DICE of the tumor for almost all the high breathing peaks, i.e. the points farthest from the midp, upwards. Let's try to display now the overlapping masks of the liver and the tumor for the best and worst DICE. The

best tumor DICE is obtained at iteration 151, for a phase 30 detected by the metric. The worst tumor DICE is obtained at iteration 202, for a phase 50 detected by the metric.

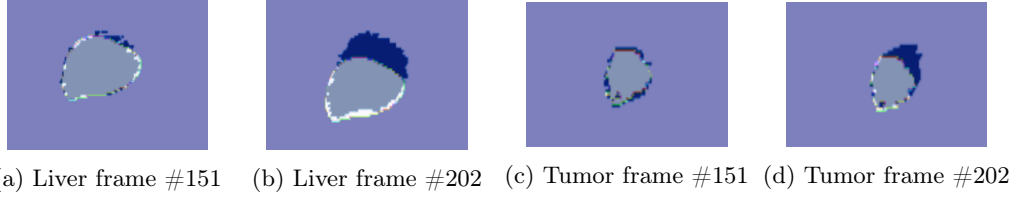


Figure 6.25: Patient 1 liver and tumor masks superposed for the best (6.25a and 6.25c) and worst (6.25b and 6.25d) DICE result of the continuous sequence. White masks are the simulated masks while blue masks are the GT masks.

From what we obtained, it can be seen that the poor results of frame 202 are due to a greater deformation of the liver, and thus of the tumor. When the lung deflates a lot, it rises strongly, stretching the liver in the z direction. This is not taken into account by the model extrapolation method. This method simply moves the liver, keeping the extreme phase deformation field (phase 50% here), along the average displacement direction. So an improvement of the extrapolation method would be to calculate a deformation field that takes into account a factor proportional to the amplitude of the extrapolation. Since patient 1 breathes a lot out of planning range, we have a lot of strong extrapolations, for which the tumor was misrepresented. This explains the poorer overall results.

If we now look at patient 2, we notice that he breathes mostly within the respiratory cycle, still according to the z -position of the tumor COM. The only exceptions are large breathing peaks in the middle of the sequence, where the patient inhales more deeply, bringing the liver lower. We applied the same procedure for patient 2, and we obtained the best tumor DICE at iteration 104, for an extrapolated phase 0, 125% detected by the metric. The worst tumor DICE is obtained at iteration 77, for a phase 0 and an extrapolation ratio of 200% detected by the metric.

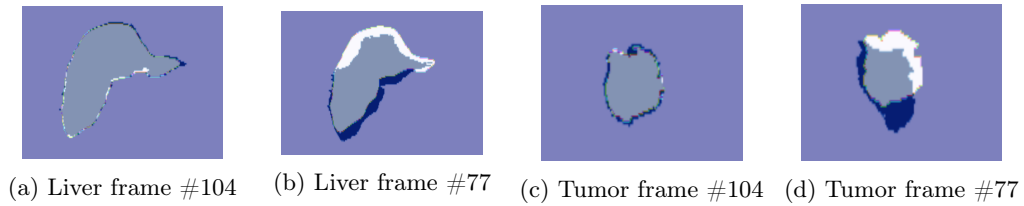


Figure 6.26: Patient 2 liver and tumor masks superposed for the best (6.26a and 6.26c) and worst (6.26b and 6.26d) DICE result of the continuous sequence. White masks are the simulated masks while blue masks are the GT masks.

We observe the same phenomenon as for patient 1 but for the other extreme phase, i.e. when the lung is strongly inflated. It then crushes the liver (as in the figure 6.26b) which leads to a deformation of the tumor (Figure 6.26d). The difference with patient 1 is that since patient 2 breathes more often inside the planning range, this phenomenon was observed only for the few large extrapolations.

Finally, patient 3 is the most regular with respect to his planning cycle, and is also the one for whom we observe the best results. His breathing during the continuous sequence is almost always within the range of

the planning cycle. However, we notice an apnea towards the middle of the sequence, i.e. the patient stops breathing in the upper lung position (end of expiration). And we observe a small peak of breathing just after. The patient 3 took a deep breath after his apnea. For patient 3, we have also displayed the masks of the best and worst DICE tumors obtained. We obtained the best tumor DICE at iteration 82, for an extrapolated phase 50, 175% detected by the metric. The worst tumor DICE is obtained at iteration 129, for a phase 0, 200% detected by the metric.

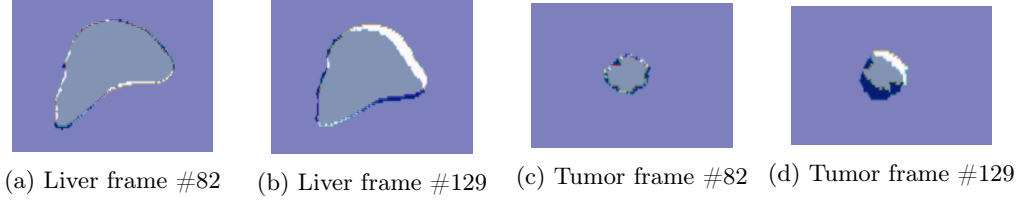


Figure 6.27: Patient 3 liver and tumor masks superposed for the best (6.26a and 6.26c) and worst (6.26b and 6.26d) DICE result of the continuous sequence. White masks are the simulated masks while blue masks are the GT masks.

The results show that the worst tumor DICE is obtained during the peak (strong inspiration). When we look at the tumor masks we observe a rather identical shape (Figure 6.27d). We are in a different situation than patients 1 and 2. Here, we simply did not extrapolate far enough. We can therefore create phases of greater extrapolation, for example an extrapolation of 250 %.

6.6 Real-Time Validation

We wanted to see if the method developed in this work was fast enough to be used in real-time. We therefore tried to calculate the total execution time of the chain (from the acquired fluoroscopic image to the restitution of the 3D image). We obtained an average execution time of 0.025 sec for an iteration of the patient 1 continuous sequence. According to [106], the image acquisition rate for fluoroscopy can reach 30 images/sec. We can therefore consider a maximum delay between two successive frames being equal to 0.033 sec. For the patient 1 we had to take a larger cropped section than for patients 2 and 3, so this represents our worst case. This method was performed on our personal laptops. Therefore it may have slower computation speed than professional ones used for medical technologies. As the execution time obtained is shorter than the acquisition time of a fluoroscopic image, it can be concluded that the method is suitable for real-time use.

7 Limitations and Improvements

Now that all the results have been presented, we can take a step back and try to identify the weaknesses of the method presented in this master thesis. To counteract these imperfections, suggestions for improvement will also be provided.

About the first part of our method, the metric computation, several improvements can be considered. First of all, the pre-processing of the images could be improved. Sometimes the framing we have chosen has been determined because of artifacts that prevent us from taking a potentially more interesting one.

In addition, lung segmentation could also be improved. Other criteria can be added to the region growing algorithm to avoid missing points in the middle of the region due to noise or artifacts. We could also make the choice of the first point more robust in case of noise. It could be that the first point chosen is an outlier and therefore distorts the result of the segmentation, even if the case has not been encountered here.

A third improvement concerns the offset that must be made on the metric obtained for the planning sequence (section 6.1). We observed that the curve described by our metric sometimes presents a significant difference of level with the continuous sequence compared to the curve obtained with the mass center of the liver. In this work, we used one of the mass centers of the continuous sequence to adapt our curve but this information will not be available in reality. One idea to remedy this would be to use the midp image which is available from the planning sequence, and which moreover served to establish the continuous sequence. Our metric can be calculated on the midp and we would also have the liver center of mass. With this point, we would be able to calculate the offset as we did with a point in the continuous sequence. We have to keep in mind the reason why this difference occurs. This should mainly come from the fact that the planning images are reconstructed and averaged images while those in the continuous sequence are distorted images based on the midp which exhibits better quality (thanks to various image processing techniques). This can lead to a potential quality difference in these two datasets. However, since we tried to reproduce realistic projections from the continuous sequence, we should face the same kind of difference in a real situation with fluoroscopic images.

Another thing to keep in mind is the noise. Even if we made a first test in order to verify that our method can be robust to its impact, a more complete validation is to take into consideration. As mentioned in section 3.1.4, some papers present some ideas to implement a noise modelization which could better suit the reality. Moreover, there are also researches to find a better way to overcome the impact of noise in fluoroscopy, those could be investigated and implemented if necessary.

The last improvement concerning this part of the method is about the extrapolation. At this stage, the latter is calculated between two successive phases: the extreme phase (p0, p50, or p90), and an artificial phase which is defined as the closest extreme phase with the addition or subtraction of 100 pixels (depending on the direction of the extrapolation). However, to facilitate the calculation of the 3D model offset, necessary for the extrapolation, and to avoid more steps and therefore potential sources of errors, it might be wise to calculate this percentage from the midp rather than between these two successive phases. This could be done simply by calculating the metric on the midp image and then returning the percentage distance between the metric calculated for a given respiratory state and the one calculated for that midp image. In addition, this change would also avoid creating the artificial phase (with 100 pixels difference) that does not really

correspond to anything in reality.

Concerning the finite element model, two problems can be raised. The first one concerns the choice of the biomechanical parameters. Here we have taken values mentioned in scientific articles, and we have considered these same values for the 3 patients. However, in practice, these parameters are very likely to vary from one patient to another, especially according to the state of fibrosis of the liver. It could be interesting to take the main parameter (Young's modulus, which characterizes elasticity) from each patient individually. For this, there is a non-invasive technique, the Fibroscan (section 5.2.3).

Then, as explained in the previous section, it was realized that the respiratory states identified outside the scope of the planning by the metric were not always very well estimated by the model. In other words, the model had difficulty representing extrapolations. The method developed here consisted in making a simple translation of the extreme phase of the schedule (in general phase p0 or p50) and then applying the deformation field corresponding to this phase. This technique did not take into account the different anatomical barriers to displacement. Thus the liver (and the tumor) were sometimes modeled too far away, where it could not physically be. A solution would be to model in finite elements, in addition to the liver, the interfaces of its adjacent organs (diaphragm and therefore lung, rib cage, stomach, etc). Thus, the liver will stop at these interfaces and be correctly deformed instead. Another solution would be to find a way to predict these extrapolation phases in the planning cycle. One could plan a 4DCT of planning for a normal respiratory cycle, and another for a more intense respiratory cycle where the patient is instructed to inhale and exhale more deeply. This would allow us to know the deformation fields of the extrapolated phases with respect to the planning.

8 Conclusion

The monitoring of tumors located in organs that move according to the respiratory cycle is still a challenge in proton therapy. Unlike conventional radiotherapy, proton therapy allows a much more precise irradiation, but therefore also a much more concentrated energy. It is thus all the more important to know the precise location of the tumor during treatment, in order to irradiate the cancerous tissue with precision, and thus to preserve the surrounding healthy tissue from destructive radiation.

The objective of this work was to establish a general method that would allow on the one hand to follow in real-time the respiratory state of the patient, and on the other hand to be able to estimate a 3D position of the tumor corresponding to this respiratory state. To build our method, we had available, for each three patients, a planning 4DCT, modeling an average respiratory cycle.

The two main axes of this thesis, namely the metric and the finite element liver model, have been developed independently and then combined in a general method. The metric was used to determine the patient's respiratory state based on a 2D projection, using a region growing of the lung. The model allowed simulating upstream the liver displacements for different respiratory states (planning phase and inter/extrapolation), using the intrinsic biomechanical characteristics of the tissues. We combined these methods by working on the continuous sequence, a 4DCT sequence that mimics a more random breathing of the patient, with successive breathing cycles of varying intensities. This continuous sequence was our "real-time" sequence. For each iteration of this continuous sequence, we were able to determine the respiratory state and choose the results corresponding to the closest phase precomputed by the model. The results of the DICE and COME coefficients showed that we succeeded in tracking the motion of the tumor. For the mean tumor DICE, we obtained for patient 1, 2 and 3 respectively 0.8176 ± 0.0547 , 0.8755 ± 0.0333 and 0.7906 ± 0.0446 . For the mean COME, we obtained patient 1, 2 and 3 respectively 3.48 ± 2.11 mm, 1.86 ± 1.48 mm and 2.01 ± 1.12 mm. Moreover, we were able to validate the possibility to use the method in real-time. The total computation time, from a fluoroscopic frame to the restitution of a 3D image, that we obtained is 0.025 sec compared to a fluoroscopy acquisition time of 0.033 sec [106].

After the realization of this work, we highlighted the interest to improve some parts in order to optimize our method. In this context, some tracks to explore were also suggested. A good extrapolation technique, as well as a good calibration, could have a significant positive impact on the results and are therefore the most important aspects to be investigated. Further research on the noise involved would also be needed to get closer to real fluoroscopic quality.

We were able to demonstrate that our method deserved to be investigated more in-depth. This approach combining both image processing and finite element reconstruction brings an innovative side to the tumor motion tracking field of research. Solutions responding to our areas for improvement can already be found in the literature (noise treatment, segmentation, consideration of the anatomy for finite elements, etc). This would make it possible to improve the method described in this document fairly quickly.

All this, we hope, will contribute to the advancement of research in the field of proton therapy. By reducing the radiation margins, we could decrease the amount of healthy tissue affected and thus prevent the patient from experiencing unwanted side effects. The improvement of the method must still be done, as well as clinical tests thereafter, nevertheless this direction can be investigated in order to have the possibility of

improving the life quality of patients who will undergone this type of treatment.

References

- [1] World Health Organization. *Cancer*. <https://www.who.int/en/news-room/fact-sheets/detail/cancer>. Seen on April 15, 2021.
- [2] International Agency for Research on Cancer. *World Cancer Report 2020*. Christopher P. Wild, Elisabete Weiderpass, Bernard W. Stewart, 1981. ISBN: 978-92-832-0448-0. URL: <https://publications.iarc.fr/586>.
- [3] NHS. *Radiotherapy, overview*. URL: <https://www.nhs.uk/conditions/radiotherapy/> (visited on 04/16/2021).
- [4] NIH. *Radiation Therapy to Treat Cancer*. URL: <https://www.cancer.gov/about-cancer/treatment/types/radiation-therapy> (visited on 04/16/2021).
- [5] Cancer.net Editorial Board. *Proton therapy*. 2018. URL: <https://www.cancer.net/navigating-cancer-care/how-cancer-treated/radiation-therapy/proton-therapy> (visited on 05/20/2021).
- [6] IASLC editor(s). *Proton-Beam Therapy Versus Photon-Beam Therapy: The Debate Continues*. 2017. URL: <https://www.iaslc.org/iaslc-news/ilcn/proton-beam-therapy-versus-photon-beam-therapy-debate-continues> (visited on 05/20/2021).
- [7] Protom. *The Bragg peak*. 2018. URL: <https://www.protominternational.com/2018/06/bragg-peak/> (visited on 03/22/2021).
- [8] NCI staff. *Radiation Therapy Side Effects*. 2018. URL: <https://www.cancer.gov/about-cancer/treatment/types/radiation-therapy/side-effects> (visited on 04/05/2021).
- [9] NCI staff. *Is Proton Therapy Safer than Traditional Radiation?* 2020. URL: <https://www.cancer.gov/news-events/cancer-currents-blog/2020/proton-therapy-safety-versus-traditional-radiation> (visited on 04/05/2021).
- [10] Lu HM St James S Grassberger C. “Considerations when treating lung cancer with passive scatter or active scanning proton therapy”. In: *Transl Lung Cancer Res* 7 (Apr. 2018), pp. 210–215.
- [11] Demcom Nymus3D. *Proton Therapy Concepts*. YouTube. July 2017. URL: https://www.youtube.com/watch?v=V_6KEDw7xD8&ab_channel=DemconNymus3D.
- [12] Lu HM et al Kooy HM Clasio BM. “Case study in proton pencil-beam scanning delivery”. In: *Int J Radiat Oncol Biol Phys* 76 (June 2010), pp. 624–630.
- [13] Paul J Keall Yelin Suh Sonja Dieterich. “Geometric uncertainty of 2D projection imaging in monitoring 3D tumor motion”. In: *National Library of Medicine NIH* 52 (June 2007), pp. 3439–54.
- [14] Sonja Dieterich et al. “Locating and targeting moving tumors with radiation beams”. In: *Medical Physics* 35.12 (2008), pp. 5684–5694. URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3020593>.
- [15] Philippe Giraud and Annie Houle. “Respiratory Gating for Radiotherapy: Main Technical Aspects and Clinical Benefits”. In: *ISRN Pulmonology* 2013 (Mar. 2013). DOI: 10.1155/2013/519602.
- [16] Sai Vemprala et al. “Real-Time Tumor Tracking for Pencil Beam Scanning Proton Therapy”. In: (2018), pp. 4434–4440.

- [17] Twyla R. Willoughby et al. "Target localization and real-time tracking using the Calypso 4D localization system in patients with localized prostate cancer". In: *International Journal of Radiation Oncology*Biological*Physics* 65.2 (2006), pp. 528–534. ISSN: 0360-3016. DOI: <https://doi.org/10.1016/j.ijrobp.2006.01.050>. URL: <https://www.sciencedirect.com/science/article/pii/S0360301606002537>.
- [18] James M. Balter et al. "Accuracy of a wireless localization system for radiotherapy". English (US). In: *International Journal of Radiation Oncology Biology Physics* 61.3 (Mar. 2005). Copyright: Copyright 2008 Elsevier B.V., All rights reserved., pp. 933–937. ISSN: 0360-3016. DOI: 10.1016/j.ijrobp.2004.11.009.
- [19] Patrick Kupelian et al. "Multi-institutional clinical experience with the Calypso System in localization and continuous, real-time monitoring of the prostate gland during external radiotherapy". In: *International Journal of Radiation Oncology*Biological*Physics* 67.4 (2007), pp. 1088–1098. ISSN: 0360-3016. DOI: <https://doi.org/10.1016/j.ijrobp.2006.10.026>. URL: <https://www.sciencedirect.com/science/article/pii/S0360301606032767>.
- [20] Shinichi Shimizu et al. "Fluoroscopic real-time tumor-tracking radiation treatment (RTRT) can reduce internal margin (IM) and set-up margin (SM) of planning target volume (PTV) for lung tumors". In: *International Journal of Radiation Oncology Biology Physics - INT J RADIAT ONCOL BIOL PHYS* 48 (Dec. 2000), pp. 166–167. DOI: 10.1016/S0360-3016(00)80127-3.
- [21] Shinichi Shimizu et al. "Use of an implanted marker and real-time tracking of the marker for the positioning of prostate and bladder cancers". In: *International Journal of Radiation Oncology*Biological*Physics* 48.5 (2000), pp. 1591–1597. ISSN: 0360-3016. DOI: [https://doi.org/10.1016/S0360-3016\(00\)00809-9](https://doi.org/10.1016/S0360-3016(00)00809-9). URL: <https://www.sciencedirect.com/science/article/pii/S0360301600008099>.
- [22] Shinichi Shimizu et al. "Detection of lung tumor movement in real-time tumor-tracking radiotherapy". In: *International Journal of Radiation Oncology*Biological*Physics* 51.2 (2001), pp. 304–310. ISSN: 0360-3016. DOI: [https://doi.org/10.1016/S0360-3016\(01\)01641-8](https://doi.org/10.1016/S0360-3016(01)01641-8). URL: <https://www.sciencedirect.com/science/article/pii/S0360301601016418>.
- [23] Hiroki Shirato et al. "Physical aspects of a real-time tumor-tracking system for gated radiotherapy". In: *International Journal of Radiation Oncology*Biological*Physics* 48.4 (2000), pp. 1187–1195. ISSN: 0360-3016. DOI: [https://doi.org/10.1016/S0360-3016\(00\)00748-3](https://doi.org/10.1016/S0360-3016(00)00748-3). URL: <https://www.sciencedirect.com/science/article/pii/S0360301600007483>.
- [24] Ross Berbeco et al. "Integrated radiotherapy imaging system (IRIS): Design considerations of tumour tracking with linac gantry-mounted diagnostic x-ray systems with flat-panel detectors". In: *Physics in medicine and biology* 49 (Feb. 2004), pp. 243–55. DOI: 10.1088/0031-9155/49/2/005.
- [25] Martin J Murphy. "Tracking moving organs in real time". In: *Seminars in Radiation Oncology* 14.1 (2004). High-Precision Radiation Therapy of Moving Targets, pp. 91–100. ISSN: 1053-4296. DOI: <https://doi.org/10.1053/j.semradonc.2003.10.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1053429603000869>.
- [26] Achim Schweikard, Hiroya Shiomi, and John Adler. "Respiration tracking in radiosurgery". In: *Medical physics* 31.10 (Oct. 2004), pp. 2738–2741. ISSN: 0094-2405. DOI: 10.1118/1.1774132.
- [27] Ivan Buzurovic et al. "A robotic approach to 4D real-time tumor tracking for radiotherapy". In: *Physics in medicine and biology* 56 (Mar. 2011), pp. 1299–318. DOI: 10.1088/0031-9155/56/5/005.

- [28] Yvette Seppenwoolde et al. “Accuracy of tumor motion compensation algorithm from a robotic respiratory tracking system: a simulation study”. In: *Medical physics* 34.7 (July 2007), pp. 2774–2784. ISSN: 0094-2405. DOI: 10.1118/1.2739811. URL: <https://doi.org/10.1118/1.2739811>.
- [29] Ross Berbeco et al. “Integrated radiotherapy imaging system (IRIS): Design considerations of tumour tracking with linac gantry-mounted diagnostic x-ray systems with flat-panel detectors”. In: *Physics in medicine and biology* 49 (Feb. 2004), pp. 243–55. DOI: 10.1088/0031-9155/49/2/005.
- [30] Byungchul Cho et al. “A monoscopic method for real-time tumour tracking using combined occasional X-ray imaging and continuous respiratory monitoring”. In: *Physics in medicine and biology* 53 (July 2008), pp. 2837–55. DOI: 10.1088/0031-9155/53/11/006.
- [31] B Raaymakers et al. “Integrating a MRI scanner with a 6 MV radiotherapy accelerator: Dose deposition in a transverse magnetic field”. In: *Physics in medicine and biology* 49 (Oct. 2004), pp. 4109–18. DOI: 10.1088/0031-9155/49/17/019.
- [32] A. Hsu et al. “Feasibility of using ultrasound for real-time tracking during radiotherapy”. In: *Medical Physics* 32.6Part1 (2005), pp. 1500–1512. DOI: <https://doi.org/10.1118/1.1915934>. URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.1915934>.
- [33] Nikhil Bhagat et al. “Complications Associated with the Percutaneous Insertion of Fiducial Markers in the Thorax”. In: *Cardiovascular and interventional radiology* 33 (Dec. 2010), pp. 1186–91. DOI: 10.1007/s00270-010-9949-0.
- [34] RadiologyInfo.org authors. *Fiducial Marker Placement*. 2019. URL: <https://www.radiologyinfo.org/en/info/fiducial-marker> (visited on 06/12/2021).
- [35] Zyp; Noelle et al. “Stability of Markers Used for Real-Time Tumor Tracking After Percutaneous Intrapulmonary Placement”. In: *International journal of radiation oncology, biology, physics* 81 (Feb. 2011), e75–81. DOI: 10.1016/j.ijrobp.2010.12.026.
- [36] Elisabeth Weiss. “Long-Term Safety and Stability of Gold Coil Fiducial Markers in Non-Small-Cell Lung Cancer Image-Guided Radiotherapy”. In: *International Journal of Radiology Radiation Therapy* 3 (May 2017). DOI: 10.15406/ijrrt.2017.03.00049.
- [37] Tanaka; Osamu et al. “The advantage of iron-containing fiducial markers placed with a thin needle for radiotherapy of liver cancer in terms of visualization on MRI: an initial experience of Gold Anchor”. In: *Radiology Case Reports* 12 (Apr. 2017). DOI: 10.1016/j.radcr.2017.03.014.
- [38] Zhu; Xiaoshi et al. “Tradeoffs of integrating real-time tracking into IGRT for prostate cancer treatment”. In: *Physics in medicine and biology* 54 (Aug. 2009), N393–401. DOI: 10.1088/0031-9155/54/17/N03.
- [39] J. Rottmann; M.Aristophanous; A. Chen; L. Court; R. Berbeco. “A multi-region algorithm for markerless beam’s-eye view lung tumor tracking.” In: *Physics in medicine and biology* 55 (2010), pp. 5585–5598. DOI: 10.1088/0031-9155/55/18/021.
- [40] J. Rottmann; P. Keall; and R. Berbeco. “Markerless EPID image guided dynamic multi-leaf collimator tracking for lung tumors”. In: *Physics in medicine and biology* 58 (2013), pp. 4195–4204. DOI: doi: 10.1088/0031-9155/58/12/4195.
- [41] Xie; Yaoqin et al. “Tissue feature-based intra-fractional motion tracking for stereoscopic X-ray image guided radiotherapy”. In: *Physics in medicine and biology* 58 (May 2013), pp. 3615–3630. DOI: 10.1088/0031-9155/58/11/3615.

- [42] Ichiji; Kei et al. “A key-point based real-time tracking of lung tumor in X-ray image sequence by using difference of Gaussians filtering and optical flow”. In: *Physics in Medicine and Biology* 63 (Aug. 2018). DOI: 10.1088/1361-6560/aada71.
- [43] Brock et al. “Accuracy of finite element model-based multi-organ deformable image registration”. In: *Medical Physics : The International Journal of Medical Physics Research and Practice* (May 2005). URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.1915012>.
- [44] Eom et al. “Predictive modeling of lung motion over the entire respiratory cycle using measured pressure-volume data, 4DCT images, and finite-element analysis”. In: *Medical Physics : The International Journal of Medical Physics Research and Practice* (July 2010). URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3455276>.
- [45] Werner et al. “Patient-specific finite element modeling of respiratory lung motion using 4D CT image data”. In: *Medical Physics : The International Journal of Medical Physics Research and Practice* (Apr. 2009). URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3101820>.
- [46] Tehrani et al. “Sensitivity of tumor motion simulation accuracy to lung biomechanical modeling approaches and parameters”. In: *Physics in Medicine Biology* (Nov. 2015). URL: <https://iopscience.iop.org/article/10.1088/0031-9155/60/22/8833>.
- [47] Marchesseau et al. “Non linear Biomechanical Model of the Liver: Hyperelastic Constitutive Laws for Finite Element Modeling”. In: (Dec. 2017). URL: <https://hal.inria.fr/hal-01536406>.
- [48] Ghahfarokhi et al. “Developing a deformable model of liver tumor during breathing to improve targeting accuracy in image-guided therapy using finite element simulation”. In: *Scientia Iranica* (Apr. 2019). URL: https://www.researchgate.net/publication/332632948_Developing_a_deformable_model_of_liver_tumor_during_breathing_to_improve_targeting_accuracy_in_image_guided_therapy_using_finite_element_simulation.
- [49] COCIR. *Medical Imaging*. URL: <https://www.cocir.org/our-industry/medical-imaging.html> (visited on 05/27/2021).
- [50] Innovatemedtec. *Medical Imaging*. URL: <https://innovatemedtec.com/digital-health/medical-imaging> (visited on 05/27/2021).
- [51] Tim Newman. *Are X-rays really safe*. Medically reviewed by William Morrison, M.D. 2018. URL: medicalnewstoday.com/articles/219970 (visited on 05/27/2021).
- [52] Kerckhofs G.; Lee J.; Macq B.; Peeters F. *Lectures Notes of LGBIO2050 : Medical Imaging*. Louvain School of Engineering, Belgium. 2021.
- [53] Jeffrey S. Klein. *Conventional radiography*. 2019. URL: <https://www.pulmonologyadvisor.com/home/decision-support-in-medicine/pulmonary-medicine/conventional-radiography/> (visited on 05/27/2021).
- [54] Biology Online. *Radiodensity definition*. 2021. URL: <https://www.biologyonline.com/dictionary/radiodensity> (visited on 05/27/2021).
- [55] Wikipedia contributors. *Hounsfield scale* — *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Hounsfield_scale&oldid=1013565327. [Online; accessed 27-May-2021]. 2021.

- [56] Paul Kinahan, B. Hasegawa, and T. Beyer. “X-ray-based attenuation correction for positron emission tomography/computed tomography scanners.” In: *Seminars in nuclear medicine* 33 3 (2003), pp. 166–79.
- [57] Rodrigo Viana et al. “Heterogeneity correction in the construction of optimized planning in radiotherapy using Linear Programming”. In: *Pesquisa Operacional* 31 (Dec. 2011), pp. 565–578. DOI: 10.1590/S0101-74382011000300009.
- [58] Valentin E. Sinitsyn and Stephan Achenbach. “Electron Beam Computed Tomography (EBCT)”. In: *Coronary Radiology*. Ed. by Matthijs Oudkerk. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 137–165. ISBN: 978-3-662-06419-1. DOI: 10.1007/978-3-662-06419-1_8. URL: https://doi.org/10.1007/978-3-662-06419-1_8.
- [59] et al. Dr Zemar Vajuhudeen Dr Matt A. Morgan. *Helical CT image acquisition*. 2021. URL: <https://radiopaedia.org/articles/helical-ct-image-acquisition-1> (visited on 05/27/2021).
- [60] Jeffrey S. Klein. *Multidetector computed tomography*. 2019. URL: <https://www.pulmonologyadvisor.com/home/decision-support-in-medicine/pulmonary-medicine/multidetector-computed-tomography/> (visited on 05/27/2021).
- [61] William C. Scarfe and Allan G. Farman. “What is Cone-Beam CT and How Does it Work?” In: *Dental Clinics of North America* 52.4 (2008). Contemporary Dental and Maxillofacial Imaging, pp. 707–730. ISSN: 0011-8532. DOI: <https://doi.org/10.1016/j.cden.2008.05.005>. URL: <https://www.sciencedirect.com/science/article/pii/S001185320800044X>.
- [62] AICA Orthopedics. *What is the Difference Between an X-Ray and Fluoroscopy*. 2019. URL: <https://aica.com/whats-the-difference-between-an-x-ray-and-fluoroscopy/> (visited on 05/28/2021).
- [63] IAEA Human Health Campus. *Conventional and digital fluoroscopy*. URL: <https://humanhealth.iaea.org/HHW/MedicalPhysics/DiagnosticRadiology/PerformanceTesting/Conventionalanddigitalfluoroscopy/index.html> (visited on 05/28/2021).
- [64] Isabel Rodrigues, Joao Sanches, and Jose Bioucas-Dias. “Denoising of medical images corrupted by Poisson noise”. In: *2008 15th IEEE International Conference on Image Processing*. 2008, pp. 1756–1759. DOI: 10.1109/ICIP.2008.4712115.
- [65] Kirti Thakur et al. “Poisson noise reduction from X-ray medical images using modified Harris operator and wavelet domain thresholding”. In: *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. 2015, pp. 568–572. DOI: 10.1109/IIC.2015.7150806.
- [66] Dang Thanh, Surya Prasath, and Hieu Le Minh. “A Review on CT and X-Ray Images Denoising Methods”. In: *Informatica* 43 (June 2019), pp. 151–159. DOI: 10.31449/inf.v43i2.2179.
- [67] E. Manson et al. “Image Noise in Radiography and Tomography: Causes, Effects and Reduction Techniques”. In: 2019.
- [68] James Seibert and Richard Morin. “The standardized exposure index for digital radiography: An opportunity for optimization of radiation dose to the pediatric population”. In: *Pediatric radiology* 41 (May 2011), pp. 573–81. DOI: 10.1007/s00247-010-1954-6.
- [69] Sangyoon Lee, Min Seok Lee, and Moon Gi Kang. “Poisson–Gaussian Noise Analysis and Estimation for Low-Dose X-ray Images in the NSCT Domain”. In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18041019. URL: <https://www.mdpi.com/1424-8220/18/4/1019>.

- [70] Walid Sharabati and Bowei Xi. “A neighborhood regression approach for removing multiple types of noises”. In: *EURASIP Journal on Image and Video Processing* 2018 (Mar. 2018). DOI: 10.1186/s13640-018-0259-9.
- [71] Ajay Boyat and Brijendra Joshi. “A Review Paper: Noise Models in Digital Image Processing”. In: *Signal Image Processing : An International Journal* 6 (May 2015). DOI: 10.5121/sipij.2015.6206.
- [72] Sangyoon Lee, Min Lee, and Moon Kang. “Poisson-Gaussian Noise Analysis and Estimation for Low-Dose X-ray Images in the NSCT Domain”. In: *Sensors (Basel, Switzerland)* 18 (Mar. 2018). DOI: 10.3390/s18041019.
- [73] V. Legat. *Lectures Notes of LEPL1110 : Introduction to Finite Elements*. Louvain School of Engineering, Belgium. 2021.
- [74] G. P. Nikishkov. *Introduction to the Finite Element Method - Lecture notes*. University of Aizu, Japan. 2004.
- [75] X. Desroches J.P. Lefebvre. *Shape Functions of Isoparametric Elements*. EDF-R&D/AMA, Code Aster. 2003.
- [76] H. Delingette S. Marchesseau S. Chatelin. “Nonlinear Biomechanical Model of the Liver”. In: *Biomechanics of Living Organs*, 2017. Chap. 11, pp. 243–265.
- [77] Y. Zhang & al. *Enhancing liver tumor localization accuracy by prior-knowledge-guided motion modeling and a biomechanical model*. 2019. DOI: 10.21037/qims.2019.07.04.
- [78] M. Amestoy S. Forest. *Mécanique des Milieux Continus*. Ecole des mines de Paris. 2015-2016.
- [79] Unknown. *Finite Strain Theory*. Wikipedia. URL: https://en.wikipedia.org/wiki/Finite_strain_theory#Deformation_tensors (visited on 04/08/2021).
- [80] L. Zhang J. Cheng. “A General Approach to Derive Stress and Elasticity Tensors for Hyperelastic Isotropic and Anisotropic Biomaterials”. In: (June 2018).
- [81] R. Cimrman & al. *SfePy Documentation*. Version 2020.4. 2020.
- [82] Damien Dasnoy-Sumell; Kevin Souris; G. Van Ooteghem; Benoit Macq. “Continuous real time 3D motion reproduction using dynamic MRI and precomputed 4DCT deformation fields”. In: *Journal of Applied Clinical Medical Physics* (July 2020). URL: <https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/acm2.12953>.
- [83] Gürsoy D; De Carlo F; Xiao X; and Jacobsen C. “Tomopy: a framework for the analysis of synchrotron tomographic data.” In: *Journal of Synchrotron Radiation* 21 (Aug. 2014), pp. 1188–1193.
- [84] Thomas Kluyver et al. “Jupyter Notebooks – a publishing format for reproducible computational workflows”. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Ed. by F. Loizides and B. Schmidt. IOS Press. 2016, pp. 87–90.
- [85] *Anaconda Software Distribution*. Version Vers. 2-2.4.0. 2020. URL: <https://docs.anaconda.com/>.
- [86] Shreenidhi Sudhakar. *Histogram Equalization*. 2017. URL: <https://towardsdatascience.com/histogram-equalization-5d1013626e64> (visited on 05/24/2021).
- [87] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).
- [88] Collaborative OpenSource Python Module. *scikit-image*. URL: https://scikit-image.org/docs/dev/api/skimage.measure.html?highlight=marching_cubes#skimage.measure.marching_cubes (visited on 04/10/2021).

- [89] Collaborative OpenSource Python Module. *numpy-stl*. URL: <https://pypi.org/project/numpy-stl/> (visited on 04/10/2021).
- [90] Unknown. *Marching cube algorithm*, *Wikipedia*. URL: https://fr.wikipedia.org/wiki/Marching_cubes (visited on 04/12/2021).
- [91] A. Muntoni; P. Cignoni; Opensource module. *pymeshlab*. URL: <https://pypi.org/project/pymeshlab/> (visited on 04/15/2021).
- [92] Opensource software ISTI & CNR. *MeshLab*. URL: <https://www.meshlab.net/> (visited on 04/15/2021).
- [93] P. Cignoni; F. Ganovelli. *Visualization and Computer Graphics Library*. URL: <http://www.vcglib.net/> (visited on 04/15/2021).
- [94] B. Curless; M. Levoy. *A Volumetric Method for Building Complex Models from Range Images*. 1996. DOI: <https://doi.org/10.1145/237170.237269>.
- [95] Alec Jacobson, Daniele Panozzo, et al. *libigl; A simple C++ geometry processing library*. <https://libigl.github.io/>. 2018.
- [96] Unknown. *wildmeshing: python interface to triangulate and tetrahedralize*. <https://wildmeshing.github.io/python/>.
- [97] Nico Schlomer. *meshio: module to read and write meshes*. <https://pypi.org/project/meshio/1.8.9/description>.
- [98] LMU UCLouvain UPENN and IBA. *OpenReggui : Registration Graphical User Interface*. Matlab-based medical image processing software. 2016. URL: <https://openreggui.org/>.
- [99] T. Hartkens; PhD W. R. Crum; DPhil and D. L. G. Hill; PhD. “Non-rigid image registration: theory and practice”. In: *The British Journal of Radiology* 77 (2004), pp. 140–153. DOI: 10.1259/bjr/25329214.
- [100] F. Oberti N. Boucher J. Boursier and V. de Ledinghen. *Evaluate fibrose, why ? how ? for whom ?* 2010. URL: <https://www.fmcgastro.org/postu-main/archives/postu-2010-paris/evaluer-la-fibrose-pourquoi-comment-pour-qui-2/> (visited on 04/19/2021).
- [101] J-F. Blanc and al. *Hepatocellular Carcinoma*. 2019. URL: <https://www.snfge.org/content/7-carcinome-hepatocellulaire-cancer-primitif-du-foie> (visited on 04/19/2021).
- [102] S.; Liverani; C. et al. Islam M.T.; Tang. “Non-invasive imaging of Young’s modulus and Poisson’s ratio in cancers in vivo”. In: *Sci Rep* 10, 7266 (2020). DOI: <https://doi.org/10.1038/s41598-020-64162-6>.
- [103] P. Ramachandran and G. Varoquaux. “Mayavi: 3D Visualization of Scientific Data”. In: *IEEE Computing in Science Engineering* (2011), pp. 40–51. URL: <https://docs.enthought.com/mayavi/mayavi/index.html>.
- [104] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [105] Sumin Lee et al. “Evaluation of Hepatic Toxicity after Repeated Stereotactic Body Radiation Therapy for Recurrent Hepatocellular Carcinoma using Deformable Image Registration”. In: *Scientific Reports* 8 (Nov. 2018). DOI: 10.1038/s41598-018-34676-1.
- [106] Eric Gingold. *Modern Fluoroscopy Imaging Systems*. URL: <https://www.imagewisely.org/Imaging-Modalities/Fluoroscopy/Modern-Imaging-Systems> (visited on 06/13/2021).

Appendices

A Shape Functions

For **dimension one**, we have define a number of nodes per element, then the Lagrangian polynomials are used to generate shape functions. N nodes gives N shape functions, where each shape function is associated to its node so as to be 1 at the node and 0 at the other nodes and cannot be a linear combination of other shape functions.

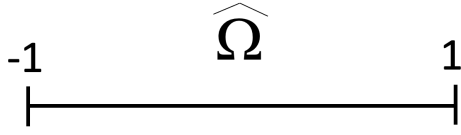


Figure A.1: Parent Element Ω $[-1;+1]$

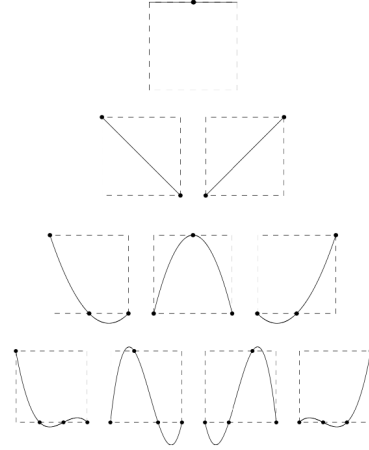


Figure A.2: Shape function for 1D element : constant, linear, quadratic and cubic forms [73]

Therefore the expression of the shape functions of the different degree is the following, for a point ξ inside the parent element $\hat{\Omega}$:

Degree zero, $N = 1$:

$$\phi(\xi) = 1$$

Degree one, $N = 2$:

$$\phi_1(\xi) = (1 - \xi)/2$$

$$\phi_2(\xi) = (1 + \xi)/2$$

Degree two, $N = 3$:

$$\phi_1(\xi) = -\xi(1 - \xi)/2$$

$$\phi_2(\xi) = \xi(1 + \xi)/2$$

$$\phi_3(\xi) = (1 - \xi)(1 + \xi)$$

Here the nodes are chosen at the ends, and for $N = 3$ at the center of the element, but they could very well have been taken at different places.

Then, for **dimension three**, we take our parent element, which is a tetrahedron and we define two *degree* for our subject ¹². The python module SFEPY that is used in these thesis does not give information about the shape functions they use. What is shown here was found in [75] as an example of what the shape functions can be for tetrahedrons. The parent element tetrahedron here is defined with his four vertices at $[0,0,0]$, $[1,0,0]$, $[0,1,0]$ and $[0,0,1]$. For the "second" *degree*, nodes in the middle of each edge are added.

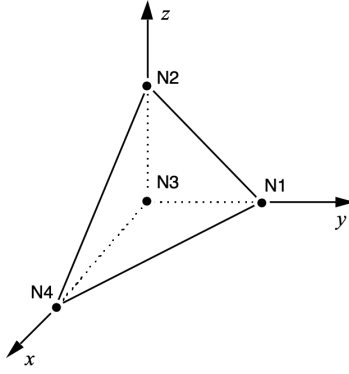


Figure A.3: Tetrahedron and its nodes for degree 1 [75]

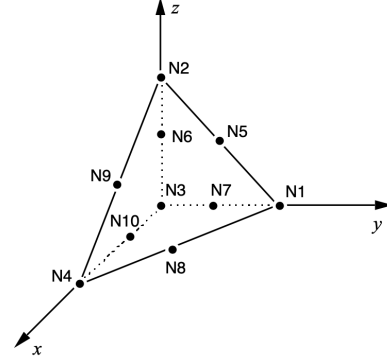


Figure A.4: Tetrahedron and its nodes for degree 2 [75]

Then shape functions expression is :

Degree zero, $N = 4$:

$$\phi_1(\xi, \eta, \zeta) = \eta$$

$$\phi_2(\xi, \eta, \zeta) = \zeta$$

$$\phi_3(\xi, \eta, \zeta) = 1 - \xi - \eta - \zeta$$

$$\phi_4(\xi, \eta, \zeta) = \xi$$

Degree one, $N = 10$:

$$\phi_1(\xi, \eta, \zeta) = \eta(2\eta - 1)$$

$$\phi_2(\xi, \eta, \zeta) = \zeta(2\zeta - 1)$$

$$\phi_3(\xi, \eta, \zeta) = (1 - \xi - \eta - \zeta)(1 - 2\xi - 2\eta - 2\zeta)$$

$$\phi_4(\xi, \eta, \zeta) = \xi(2\xi - 1)$$

$$\phi_5(\xi, \eta, \zeta) = 4\eta\zeta$$

$$\phi_6(\xi, \eta, \zeta) = 4\zeta(1 - \xi - \eta - \zeta)$$

¹²The degrees defines here does not really correspond to the degree of shape functions, but are rather representative of the increase in the number of nodes. There are indeed several ways to define shape functions with different degrees. What is important is that the functions cannot be linear combinations of others

$$\phi_7(\xi, \eta, \zeta) = 4\eta(1 - \xi - \eta - \zeta)$$

$$\phi_8(\xi, \eta, \zeta) = 4\xi\eta$$

$$\phi_9(\xi, \eta, \zeta) = 4\xi\zeta$$

$$\phi_{10}(\xi, \eta, \zeta) = 4\xi(1 - \xi - \eta - \zeta)$$

B Trilinear Interpolation

Trilinear Interpolation is a technique for interpolate a function on a regular 3D grid. It is an extension of the linear and bilinear interpolation in three dimensions.

Method

For any point located at a place of the grid, we must locate the rectangular parallelepiped in which it is located. We can perform the interpolation knowing the values at the 8 vertices of this parallelepiped. The idea is to compute the value of function at point P by assigning weights to the nodal values of the vertices according to their distance from the point P. Let x, y and z be the absolute coordinates of the point P, x_d, y_d and z_d be the coordinates of the point P inside the parallelepiped, i.e. distance from the minimum point of the parallelepiped : point A, see Figure B.1.

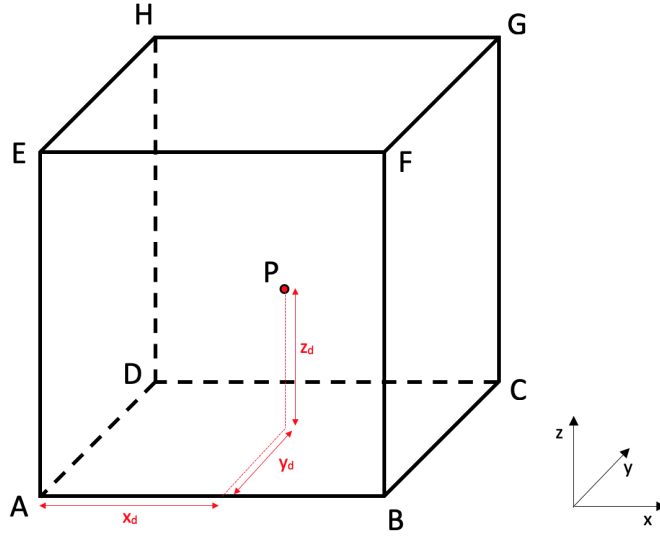


Figure B.1: Rectangular parallelepiped disposition, in which lies point P

We have :

$$x_d = x - x_A \quad y_d = y - y_A \quad z_d = z - z_A \quad (33)$$

Then we can compute weights, value between 0 and 1 along the three axis. Weights are ratio's of distance in the cube to the length of the corresponding parallelepiped edge.

$$w_x = \frac{x_d}{||AB||} \quad w_y = \frac{y_d}{||AE||} \quad w_z = \frac{z_d}{||AD||} \quad (34)$$

After that we can do the interpolation : first along x, then along y and finally along z. Let f be the function we want to interpolate.

In x direction :

$$f_{AB} = (1 - w_x)f_A + w_x f_B$$

$$f_{DC} = (1 - w_x)f_D + w_x f_C$$

$$f_{EF} = (1 - w_x)f_E + w_x f_F$$

$$f_{HG} = (1 - w_x)f_H + w_x f_G$$

In y direction :

$$f_{ABCD} = (1 - w_y)f_{AB} + w_y f_{DC}$$

$$f_{EFGH} = (1 - w_y)f_{EF} + w_y f_{HG}$$

In z direction :

$$f_P = (1 - w_z)f_{ABCD} + w_z f_{EFGH}$$

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

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