

The definitive version is available at <http://ieeexplore.ieee.org/>.

F. P. Vidal and P.-F. Villard and É. Lutton: Tuning of Patient Specific Deformable Models using an Adaptive Evolutionary Optimization Strategy. *IEEE Transactions on Biomedical Engineering* (October 2012), IEEE. Volume 59, Issue 10, pp. 2942-2949.

DOI: 10.1109/TBME.2012.2213251

Keywords: Evolutionary computation, inverse problems, medical simulation, adaptive algorithm.

IEEE Terms: Adaptation models, Biological system modeling, Computational modeling, Deformable models, Genetics, Liver, Optimization.

```
@article{Vidal2012IEEETransBiomedEng,
  author = {F. P. Vidal and {P.-F.} Villard and \ 'E. Lutton},
  title = {Tuning of Patient Specific Deformable Models using an Adaptive
    Evolutionary Optimization Strategy},
  journal = {IEEE Transactions on Biomedical Engineering},
  year = 2012,
  volume = 59,
  pages = {2942-2949},
  number = 10,
  month = oct,
  abstract = {We present and analyze the behavior of an evolutionary algorithm
    designed to estimate the parameters of a complex organ behavior model. The
    model is adaptable to account for patients specificities. The aim is
    to finely tune the model to be accurately adapted to various real patient
    datasets. It can then be embedded, for example, in high fidelity
    simulations of the human physiology. We present here an application focused
    on respiration modeling. The algorithm is automatic and adaptive.
    A compound fitness function has been designed to take into account for
    various quantities that have to be minimized. The algorithm efficiency is
    experimentally analyzed on several real test-cases:
    i) three patient datasets have been acquired with the breath hold protocol,
    and ii) two datasets corresponds to 4D CT scans. Its performance is
    compared with two traditional methods (downhill simplex and conjugate
    gradient descent), a random search and a basic realvalued genetic algorithm.
    The results show that our evolutionary scheme provides more significantly
    stable and accurate results.},
  doi = {10.1109/TBME.2012.2213251},
  pmid = {22907958},
  keywords = {Evolutionary computation, inverse problems, medical simulation,
    adaptive algorithm},
  publisher = {IEEE}
}
```

Tuning of Patient Specific Deformable Models using an Adaptive Evolutionary Optimization Strategy

F. P. VIDAL¹, P.-F. VILLARD², and É. LUTTON³

¹ University of Wales, Bangor, UK,

² LORIA, University of Lorraine, France,

³ AVIZ, INRIA Saclay-Île-de-France, France

Abstract

We present and analyze the behavior of an evolutionary algorithm designed to estimate the parameters of a complex organ behavior model. The model is adaptable to account for patients specificities. The aim is to finely tune the model to be accurately adapted to various real patient datasets. It can then be embedded, for example, in high fidelity simulations of the human physiology. We present here an application focused on respiration modeling. The algorithm is automatic and adaptive. A compound fitness function has been designed to take into account for various quantities that have to be minimized. The algorithm efficiency is experimentally analyzed on several real test-cases: i) three patient datasets have been acquired with the breath hold protocol, and ii) two datasets corresponds to 4D CT scans. Its performance is compared with two traditional methods (downhill simplex and conjugate gradient descent), a random search and a basic realvalued genetic algorithm. The results show that our evolutionary scheme provides more significantly stable and accurate results.

Keywords: Evolutionary computation, inverse problems, medical simulation, adaptive algorithm.

1 Introduction

High fidelity simulations of the human physiology require complex bio-mechanical models. Such models need also to be adaptable to account for patient's specificities. They can be used in various medical contexts, e.g. reducing motion artifacts in positron emission tomography (PET) and cone-beam computed tomography (CBCT) to improve the image quantification [15, 32], accurate dose calculation in radiotherapy treatment planning [16], and high fidelity computer-based training simulators [22]. The calibration and parametrization of the models are therefore critical to preserve realism and numerical accuracy. For training simulators, parameters are, however, often manually tuned using trial and error. This task is time consuming and it is not possible to ascertain that the results are optimal. Throughout the paper, we will consider patient specific respiration modeling with the deformation of the diaphragm and liver as the application example [30]. For each patient, 15 parameters need to be finely tuned. The parametrization of the model is considered as an inverse problem. It corresponds to fitting an analytic model with 15 parameters to experimental data. An optimization technique is deployed to solve the problem. Our approach makes use of an *ad-hoc* evolutionary algorithm (EA) that is able to successfully explore a search space with 15 dimensions (15-D). We choose an evolutionary framework because i) EAs can be used when little is known about the function to optimize, e.g. when no derivative is known, ii) this function does not need to be very smooth, iii) EAs can work with any search space, and iv) EAs are less likely to stop at local optima than classical deterministic optimisation methods. Our objective is to propose a fully automatic and adaptive methodology that leads to significantly better tuning. Our approach is

generic and can be generalized to other models when gold truth is available and the discrepancies between the model outputs and gold truth can be numerically assessed.

Section 2 briefly presents related work. The analytic model of respiration is described in Section 3. Details about the evolutionary algorithm are provided in Section 4. In Section 5, the performance of the algorithm is analyzed and compared with the performance of other methods. Section 6 presents a discussion about the limitations of our approach. Some conclusions are finally presented in Section 7.

2 Related Work

2.1 Optimization Based on Artificial Evolution

Artificial Evolution is the generic name of a large set of techniques that rely on the computer simulation of natural evolution mechanisms. Since the pioneering works of A. Fraser, H.-J. Bremermann, and after them, J. Holland and I. Rechenberg, Artificial Darwinism techniques have progressively gained a major importance in the domain of stochastic optimization and artificial intelligence [8].

Their basic idea is to copy, in a very rough manner, the principles of natural evolution, which let a population be adapted to its environment. According to Darwin’s theory, adaptation is based on a small set of very simple mechanisms: random variations, and survival/reproduction of the fittest individuals. Computer scientists have transposed this scheme into optimization algorithms. Their major advantage is to make only a few assumptions on the function to be optimized (there is no need to have a continuous or derivable function for instance). In short, Evolutionary Optimization considers a population of potential solutions exactly as a population of real individuals who lives, fights and reproduces in a natural environment. The pressure of this environment is replaced by an “optimization” pressure: the function to be optimized is considered as a measurement of the adaptation of the individual to its environment. In this way, individuals that reproduce are the best ones with respect to the problem to be solved. The reproduction consists in generating new solutions using a variation scheme (the genetic operators) that, by analogy to nature, is called mutation if it involves one individual, or crossover if it involves several individuals.

Evolutionary optimization techniques are particularly well suited to complex problems, where classical methods fail, e.g. due to the irregularity of the function or to the complexity of the search space. The versatility of the evolutionary framework has produced a variety of different optimization techniques suited for various purposes (multi-objective, interactive, cooperative) aimed at exploring different search spaces (discrete, combinatorial, continuous, tree-based, graph-based, grammar-based, constrained, limited or infinite). The major reason of this success is the tunable combination of oriented and random search mechanisms embedded in an evolutionary algorithm: it allows injecting *a priori* incomplete information in the genetic operators, while letting some other more unpredictable components be randomly searched.

Success stories in various application domains prove the power of these techniques as long as they rely on this “partial” randomness ability: the most successful applications are often based on hybridization of classical and evolutionary techniques. In such cases evolution is used as an “orchestra conductor”, that combines *a priori* information, and classical “local” optimizers in the evolution loop.

Considering evolutionary optimization as a “black box” is not a good strategy in general because one may lose an opportunity to adapt to the problem. Adapting the evolutionary mechanisms to the specificities of the problem usually improves the efficiency of the algorithms and reduces its computation time. In particularly difficult problems it is always helpful to compare evolutionary approaches to a pure random optimization to evaluate the improvement due to the “intelligence” set in the genetic operators. This is what we do in Section 5. In our context, the random search algorithm generates a given number of sets of 15 random parameters [29]. The set of parameters that provides the lowest fitness is extracted and corresponds to the solution of the optimization problem.

2.2 Medical Applications of Artificial Evolution

The use of evolutionary techniques in medicine is not a recent trend: J.T. Alander clearly shows the growth of interest for this topic since 1990 in the EA community [2]. In recent reviews, S.L. Smith and S. Cagnoni [27, 28] identified five main trends in this domain:

- medical imaging and signal processing (segmentation, registration, change detection, correction, denoising, 3D reconstruction)
- medical data and patient records mining
- clinical expert systems and knowledge-based expert systems
- therapy (tests, implants tuning)
- modeling and simulation

It is worth to note that these various applications take advantage of the flexibility of the evolutionary scheme, and rely on many variants of it (cooperation, multi-objective, agent-based approaches). It is also possible in many cases to control the trade-off between computational cost and precision of the results via a simple parameter tuning.

The work presented here is related to the first and last items of the trend list presented above. In the literature, model fitting usually relies on generic models, such as deformable image models in 2D [5, 18], and their extensions in 3D [19, 13]. Similar work on 3D surface registration [6] is also dealing with generic models for segmentation purposes.

Once again for a segmentation task, Heimann *et al.* address the problem of liver surface modeling [10]. They use statistical shape models combined with a freely deformable, energy-based snake model, and were able to obtain an average surface distance of about 1.6 mm in comparison to manual reference segmentations.

Here, we are relying on an accurate respiration model (see Section 3) that necessitates the simultaneous fitting of two surface models, for the diaphragm and the liver. In this paper, the evolutionary engine that has been developed is an attempt to deal with these two objectives by linear combination, with weights that adjust dynamically to balance the relative importance of the two objectives (see Section 4.2). To our best knowledge this problem has never been tackled using evolutionary computing.

2.3 Respiration Models

Respiratory models can be used in various contexts. The method depends on the application that is targeted. In radiotherapy, deformable models have to be accurate and focused on the targeted anatomical structures (e.g. tumor or organs at risk). Such models are generally based on continuous laws of mechanics, and equations are resolved using the finite-element technique [1]. In computer graphics animation, the reality is not the focus. It is rather the impression of realism. Therefore many details and organs should be modeled. In [33], diaphragm, ribs, intercostal muscles and the external skin are modeled and the deformation is achieved by using the mass-spring technique [23]. In medical simulation, the realism as well as the precision are needed to teach the users the real conditions of an operation. However the key challenge is often the computing time, e.g. both the visual and haptic renderings require a high refresh rate. In [14], the method is entirely based on geometrical constraints and the viscera are only modeled using a single envelope that wraps all the organs. We propose here a method that takes into account the motion of each organ due to the respiration using a geometrical method based on mechanical-based parameters.

3 Respiration Model

3.1 Anatomy and Physiology Analysis

Respiration is a complex process that mainly involves two muscles: the diaphragm and the intercostal muscles [21]. Respiration has also an indirect influence on all the organs of the abdominal cavity, in particular the liver. It is attached to the diaphragm by the falciform ligament, and hereby follows its deformation and motion. The diaphragm is composed of two parts: i) the tendon where the lungs and the heart lie, and ii) the muscle part that contracts and relaxes. The diaphragm is attached to the lower ribs and to the spine. In our case, the patient is lying on his/her back. We can therefore assume that the influence of the intercostal muscle is negligible.

3.2 Soft-Tissue Deformation

We will mainly focus here on i) the liver, which is the treated organ, and ii) the diaphragm, which is the active muscle as previously seen. We choose to compute the deformation using the generalized ChainMail extension [9]. The main advantage of this model is its small computing time. Instead of computing the deformation field based on time integration of the forces as other physically-based methods [20], it only uses geometrical equations that are quickly computed. It also preserves the use of parameters that have links with bio-mechanical approaches: the compression (S_i), the stretching (Str_i) and the shearing (Sh_i) where i designates the concerned organ.

To decompose the diaphragm into two parts as pointed out in Section 3.1, we choose the Cartesian equation of a plane ($a.x + b.y + c.z + d = 0$). As boundary conditions, we chose to impose a null displacement when the diaphragm is close to the ribs (defined by a distance D_{ribs}) and to impose a uniform 3D force to the whole tendon part (defined by (F_x, F_y, F_z)). The muscle part should deform. This deformation is governed here by the muscle intrinsic elasticity. The muscle stretches from the point attached to the ribs to the moving tendon.

The liver attachments to the diaphragm are modeled by a distance (D_{diaph}) determining which points directly follow the diaphragm. This rigid link simulates the compression of the liver by the diaphragm during the respiration process. The other parts of the liver deforms following the ChainMail algorithm.

3.3 Parameter Analysis

Various parameters have been extracted from the model previously described (see Fig. 1). These parameters are unique to each patient and need to be individually customized. There are bio-mechanical parameters, anatomy-based geometrical constrains and respiration pattern information. More details on these parameters can be found in [29].

3.4 Model Evaluation

To optimize the model parameters we need datasets of different patients but also metrics to evaluate the accuracy of the respiration model. The simulation always starts at the real inspiration state (Geometry G_R^I) extracted from the real data. The aim is to reach the real expiration state (Geometry G_R^E) also extracted from the real data. In the 4D CT scan case (Geometry G_R^t), going through intermediate states t is also achievable. The metrics assesses the accuracy of a simulated geometry G_S^t compared to the ground truth geometry extracted at the same time step t .

To compute the difference between two geometries, we choose to analyze for each mesh vertex the point-to-surface distance according to [3]. It is based on a distance measure $d(p, G')$ between a point p belonging to a surface G and a surface G' as follows:

$$d(p, G') = \min_{p' \in G'} \|p - p'\| \quad (1)$$

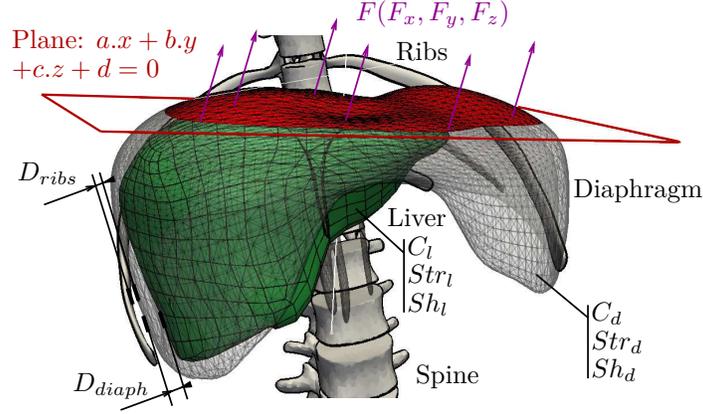


Figure 1: List of the respiration model parameters.

In this application, the root mean square error ($\mathcal{E}_{RMS}(G_S^t, G_R^t)$) defined by Eq. (2) is used.

$$\mathcal{E}_{RMS}(G_S^t, G_R^t) = \sqrt{\frac{1}{G_S^t} \iint_{p \in G_S^t} d(p, G_R^t)^2 dS} \quad (2)$$

4 Optimization Algorithm

An evolutionary algorithm is used to finely tune the parameters of the deformable model. The core of the genetic engine has been written in C++ using templates. It makes the approach generic, so that it can be used for other optimization problems. However, for each specific optimization problem, it is necessary to implement the corresponding genotype (e.g. the search space), some of the genetic operators (e.g. mutation and crossover), as well as the fitness function. This section describes in detail how the EA has been implemented.

4.1 Genotype

Fifteen parameters in total have been identified in Sections 3.2 and 3.3 and are illustrated in Fig. 1. The evolutionary algorithm explores this 15-D search space. Each parameter corresponds to a real number, whose boundary values are known. The genome of individuals is therefore made of 15 floating point numbers.

4.2 Adaptive Fitness Function

A metric ($\mathcal{E}_{RMS}(M_0, M_1)$) is presented in Section 3.4 to evaluate the discrepancies between two polygon meshes M_0 and M_1 . It is therefore possible to quantify the difference between the mesh ($G_S^t(i)$) simulated using the deformable model with the parameters corresponding to Individual i , and the real mesh (G_R^t) extracted from the patient's dataset at state t . For each individual, two metrics are computed (one for the diaphragm, and one for the liver). These values can be used to define the fitness function (*fitness*) corresponding to i . The optimization consists in minimizing *fitness*. The simplest function is:

$$rClfitness(i) = \alpha \mathcal{E}_{RMS}(G_S^t(i, diaph), G_R^t(diaph)) + (1 - \alpha) \mathcal{E}_{RMS}(G_S^t(i, liver), G_R^t(liver)) \quad (3)$$

with $0 \leq \alpha \leq 1$ to give more or less weight to the diaphragm or the liver. Selecting the value of α is not trivial because the numerical quantity of the error for the diaphragm and the liver can be significantly different. If the same weight is applied to both tissue error measurement (i.e. α

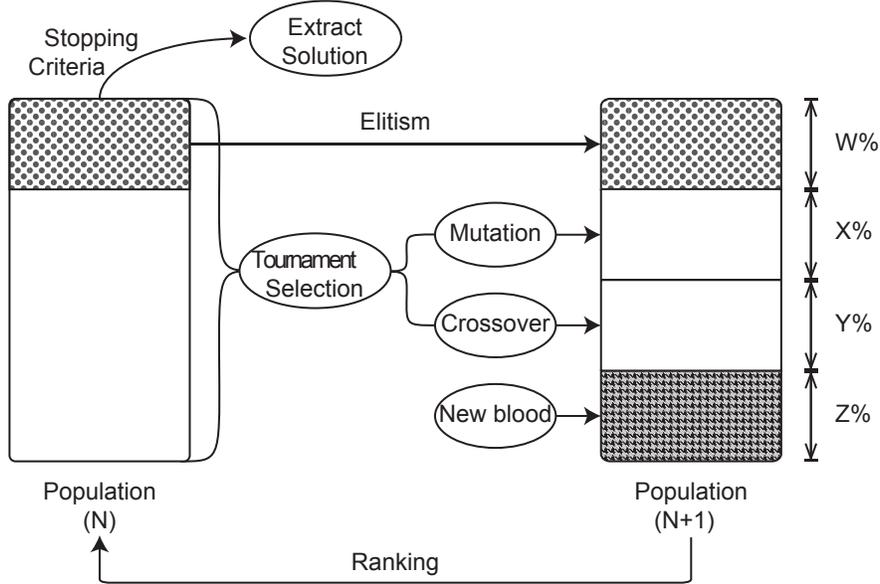


Figure 2: Evolutionary loop.

is equal to 0.5), the predominant quantity will then have more influence on the optimization process. We would actually expect instead the importance of both tissues to be the same during the minimization. In an application such as the real-time simulation of liver puncture, a higher level of fidelity is required for the liver than the diaphragm. Scaling factors on errors are introduced to give the same relative weight to the diaphragm and liver. Eq. 3 becomes:

$$\begin{aligned}
 rClfitness(i) = & \frac{\alpha}{\mathcal{E}_{RMS}^{diaph}} \mathcal{E}_{RMS} (G_S^t(i, diaph), G_R^t(diaph)) + \\
 & \frac{(1-\alpha)}{\mathcal{E}_{RMS}^{liver}} \mathcal{E}_{RMS} (G_S^t(i, liver), G_R^t(liver))
 \end{aligned} \tag{4}$$

with $\mathcal{E}_{RMS}^{liver}$ and $\mathcal{E}_{RMS}^{diaph}$ the error metrics of the best individual provided by the previous generation for the liver and diaphragm respectively. For each iteration of the evolution loop, these metrics are updated. For practical reasons, α is rescaled so that the sum $\frac{\alpha}{\mathcal{E}_{RMS}^{diaph}} + \frac{(1-\alpha)}{\mathcal{E}_{RMS}^{liver}}$ is equal to one.

4.3 Genetic Engine

An elitist generational genetic approach is used (see Fig. 2). For each optimization loop performed by the genetic engine, a new population based on the previous generation is created. The population size is made constant. Each gene of an individual is initialized using an uniform distribution to randomly chose a value within its range of validity. During the optimization loop, a genetic operator is randomly chosen to create a new individual. Genetic operators are: i) *elitism*, ii) *mutation*, iii) *crossover*, and iv) *new blood*. They are described in detail below. Each operator is assigned a probability of occurrence, W , X , Y , and Z respectively. The sum of the operators' probability is 100%. Some operators (e.g. the mutation and the crossover) require one or several individuals of the previous generation to be randomly chosen for the reproduction (i.e. the creation of a new individual based on the chosen individual(s)). This process is performed by the *selection* operator. The optimization stops when a stopping criteria is reached.

4.3.1 Tournament Selection

During the evolution, individuals are selected using tournaments. For each selection, n individuals are randomly chosen, without any bias. The best individual amongst the n chosen ones is the individual who is selected.

4.3.2 Elitism

$W\%$ best individuals of the previous generation are kept in the new population. In practice, even if W is equal to 0% , the best individual is kept. In this case, if nothing better is found, the best individual is not lost.

4.3.3 Mutation

$X\%$ of the new individuals are created by mutating individuals of the previous generation. It mimics mutations in a genomic sequence that are observed in molecular biology and genetics. For each mutation, an individual is selected using the selection operator. The genotype of the selected individual is copied into the genotype of the new individual. The numerical value of each gene is then slightly altered to introduce spontaneous and random changes. The range of possible random changes is controlled by the value σ . In classical implementations, this value is fixed. However theoretical studies have shown that a variable, adaptive or self-adaptive mutation is beneficial in optimization [4]. Adaptive mutation is actually a scheme that has been made experimentally evident in natural population (see for instance [12] about stress mutation in bacteria populations). Here, we have chosen to use an adaptive strategy for σ , in the style of stress mutation. The adaptation rule produces greater σ for “bad individuals” (i.e. large fitness) and smaller ones for “good individual” (small fitness). The idea is to favor large exploration around bad individuals, whilst performing fine tuning in the vicinity of good individuals. σ is then controlled depending on the fitness value. It makes use of a scaled and stretched cosine function:

$$\sigma(f) = \begin{cases} \sigma_{min}, & f < fit_{min} \\ \sigma_{max}, & f > fit_{max} \\ \sigma_{min} + (\sigma_{max} - \sigma_{min}) \times \\ \quad \frac{\cos\left(\pi \times \frac{f - fit_{min}}{fit_{max} - fit_{min}}\right) + 1.0}{2.0}, & \text{otherwise} \end{cases} \quad (5)$$

f corresponds to the fitness of the individual who will undergo a mutation. $\sigma(f)$ smoothly varies between the smaller (fit_{min}) and the larger (fit_{max}) fitness thresholds respectively. If f is smaller than fit_{min} , σ is then σ_{min} ; if the individual’s fitness is greater than fit_{max} , σ is then σ_{max} (with σ_{min} and σ_{max} two constant values set by the user).

Mutation is performed via the addition of a shift *offset* to the genome’s value, whose value is:

$$offset = \frac{range}{2} \times \sigma(Fitness(i)) \times k \quad (6)$$

with *range* the interval size of the valid values of the genomes, and $\sigma(Fitness(i))$ the amount of mutation corresponding to the fitness of Individual i , and k a random number in the interval $[-1, 1]$.

4.3.4 Blend Crossover

$Y\%$ of the new individuals are created using BLX (for blend crossover) [11]. For each crossover operation, two individuals (i_1 and i_2) are selected using the selection operator. Each component (or chromosome) of the genome of the new individual will be given by a random combination of the corresponding chromosome of i_1 and i_2 as follows:

$$C = kC_1 + (1 - k)C_2 \quad (7)$$

with C the numerical value of the chromosome of the new individual, C_1 the numerical value of i_1 ’s chromosome, C_2 the numerical value of i_2 ’s chromosome, and k a random number between 0 and 1. For each chromosome, a new k value is chosen.

Name	Image size	Spacing [mm ³]	Protocol
Patient A	512×512×136	1.08×1.08×2.5	Breath hold
Patient B	512×512×75	0.98×0.98×5	Breath hold
Patient C	512×512×139	1.17×1.17×2	Breath hold
Patient D	512×512×141	0.98×0.98×2	4D CT scan
Patient E	512×512×287	0.71×0.71×1	4D CT scan

Table 1: Patient dataset properties.

4.3.5 New Blood

$Z\%$ of individuals are replaced by new random individuals. Such individuals are created using the same method as during the initialization. This operator allows “new blood” to continually enter the population. It corresponds to a “pure random search” component and maintains a minimal diversity level in the population.

4.3.6 Stopping Criteria

Stagnation is the stopping criteria that has been chosen, i.e. the optimization process is stopped when artificial evolution does not improve the result. In practice, when s successive evolution loops provide exactly the same best individual, the optimization process is stopped, and this individual is the solution to the optimization problem.

5 Results and Validation

This section shows the performance of the evolutionary algorithm in term of accuracy. The results are compared with the outputs of: i) a pure random search to evaluate the improvement provided by the genetic operators, ii) a basic real-valued genetic algorithm (GA) used as a black box evolutionary optimization to assess the efficiency of our new genetic operators, and iii) more traditional methods – Downhill simplex method [24] and Powell’s conjugate gradient descent method [26] – for further comparisons. For each tested case, every stochastic optimization process has been repeated 15 times. For each optimization process, the errors were recorded and taken into account during the statistical tests presented in this section. It allows us to ascertain the effectiveness and usefulness of the evolutionary algorithm, i.e. to demonstrate that it outperforms the brute force algorithm, the black box EA, and the classic methods.

5.1 Input Data

Abdominal images with respiration information are routinely acquired during radiotherapy treatment planning when knowledge of the tumor displacement is required to perform the dose calculations. Five patient specific datasets have been selected (see Table 1). Three datasets (Patients A, B and C) have been acquired with the “breath hold” protocol, i.e. with only two time steps corresponding to the inhale and exhale states. The patients are asked to hold their breath following the “ABC” protocol [31]. Two datasets (Patients D and E) correspond to 4D CT scans with ten time steps each. The data was acquired over the respiratory cycle while the patient breathes normally. For Patients A, B and C, a single optimization problem each needs to be solved. For patients with 4D CT scans, one optimization problem per time step needs to be solved. Note that for such patients, the values presented in this section correspond to average values over the 10 time steps.

Every CT scan has been segmented to extract the organs that are required by the simulation model. Medical expert collaborators have manually segmented the data using a graphic tablet. The contouring process has been performed using ITKSnap (<http://www.itksnap.org/> (last accessed: 31/05/2012)). Polygon meshes were then exported after using the Marching Cube algorithm [17]. Meshes were decimated and smoothed to have about 2,000 vertices per organ. The manual segmentation process of a CT scan takes on average 2 hours per liver and 3 hours per

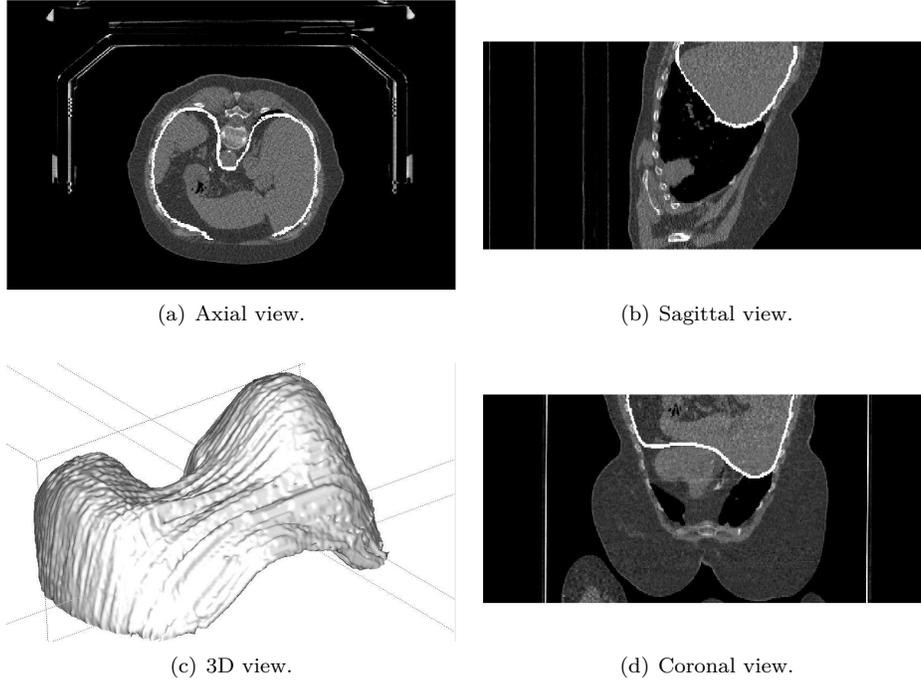


Figure 3: Diaphragm manual segmentation (here in white).

population size (n)	50 and 200
α	33%
tournament size	5% of the population size
elitism (W)	9%
mutation probability (X)	55%
crossover probability (Y)	35%
new blood probability (Z)	1%
σ_{min}	0.001
σ_{max}	0.2
stopping criteria (s)	10

Table 2: Evolutionary algorithm parameters.

diaphragm. Fig. 3 shows an input medical image with the manual segmentation of Patient B’s diaphragm. In total 26 3D CT scans were segmented.

5.2 Evolutionary Algorithm Parameters

Table 2 provides a summary of the algorithm’s parameters. Two different sizes of population (50 and 200) are used to assess the stability of the algorithm. α is set to 33% to give more weight to the liver than the diaphragm, without neglecting the diaphragm.

5.3 Performance Comparison

Box plots have been produced for each patient. Fig. 4 presents the results for Patient B as an illustration. Figures for other patients present the same features. It shows the errors for both the diaphragm and the liver after optimization using the different methods. A population of 200 individuals provides relatively stable results with the *Ad hoc* EA. This population size has therefore been used for further statistical analysis with other methods. Note that for each patient care is given to use the same amount of computing time for the random search than our *ad-hoc*

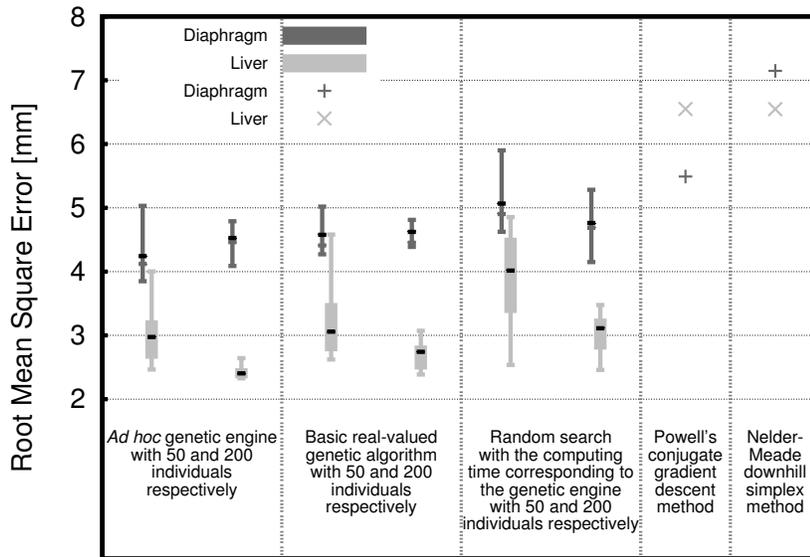


Figure 4: Root mean square errors for the Patient B’s diaphragm and liver using the *Ad hoc* EA, pure random search, basic real-valued GA, downhill simplex, and conjugate gradient descent methods.

	Patient A	Patient B	Patient C	Patient D	Patient E
<i>Ad hoc</i> EA	15093	5933	8320	6753	11493
Pure random search	15093	5933	8320	6753	11493
Basic real-valued GA	26053	16107	37253	13928	∞
Downhill simplex	1135	3012	965	690	3010
Conjugate gradient descent	4351	7298	2907	4390	17867

Table 3: Average number of fitness evaluations.

evolutionary algorithm (see Table 3 for the numerical values). In one case (Patient E), the basic real-valued genetic algorithm fails to converge in an acceptable time.

Fig. 5 presents the evolution of the diaphragm weight ($\alpha/\mathcal{E}_{RMS}^{diaph}$) in the fitness function of Patient A (note that the weight of the liver is complimentary and it corresponds to $1 - \alpha/\mathcal{E}_{RMS}^{diaph}$). The weights are updated for each iteration of the evolution loop (see Section 4.2). It shows that the final weight are consistent amongst the 15 runs. For each run, the shape of the curve is similar. The same phenomena are observed for every patient. It is therefore acceptable to use the “evolved” fitness function for comparison with other optimization methods when applicable.

Table 4 shows the average root mean square error (\mathcal{E}_{RMS}) of the liver and diaphragm for each patient and for each optimization technique. It also presents the result of a non-parametric statistical hypothesis test – the Wilcoxon signed-rank test – that is used to compare the performance of our *Ad hoc* EA with other methods. The size of the samples is 15. If the value of the Wilcoxon signed-rank test (\mathcal{W}) is negative, then the *Ad hoc* EA performs better than the other algorithm. If \mathcal{W} is close to 0, then both algorithms are performing similarly. The results show that only our evolutionary algorithm can minimize successfully both the error of the liver and diaphragm. The only positive \mathcal{W} value has been obtained for Patient A’s diaphragm using the gradient descent method. However the corresponding value for the liver was negative. Classic optimization methods fail to explore the 15-D search space to minimize the two error values. This is particularly true for the downhill simplex method. The basic real-valued genetic algorithm is very slow and fails to converge in some cases (see Patient E in Table 3). In our application, artificial evolution

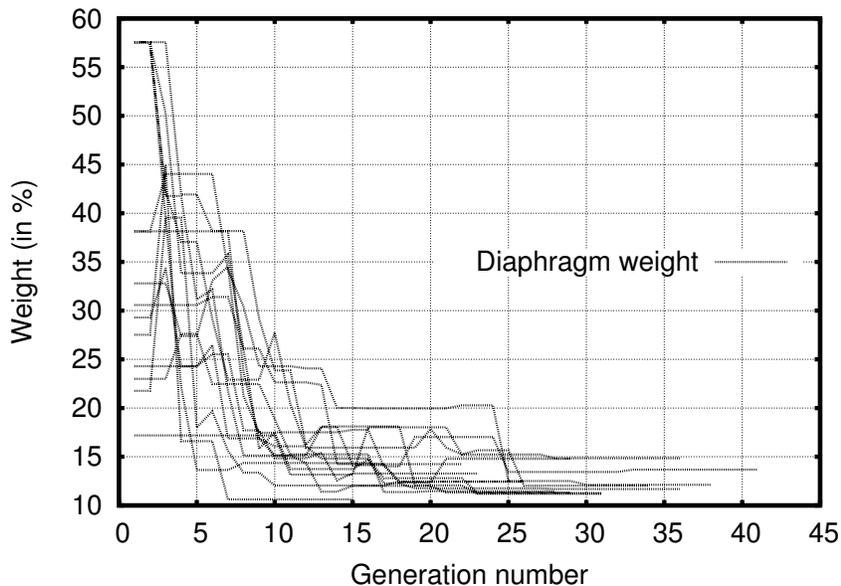


Figure 5: Evolution of the diaphragm weight, $\alpha/\mathcal{E}_{RMS}^{diaph}$, for 15 optimization runs for Patient B.

		Patient A		Patient B		Patient C		Patient D		Patient E	
		$\overline{\mathcal{E}_{RMS}}$	\mathcal{W}								
<i>Ad hoc</i> EA	Diaph	2.05	N.A.	4.53	N.A.	2.06	N.A.	3.63	N.A.	8.84	N.A.
	Liver	2.95	N.A.	2.42	N.A.	2.85	N.A.	2.16	N.A.	4.43	N.A.
Pure random search	Diaph	2.27	-96	4.77	-84	2.17	-102	4.06	-93	9.30	-45
	Liver	3.18	-88	3.02	-118	2.94	-120	2.24	-42	4.87	-34
Basic real-valued GA	Diaph	2.21	-102	4.58	-18	2.10	-76	3.81	-12	N.A.	N.A.
	Liver	3.13	-44	2.69	-120	2.88	-78	2.45	-35	N.A.	N.A.
Downhill simplex	Diaph	3.67	-120	7.15	-120	9.65	-120	5.22	-119	9.95	-68
	Liver	21.89	-120	6.55	-120	13.04	-120	8.39	-120	12.72	-120
Conjugate gradient descent	Diaph	2.00	96	5.49	-120	2.33	-120	5.33	-112	9.91	-33
	Liver	5.96	-120	6.55	-120	5.47	-120	3.55	-109	6.41	-50

Table 4: Performance comparison of the *Ad hoc* evolutionary algorithm, pure random search, basic real-valued genetic algorithm, downhill simplex, and conjugate gradient descent methods. $\overline{\mathcal{E}_{RMS}}$ is the average root mean square error in millimeters. \mathcal{W} is Wilcoxon signed-rank test between the *Ad hoc* EA and the other methods.

can effectively explore a large search space without *a priori* knowledge.

Table 5 presents the resulting parameters found by our method for the 15 optimization runs of each patient. The average value and the standard deviation are given for each parameter. Some parameters found over 15 runs are relatively close (e.g. the diaphragm plane (a , b , c and d)) whilst others may significantly differ (e.g. D_{ribs} and Str_l).

6 Discussion

The comparison with pure random search provides a good assessment of the added value of the genetic engine, i.e. the improvement due to the random search oriented by the genetic operators over a “blind” random search. The pure random search used the “evolved” fitness function for comparison. It represents a disadvantage for the evolutionary approach. It is difficult to evaluate the algorithmic cost of the adaptation of the fitness function in the evolutionary algorithm. It is therefore not considered in the comparisons displayed in Fig. 4 and Tables 3 and 4. Despite

	Patient A	Patient B	Patient C	Patient D	Patient E
a	0.0 ± 0.1	0.0 ± 0.1	0.0 ± 0.1	0.0 ± 0.1	0.0 ± 0.1
b	0.0 ± 0.1	0.0 ± 0.1	0.0 ± 0.1	0.0 ± 0.1	0.0 ± 0.1
c	-0.9 ± 0.0	-0.9 ± 0.0	-0.9 ± 0.0	-0.9 ± 0.0	0.9 ± 0.0
d	114.1 ± 7.8	113.7 ± 9.2	110.8 ± 13.5	117.3 ± 10.1	-99.5 ± 14.8
C_d	0.4 ± 0.2	0.4 ± 0.2	0.6 ± 0.2	0.5 ± 0.2	0.4 ± 0.2
Str_d	5.1 ± 1.3	5.2 ± 1.8	5.6 ± 1.9	5.2 ± 2.1	5.5 ± 1.7
Sh_d	4.7 ± 1.7	3.9 ± 2.0	4.9 ± 2.0	5.0 ± 2.4	4.5 ± 1.9
F_x	0.6 ± 0.6	3.9 ± 0.4	0.6 ± 0.1	2.0 ± 2.3	0.3 ± 2.7
F_y	3.5 ± 0.2	1.8 ± 0.7	6.2 ± 0.7	-2.4 ± 2.0	-1.3 ± 2.8
F_z	-5.8 ± 1.0	-8.1 ± 0.5	-7.9 ± 0.4	1.3 ± 1.8	4.2 ± 3.9
D_{ribs}	19.7 ± 14.2	15.9 ± 13.6	19.9 ± 5.1	11.3 ± 7.8	11.7 ± 10.6
C_l	0.5 ± 0.1	0.4 ± 0.2	0.5 ± 0.2	0.5 ± 0.2	0.5 ± 0.2
Str_l	8.2 ± 6.8	5.9 ± 3.7	16.7 ± 5.5	5.2 ± 2.0	4.9 ± 1.9
Sh_l	1.5 ± 1.7	4.3 ± 2.6	5.9 ± 1.9	4.6 ± 2.4	2.6 ± 1.6
D_{diaph}	3.3 ± 0.7	1.3 ± 0.7	0.7 ± 0.3	4.1 ± 2.2	4.0 ± 0.7

Table 5: Parameters after optimization using the *Ad hoc* EA.

this structural disadvantage, the evolutionary approach surpasses the random search. For a given amount of computation time, the best results are obtained using our evolutionary algorithm.

Our genetic engine is also more efficient than the basic real-valued genetic algorithm used as a black box optimizer. Our method provides more accurate results using less computational power. Our algorithm also outperforms the classic optimization methods that were used for comparison purposes.

The diaphragm errors are larger than liver errors for Patients B, D and E. This is due to the fact that the spatial or time resolution of the initial datasets was lower than in the other test cases. For Patient B, the slice thickness is 5 mm, i.e. much larger than the diaphragm thickness (less than 2 mm). Patients D and E make use of 4D CT data. In such cases, the inaccuracies are due to the motion artifact (known for blurring tissue edges). For Patients B, D and E, the reconstructed surface of the diaphragm is then over-smoothed. The quality of the geometrical models is then less accurate. The diaphragm is a relatively thin tissue. Inaccuracies are attenuated for the liver as it is a much larger organ than the diaphragm.

Let’s analyze the value of parameters presented in Table 5. The plane coefficients (a and b) are very close to zero. It was to be expected as it roughly corresponds to a transverse plane. The force (F_x, F_y, F_z) is mainly orthogonal to this plane. The ChainMail parameters are very high for both organs. This is due to the large deformations that occur: the contraction and relaxation of diaphragm muscle as well as the soft-tissue behavior of the liver. Finally, the attachment distance is higher from the diaphragm to the ribs (D_{ribs}) than from the liver to the diaphragm (D_{diaph}). This is because D_{diaph} mimics a very thin ligament attaching the liver to the diaphragm.

Finally, Fig. 6 shows 3D plots of surface meshes for all the patients. The printed color depends on a lookup table (LUT) that corresponds to the localized error. Its range varies from blue for no error to red for the maximum error. The motion is fairly well compensated using our genetic algorithm.

7 Conclusion and Future Work

We have presented an artificial evolution strategy to finely tune the parameters of a multidimensional model of respiration with soft tissue deformations. Its efficiency has been validated using five datasets of real patients (that is 23 different optimization problems in total). The advantage of artificial evolution over the downhill simplex, the conjugate gradient descent, the purely random search, and a black box basic real-valued genetic algorithm has also been demonstrated. Results obtained using our artificial evolution framework were both more accurate and more stable.

The proposed evolutionary optimization is adaptive in two ways: i) mutation is adapted based on fitness, and ii) the weight the two-objectives compound fitness is automatically balanced along calculations.

We also demonstrated that this compound fitness function effectively takes into account various

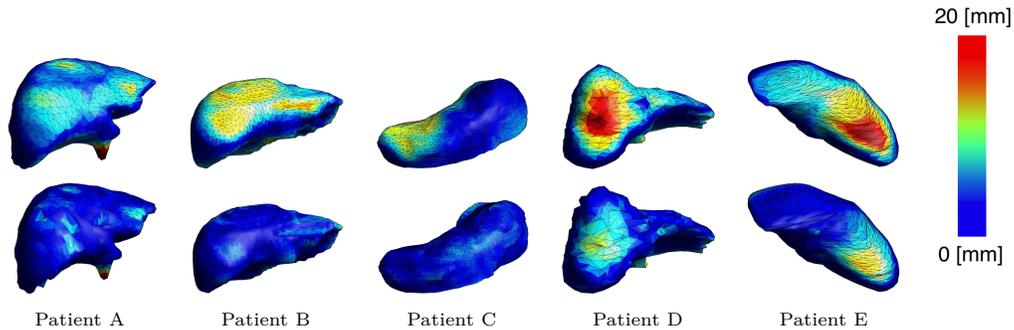


Figure 6: 3D plots of surface meshes with localized errors. The first row shows the initial difference map between real inhale and real exhale states. The second row shows the difference between real exhale and simulated exhale with our genetic algorithm.

properties of the model, e.g. minimizing several error values for the liver and the diaphragm. The approach can be generalized to other models when gold truth is available and the discrepancies between the model outputs and gold truth can be numerically assessed.

The current solution that is to balance the different objectives in a single fitness function can be revisited. We will investigate other composition strategies (e.g. multiplicative instead of additive). Other evolutionary approaches will also be considered. A cooperative-coevolutionary approach [25] can be used as the problem we presented here includes most of the features that have been identified to be difficult to solve using single-population evolutionary algorithms but it is efficiently addressable using cooperative-coevolution:

- the search space is complex (e.g. 15-D)
- the problem can be split (e.g. minimizing the error for both the liver and the diaphragm)
- there is a strong interdependence between the components of the solution (e.g. the diaphragm is influencing the liver)

Also a Classical multi-objective evolutionary approach like the famous NSGA-II [7] will be also considered for dealing with multiple objectives. Since these methods are able to perform an optimization without setting any priority between the objectives, they provide a set of potential compromises between the various objectives, the Pareto set. To be applicable in our context it is necessary to select a single acceptable solution from the Pareto set at the end of the process. The superiority of such a strategy in our context is not obvious, as the time spent to obtain a good approximation of the whole Pareto set may be better used to perform calculations focused on a single compound and well chosen objective.

References

- [1] A. Al-Mayah, J. Moseley, M. Velec, and K. Brock. Effect of friction and material compressibility on deformable modeling of human lung. In *Proceedings of the 4th international symposium on Biomedical Simulation, ISBMS '08*, pages 98–106. Springer-Verlag, 2008.
- [2] J. T. Alander. Indexed bibliography of genetic algorithms in medicine. Report 94-1-MEDICINE, University of Vaasa, Department of Electrical Engineering and Automation, 2007. Updated 2010/01/15.
- [3] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. MESH: Measuring Errors between Surfaces using the Hausdorff distance. In *Proc IEEE Int Conf on Multimedia and Expo*, volume I, pages 705–708, 2002.

- [4] T. Bäck. Optimal mutation rates in genetic search. In *Proc 5th Int Conf on Genetic Algorithms*, pages 2–8, 1993.
- [5] L. Ballerini. Genetic snakes for color images segmentation. In *Applications of Evolutionary Computing: EvoWorkshops 2001*, volume 2037 of *Lect Notes Comput Sc*, pages 268–277, 2001.
- [6] C. K. Chow, H. T. Tsui, and T. Lee. Surface registration using a dynamic genetic algorithm. *Pattern Recogn*, 37(1):105–117, 2004.
- [7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T Evolut Comput*, 6(2):182–197, 2002.
- [8] D. B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, pages 106–107. Wiley-IEEE Press, 3 edition, Dec. 2005.
- [9] S. F. Gibson. 3D chainmail: a fast algorithm for deforming volumetric objects. In *Proc Symp on Interactive 3D Graphics*, pages 149–154, 1997.
- [10] T. Heimann, S. Münzing, H.-P. Meinzer, and I. Wolf. A shape-guided deformable model with evolutionary algorithm initialization for 3D soft tissue segmentation. In *Inf Process Med Imaging*, pages 1–12, 2007.
- [11] F. Herrera, M. Lozano, and A. M. Sánchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *Int J Intell Syst*, 18(3):309–338, 2003.
- [12] M. N. Hersh, R. G. Ponder, P. J. Hastings, and S. M. Rosenberg. Adaptive mutation and amplification in *Escherichia coli*: two pathways of genome adaptation under stress. *Res Microbiol*, 155(5):352–9, 2004.
- [13] A. Hill, A. Thornham, and C. J. Taylor. Model-based interpretation of 3D medical images. In *Br Mach Vis Conf*, pages 339–348. BMVA Press, 1993.
- [14] A. Hostettler, S. A. Nicolau, Y. Rémond, J. Marescaux, and L. Soler. A real-time predictive simulation of abdominal viscera positions during quiet free breathing. *Progress in Biophysics and Molecular Biology*, 103(2 - 3):169 – 184, 2010.
- [15] F. Lamare, M. J. Ledesma Carbayo, T. Cresson, G. Kontaxakis, A. Santos, C. Cheze Le Rest, A. J. Reader, and D. Visvikis. List-mode-based reconstruction for respiratory motion correction in PET using non-rigid body transformations. *Phys Med Biol*, 52(17):5187–5204, 2007.
- [16] J. H. Lewis and S. B. Jiang. A theoretical model for respiratory motion artifacts in free-breathing CT scans. *Phys Med Biol*, 54(3):745–755, 2009.
- [17] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, 1987.
- [18] C. McIntosh and G. Hamarneh. Spinal crawlers: Deformable organisms for spinal cord segmentation and analysis. In *Med Image Comput Comput Assist Interv*, volume 4190 of *Lect Notes Comput Sc*, pages 808–815, 2006.
- [19] C. McIntosh and G. Hamarneh. Vessel Crawlers: 3D physically-based deformable organisms for vasculature segmentation and analysis. In *Conf Comput Vis Pattern Recognit*, pages 1084–1091, 2006.
- [20] U. Meier, O. Lopez, C. Monserrat, M. C. Juan, and M. Alcaniz. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, 77(3):183–197, 2005.

- [21] K. Moore, A. F. Dalley, and A. M. R. Agur. *Clinically Oriented Anatomy*. Lippincott Williams & Wilkins, 6 edition, 2009.
- [22] P. J. Morgan, D. Cleave-Hogg, S. Desousa, and J. Lam-McCulloch. Applying theory to practice in undergraduate education using high fidelity simulation. *Med Teach*, 28(1):e10–e15, 2006.
- [23] L. P. Nedel and D. Thalmann. Real time muscle deformations using mass-spring systems. In *Proceedings of the Computer Graphics International 1998*, pages 156–165, 1998.
- [24] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.
- [25] C. A. Pena-Reyes and M. Sipper. Fuzzy CoCo: A cooperative-coevolutionary approach to fuzzy modeling. *IEEE Trans. Fuzzy Syst.*, 9(5):727–737, 2001.
- [26] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7(2):152–162, 1964.
- [27] S. L. Smith and S. Cagnoni. Introduction to the special issue on medical applications of genetic and evolutionary computation. *Genet Program Evol M*, 8(4):297–299, Dec. 2007.
- [28] S. L. Smith and S. Cagnoni. *Genetic and Evolutionary Computation: Medical Applications*. John Wiley & Sons, 2011.
- [29] P.-F. Villard, F. P. Vidal, F. Bello, and N. W. John. A method to compute respiration parameters for patient-based simulators. In *Medicine Meets Virtual Reality 19 - NextMed*, volume 173 of *Stud Health Technol Inform*, pages 529–533, 2012.
- [30] P.-F. Villard, F. P. Vidal, C. Hunt, F. Bello, N. W. John, S. Johnson, and D. A. Gould. A prototype percutaneous transhepatic cholangiography training simulator with real-time breathing motion. *Int J Comput Assist Radiol Surg*, 4(6):571–578, 2009.
- [31] J. W. Wong, M. B. Sharpe, D. A. Jaffray, V. R. Kini, J. M. Robertson, J. S. Stromberg, and A. A. Martinez. The use of active breathing control (ABC) to reduce margin for breathing motion. *Int. J. Radiation Oncology Biol. Phys.*, 44(4):911–919, 1999.
- [32] Q. Zhang, Y.-C. Hu, F. Liu, K. Goodman, K. E. Rosenzweig, and G. S. Mageras. Correction of motion artifacts in cone-beam CT using a patient-specific respiratory motion model. *Med Phys*, 37(6):2901–2909, 2010.
- [33] V. B. Zordan, B. Celly, B. Chiu, and P. C. DiLorenzo. Breathe easy: model and control of simulated respiration for animation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '04, pages 29–37, 2004.