Neural Processing Letters (2021) 53:2221–2241
https://doi.org/10.1007/s11063-021-10512-y

Performance of Genetic Algorithm and Levenberg Marquardt Method on Multi-Mother Wavelet Neural Network Training for 3D Huge Meshes Deformation: A Comparative Study

Naziha Dhibi[1] · Chokri Ben Amar[2]

Accepted: 3 April 2021 / Published online: 14 April 2021

Abstract
We propose, in this paper, a novel technique for large Laplacian boundary deformations using estimated rotations. The introduced method is used in the domain of Region of Interest (ROI) to align features of mesh based on Multi Mother Wavelet Neural Network (MMWNN) structure found in several mother wavelet families. The wavelet network allows the alignment of the characteristic points of the original mesh towards the target mesh. The key component of our correspondence scheme is a deformation energy that penalizes geometric distortion, encourages structure preservation and simultaneously allows mesh topology changes. To ensure the design of wavelet neural network architecture, an optimization algorithm should be applied to estimate and optimize the network parameters. Therefore, we compare our approach of 3d mesh deformation using MMWNN architecture based on genetic algorithm and our approach relying on Levenberg-Marquardt Method. We also discuss the existing comparison metrics for static and deformed triangle meshes employing the two mentioned approaches. Besides, we enumerate their strengths, weaknesses and relative performance.

Keywords Genetic algorithm · Levenberg Marquardt · Multi mother wavelet neural network · 3D mesh deformation

1 Introduction

With technological advances in multimedia, telecommunication, hardware and computer design, the use of 3D data is now well established in several industrial domains, such as scientific visualization, entertainment, computer-aided design, etc. Mesh-based deformation has recently developed, which allowed the interactive deformation of the objects while preserving

✉ Naziha Dhibi
dhibi.naziha@gmail.com

Chokri Ben Amar
chokri.benamar@tu.edu.sa

[1] REGIM: Research Groups on Intelligent Machines, National School of Engineers (ENIS), University of Sfax, 3038 Sfax, Tunisia

[2] Department of Computer Engineering, College of Computers and Information Technology, Taif Univerity, Taif, Saudi Arabia

various types of constraints. The deformation of 3D surface mesh starts by the displacement of the control vertices. It consists in setting the target positions of the control vertices (directly or by using an affine transformation applied to a control vertex or a range of control vertices). Then, a rotation or a translation of a control vertex is constantly applied on its last target position set. In this technique, the object is encoded, using differential equations, and the mesh is deformed by modifying the boundary conditions of the differential equations. In fact, it is a fundamental modeling operation used to change the local shape of a mesh. Deformation transfer applies the deformation showed by a source triangle mesh onto a different target triangle mesh. It computes the set of transformations induced by the deformation of the source mesh and maps the transformations through the correspondence from the source to the target. In the area of function approximation, 3D system modelling the wavelet network and the neural network are generally used as non-parametric estimators. Mostly, the neural network modeling consists of two steps: determining the network structure and adjusting the network parameters using back-propagation (BP) algorithms. For example, in [1–3] wavelet networks represent an implementation of wavelet decomposition. This recently-developed technique, is an efficient tool utilized in many applications applied in the signal processing domain; such as function approximation and 3D data modeling [4,5]. The wavelet network structure has also been applied in nonlinear system identification [6–8]. In the last decades, many research studies on the wavelet network algorithms have been carried out. We proposed in [9] a novel and an efficient framework of 3D mesh deformation, based on multi-mother wavelet neural network architecture and trust region spherical parameterization to edit the mesh, was proposed. These processes were considered, as an approximation tool for feature alignment between the source and the target models considered in the deformation process. To initialize the weights of wavelet neural network [10], many algorithms were developed. The main goal of these algorithms is to accelerate the convergence [11,12] and to adapt the wavelet network architecture [11] when the error back-propagation is applied to the training algorithms. To train the wavelet network, other techniques, such as Kalman filter [13], genetic algorithms [14–16] and immune algorithms [17], were developed. These methods stem mainly from the typical neural network. Some other studies used the excellent wavelets properties [18] in the frequency domain though they have accelerated convergence, avoided local minimum and overcome over fitting to some extent.

In [9] we, we proposed a 3D mesh deformation technique based on multi-mother wavelet neural network architecture relying on Gradient-based Optimization methods, to estimate the network parameters, optimize them and maximize the probability of selecting the best wavelets that approximates better the signal. The first step of applying the classic forming algorithms consists in using a predetermined network architecture that may be insufficient or too complicated. In addition, the resolution of wavelet neural network training problems is described by their perceived inability to escape local optima. In the current manuscript, we develop new learning algorithms using the improved version of Levenberg–Marquardt technique and genetic algorithm to solve the problem of best wavelet selection and that of determining the wavelets number to be selected to construct the network, So, we present a comparative study between our results obtained in [9] and those provided by applying our proposed deformation technique based on multi-mother wavelet neural network architecture using Genetic Algorithm and Levenberg–Marquardt method. The main objective of our work is to overcome the inability to escape of local optima when implemented on high dimensional meshes, and to estimate more advanced visual similarity measures for meshes with noise or large-scale deformations. The key component of our correspondence scheme is a deformation energy that penalizes geometric distortion, encourages structure preservation and simultaneously allows mesh topology changes.

2 Wavelet Networks

Wavelet networks combine some of the useful classification properties of neural networks with the localization and feature extraction properties of wavelets. They replace the global sigmoidal activation units of the classic feedforward wavelet network with wavelets, while preserving the networks universal approximation property. For function approximation and 3D modelling, both wavelet network and neural network are extensively used as non-parametric estimators. Generally, the wavelet network modeling consists of two steps: determining the network structure and adjusting the network parameters using a back-propagation algorithm. A new mapping network, called wavelet neural network or wavenets (WNN), is proposed by combining the wavelet transform theory with the basic concept of neural networks as an alternative to feedforward neural networks for approximating arbitrary nonlinear functions. The fundamental idea behind applying wavelets is to analyze the signal at different scales or resolutions, which is called multiresolution analysis [19,20]. Wavelets are a class of functions used to localize a given signal in both space and scaling domains. A family of wavelets can be constructed from a mother wavelet. Compared to Windowed Fourier analysis, a mother wavelet is stretched or compressed to change the size of the window. In this case, large wavelets give an approximate image of the signal, while smaller wavelets zoom in on small details. Therefore, wavelets can be automatically adapted to both high-frequency and low-frequency components of the signal having different window sizes. A wavelet network with one output y, N_i inputs $x1, x2, \ldots, x_{N_i}$ and N wavelets can be parameterized as follows:

$$\hat{y}(x) = \sum_{i=1}^{N_p} \omega_i \Psi_i(x) + \sum_{k=0}^{N_i} a_k x_k + b \tag{1}$$

where $\Psi_i(x) = \Psi\left(\frac{x-b_i}{a_i}\right)$

$x = [x_1, x_2, \ldots, x_{Ni}]^T$ is the input vector (x_k is the sampled input). The network parameters are $\omega_i \in \Re$, $d_i \in \Re^*$ and $t_i \in \Re$, correspond respectively to the wavelet coefficient, dilation parameter and translation parameter. Concerning the other parameters: $a_k \in \Re$, $b \in \Re$ are the linear coefficients and the biais term. The Wavelets Ψ_i are dilated and translated versions of a single function Ψ termed the "mother wavelet": $\Re^d \rightarrow \Re$

$$\Psi_i(x) = \Psi(d_i(x - t_i)) \tag{2}$$

with

$$\Psi(d_i(x - t_i)) = \frac{1}{\sqrt{d_i}} \Psi\left(\frac{x - t_i}{d_i}\right) \tag{3}$$

The role of the wavelet network is to construct a discretized wavelet family by adapting the network parameters to the displacement data. To obtain the best set of wavelets to be used in the network, these steps are followed. Firstly, an initial set of wavelets is defined on the basis of the available data. Secondly, considering the wavelets as regression candidates, a regressor selection algorithm is applied to identify the members of the wavelet set. Finally, to further enhance the quality of the used model [21], a training algorithm can be applied to adjust the network parameters.

The latter is estimated by an optimization algorithm. The wavelet network learning implements then iterative methods such as back-propagation.

3 Training Algorithms: Multi-Mother Wavelet Neural Network (MMWNN)

Wavelet neural networks which employ wavelets as basis function have various interesting properties including fast training and good generalization performance. Several approaches have been recently proposed for structure selection and wavelet neural networks training based on the limited band of wavelet networks, in which the input weights are determined by the sampling period or the frequency band of the target function (if available). This approach has shown global convergence and has allowed avoiding overfitting for non-noisy equi-spaced samples. In many practical situations, a finite number of samples of the target function are known and there is not a priori information about the frequency contents and frequency band of the target function. Without using information about the target signal and relying on the sampling period to find the input weights of neural network causes complex structure and a serious overfitting. To overcome this problem, a suitable model selection approach for complexity control and overfitting prevention should be used.

Once the initialization set W of wavelet basis functions has been constructed, the next step is to select the best M-size subset (M < L) of wavelet basis functions in W to estimate $f(x)$. However, in general, a search through all the M-size subsets is a computationally expensive procedure suffering from a combinatorial optimization problem [4].

One nonlinear optimization technique consists in using genetic algorithms which have been utilized successfully by Echauz et al. in [4], for radial wavelet networks, and Chen in [22] for RBF networks. In fact, genetic algorithms can provide optimal or near-optimal network topologies, at the expense of extensive computational requirements [22]. The proposed network structure is similar to that of the classic network, but it has some differences. Indeed, the classic network uses dilation and translation versions of only one mother wavelet and it is constructed by the new version by implementing several mother wavelets in the hidden layer [23,24]. The objective of the new architecture is to maximize the potentiality of wavelet selection [25] that approximates better the signal.

The new wavelet network structure with one output f, can be expressed by the following equation:

$$\hat{y}(x) = \sum_{i=1}^{N_1} W_i^1 \Psi_i^1(x) + \sum_{i=1}^{N_2} W_i^2 \Psi_i^2(x) + \cdots + \sum_{i=1}^{N_M} W_i^M \Psi_i^M(x) + \sum_{k=0}^{N_i} a_k x_k \tag{4}$$

we propose a new training procedure that permits to make:

- The initialization of weights, as well as translations and dilations of the wavelet network (generation of the Multi-mother).
- An automatic selection and increment of the wavelet in the hidden layer.
- The choice of the optimal wavelet numbers opt N_{opt}.
- The update and construction of the wavelet network. Since the new wavelet network structure based on several mother wavelets families is defined, we can ask the question of model construction, consisting of wavelet network for a given process. The parameters that determine the network construction are:
- The values that give the different network parameters: the structural parameters of wavelets, and the linear coefficient.
- The necessary wavelet numbers to reach a wanted performance.

The essential difficulty resides in the determination of the network parameters. Since these parameters take discreet values, conceive wavelets selection methods can be applied in a discreet wavelets library. The performance depends on the initial choice of the wavelets library as well as on a discriminating selection in this library. Thus, the problem of selecting the number of wavelets in the hidden layer is presented here. The criteria considered to calculate the parameters already proposed in the literature are often developed for a particular modeling framework under restrictive hypotheses and using important approximations. The improvement of the developed learning algorithms can provide good solutions to solve the problem of best wavelet selection and that of determining the wavelets number to be selected to construct the network. The multi wavelet families library is constructed by forming several mother wavelet families for the network construction (M wavelets). The library elements, generated by distributing the parameters on a dyadic grid, are grouped in levels according to the dilation and translation parameters. This architecture is to maximize the potentiality of wavelet selection that approximates better the signal. This choice allows not only enriching the library, but also getting a better performance for a given wavelet number, by using regressor selection techniques to choose the best wavelets is often shorter than the training of the dilations and translations. Therefore, the supplementary cost introduced by different versions can be acceptable.

In the current study our proposed architecture has a new structure, made up of multi-mother wavelet families, a hidden layer formed by N wavelets of M mother wavelets. The components of each mother wavelet families belong to a family of wavelets having different sizes.

To design our wavelet neural network architecture, an optimization algorithm should be applied to estimate and optimize the network parameters.

In this context, the genetic algorithm is considered as an optimization algorithm used to treat important research spaces, while the Levenberg–Marquardt algorithm is one of the most efficient training algorithms utilized in neural network modeling. Gradient-based Optimization Methods have been applied with the improved version of Levenberg–Marquardt technique. It is based on steepest descent principle which is relatively easy to implement. However, the error surface of wavelet network training usually contains planes with a gentle dope due to the squashing functions commonly used in wavelet networks. This gradient is too small for weights to move rapidly on these planes, which reduces the rate of convergence. The rate of convergence could also be very slow when the steepest descent method encounters “narrow valley” in the error surface where the direction of gradient is close to the perpendicular direction of the valley. The update direction oscillates back and forth along the local gradient. Since supervised learning of wavelet networks can be viewed as a function optimization problem, higher order optimization methods employing gradient information can be adopted in wavelet network training to enhance convergence rate. The genetic algorithm is a global search technique (i.e. stochastic metaheuristics) utilized to find approximate solutions to optimization and search problems. It is also an efficient, and effective techniques employed in both optimization and machine learning applications. In our experiments, we used the genetic algorithm to overcome the inability of the signal to escape local optima when implemented on high dimensional data set and to ensure the design of wavelet neural network meshes. This algorithm minimized the error between the original object and the target object and gave good results in several cases. Our main objective is to present a genetic algorithm that makes it possible to decrease or to increase the dimension of the multi-mother wavelet library and to evade local minima; thanks to the ability of genetic algorithm to examine the entire search space. This method may plainly initialize the wavelet network facilitating the employment of the gradient descent.

4 Our Training MMWNN Using Levenberg–Marquardt Algorithm for 3D Mesh Deformation

Lately, various,researchers applied a newly developed wavelet neural network, to function approximations [2,5,26] and, more recently, to nonlinear time series predictions [27–29]. All these studies indicated that wavelet network structures were better than either back-propagation neural network (BPNN) and radial basis function neural network (RBFNN) structures at approximating nonlinearities, generated better prediction and approximation results. In this study, we introduce a novel multi-mother wavelet neural network based on Levenberg method for 3D huge mesh deformation technique. The Levenberg–Marquardt algorithm is used to train the wavelet network structure. This training process can be considered as an unconstrained nonlinear optimization problem that can be founded by several optimization tools. In our study the Levenberg–Marquardt method is used to train the presented wavelet network prediction architectures. The Levenberg–Marquardt algorithm is founded on the standard Newton method and has been exposed to be a very powerful optimization technique [30,31]. In [30] Hagan et al. found that the Levenberg–Marquardt algorithm is more efficient than the popular conjugate gradient algorithm, and in various cases it can converge when the conjugate gradient algorithm cannot. Our preliminary experiment also shows that the Levenberg–Marquardt algorithm is the most efficient algorithm among those, such as gradient descent, gradient descent with momentum, and gradient descent with adaptive learning rate.

The Levenberg–Marquardt (LM) algorithm is an iterative technique that locates the minimum of a multivariate function that is expressed as the sum of squares of non-linear real-valued functions. It has become a standard technique for non-linear least-squares problems, widely adopted in a broad spectrum of disciplines. LM can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is far from the correct one, the algorithm behaves like a steepest descent method, slow, but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method.

The Levenberg–Marquardt Optimization algorithm is a virtual standard in nonlinear optimization that notably outperforms gradient descent and conjugate gradient methods for medium sized problems. It is a pseudo-second order algorithm which means that it works with only function evaluations and gradient information but it estimates the Hessian matrix using the sum of outer products of the gradients.

Gradient-based optimization algorithms were used with the improved version of Levenberg–Marquardt method.

It is founded on steepest descent principle which is comparatively easy to implement. The only variance is in the introduction of a parameter λ , called Levenberg–Marquardt parameter, enabling to stabilize the Newton method. This parameter λ is updated automatically, based on the convergence of every iteration. Stabilization is feasible through a reiterative process (if one iteration diverges, it starts again by increasing the parameter λ to obtain a convergent iteration). However, this process has a strong divergence when approaching the optimum inherent in Newton method. The divergence can be reduced. Succeeding Newton method, it is suggested to switch automatically to the conjugate gradient method when the phenomenon of divergence appears.

First, we must note that initialization step is so necessary: that if we have a good initialization, the local minimum problem can be avoided, it is sufficient to select the best regressions (the best based on the training data) from a finished set of regressors. The initialization of weights, as well as translations and dilations of the wavelet network (generation of the Multi-mother), Now, the wavelet network is considered with the linear coefficients a_k and the biais term. In order to more easily capture linear properties in regressions, the terms a_k and the biais are always first selected. One a_{jm} and b_{jm} are obtained from the initialization by a dyadic grid, they are used in computing a least squares solution.

Training the wavelet network models is to determine $\beta = \{w(0), w(m), u(j), a_{jm}, b_{jm}\}$ where $w(0)$ is the biais; $w(m)$ is the weights connecting input layer and output layer, $m = 1, 2, \ldots, In; u(j)$ is the weights connecting hidden layer and output layer, $j = 1, 2, \ldots, Hd$; a_{jm} is the dilation coefficient for the jth hidden neuron and the mth input neuron; b_{jm} is the translation coefficient for the jth hidden neuron and the mth input neuron such that the discrepancy between the observed data and the predictions is minimized. This training process may be regarded as an unconstrained nonlinear optimization problem that can be solved by various optimization tools.

For training our multi-mother wavelet network architecture, we want to minimize the training error represented by Eq. 5.

$$E(\beta) = \frac{1}{2}\sum_{i=1}^{M} e_i^2 = \frac{1}{2}\sum_{i=1}^{M}(y(i) - \hat{y}(i))^2 \tag{5}$$

where $\hat{y}(i)$, $y(i)$ is the predicted and observed outputs respectively for the ith input, M is the number of training inputs; and $E(\beta)$ is the training error to be minimized. The variables to be optimized are the parameters in vector β. The training error $E(\beta)$ is minimized by refining these parameters. The Levenberg–Marquardt algorithm uses Eq. 6 to search for the best parameters [30,31].

$$\beta_{k+1} = \beta_k - (J^T J + \lambda_k I)^{-1} J^T e \tag{6}$$

where I is the identity matrix that has the same size as $J^T J$, λ_k is the constant parameter, $e = [e_1 e_2 e_3, \ldots, e_M]^T$ is the error vector and the Jacobian matrix is:

$$J = \begin{bmatrix} \frac{\partial_{e_1}}{\partial_{w(0)}} & \frac{\partial_{e_1}}{\partial_{w(m)}} & \frac{\partial_{e_1}}{\partial_{u(j)}} & \frac{\partial_{e_1}}{\partial_{a_{jm}}} & \frac{\partial_{e_1}}{\partial_{b_{jm}}} \\ \frac{\partial_{e_2}}{\partial_{w(0)}} & \frac{\partial_{e_2}}{\partial_{w(m)}} & \frac{\partial_{e_2}}{\partial_{u(j)}} & \frac{\partial_{e_2}}{\partial_{a_{jm}}} & \frac{\partial_{e_2}}{\partial_{b_{jm}}} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial_{e_M}}{\partial_{w(0)}} & \frac{\partial_{e_M}}{\partial_{w(m)}} & \frac{\partial_{e_M}}{\partial_{u(j)}} & \frac{\partial_{e_M}}{\partial_{a_{jm}}} & \frac{\partial_{e_M}}{\partial_{b_{jm}}} \end{bmatrix} \tag{7}$$

We require to calculate the Jacobian matrix and define the parameter λ_k in Eq. 6. Based on the proposed wavelet network structure, using mother wavelet, the expression for every element in the Jacobian matrix are:

$$\frac{\partial_{e_i}}{\partial_{w(0)}} = -1,$$
$$\frac{\partial_{e_i}}{\partial_{w(m)}} = -x(m),$$
$$\frac{\partial_{e_i}}{\partial_{u(j)}} = -\Psi_j(X(i)),$$

$$
\begin{aligned}
&\frac{\partial_{e_i}}{\partial_{b_{jm}}} = -u(j)\Psi_j(X(i))\left(\frac{1}{Z_{jm}} - Z_{jm}\right)\left(\frac{-1}{a_{jm}}\right),\\
&\frac{\partial_{e_i}}{\partial_{a_{jm}}} = \frac{\partial_{e_i}}{\partial_{b_{jm}}} Z_{jm},\\
&Z_{jm} = \frac{x_m(i) - b_{jm}}{a_{jm}}, and\\
&\quad i = 1, 2, \ldots, M, \quad j = 1, 2, \ldots, Hd, \quad m = 1, 2, \ldots, In
\end{aligned}
\tag{8}
$$

By the defined Jacobian matrix, our multi-mother wavelet network prediction structure can be trained for 3D mesh deformation following the process exposed in Fig. 1. In this figure, Mu, Mu_{Inc}, and Mu_{max} are parameters used to control the training process. The training process of our multi mother wavelet neural networks can appropriate the underlying nonlinear trend in the training data set, but it can also learn the noise contained in the data and lead to overfitting. This training is based on the formation of a training set consisting of pairs inputs/outputs (x_k, y_k). The training algorithm aims to reduce the error of the wavelet network on the training set examples, adjusting the network parameters. It then implements iterative methods such as back-propagation. Besides the choice of selection method, we must choose an optimization algorithm to optimize the parameters. From the Levenberg method, originally scheduled for solving linear systems by least squares. Marquart modifies the algorithm to enable solving nonlinear systems of statistical problems.The method combines two algorithms into one. The simple gradient method (the steepest descent method) which is merely a less accurate estimate but has a slow convergence and the Gauss-Newton which has a quadratic convergence but requires an initial vector near to the solution. Thus, leading the gradient method to a local optimum makes the Gauss-Newton ineffective since the Jacobian becomes singular. One frequently used way of preventing overfitting and improving generalization is to present a validation error term, and the best network parameters in vector β are selected founded on validation errors rather of training errors. With a good initialization of the network parameters and with the proposed optimization scheme using LM algorithm to adjusting the network parameters the efficiency of training increases to have a good 3D mesh deformation process. As we previously noted, classical approaches begin often with predetermined wavelet networks. Consequently, the network is often insufficient. After that, new works are used to construct a several mother wavelets families library for the network construction every wavelet has different dilations following different inputs. This choice has the advantages of enriching the library, and offering a better performance for a given wavelets number. Supervised learning of wavelet networks can be viewed as a function optimization problem, higher order optimization methods using gradient information can be adopted in wavelet network training to improve the rate of convergence.

5 Our Training MMWNN Using Genetic Algorithm for 3D Mesh Deformation

In our work we presents the implementation of the genetic algorithm in the multi-mother wavelet network training which aims at searching for an optimal or near optimal solution to the 3D objects deformation problem. The idea is to integrate genetic algorithms into the wavelet network to avoid both insufficiency and local minima in the 3D mesh deformation technique. Our wavenet learning algorithms consist of two processes: the self-construction of networks and the minimization error.

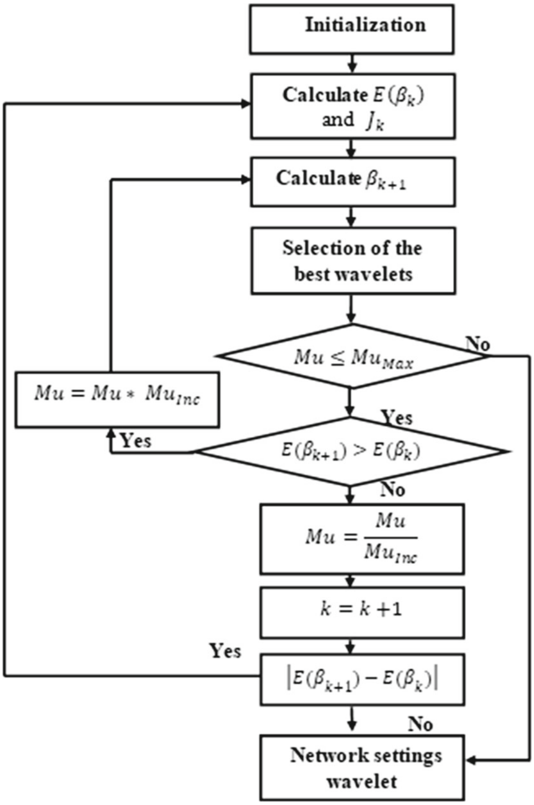

Initialization
Calculate $E(\beta_k)$ and J_k
Calculate β_{k+1}
Selection of the best wavelets
$Mu \leq Mu_{Max}$
No
Yes
$Mu = Mu * Mu_{Inc}$
Yes
$E(\beta_{k+1}) > E(\beta_k)$
No
$Mu = \frac{Mu}{Mu_{Inc}}$
$k = k + 1$
Yes
$|E(\beta_{k+1}) - E(\beta_k)|$
No
Network settings wavelet

Fig. 1 Multi-mother wavelet neural network training based Levenberg–Marquardt method

5.1 Network Initialization Parameter

Once t_i and d_i are obtained from the initialization by a dyadic grid, they are used in computing a least square solution for w, a, b. The variable Ni represents the number of displacement pairs data. Using families of wavelets, we have a library that contains Nl wavelets. To every wavelet Ψ_{ji} we associate a vector whose components are the values of this wavelet according to the examples of the training sequence. Now, the wavelet network is considered with the linear coefficients a_k and the biais term b. In order to more easily capture linear properties in regressions, the terms a_k and b are always first selected. One t_i and d_i are obtained from the initialization by a dyadic grid, they are used in computing a least squares solution for w, a, b.

$$\begin{bmatrix} \Psi_1^1(X_1) & \cdots & \Psi_N^M(X_1) & x_1 & 1 \\ . & \cdots & . & . & . \\ . & \cdots & . & . & . \\ . & \cdots & . & . & . \\ . & \cdots & . & . & . \\ \Psi_1^1(X_{N_i}) & \cdots & \Psi_N^M(X_{N_i}) & x_{N_i} & 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ . \\ . \\ w_L \\ a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ . \\ . \\ . \\ . \\ y_{N_i} \end{bmatrix} \tag{9}$$

where $y_1, \ldots, y_{Ni}$ are the sampled outputs in Z and $L = M * N$.

5.2 Selection of the Best Wavelets

A wavelet library having several wavelet families is more voluminous than the one that possesses the same wavelet mother. It implies a more elevated calculation cost during the selection process. Because the model size M is usually excessively large, the subset model selection is required. The optimal subset selection techniques are computationally prohibitive and impractical. The practical method is the forward selection, and the OLS procedure (Orthogonal Least Squares) which are an efficient implementation of this subset selection procedure. For the best selection, we propose an improved version of OLS procedure for the subset model selection. Nevertheless, using regressor selection techniques to choose the best mother wavelets is often shorter than the training of the dilations and translations; the supplementary cost introduced by different versions can be therefore acceptable. The library being constructed, a selection method is applied in order to determine the most meaningful wavelet for modelling and deforming 3D mesh. Generally, the wavelets in W are not all meaningful to estimate the signal. Lets suppose that we want to construct a wavelets network with M wavelets, the problem is to select M wavelets from the wavelet library W. The proposed selection is based on Orthogonal Least Square (OLS) [32,33]. A selection method is applied in order to determine the most meaningful wavelet to model the considered signal. Generally, the regressors in V_{MW} are not all meaningful to estimate the signal. Let's suppose that we want to construct a wavelet network $f(x)$ with N_{MW} wavelets, the problem is how to select N_{MW} wavelets from V_{MW}. To the first iteration, the signal is defined as $Y = [y_1, \ldots, y_{Ni}]$ and the regressors vectors are the $V_{MW}(t, d)$ defined by Eq. 10. The selected regressor is the one for which the absolute value of the cosine with the signal Y is maximal. We constitute a matrix that is constituted of V_{MW} of blocks of the vectors representing the wavelets of every mother wavelet where the expression is:

$$V_{MW} = \{V_i^j\}_{i=[1,\ldots,N],\, j=[1,\ldots,M[} \tag{10}$$

The V_{MW} matrix is defined as follows:

$$V_{MW} = \begin{vmatrix} V_1^1(x_1) & \cdots & V_N^1(x_1) & \cdots & V_1^M(x_1) & \cdots & V_N^M(x_1) \\ \vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\ \vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\ \vdots & \ldots & \vdots & \ldots & \vdots & \ldots & \vdots \\ V_1^1(x_{Ni}) & \cdots & V_N^1(x_{Ni}) & \cdots & V_1^M(x_{Ni}) & \cdots & V_N^M(x_{Ni}) \end{vmatrix} \tag{11}$$

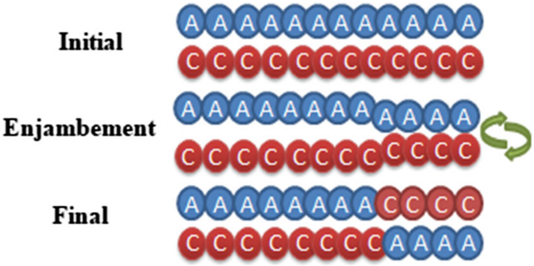
5.3 Change of the Library Dimension

5.3.1 Crossover Operators

This algorithm used two crossover operators: One of them changes the number of columns of chromosome so it changes the number of mother wavelets and introduce in the library a new version of wavelets issued of the new mother wavelet. The second operator does not change the number of columns of each chromosome.

The Crossover1 Operator: After the selection of the two chromosomes to which we will apply this operator, we choose an arbitrary position a in the first chromosome and a position b in the second according to a. After that, we exchange the second parts of the two chromosomes (Fig. 2).

Fig. 2 Crossover1 operator



The Crossover2 Operator: For the second operator, we choose an arbitrary position c in the first chromosome and a position d in the second chromosome according to c. Let $Min_{point} = Min(c, d)$. First, we change the values of c and d to Min_{point}. Then, we exchange the second parts of the two chromosomes. In this case, we have necessarily first children having the same length as the second chromosome and the second children having the same length as the first chromosome.

5.3.2 Mutation Operator

Generally, the initial population does not have all the information that is essential to the solution. The goal of applying the mutation operator is to inject some new information (wavelets) into the population. Mutation consists in changing one or more gene(s) in chromosome chosen randomly according to a mutation probability pm. Nevertheless, the muted gene may be the optimal one therefore, the new gene will not replace the old but it will be added to this chromosome.

5.4 Change of Settings Wavelets

In this step, we have a uniform crossover operator and a mutation operator applied to structural parameters of wavelet network (translations and dilatations).

5.4.1 Uniform Crossover Operator

Let $T = (t_1, t_2, \ldots, t_N)$ the vector representing the translations: A coefficient is chosen and a vector $T = (t'_1, t'_2, \ldots, t'_N)$ is constructed as follows:

$$t_1 = \alpha.t_1 + (1 - \alpha).t_2 \tag{12}$$

$$t_2 = \alpha.t_2 + (1 - \alpha).t_1 \tag{13}$$

where α is a real random value chosen in [-1 1]. The same operator is applied to the vector dilation D.

5.4.2 Mutation Operator

After crossing operation, the string is subject to mutation. We consider the optimal wavelet, we reset the network with these wavelets that will replace the old in the library and the optimization algorithm will be continued using new wavelets.

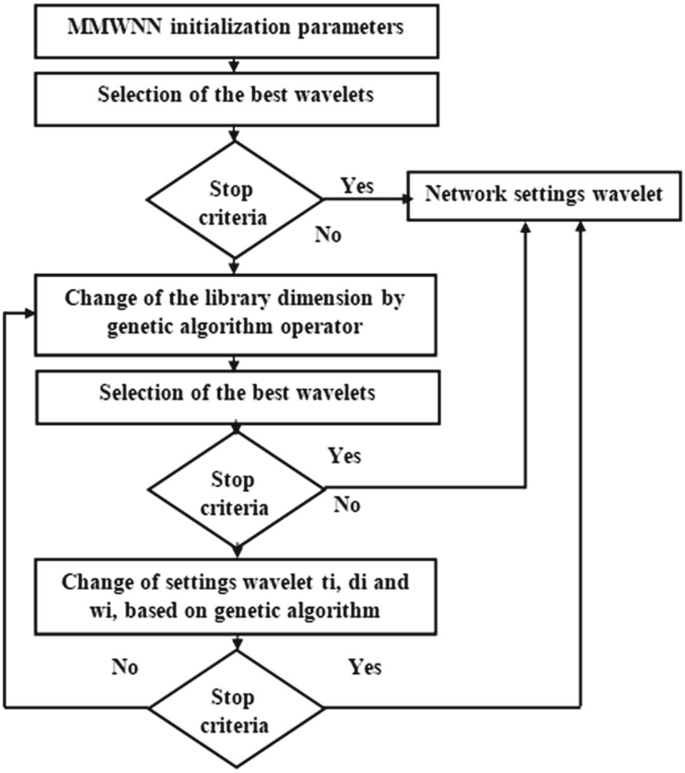

MMWNN initialization parameters
Selection of the best wavelets
Stop criteria
Yes
No
Network settings wavelet
Change of the library dimension by genetic algorithm operator
Selection of the best wavelets
Stop criteria
Yes
No
Change of settings wavelet ti, di and wi, based on genetic algorithm
No
Yes
Stop criteria

Fig. 3 Multi-mother wavelet neural network training based genetic algorithm

Finally, after N iterations, we construct a wavelets network for the approximation signal Y.

A genetic algorithm that allows you to both reduce or increase the size of the mother wavelet library and then escape local minima.

This approach therefore makes it possible to properly initialize the wavelet network, which then facilitates the task for the gradient.

Our proposed algorithm is resumed in Fig. 3.

6 Results and Simulation

To ensure the design of our multi-mother wavelet neural network architecture, we used an optimization algorithm and performed a comparative study between genetic algorithm and Levenberg–Marquardt algorithm. We employed a multi-mother library wavelet network based on the Levenberg–Marquardt algorithm and on genetic algorithm for 3D mesh approximation and deformation.

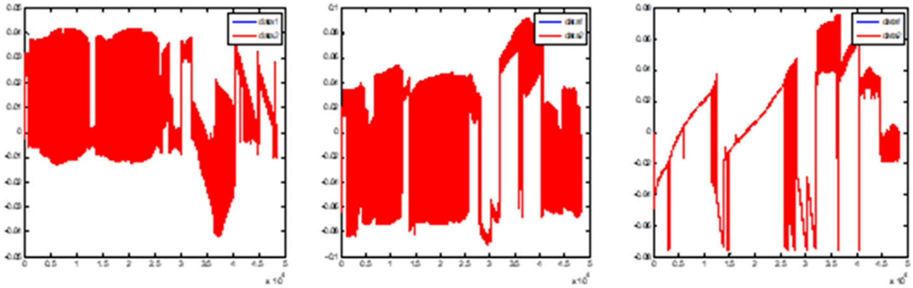


Fig. 4 Approximations of X, Y and Z vertex of bunny object

6.1 3D Mesh Modeling Results

We further demonstrated and evaluate our approach in the 3D mesh modeling context. The complexity of 3D modeling object is directly related to the number of wavelets selected and the iterations to build the network.

To assess the performance of the two approaches, we utilized the mean square error (MSE) and the normalized square root of the mean square error (NSRMSE) for 3D mesh modelling which is defined as follows:

$$MSE = \frac{1}{N_i} \sum_{k=1}^{N_i} (O_N(x_{NK}, y_{NK}, z_{NK}) - O(x_K, y_K, z_K))^2 \tag{14}$$

$$NSRMSE = \sqrt{\frac{\sum_{k=1}^{N_i} (O_N(x_{NK}, y_{NK}, z_{NK}) - O(x_K, y_K, z_K))^2}{\sum_{k=1}^{N_i} (\bar{O} - O(x_K, y_K, z_K))^2}} \tag{15}$$

O is the approximate object, K is the number of observation and $\bar{O} = \sum_{k=1}^{N_i} O(x_K, y_K, z_K)$ is the results of the 3D representation, from the multi mother wavelet library. The 3D wavelet can be constructed as the separable 1D wavelet product by applying a 1D wavelet analysis in three directions of space:

$$\Psi_i^j(x, y, z) = \Psi_i^j(x) * \Psi_i^j(y) * \Psi_i^j(z) \tag{16}$$

The following figures represent the approximations of vertex X, Y and Z of bunny object by our MMWNN (Fig. 4):

We compare, in this section, some experimental results of obtained by applying the genetic algorithm MMWNN-GA with those provided by the Levenberg–Marquardt algorithm MMWNN-LM to ensure the design of wavelet neural network for 3D mesh approximation and modeling.

Tables 1 and 2 give the mean square error and the final normalized square root of the mean square error respectively after 20 training iterations using MMWNN, MMWNN-LM and MMWNN-GA constructed with 15 wavelets in hidden layer and based on several wavelets (Beta, Polywog1, Mexican Hat, Slog1 and Polywog1). The optimal wavelet number was fixed by our MMWNN algorithm. In fact, MMWNN structure is as an interesting alternative to classic wavelet network. It is effectively used to construct the network by several mother wavelets and solve the problem of choosing the best wavelet mother that can better approximate the signal.

Table 1 MSE for different activation wavelet functions using MMWNN, MMWNN and MMWNN-GA

Objects	MMWNN	MMWNN-LM	MMWNN-GA
Horse	4.74990e–2	1.95420e–4	1.47022e–4
Elephant	2.91680e–1	6.25610e–3	4.32540e–3
Flamingo	8.95610e–3	4.08580e–5	5.36550e–6
Face	9.65520e–2	4.68210e–4	4.54022e–4

Table 2 NSRMSE for different activation wavelet functions using MMWNN, MMWNN and MMWNN-GA

Objects	MMWNN	MMWNN-LM	MMWNN-GA
Horse	5.64909e–1	6.449e–2	1.72912e–2
Elephant	8.95486e–2	4.518e–3	2.44860e–3
Flamingo	4.25420e–2	1.554e–2	3.58423e–1
Face	6.44980e–3	4.84522e–4	2.71410e–4

For the MMWNN-GA solution, we applied the genetic algorithm to overcome the inability of the signal to escape local optima when implemented on high dimensional data set and to ensure the design of wavelet neural network meshes. For MMWNN-LM, the Levenberg–Marquardt algorithm is used to train the wavelet network models because it is more efficient than the other algorithms based on gradient descent and it is one of the most efficient training algorithms for neural network modeling. Having the updating rule, the Jacobian matrix needs to be computed and a training process should be designed. A good method used for weights adaptation is the Levenberg–Marquardt algorithm. It is a combination of the gradient descendent rule and the Gauss-Newton method.

We can see, in tables 1 and 2, that MMWNN-GA are more suitable for 3D approximation, compared to the Levenberg method and MMWNN structure. For example, to approximate Horse object using MMWNN-GA after 20 iterations over 1.95420e – 4, the Levenberg method gave an MSE equal to 1.47022e – 4; whereas MMWNN structure provided 4.74990e – 2. The genetic algorithm has a sound theoretical basis and guarantee convergence for most of the smooth functions, which may reduces divergence.

we applied genetic algorithm to improve the robustness of gradient-descent algorithms. We also presented a genetic algorithm for the design of wavelet network in [34,35]. The problem was to find the optimal network structure and parameters. In order to determine the optimal network, the developed algorithm modified the number of wavelets in the library. In MMWNN-LM structure, Gradient-based optimization methods were applied with the improved version of Levenberg–Marquardt technique, this gradient is too small for weights to move rapidly on these plans, which minimizes the rate of convergence. The rate of convergence can also be very slow when the steepest descent method encounters “narrow valley” in the error surface where the direction of gradient is close to the perpendicular direction of the valley. The good performance of the algorithm is achieved by evolving the initial population and by using operators that alter the structure of the wavelets library. Comparing our algorithm with the classical ones, our results show significant improvement either with MMWNN-GA or with MMWNN-LM in the resulting performance and topology.

Figure 5 represents the evolution of the MSE when the number of wavelets in the hidden layer is increased from 15 to 100 wavelets using MMWNN, MMWNN-LM and MMWNN-GA.

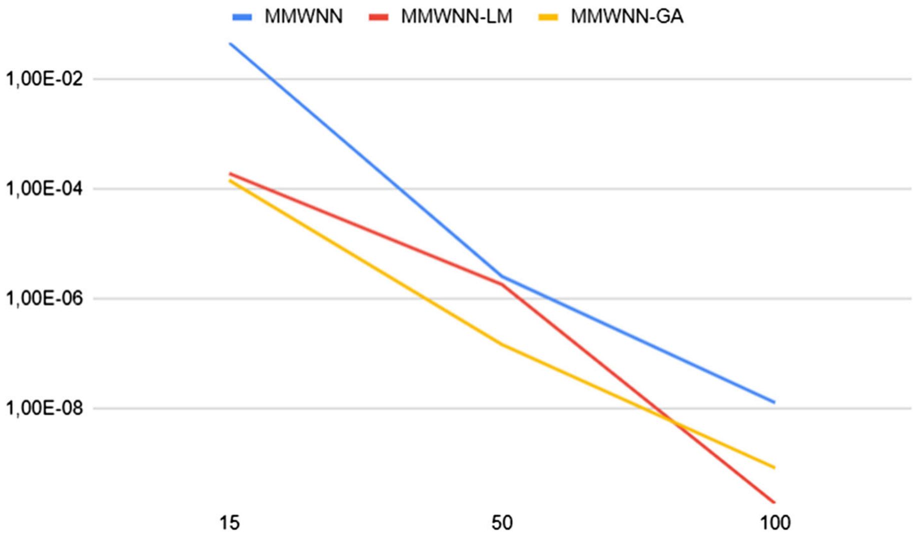


Fig. 5 Evolution of the MSE according to the number of wavelets by MMWNN, MMWNN-LM and MMWNN-GA

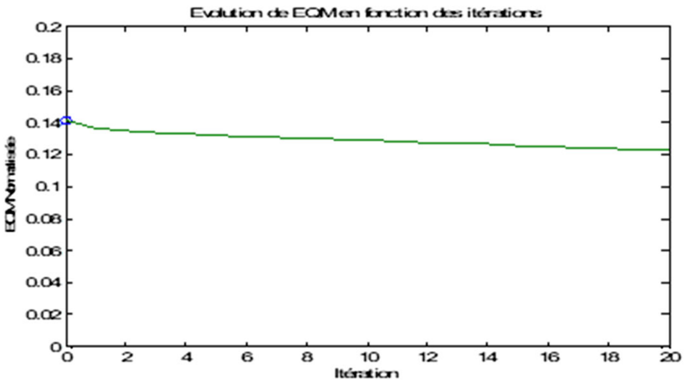


Fig. 6 Evolution of the MSE according to the number of iterations

From the simulation results we see that increasing the wavelet number in the hidden layer increases not only the approximation capacity, but also time cost and algorithm complexity increase considerably. The procedure allows to specify the selected wavelets for every mother wavelet. Therefore, when we increase the number of wavelets the MSE given by the MMWNN-GA network are weaker than those obtained by the MMWNN and very close to the MSE by MMWNN-LM .

In addition to that, to ameliorate these criteria :MSE and NSRMSE, we can increase the iterations number in the training stage.

Figure 6 gives the evolution of the MSE according to the number of iterations:

6.2 3D Mesh Deformation Results

For 3D mesh deformation the interpolation between objects can be realised by determining the trajectory of the corresponding vertices on the representation obtained in the previous step. The simplest way to interpolate between these points is a linear interpolation.

The interpolation in the wavelet domain makes it possible to control interpolation starting time and speed at various resolutions. In fact, multi-mother wavelet neural network have achieved great success in different areas of computer science. Compared with 2D images, 3D shapes are more difficult to process, due to their irregular connectivity and limited data availability.

The 3D deformation sequence consists of a large number of frames; each of which represents a large static 3D object.

A deformed mesh with fixed connectivity can be represented by a $3F \times N$ matrix A, where F is the number of frames and N the number of vertices:

$$A = \begin{pmatrix} v_1^{(1)} & \cdots & v_N^{(1)} \\ \vdots & \ddots & \vdots \\ v_1^{(F)} & \cdots & v_N^{(F)} \end{pmatrix} \tag{17}$$

The approximation error is typically reported in terms of the following error metric:

$$100\%.\frac{||A - A_{approx}||_F}{||A - A_{timeAverage}||_F} \tag{18}$$

where $A_{timeAverage} \in \Re^{3F \times N}$ is a matrix indicating a static mesh (average of all keyframes). Unfortunately, this metric is sensitive to global motion applied to the entire mesh.

For example, adding linear motion increases the denominator $||A - A_{timeAverage}||_F$ but leaves the numerator

$100\%.||A - A_{approx}||_F$ unchanged because skinning can trivially reproduce translation (by simply adding the translation to all bone transformations). For example, an on-spot walking sequence will have higher error than exactly the same animation where the character moves forward.

Therefore, we propose to simply use $\sqrt{3NF}$ as the normalizing factor, obtaining:

$$E_{RMS} = 100\%.\frac{||A - A_{approx}||_F}{\sqrt{3NF}} \tag{19}$$

A deformed mesh with fixed connectivity can be represented by a 3F × N matrix A, where F is the number of frames and N the number of vertices. Since $3NF$ is the total number of elements of matrix A, E_{RMS} is simply the average error per element (scaled by 1000 for convenience). To be able to compare E_{RMS} between objects with different proportions, we uniformly scale every animation so that its first frame is tightly enclosed by a unit ball.

To evaluate the effectiveness of our proposed method, we use the E_{RMS} generalization ability error (root mean square) to ensure the comparison on the set of tests of our method with several state-of-the-art methods in term of 3D mesh deformation process, including sparse localized deformation component (SPLOCS) with edge lengths and dihedral angles [36], SPLOCS with the feature from [37], and Mesh-Based Autoencoders for Localized Deformation Component Analysis [38]. The Variation of E_{RMS} in terms of 3D mesh deformation is presented in Table 3.

Obviously, the existing methods cannot cope with such dataset with large deformations and they suffer from artifacts. As a solution, our deformation algorithm presents a good

Table 3 E_{RMS} using MMWNN-LM and MMWNN-GA

Objects	[36]	[37]	[38]	MMWNN-LM	MMWNN-GA
Horse	29.609	20.199	12.960	10.032	6.441
Elephant	50.894	31.024	24.741	23.920	22.027
Flamingo	38.217	14.974	12.885	12.775	12.457
Face	8.562	2.985	2.908	2.845	2.704

Table 4 Comparative study of the execution time between our approach with MMWNN-GA and MMWNN-LM architectures

Objects	MMWNN-GA	MMWNN-LM
Horse	5412.001(s)	1025.104(s)
Elephant	2875.004(s)	410.552(s)
Flamingo	1055.825(s)	574.255(s)
Face	7856.524(s)	1035.112(s)

reconstructed object and has better quantitative reconstruction results than the deformation techniques presented in the literature, with lower reconstruction errors when sufficient components are used. It is clear that the obtained error rate was remarkably reduced and the performance of Levenberg–Marquardt and genetic algorithm on the synthesized deformed mesh allowed minimizing the distortion without losing the initial data, reaching the control mesh. We can see also that the error was minimized when we used the wavelet network by the genetic algorithm MMWNN-GA and MMWNN-LM, compared to state-of-the-art methods.

Besides, it is obvious that the error rate decreased by applying the wavelet network and the genetic algorithm MMWNN-GA, in comparison with MMWNN-LM. However, it is almost the same when the two methods were applied, except for the horse object for which we obtained an E_{RMS} of 10.0321, using MMWNN-LM, against an E_{RMS} equal to 6.4412 using MMWNN-GA. We may conclude that the horse object contains considerable deformations and suffers from artifacts with large scale rotations. To solve these limitations, the genetic algorithm is a good alternative solution that may treat better this type of objects, compared to Levenberg–Marquardt method.

Generally, our deformation algorithm based on MMWNN-GA and MMWNN-LM provides a good reconstructed object and allows obtaining better quantitative reconstruction results than the existing deformation techniques, with lower reconstruction errors when sufficient components are used.

Table 4, reveals that MMWNN-LM clearly minimizes the time necessary to ensure the deformation process, in comparison with the MMWNN-GA architecture. For example, for Elephant object, the simulation time is reduced from 2875.004(s), using MMWNN-GA architecture, to 410.552(s), employing MMWNN-LM. In our approach, the 3D deformation sequence consists of a large number of frames; each of which represents a large static 3D mesh, characterized by a search space with an increasing number of dimensions. The wavelet network permitted the characteristic points of the original mesh to be aligned towards the target mesh.

The genetic algorithm is an optimization algorithm applied to treat important research spaces. However, it requires more calculations than the other meta-heuristic algorithms (notably the evaluation function).

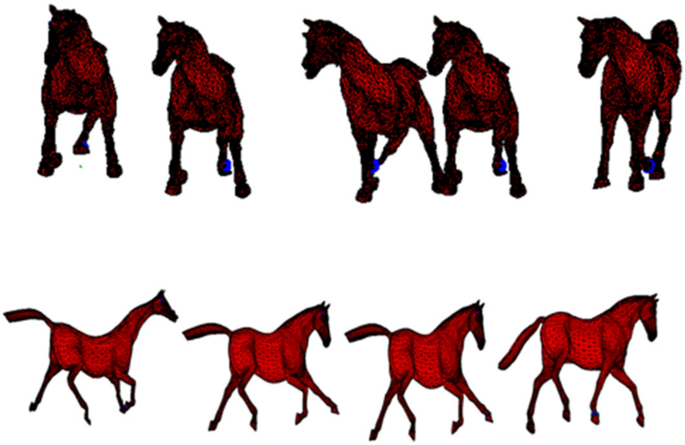


Fig. 7 Comparison of deformation components located in similar areas for Horse object by MMWNN-GA (first row) and MMWNN-LM (second row)

On the other hand, the Levenberg–Marquardt algorithm is one of the most efficient training algorithms for neural network modeling, and therefore for training a feed-forward neural network with the Levenberg–Marquardt algorithm. Having the updating rule, the Jacobian matrix needs to be computed and a training process should be designed. This algorithm combines the ability of both methods (i.e., convergence from any initial state as in the case of gradient descent, and rapid convergence near the vicinity of the minimum error as in the case of Gauss-Newton method) while avoiding their drawbacks.

We used the genetic algorithm to ensure the deformation of the 3D meshes and we got good results. However, for 3D huge meshes genetic algorithm requires less information about the problem and designing an objective function and getting the representation and operators right can be difficult and computationally expensive. In this case, we talk about time-consuming modeling. The Levenberg–Marquardt method, that can handle noisy objective function values as well as random models, provided sufficient accuracy. As the probability of accurate function estimates and models is sufficiently large, we assume that the proposed algorithm converges globally to a first-order stationary point of the objective function with probability one. For this reason, local optimization techniques, such as the Levenberg method, are used mainly for their computational and processing time. However, due to the slowness of the execution time, in many cases, the genetic algorithm diverged and did not give accepted reconstructions when the search space of the best wavelet was enlarged. Thus, gradient-based optimization methods were applied with the improved version of Levenberg–Marquardt technique.

Figures 7, 8, 9 and 10 represent our training results obtained using 15 wavelets in the hidden layer for the 3D deformation of the four objects (Scape, Horse, Elephant and Flamingo). These findings were provided by the MMWNN-GA and MMWNN-LM.

7 Conclusion

In our approach, the 3D deformation sequence consists of a large number of frames; each of which represents a large static 3D mesh. We used a wavelet network to align the char-

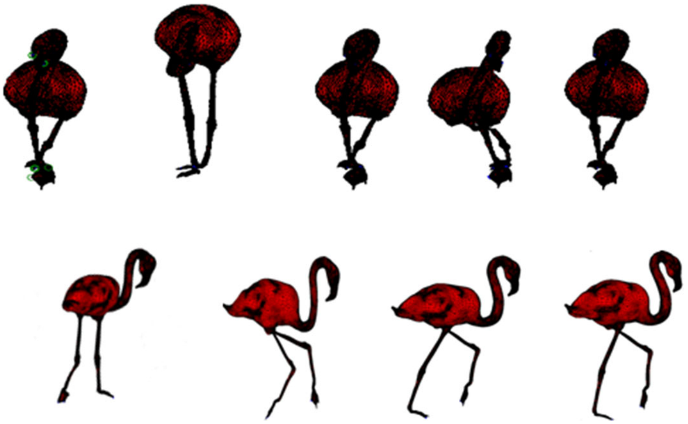


Fig. 8 Comparison of deformation components located in similar areas for Flamingo object by MMWNN-GA (first row) and MMWNN-LM (second row)

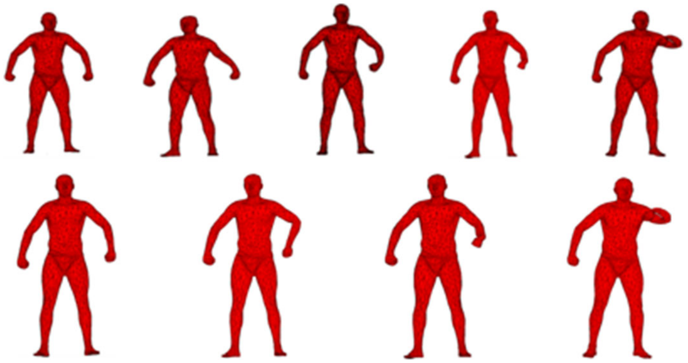


Fig. 9 Comparison of deformation components located in similar areas for Scape object by MMWNN-GA (first row) and MMWNN-LM (second row)

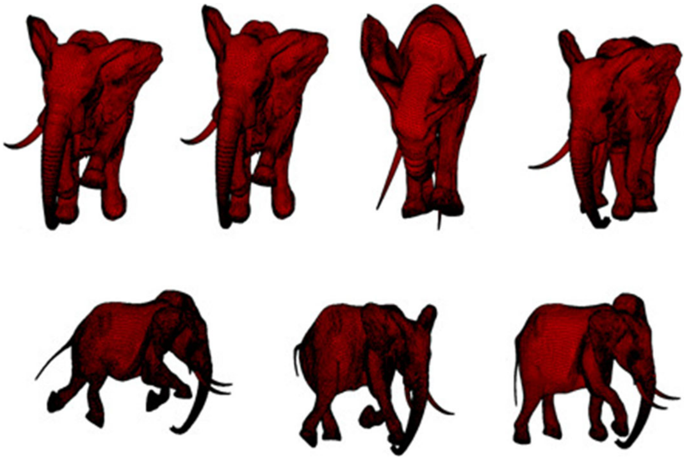


Fig. 10 Comparison of deformation components located in similar areas for Elephant object by MMWNN-GA (first row) and MMWNN-LM (second row)

acteristic points of the original mesh and to aligned towards the target mesh. To ensure the design of wavelet neural network architecture, an optimization algorithm was used to estimate and optimize the network parameters. The applied genetic algorithm, which is an optimization algorithm, allowed treating important research spaces, and the Levenberg Marquardt algorithm showed its efficiency in neural network modeling. A comparative study between genetic algorithm and Levenberg–Marquardt algorithm was also presented in this article. Experimental results reveal that MMWNN-GA and MMWNN-LM provided a good reconstructed object and better quantitative reconstruction results, compared to the existing deformation techniques, with lower reconstruction errors when sufficient components were used. For object containing considerable deformations and suffering from artifacts with large scale rotations, the genetic algorithm treated better this type of objects, in comparison with Levenberg–Marquardt method. On the other hand, genetic algorithm requires more calculations than Levenberg–Marquardt algorithm and it is computationally expensive, i.e. it is time-consuming modelling algorithm. So, MMWNN-LM minimized more remarkably the time necessary to ensure the deformation, in comparison with the MMWNN-GA architecture.

In our future work, using MMWNN-GA structure, we will apply a simplification method to reduce the simulation time of deformation for all high-resolution objects.

References

1. Daubechies I (1992) Ten lectures on wavelets. In: CBMS-NSF regional conference series in applied mathematics, pp 1–350
2. Zhang Q, Benveniste A (1992) Wavelet Networks. IEEE Trans Neural Netw 3:889–898
3. Bouchrika T, Zaied M, Jemai O, Ben Amar C (2012) Ordering computers by hand gestures recognition based on wavelet networks. In: 2nd international conference on communications computing and control applications, CCCA12, pp 1–6
4. Echauz J, Vachtsevanos G (1995) Wavelet neural networks for EEG modeling and classification. Math J
5. Zhang J, Walter GG, Miao Y, Wayne Lee W (1995) Wavelet neural networks for function learning. IEEE Trans. Signal Process 43:1485–1496
6. Lin CJ (2009) Nonlinear systems control using self-constructing wavelet networks. Appl Soft Comput 9:71–79
7. Karimi H, Lohmann B, Moshiri B, Jabedar-Maralani P (2006) Wavelet-based identification and control design for a class of nonlinear systems. Int J Wavel Multiresol Inform Process 4:213–226
8. Billings SA, Wei HL (2005) A new class of wavelet networks for nonlinear system identiffication. IEEE Trans Neural Netw 16:862–874
9. Dhibi N, Elkefi A, Bellil W, Ben Amar C (2016) 3D high resolution mesh deformation based on multi library wavelet neural network architecture. 3D Res J 7:408–416
10. Oussar Y, Dreyfus G (2002) Initialization by selection for wavelet network training. Neurocomputin 34:131–143
11. Bodyanskiy Y, Lamonova N, Pliss I, Vynokurova O (2005) An adaptive learning algorithm for a wavelet neural network. Exp Syst 22:235–240
12. Jinhua X, Ho DWC (2005) A constructive algorithm for wavelet neural networks. In: International conference on natural computation. Changsha, China, pp 730–739
13. Qingmei S, Yongchao G (2003) A stepwise updating algorithm for multiresolution wavelet neural networks. In: Proceedings of the international conference on wavelet analysis and its applications (WAA). Chongqing, China, pp 633–638
14. Kai C, Yoshio H (2004) An adaptive hybrid wavelet neural network and its application. In: IEEE international conference on robotics and biomimetics, pp 73–86
15. Seong-Joo K, Yong-Taek K, Jae-Yong S, Hong-Tae J (2002) Design of the scaling-wavelet neural network using genetic algorithm. In: Proceedings of the international joint conference on neural networks. Honolulu, pp 2174–2179
16. ChangGyoon L, Kangchui K, Eungkon K (2004) Modeling for an adaptive wavelet network parameter learning using genetic algorithms. In: International conference on modeling and simulation, CA, USA, pp 55-5-9

17. Min H, Baotong C (2005) A novel learning algorithm for wavelet neural networks. In: International conference on natural computation. ICNC, Changsha, China, pp 1–7
18. Mehra M, Rathish Kumar BV (2007) Error estimates for time accurate wavelet based schemes for hyperbolic partial differential equations. Int J Wavel Multiresol Inform Process 5:657–678
19. Mallat SG (1989) A theory for multi-resolution signal decomposition: the wavelet representation. IEEE Trans Patt Anal Mach Int 11:674–693
20. Valette S, Prost R (2004) Wavelet-based multiresolution analysis of irregular surface meshes. IEEE Trans Vis Comput Graph 10:113–122
21. Wang YS, Lee CR (2009) Evaluation of nonstationary vehicle passing loudness based on an antinoise wavelet pre-processing neural network mode. Int J Wavel Multiresol Inform Process 7:459–480
22. Chen S, Wu Y, Luk BL (1999) Combined genetic algorithm optimization and regularized orthonormal least squares learning for radial basis function networks. IEEE Trans Neural Netw 10:1239–1243
23. Othmani M, Bellil W, Ben Amar C, Alimi AM (2012) A novel approach for high dimension 3D object representation using multi-mother wavelet network. Int J Multimed Tools Appl 59:7–24
24. Othmani M, Bellil W, Ben Amar C, Alimi AM (2010) A new structure and training procedure for multi-mother wavelet networks. Int J Wavelets Multiresolut Inf Process 8(1):149–175
25. Ruqiang Y, Gao Robert X (2009) Base wavelet selection for bearing vibration signal analysis. Int J Wavel Multiresol Inform Process 7:411–426
26. Oussar Y, Rivals I, Personnaz L, Dreyfus G (1998) Training wavelet networks for nonlinear dynamic input-output modeling. Neurocomputing 20:173–188
27. Kardanpour Z, Hemmateenejad B, Khayamian T (2005) Wavelet neural network-based QSPR for prediction of critical micelle concentration of Gemini surfactants. Analytica Chimica Acta 531:285–291
28. Cui WZ, Zhu Ch, Zhao H (2005) Prediction of thin film thickness of field emission using wavelet neural networks. Thin Solid Films 473:224–229
29. Chen Y, Yang B, Dong J (2006) Time-series prediction using a local linear wavelet neural network. Neurocomputing 69:449–465
30. Hagan MT, Menhaj MB (1994) Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Netw 5:989–993
31. Ni Ampazis, Perantonis SJ (2002) Two highly efficient second order algorithms for training feedforward networks. IEEE Trans Neural Netw 13:1064–1074
32. Titsias MK, Likas AC (2001) Shared kernel model for class Conditional density Estimation. IEEE Trans Neural Netw 12:987–996
33. Colla V, Reyneri LM, Sgarbi M (1999) Orthogonal least squares algorithm Applied to the initialization of multi-layer perceptrons. In: European symposium on artificial neural networks. Bruges, Belgium, pp 363–369
34. Dhibi N, Ben amar C (2019) Multi-mother wavelet neural network training using genetic algorithm-based approach to optimize and improves the robustness of gradient-descent algorithms: 3D mesh deformation application. In: IWANN,pp 98–108
35. Dhibi N, Ben Amar C (2019) Multi-mother wavelet neural network based on genetic algorithm and multiresoluion analysis for fast 3d mesh deformation. IET Image Process 13:2480–2486
36. Wang Y, Guiqing L, Zhichao Z, Huayun H (2016) Articulated-motionaware sparse localized decomposition. Comp Graph Forum 36:247–259
37. Florian B, Gemmar P, Hertel F, Goncalves J, Thunberg J(2016) Linear shape deformation models with local support using graph-based structured matrix factorisation. In: CVPR, pp 5629–5638
38. Tan Q, Gao L, Lai Y-K, Yang J, Xia S (2018) Mesh-based autoencoders for localized deformation component analysis. In: Thirty-second AAAI conference on artificial intelligence, vol 32, issue 1, pp 2452–2459

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.