

# Parallel Implementation of Vascular Network Modeling

Krzysztof Jurczuk and Marek Krętownski

Faculty of Computer Science, Białystok Technical University  
Wiejska 45a, 15-351 Białystok, Poland  
{kjurczuk,mkret}@wi.pb.edu.pl

**Abstract.** The paper presents modeling of the vascular system in a parallel environment. The aim of this approach is to accelerate the simulation of vascular network growth and make it closer to analogous real life processes. We concentrated on the perfusion process and made an attempt to parallelize the process of connecting ischemic macroscopic functional units to existing vascular systems. The proposed method was implemented on a computing cluster with the use of the MPI standard. The results show that it is possible to gain a significant speedup that allows us to make simulations for a greater number of macroscopic functional units and vessels in a reasonable time, which increases the possibility to create more complex and more precise virtual organs.

## 1 Introduction

The human body is characterized by high complexity. It can be observed on each level, starting from molecules, cells and ending on organs and the whole organism [3]. Moreover, a lot of internal mechanisms are parallel or even distributed. These factors are the main reasons why the modeling of living systems is becoming more and more important. The modeling provides new ways to better understand complex interactions between elementary mechanisms and behaviors of the whole organs.

One of the main difficulties in model designing is the necessity to capture the most essential properties of the system and disregard the elements whose role is insignificant. It is not easy to choose appropriate simplifications, which applies especially to living organisms. Too simple models can be useless but, on the other hand, too elaborate models can be ineffective in practical cases, which means that the computations cannot be done in a reasonable time. Therefore, it appears natural to attempt to use parallel computing in the modeling of living organisms, especially vascular networks. Implementations in a parallel environment can accelerate the simulation process and allow us to introduce more sophisticated details.

In this paper, we focus on the modeling of vascular systems. They are very important in the detection processes of various pathological anomalies, because changes in vascular structures can be directly caused by diseases. Most of these modifications appear in medical images, especially when the contrast product

is administrated. Vessels play a key role in a contrast material propagation and they are one of the most visible structures in dynamic images. Therefore, the modeling of vascular systems can support the development of methods to detect early indicators of diseases and help to understand the mechanisms of image formation. Many vascular models have been proposed, e.g. Constrained Constructive Optimization (COO) method for an arterial tree generation [11], an algorithm of arterial tree growth inside a defined and gradually expanding shape [1], improved CCO method to simulate the coronary tree [6] or a fractal model [13]. According to our knowledge, all of them use the sequential algorithm to develop vascular systems.

In our previous studies [2] and [7], we used the physiological modeling as a way to better understand medical images (both CT and MRI) and to find some image markers of pathologies. In the research we also made use of the sequential algorithm to generate a virtual organ of liver (represented by three vascular trees and parenchyma) [8], CT simulator and MRI virtual scanner implemented in a parallel environment. In this paper, however, we propose a parallel algorithm of the vascular network growth, based on the previously used sequential algorithm. We concentrated on the perfusion process and parallelized the process of connecting new cells to existing vascular trees. The aim of this research is to accelerate the simulation of the vascular network growth and bring it as close to the real, analogous process as possible.

The rest of the paper is organized as follows. In the next section the organ model with sequential algorithm of vascular system development is briefly recalled. Whereas, in Sect. 3 the parallel algorithm of the same vascular growth process is presented. An experimental validation of the presented approaches is performed in Sect. 4. The conclusion and some plans for future research are sketched in the last section.

## 2 Model Description

In its generic form [8], the discussed model was constructed for the modeling of internal organs which develop by a division of their structural elements. But it should be emphasized that it is oriented towards an image generation. Therefore, the model concentrates on elements which are directly visible in images or have a significant influence on image analysis. The main components of the model are: the tissue and the vascular network that perfused it.

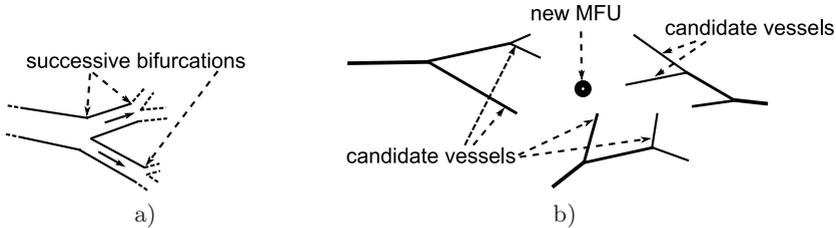
Most of features are not linked with any internal organ. However, it is very hard to model particular organs without some kind of specialization. Therefore, the model expresses the specificity of liver, as it is one of the most important organs. It plays a major role in the metabolism and has a number of functions in the body, including glycogen storage, decomposition of red blood cells, plasma protein synthesis, and detoxification [12]. Moreover, it possesses an unique organization of the vascular network with three types of trees: hepatic artery, portal vein and hepatic vein.

### 2.1 Tissue Modeling

The tissue is represented by a set of Macroscopic Functional Units (MFU) that are distributed regularly inside the specified shape. MFU is a small, fixed size part of tissue that constitutes the functional unit of the model. It is described by its class, which precises the most of its properties, both functional and structural (e.g. probability of mitosis and necrosis, blood flow rate, blood pressure, size and density). Moreover, certain parameters are described by defined distributions. This mechanism enables modeling the blood flow with more natural variability. Several classes of MFU can be defined in the organ, which allows simulating pathological changes.

### 2.2 Vascular Network Modeling

In the model, each vessel is represented by an ideal, rigid tube with a fixed radius, wall thickness, length and position. The wall thickness depends on the vessel diameter and its function. The model distinguishes vessels larger than capillaries, whereas the capillaries themselves are hidden in the MFUs. Based on this simplification, the vascular tree model assumes the form a binary tree (see Fig. 1a). It means that anastomoses, which occur sporadically, especially in pathological situations, are not taken into account. The binary trees, representing vascular trees, are built of nodes characterized by their spatial position, blood flow rate and pressure.



**Fig. 1.** Binary vascular trees: a) successive bifurcations b) perfusion process of new MFU - searching the closest vessels in three vascular trees

In the model, the blood is treated as a Newtonian fluid, with constant viscosity ( $\mu$ ), which makes it possible to calculate the pressure difference ( $\Delta P$ ) between two extremities by the Poiseuille’s law:

$$\Delta P = Q \frac{8\mu l}{\pi r^4} , \tag{1}$$

where  $l$  is the length,  $r$  is the radius and  $Q$  is the blood flow of the vessel. Moreover, at each bifurcation the law of matter conservation has to be observed:

$$Q = Q_r + Q_l , \tag{2}$$

where  $Q$  is the blood flow in a parent vessel,  $Q_r$ ,  $Q_l$  are the blood flows in descendant vessels (right and left daughter branches). It means that the quantity of blood which entering and leaving a bifurcation has to be equal. Another equation is connected with a decreasing radius of vessels in the trees where we move from proximal to distal segments of vascular trees:

$$r^\gamma = r_r^\gamma + r_l^\gamma, \quad (3)$$

where  $r$  is the radius of parent vessel,  $r_r$ ,  $r_l$  are the radiuses of descendant vessels (right and left daughter branches) and  $\gamma$  varies between 2 and 3 [5]. This morphological law describes the dependency of the mother vessel radius and the radiuses of its two daughters.

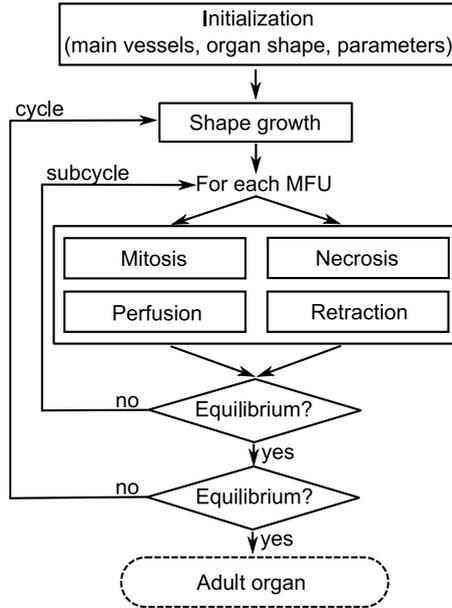
### 2.3 Sequential Vascular System Growth Algorithm

The organ growth is modeled as a analogy to a hyperplasia process (the increasing number of cells). It starts with an organ, whose size is a fraction of adult one. As it is shown in Fig. 2, after parameters initialization, in discrete time moments (called cycles), an organ enlarges its size. Therefore, between MFUs a new, empty space appears. Additionally, each cycle consists of subcycles. In each subcycle, a MFU has a certain probability to give birth to a new MFU of the same class or to die. Consequently, changes in the tissue and in the corresponding vascular network appear. The processes of the birth/perfusion and the death/retraction are repeated in each subcycle until the empty space, which can appear between cycles, is not occupied by new MFU elements. The whole process ends when organ reaches its full, adult size.

At the beginning of each subcycle, for every MFU, a few randomly chosen spatial positions of the new MFU in a neighborhood are tested. If all conditions connected with a free space and a tissue density are fulfilled, a new MFU is created. This new, small functional unit is not perfused by the existing vascular system and it is initially ischemic.

The next step is to find an optimal bifurcation point which can be used to perfuse a new MFU. First, the distances between all vessels and the new element are calculated. We choose a fixed number of the closest vessels (see Fig. 1b). Later, temporary bifurcations are created. When there are more than one tree, the algorithm chooses all possible combinations of candidate vessels (a single combination consists of one vessel from each tree). The spatial position of the bifurcation is controlled by the Downhill Simplex procedure [10] (minimization of additional blood volume needed for the new MFU perfusion [8]).

Only one combination of vessels can be used to perfuse the new MFU. Therefore, we have to choose the one which is the most appropriate from among all candidates. Additionally, the possible collisions between vessels must be checked. In the presented approach only non-crossing configurations are taken into account. Therefore, the algorithm detects intersections between perfusing vessels both from the same and different trees. Finally, for each remaining configuration the volume of the whole tree is computed. The combination with the lowest sum of volumes permanently perfuses the MFU.



**Fig. 2.** Flow chart representing two loops of events which are distinguished in the presented modeling of the organ

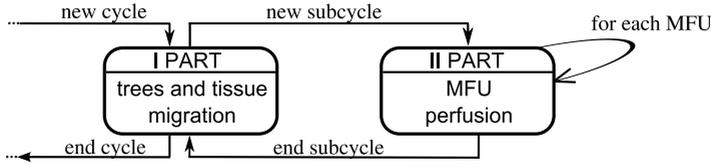
The MFUs are perfused by the vascular system in a sequential manner, one by one. This process is time consuming because for each new MFU a large number of temporary bifurcations is created. It requires many calculations to assure the consistency of the characteristics (i.e. blood flow and pressure,...) describing individual vessels.

After the reproduction process, comes the degeneration phase in which some of the MFUs can die. The algorithm of retraction is not so time consuming in comparison to the perfusion process. The vessels supplying the MFU simply retract and disappear. It requires only a single recalculation of the constraints for the vascular system.

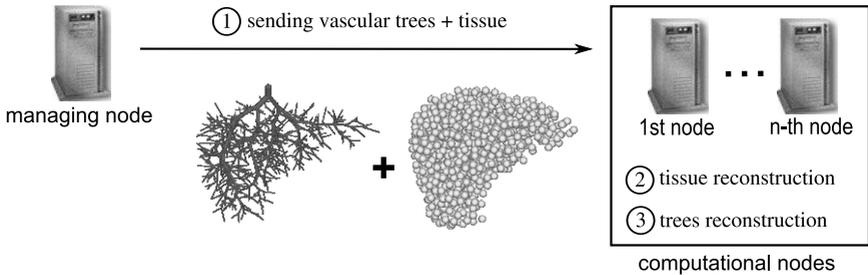
### 3 Parallel Vascular System Development

The most time consuming operation in the presented algorithm of the vascular network growth is the process of perfusion. It results from large number of MFUs, the complicated structure of vascular trees and especially by the necessity to find the optimal bifurcation. Therefore, a decision was made to spread the computations concerning the perfusion process over computational nodes. Moreover, our intention was to bring the solution closer to reality, where analogous perfusion processes can be also parallel.

The general scheme of the proposed algorithm in a parallel environment is presented in Fig. 3. The algorithm has two parts. The first (see Fig. 4) is



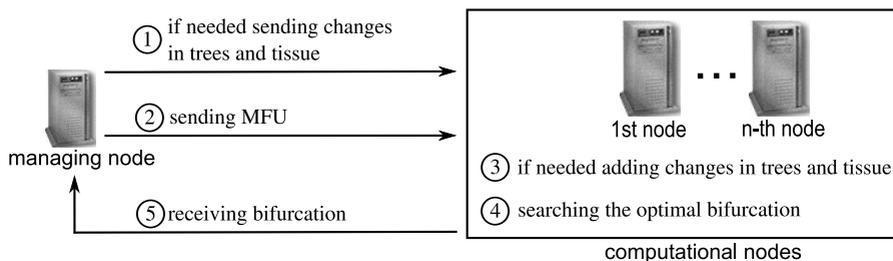
**Fig. 3.** Two parts of the parallel algorithm. The first performed at the beginning of each subcycle and connected with the trees and tissue migration. The second performed between subcycles and responsible for the distribution of the perfusion process over nodes.



**Fig. 4.** The first part of the algorithm connected with the trees and tissue migration

performed at the beginning of each subcycle while the second (see Fig. 5) does calculations between subcycles.

Each node must have the vascular system: trees and MFUs, as recent as possible. Therefore, at the beginning of each subcycle (the first part of the presented algorithm) the managing node sends the latest vascular trees and tissue represented by MFUs to the computational nodes. The vascular system can be large and complex, therefore its migration process between the processors within the framework of the message passing interface is composed of 3 steps: packing the nodes and MFUs into a flat message, sending the message and unpacking corresponding nodes and MFUs. In order to minimize the message size we choose only the parameters of the nodes that cannot be reconstructed: position in space, possession of children, individual node number and MFU class. When the computational node receives the message with the vascular trees, the remaining characteristics is restored. Almost all indispensable information about MFUs is sent with trees. Additionally, as the blood flow is unique in each MFU, we also have to transfer the value of the flow. Moreover, a lot of parameters about structure are read from input files at each node, which enables us to send quite a small package in comparison to the real size of the vascular system. We also assigned individual numbers to tree nodes and MFUs, which facilitated the process of migration, rebuilding and permanent perfusion in the managing node. To sum up, each computational node possesses identical vascular trees and MFUs after the completion of the first part of the algorithm.



**Fig. 5.** The second part of the algorithm responsible for MFUs perfusion

The second part of the algorithm is responsible for a calculation occurring between individual subcycles. First, the managing node creates the list of new MFUs which can be added to the vascular network. There is a possibility to model this process in a parallel environment but the time needed for that part of the algorithm can be neglected, as it is very short, in comparison to the perfusion time. Next, the managing node makes no attempt to find vessels in order to add new MFUs. They are sent to the computational nodes. The MFU migration is more simple and less time consuming than the entire trees, but to minimize a message within the framework of the message passing interface we only choose the essential information, namely: position in space, blood flow and MFU class. When the computational node receives the message, it tries to find the closest vessels and the optimal point to connect the received MFU to the vascular network. It makes the simulation of the perfusion process. If the searching ends with a success, the computational node does not perfuse permanently a new MFU but sends the parameters of the bifurcation to the managing node. The message contains only the position of the bifurcation point in space and numbers of perfusing vessels.

Next, when the managing node receives the parameters of the bifurcation, it checks if there have been some other changes in its trees since the last contact with the sender of a message. If there have been, it checks the changed vessels. If at least, one vessel that was changed, is on the list of vessels to perfuse the new element, the MFU is rejected. But in the other case, the managing node permanently joins the new MFU and broadcasts all new changes that occurred between previous and present contact with the sender of the message. The migration of changes is less time consuming than the entire trees. All vascular trees are only sent at the beginning of each subcycle. If there are more MFUs to perfuse, the managing node sends the next one.

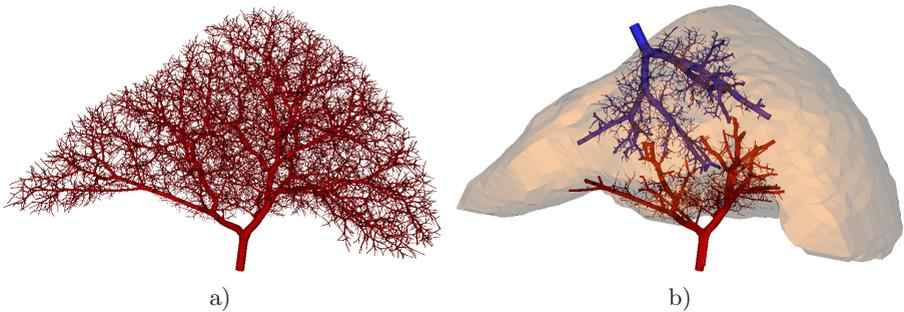
The whole vascular system has to be sent at the beginning of each subcycle as there are several other processes (e.g. degeneration and growth of organ shape) between cycles and subcycles, which has an influence on the entire trees.

The algorithm of creating new MFUs ensures that the number of rejected MFUs is small enough. Moreover, the rejected MFUs leave empty space between vessels and other MFUs, which increases a probability that the vascular network

growth algorithm will choose more macroscopic functional units in the next subcycle.

## 4 Experimental Results

This section contains a preliminary, experimental verification of the proposed algorithm in a parallel environment. The presented results were obtained from many experiments. We used the default settings for the sequential version (about 12000 MFUs). Moreover we checked the behavior of the proposed solution for large configurations with about 50000 MFUs and consequently about 300000 vessels (Fig. 6 shows a visualization of one of the obtained vascular systems).

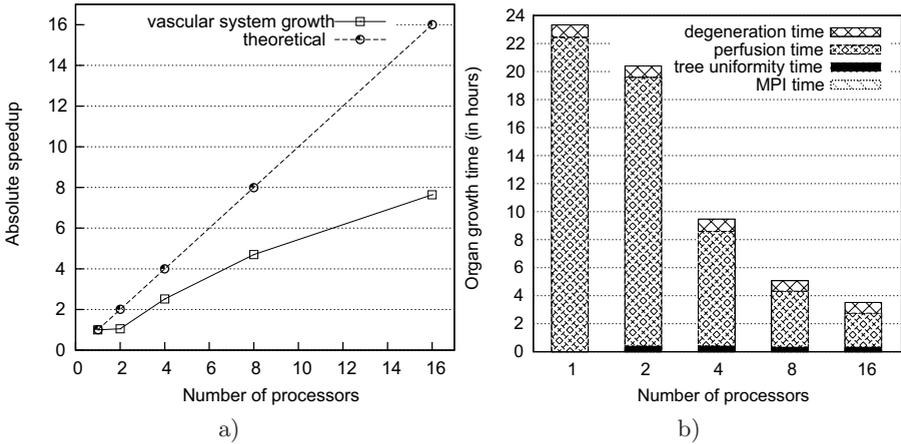


**Fig. 6.** Visualization of the adult liver (about 50000 MFUs and 300000 vessels): a) portal veins, b) main hepatic arteries, portal veins and hepatic veins with liver shape

In the experiments a cluster of sixteen SMP servers running Linux 2.6 and connected by an Infiniband network was used. Each server was equipped with two 64-bit Xeon 3.2GHz CPUs with 2MB L2 cache, 2GB of RAM and an Infiniband 10Gb/s HCA connected to a PCI-Express port. We used the MVAPICH version 0.9.5 [9] as the MPI standard implementation [4].

Figure 7a presents the obtained mean speedup. It is far from linear but from practical point of view is very satisfactory. Usually the process to obtain an organ with about 50000 MFUs on the single processor machine (3.2GHz CPU and 2GB of RAM) takes about 24 hours. On the other hand, the parallel version can simulate it approximately 8 times faster (with 16 processors). Moreover, it is worth noting that the speedup, in spite of the increasing number of MFUs, does not decrease significantly. For 16 nodes it still varies around 8.

In order to in depth explain presented results a detailed time-sharing figure is necessary and is presented for the case of 50000 MFUs (see Fig. 7b). It is clearly visible that the most time consuming operation is the perfusion process. The degeneration phase takes only a small part of the time necessary to develop the adult organ. The time connected with the MPI operations (e.g. sending and receiving messages) is insignificant in comparison to the time of other operations. In our case it was always less than 1% of the whole simulation time. Moreover,



**Fig. 7.** Efficiency of the parallel implementation: a) mean speedup for many configurations of MFUs, b) detailed time-sharing figure for the case of about 50000 MFUs

we can observe that the algorithm also spends a short period of time on the processes connected with the maintaining of uniformity of the vascular system (e.g. selection and gathering changes at the managing node and adding changes at the computational nodes).

Furthermore, it should be mentioned that we changed the memory organization at the computational nodes. To optimize the time connected with tree rebuilding we introduced a continuous memory representation. It decreased the time needed to allocate and deallocate memory. Prior to this mechanism the mean speedup for 16 nodes was equal approximately 6.7.

## 5 Conclusion

In this paper, a parallel algorithm to model the vascular network growth is investigated. It was shown that the presented solution significantly reduces the computation time, which increases the possibility to create more elaborate and precise virtual organs. This can be very useful when we use them in CT or MRI simulators. Moreover, this solution can be treated as the first step to bring the presented model closer to reality, in which the analogous processes of the vascular network growth can occur in a parallel way.

The presented algorithm is still under development. We see a lot of possible directions for future improvements and at least a few different approaches. First, we would like to introduce more decentralized solution. We also plan to implement the process of perfusion in the framework of multi-platform shared-memory parallel programming (OpenMP), which makes it possible to reduce the time period connected with tree rebuilding, waiting for new MFUs and tree uniformity.

**Acknowledgments.** This work was supported by the grant W/WI/5/08 from Białystok Technical University.

## References

1. Bézy-Wendling, J., Bruno, A.: A 3-D dynamic model of vascular trees. *Journal of Biological Systems* 7(1), 11–31 (1999)
2. Bézy-Wendling, J., Krętownski, M., Mescam, M., Jurczuk, K., Eliat, P.-A.: Simulation of hepatocellular carcinoma in MRI by combined macrovascular and pharmacokinetic models. In: 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1272–1275. IEEE Press, Washington (2007)
3. Demongeot, J., Bézy-Wendling, J., Mattes, J., Haigron, P., Glade, N., Coatrieux, J.-L.: Multiscale modeling and imaging: the challenges of biocomplexity. *Proceedings of the IEEE* 91, 1723–1737 (2003)
4. Juhasz, Z., Kacsuk, P., Kranzlmuller, D.: *Distributed and Parallel Systems: Cluster and Grid Computing*. Springer, Heidelberg (2005)
5. Kamiya, A., Togawa, T.: Optimal branching structure of the vascular trees. *Bulletin of Mathematical Biophysics* 34, 431–438 (1972)
6. Karch, R., Neumann, F., Neumann, M., Schreiner, W.: Staged growth of optimized arterial model trees. *Annals of Biomedical Engineering* 28, 495–511 (2000)
7. Krętownski, M., Bézy-Wendling, J., Coupe, P.: Simulation of biphasic CT findings in hepatic cellular carcinoma by a two-level physiological model. *IEEE Trans. on Biomedical Engineering* 54(3), 538–542 (2007)
8. Krętownski, M., Rolland, Y., Bézy-Wendling, J., Coatrieux, J.-L.: Physiologically based modeling for medical image analysis: application to 3D vascular networks and CT scan angiography. *IEEE Trans. on Medical Imaging* 22(2), 248–257 (2003)
9. Liu, J., Wu, J., Panda, D.K.: High performance RDMA-based MPI implementation over InfiniBand. *Int. Journal of Parallel Programming* 32(3), 167–198 (2004)
10. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical recipes in C. The art of scientific computing*. Cambridge University Press, Cambridge (1992)
11. Schreiner, W., Buxbaum, P.F.: Computer-optimization of vascular trees. *IEEE Trans. on Biomedical Engineering* 40(5), 482–491 (1993)
12. Sherlock, S., Dooley, J.: *Diseases of the liver and biliary system*. Blackwell Science, Malden (2002)
13. Zamir, M.: Arterial branching within the confines of fractal L-system formalism. *Journal of General Physiology* 118, 267–275 (2001)