

The structure of the software for electric impedance tomography has been presented. This structure of application construction makes it possible to carry out the real-time EIT research and can be implemented on medical and technical devices, in particular, integrated into AVL devices.

The algorithm for visualization of the results of conductivity field reconstruction was presented. Within this algorithm, there are two approaches to presenting color models and selecting colors for each particular finite element. The choice of one of these approaches depends on the needs of the study and leads either to faster performance, or to the higher quality of an image.

The algorithm of neighboring finite elements, allowing reducing the time consumed to visualize the model by uniting neighboring elements with a similar color in one polygon, has been proposed. Reducing the number of finite elements leads to a higher speed of their output on the screen.

A list of graphics libraries that can be used for the problems of visualization of the results of electric impedance tomography was presented. As a result of the research, it was found that among the analyzed libraries, the best time is demonstrated by the OpenGL library, which ensures the visualization of 0.02 s faster than in the case of the analogs. This is due to the high operation speed, which is provided by the implementation of the GPU visualization.

It was shown that the use of the proposed algorithm of neighboring finite elements actually allows reducing the time spent on displaying the model on the screen from 0.05 s to 0.03 s for the OpenGL library. At the same time, the total time spent on visualization depends on the used graphics library.

The obtained data can be used in the development of medical visualization systems, which should meet increased requirements in terms of the amount of displayed information

Keywords: electric impedance tomography, software, image reconstruction, medical visualization, graphics libraries

Received date 12.06.2020

Accepted date 17.08.2020

Published date 28.08.2020

1. Introduction

Electric impedance tomography (EIT) [1] is the method for medical visualization [2] of the distribution of specific resistance of the human body (or internal organs). The main advantage of the EIT over other types of non-invasive examination (magnetic resonance tomography, computer tomography) is the ability to assess the functional state of the object under study (organ) in real-time. Such monitoring of indicators of the patient's state makes it possible to evaluate them for prompt actions. One of the advantages of the EIT method is its applicability for problems of continuous bedside monitoring of the patient's state due to its safety and non-invasiveness. At the same time, significant results were achieved in the combined use of EIT together with the AVL devices, in particular, to monitor the air filling of lungs during the implementation of artificial ventilation.

The practical implementation of the EIT methods on medical facilities inevitably necessitates software devel-

Copyright © 2020, A. Katsupeev, G. Aleksanyan, E. Kombarova, R. Polyakov

This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0>)

opment. At the same time, one of the key problems in the development of software for electric impedance tomography systems is the visualization of the conductivity field of the object under study. For these types of tasks, the key parameters are the continuity of work and visualization speed. This is because the model of an object is split into many finite elements, the colors of which are re-calculated every time the data arrive from the measuring device.

Special software packages – graphics libraries – are used to visualize the conductivity field. The libraries under consideration should be compatible with the programming languages used in the field of medical research. Since graphics libraries have different characteristics in terms of speed and continuity of operation, it is advisable to conduct a study that can determine the optimal library for visualizing EIT results.

The relevance of the research in this area is proved by the existence of a large number of information systems both for operation in the EIT field, and for other medical visualization systems.

UDC 004.514

DOI: 10.15587/1729-4061.2020.210523

STUDY OF GRAPHICS LIBRARIES RELATED TO THE PROBLEM OF VISUALIZATION OF ELECTRICAL IMPEDANCE TOMOGRAPHY IMAGES

A. Katsupeev

PhD, Senior Lecturer*

E-mail: andreykatsupeev@gmail.com

G. Aleksanyan

PhD, Associate Professor*

E-mail: graer@yandex.ru

E. Kombarova*

E-mail: ms.ekom@mail.com

R. Polyakov*

E-mail: r.s.polyakov@gmail.com

*Department of Information

and Measuring Systems and Technologies
Federal State Budget Educational Institution
of Higher Education "Platov South-Russian State
Polytechnic University (NPI)"
Prosveshcheniya str., 132, Novocherkassk,
Russian Federation, 346428

2. Literature review and problem statement

Visualization of data, obtained from medical and biological examination of a human is a significant area. Thus, paper [3] analyzes the construction of a three-dimensional model of the object under study based on the data obtained during tomographic measurements. The specific feature of this approach is the dynamic construction of a model, depending on the individual characteristics of a patient. This does not address the task of dynamic visualization by displaying the real-time changes in an object. The problem of 3D visualization is also solved in research [4] but does not address the issues of image visualization speed. Article [5] examines the speed, at which a three-dimensional image is constructed based on images of spiral computer tomography, but the resulting speed (2–16 minutes) does not make it possible to use this approach in EIT problems, where real-time visualization is important. In paper [6], visualization occurs at a fairly fast speed (less than 1 s at certain parameters), but this speed is not sufficient to determine perfusion during EIT. In addition, the problem of visualizing the images, obtained during the computer tomography process, is considered in research [7], but the time of model visualization was not specified in the study.

Paper [8] analyzes the construction of the grid of finite elements and proposes a new approach enabling finding neighboring nodes with greater efficiency. However, this approach involves only the faster construction of the finite element model and does not involve a faster dynamic visualization of a change in the object’s state.

Article [8] examines the visualization algorithms that do not involve the construction and use of the grid of finite elements, but this approach is not currently applicable to EIT due to the limitations of the method. These limitations are related to the use of the reconstruction matrix, which makes it possible to obtain the values of the conductivity field on finite elements based on measurement data.

Thus, the presented approaches are not applicable to the EIT in real-time due to much time it takes to visualize a single frame. It is therefore appropriate to consider existing graphics libraries and to determine whether they can be used in medical and technical devices implementing the EIT methods. In addition, it is appropriate to conduct a study involving decreasing the time it takes to visualize images obtained from the EIT measurement process because the presented studies do not contain the suitable solution for dynamic visualization of the EIT.

3. The aim and objectives of the study

The aim of this paper is to study and to substantiate the selection of the necessary graphics library to address the challenges of visualizing the results of the EIT examination, taking into consideration the need to implement the capabilities of setting the measurement channel and the operation mode of the device in one window, as well as the user control of all processes of collection and processing the measurement data.

These papers are necessary when making the layout of the EIT device, as they make it possible to select the tool-

set for effective visualization of the data obtained from the functional unit.

To achieve the set goal, the following tasks should be accomplished:

- to present the structure of the application for the operation of the electric impedance tomography system in real-time, to identify and analyze the graphics libraries that can be used to display the results of electric impedance tomography for medical applications;
- to describe the algorithm of visualization of the conduction field;
- to develop the algorithm that can speed up image visualization;
- to conduct experiments comparing different graphics libraries in terms of image updating speed.

4. Description of the structure of EIT channel software

The layout of the EIT channel is a set of hardware and software solutions that can be conventionally divided into the following basic units (Fig. 1):

- functional unit (hardware part);
- functional unit (software part).

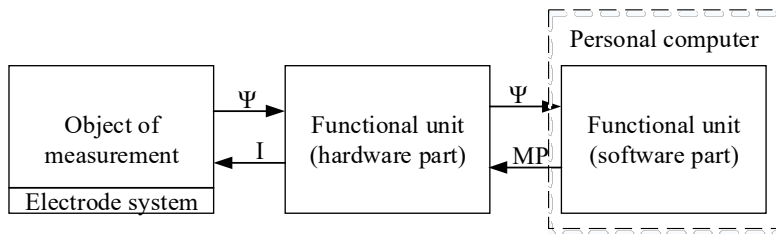


Fig. 1. Basic units of the hardware and software complex

The functional unit of the hardware part (HP) is an electronic module for current injection through the patient’s body and subsequent recording of the emerging potentials on the electrodes of the electrode system (ES). It is functionally represented by a source of high-frequency current of small amplitude, a control unit, the PC conjunction unit.

The functional unit of the software part (SP) is software modules performed on a PC: the module of interaction with the device, the module for computation of ventilation and perfusion indicators, the module of visualization of human respiratory function, the module of setting the measurement processes.

Analysis of the interaction of the explored functional units in terms of dividing into separate stages of obtaining and processing medical and technical data enables us to present the information shown in Fig. 1 in the form of a supplemented and generalized approach (Fig. 2).

The module of interaction with the device sends a request for data from the device and receives measurement data that represent the differences of potentials on measuring electrodes.

The module of computation of ventilation and perfusion indicators based on the obtained measurement data makes computations of the conductivity field.

The module of visualization of the human respiratory function visualizes the reconstructed conductivity field and ensures user-assisted interaction to control the program.

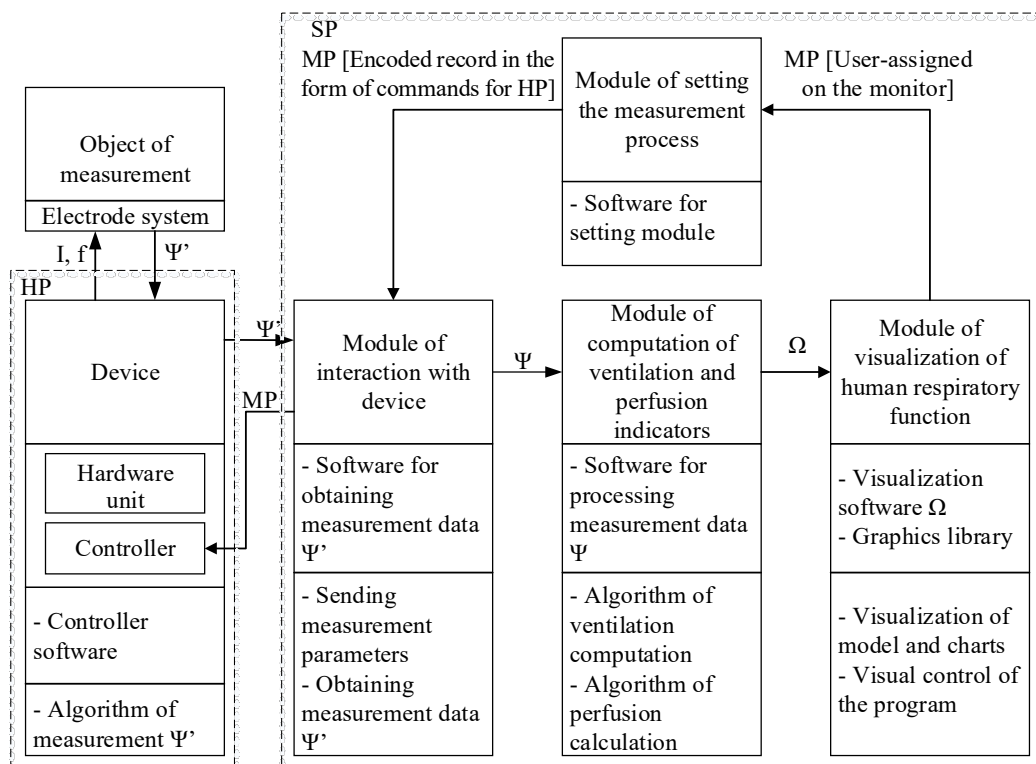


Fig. 2. Block diagram of the interaction of application units

The module of setting the measurement process implements processing the data obtained from the visualization module and transmits them to the HP of the device.

Fig. 2 represents the interaction of the application units as a set of measurement information flows (Ψ'), influencing flows (I, f), and information flows (MP, Ψ, Ω).

Influencing flows include information that is used to control the measurement process. This information is sent from a personal computer to a measuring device.

The flows of measurement information, on the contrary, are the data from the measuring device. In this case, information related to this group is not processed and is only measuring data.

Information flows are the information that is seen by a software user. This includes the measurement parameters set by the software interface and processed measurement data, based on which the values of the conductivity field are obtained.

Presented modules and flows of the application to visualize the conduction field of the examined object implement the developed algorithm of the application operation, which is shown in Fig. 3.

When the software is run, it is connected to the device (HP), after which a user sets measuring parameters MP, including the power of current I , frequency of injected current f , and signal shape (sinusoidal, triangular or rectangular). Then, NB is set – the number of belts connected to the patient’s body. In case measurements are made based on only one belt, the measurement data presented by array Ψ arrive from the device within a measuring cycle.

The reconstruction algorithm implies a mathematical apparatus for the reconstruction of distribution of electrical resistance within a biological object.

The input data for the reconstruction algorithm are differences of potentials on measurement electrodes, presented in the form of sets $\psi = \{\phi_1, \dots, \phi_n\}$, $\Psi' = \{\phi'_1, \dots, \phi'_n\}$, where:

- $\phi_i, i=1, \dots, n$, is the difference of values of potentials on measuring electrodes during the current measurement;
- $\phi'_i, i=1, \dots, n$, is the difference of values of potentials on measuring electrodes during the reference measurement;
- n is the number of measuring electrodes.

After the first measurement of potentials, the first measurement vector becomes a reference vector. Thus, the first measurement shows a monochrome picture due to the lack of differences between measurements. Indicator $\bar{\varphi}$ is computed for the first and each subsequent vector of measurements received from the device:

$$\bar{\varphi} = \frac{\sum_{i=1}^n \phi_i}{n}, \tag{1}$$

where $\phi_i, i=1, \dots, n$, is the difference of the values of potentials on measuring electrodes during current measurement; n is the number of pairs of measuring electrodes.

This indicator represents the mean value of measured potentials Ψ . Since the first measurement does not have a reference for comparison, it becomes a reference:

$$\bar{\varphi}' = \bar{\varphi}; \quad \phi'_i = \phi_i, \text{ for } i=1, \dots, n.$$

Besides the mean value of potentials, the first measurement vector also becomes a reference. For each subsequent measurement vector, the condition is verified:

$$\text{if } \bar{\varphi} < \bar{\varphi}', \text{ then } \bar{\varphi}' = \bar{\varphi}; \quad \phi'_i = \phi_i, \text{ for } i=1, \dots, n.$$

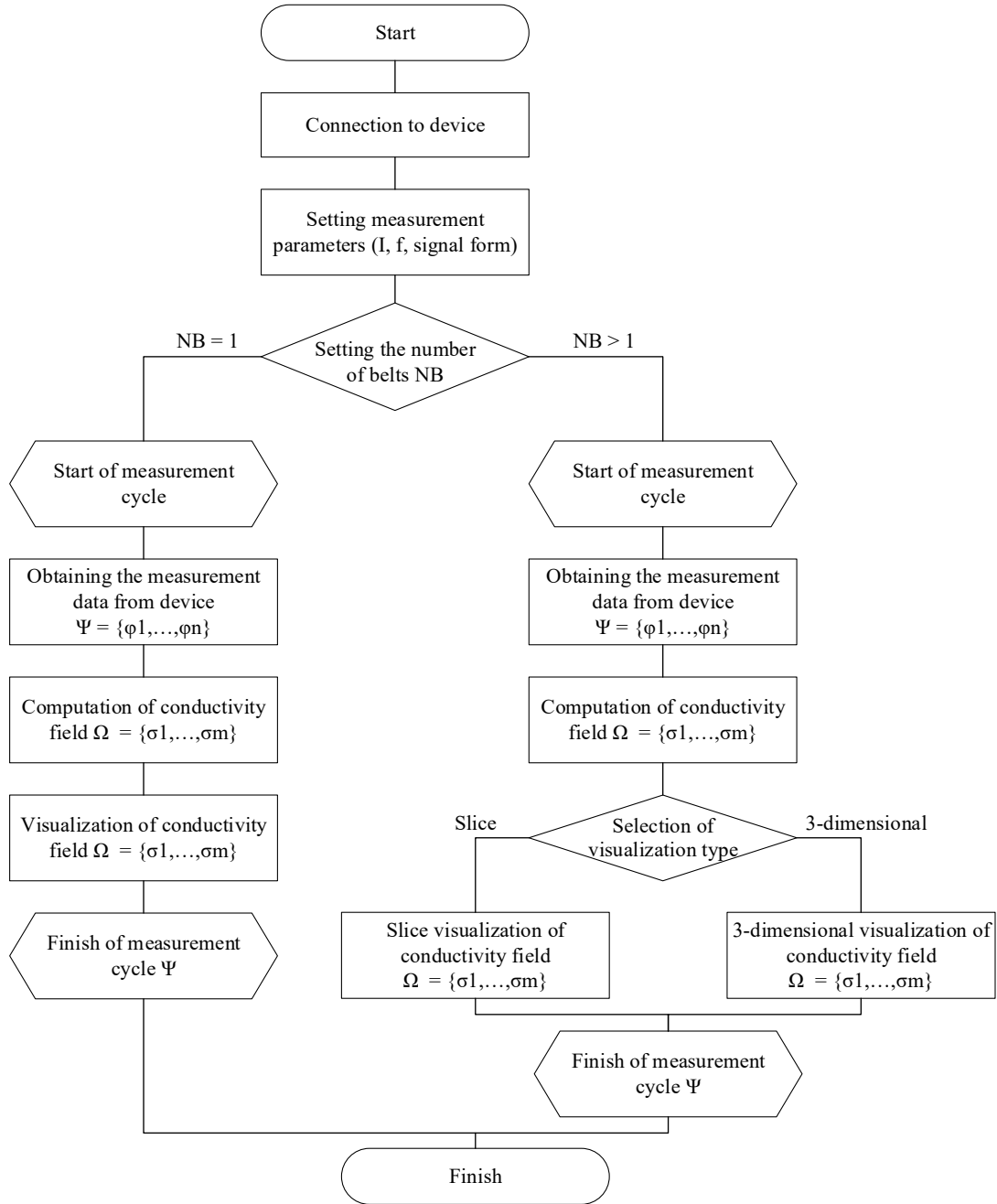


Fig. 3. Block diagram of the developed operation algorithm of conductivity field visualization software

Conductivity field for the studied object within the tomographic cross-section is computed from the following formula:

$$\Omega = H^*(\Psi' - \Psi), \tag{2}$$

where H is the pre-generated reconstruction matrix. The reconstruction matrix consists of pre-generated coefficients that serve to compute the values of the conductivity field based on the measurement vector. Thus, the number of rows of the matrix is equal to the number of finite elements; the number of columns is equal to the size of the measurement vector.

Vector Ω is a totality of the values of the conductivity field in the finite elements of the model of a reconstructed object:

$$\Omega = \{\sigma_1, \dots, \sigma_m\}, \tag{3}$$

where m is the number of finite elements of the model.

After the computation of the conductivity field is completed, it is visualized.

If the number of belts is more than one, it is possible to choose the visualization type. The first type is a slice visualization, in which each tomographic slice is displayed separately in a two-dimensional plane. The second type is three-dimensional visualization, in which a single three-dimensional model of the studied object is formed.

The principles for the construction of the software window for the medical EIT device were developed. A schematic image of the software window is shown in Fig. 4.

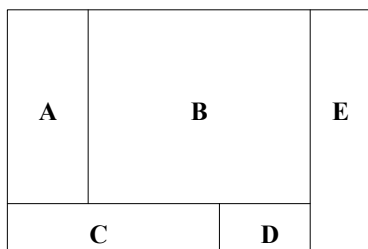


Fig. 4. A schematic image of the software units of the developed EIT channel

The window combines five main units of the interface (containers) that make it possible to organize the full operation of the device and its main software and hardware modules, shown in Fig. 2. Table 1 describes each container.

Table 1

Units of the interface of the software for EIT examination visualization and control

Unit	Description	Comments
A	Unit of reconstructed conductivity fields	Displaying reconstructed tomographic cross-sections corresponding to the number of electrode belts
B	Chart unit	General chart of ventilation and a chart for each lung
C	Setting unit	Setting measurement parameters
D	Unit of additional information	Computed values of indicators of ventilation, perfusion, ventilation-perfusion ratio
E	Control unit	Control of measurement process, exit from the program

Unit A is a container for reconstructed conductivity fields based on the data obtained from the electrode system, located on the human chest surface.

Unit B serves to display the lung ventilation charts.

Unit C is meant to set and control the measurement process.

Unit D serves to display additional information.

Unit E includes the buttons for controlling the program and measurement process. B

This interface needs to be implemented with state-of-the-art graphics visualization tools. In general, it can be concluded that these tools represent a separate extensive class of software products of dynamic and static visualization [10–13]. In this case, the main developers of these systems are Microsoft, Gnome Foundation, and Silicon Graphics, which offer ready-made solutions without a specialized focus on the use in medical applications.

This raises the practical need to select the graphics library that is most appropriate for the tasks under consideration.

Currently, the authors are carrying out comprehensive works on the development of the hardware and software EIT means for the tasks of bedside long-term monitoring of the human lungs during the AVL ventilation. To do the works related to the development of the EIT channel layout, there arises a need to develop a new user graphics interface that would take into consideration all the specific features of the conductivity field visualization, including real-time dynamic

visualization and interactive control of measurement and control parameters.

At the same time, the main criteria for choosing a graphics library are:

1) faultless operation. Since the EIT is the method of bedside monitoring of the patient's condition, the graphics shell should provide the ability to visualize a person's respiratory function over a long period;

2) the speed of visualization of finite elements. The model of the studied object consists of many finite elements, the number of which can reach several thousand. Because the color of the elements changes every time the data arrive from the measuring device, a graphics library should visualize the model at sufficient speed. This criterion is also extremely important for determining the perfusion of the patient's lungs, as the heart contraction rate is much higher than the frequency of inhalations/exhalations used to display the lungs ventilation;

3) compatibility with medical equipment. Since the EIT is often connected as an additional module to other medical devices, it is necessary to use the technology capable of integrating with other medical equipment and perform the assigned functionality;

4) the scaling possibility for further developments, in particular, for 3D visualization of the conductivity field of the subject under study;

5) download the device's CPU during visualization. Since EIT research involves a large amount of real-time computations, it makes sense to transfer visualization tasks from the CPU to the GPU.

Based on the review of the current state of the domain, the study of scientific and technical information in the area of medical visualization from open sources, the study of the patent situation, as well as taking into consideration the EIT specificity, the following effective graphics libraries were considered:

1) the Microsoft Foundation Classes (MFC) library is the library in C++ language, developed by Microsoft. This library is considered outdated and is used in a small number of projects, however, it is popular in the area of medical visualization [14];

2) the GTK library is the graphics library, used mainly in the systems based on the Linux core. Its positive qualities include the relevance of technology and cross-platforms [15];

3) the OpenGL library is the technology using the graphic processor as a means of image visualization [16];

4) the Windows Forms library is the development of the Microsoft company, allowing combined use with different programming languages [17];

5) the combined use of OpenGL with other libraries. In particular, papers [18, 19] give an example of using in medical equipment of the MFC technology including the Open GL module.

It is appropriate to consider these solutions for the possibility of their use in the EIT medical devices and to conduct experiments to assess the time of visualization of the conductivity field for each of these tools.

Thus, the structure of the software of visualization of the EIT measurements makes it possible to measure the differences of potentials recorded on the electrodes located on the patient's body, to compute the conductivity field, and visualize the tomographic cross-section in real-time. The information, measuring and influencing flows, using which

the system's modules interact, are presented. These flows enable the organization of interaction and transfer of the necessary data between the modules of the system. We developed a software operation algorithm that makes it possible to control the process of collecting and processing the measuring data from several electrode belts, as well as to implement visualization both by tomographic slices corresponding to electrode belts and by a single 3D model. A schematic representation of information units located in the software window was developed. This representation enables displaying processed measurement information in the software window and providing general control of the EIT examination process. The criteria for choosing a graphics library were presented and a range of analyzed graphics libraries was separated.

5. Visualization of reconstruction results

Visualization of the reconstruction results sets the task to translate the relative values of the conductivity field into a color scheme for the convenience of their perception by the user of the system.

There are two approaches to translating the values of the conductivity field into the values of the color scheme:

1) the approach with assigned values of a color-scheme that sets a matrix of the ranges of conductivity field values and corresponding color values in the RGB color model. The merit of this method is high operation speed, caused by the fact that no extra time is spent to compute the color for each final element. The downside of it is lower visualization accuracy compared to the second approach;

2) the approach where color values in the RGB color model are computed dynamically, depending on the values of a conductivity vector. This method takes longer to compute the value of the derived color, but at the same time, it is more accurate than the first approach.

The developed algorithm of conductivity field visualization is shown in Fig. 5.

In the first approach, the possible values of the conductivity field are split into a number of ranges, according to which the color value is assigned by the RGB color model. This creates color matrix $CS = \{cs_1, \dots, cs_m\}$, in which color c_i , consisting of the values by red, green and blue color r_i, g_i, b_i , is assigned to each range cs_i . These values are in the range $[0, 255]$, in accordance with the RGB standard.

Dynamic colors imply an approach, in which the maximum by module element of the conductivity field σ^{\max} is computed so that the condition should be satisfied:

$$\sigma^{\max} \geq |\sigma_i| \text{ for } i=1, \dots, m, \tag{5}$$

where m is the number of finite elements in the model.

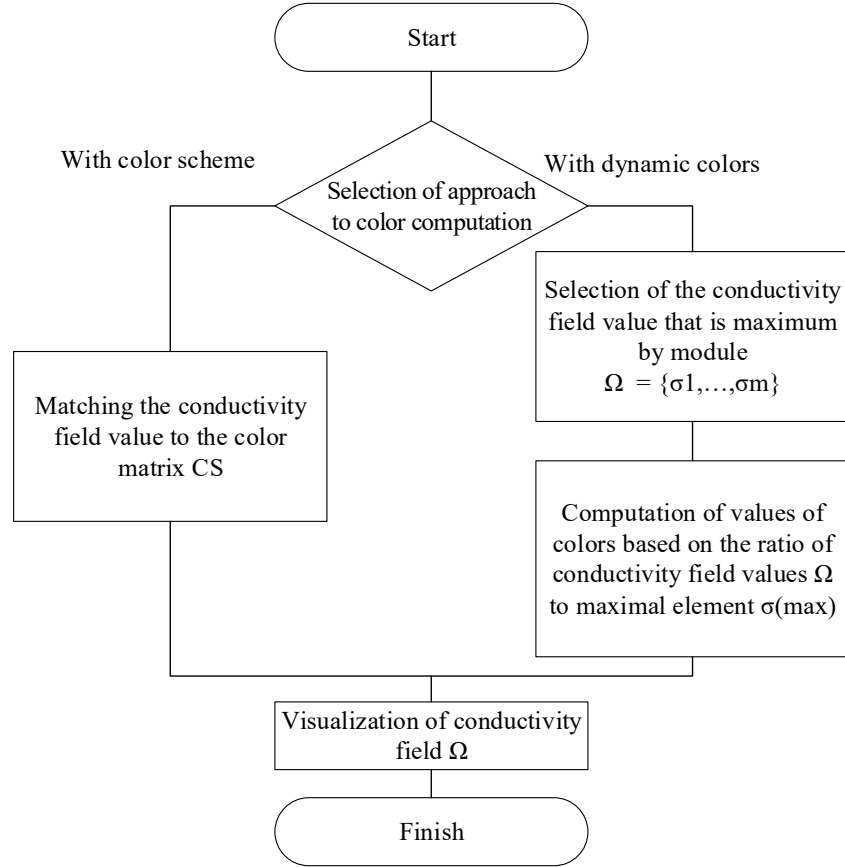


Fig. 5. The developed algorithm of visualization of the conductivity field of the research object

The color is subsequently determined dynamically for each finite element in the conductivity field. Because the colors for positive and negative values are in different parts of the color spectrum, computation for them is different:

$$1) \text{ if } \sigma_i \geq 0, \quad r_i = \frac{\sigma_i * 255}{\sigma^{\max}}, \quad g_i = 0, \quad b_i = \frac{\sigma_i * 255}{\sigma^{\max}}; \tag{6}$$

$$2) \text{ if } \sigma_i < 0, \quad r_i = 0, \quad g_i = 0, \quad b_i = \frac{\sigma_i * 255}{\sigma^{\max}}; \tag{7}$$

where σ_i is the element of the conductivity field, σ^{\max} is the maximal by module element of the conductivity field, r_i, g_i, b_i are the values of red, green, and blue colors.

In the case of positive values (these values most often include the elements affected by the heartbeat), the finite elements are colored purple. In the case of negative (lungs ventilation), the finite elements are colored blue.

As a result of the visualization of the model of finite elements, the following image is displayed on the screen (Fig. 6).

Since visualization speed is extremely important when displaying the conductivity field, it is advisable to develop an algorithm, using which it is possible to accelerate displaying finite elements on the screen. This algorithm is explored in the next section.

Thus, the algorithm of visualization of the conductivity field was presented. This algorithm makes it possible to display the reconstructed conductivity field of the examined ob-

ject on the screen. The algorithm envisages two approaches to calculating the colors of finite elements. The color-scheme approach involves using a pre-generated color matrix and matching the values of the conductivity field and this matrix. The second approach involves using dynamic determining the desired color, which includes red, green, and blue components. The advantage of the first approach is the high speed of color selection, of the second one – the accuracy of determining the necessary color.

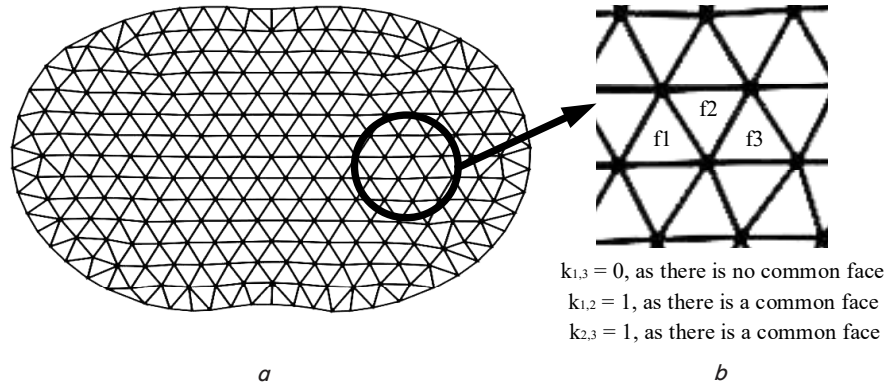


Fig. 7. The image of the grid of finite elements, into which the thoracic cavity projection is split: *a* – general view, *b* – enlarged area

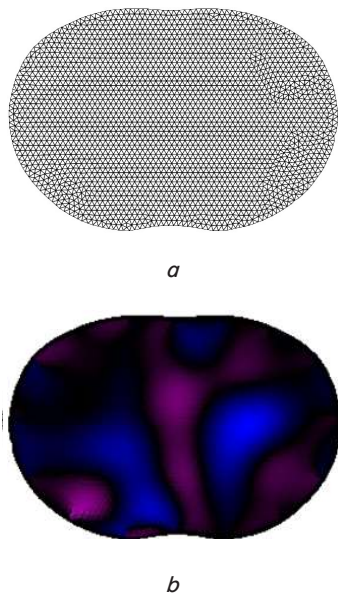


Fig. 6. Image of the model of the thoracic cavity: *a* – in the form of a grid of finite elements, *b* – during reconstruction of the conductivity field in the process of the operation of the EIT channel software

6. The algorithm of neighboring finite elements

To accelerate the display of the conductivity field, the authors proposed an algorithm of neighboring finite elements. This algorithm involves uniting the neighboring finite elements, in which color matches the specified accuracy, into enlarged polygons. It is required in order to reduce the number of polygons that need to be displayed on the screen, which allows accelerating the visualization process.

The practical implementation of the algorithm implies the execution of the following algorithm:

1) Let us assume that there is the thoracic cavity area GP, split into finite elements $f_i, i=1, \dots, m$, where m is the number of finite elements in the model. The matrix of the neighborhood of finite elements K , the lines, and columns of which are finite elements, is introduced. The element of the matrix is the neighborhood indicator $k_{i,j}$.

- $k_{i,j}=1$, if finite elements f_i and f_j have at least one common face;
- $k_{i,j}=0$, if finite elements f_i and f_j do not have any common face.

To explain the proposed approach visually, consider Fig. 7. Fig. 8, *a* shows the finite element model of the thoracic cavity cross-section. The following assumptions were made for the convenience of consideration:

- this model is universal and is not adapted to each individual patient;
- this model allows only two-dimensional reconstruction;
- this model does not consider the internal structure of the organs of a human body.

2) Precision factor ϵ , which reflects the parameter of how different shades of color can be visualized as a single color, is set. Because the color value in the RGB color model is in the range of $[0, 255]$, the value of this parameter, which is compared at step 3 with the difference between two finite elements in all three colors, is in the range of $[0, 765]$. The value of this indicator is indicated by a researcher at the start of the algorithm. At the same time, the following regularity is performed:

- at $\epsilon \rightarrow 755$, the number of finite elements in the model decreases significantly and the quality of the color display of the conductivity field of the examined object gets worse;
- at $\epsilon \rightarrow 0$, the number of finite elements in the model decreases insignificantly and the quality of the color display of the conductivity field of the examined object remains almost unchanged.

Precision factor $\epsilon=10$ was selected for this research.

3) Simultaneous satisfaction of two conditions is checked for each pair of finite elements f_i and f_j :

- $k_{i,j}=1$. This condition reflects the fact that neighborhood indicator must be equal to unity. That is, it is possible to unite only the finite elements that have a common face;

$$- (|r_i - r_j| + |g_i - g_j| + |b_i - b_j|) \leq \epsilon, \tag{8}$$

where $k_{i,j}$ is the neighborhood indicator of finite elements f_i and f_j ; ϵ is the precision factor; r_i, g_i, b_i are the values of colors of element f_i .

This condition shows that sum of differences of color values should not exceed the specified precision ϵ . That is, it is possible to unite only the elements that have the color matching the assigned precision.

4) If the conditions in p. 3 are met simultaneously, finite elements are united. Otherwise, they are not.

When uniting finite elements f_i and f_j the areas of finite elements (volume in the case of 3D reconstruction) are united, and the color of the united finite element is com-

puted as the arithmetic mean of the colors of its finite elements:

$$f_i = f_i \cup f_j; \tag{9}$$

$$r_i = \frac{r_i + r_j}{2}, \quad g_i = \frac{g_i + g_j}{2}, \quad b_i = \frac{b_i + b_j}{2}, \tag{10}$$

where r_i, g_i, b_i are the values of colors of element f_i .

Thus, we developed the algorithm that makes it possible to unite finite elements into one polygon if they are neighboring and have a similar color at assigned precision. The first step of this algorithm plots a matrix of neighborhoods of finite elements. Then, each time one calculates the conductivity field values, the neighboring elements of the conductivity field are checked for matching their color with a given accuracy. If the elements are matched, the neighboring elements are combined. This reduces the number of polygons that need to be visualized, which in turn should reduce the time it takes to display the model on the screen.

7. Results of comparison of graphics libraries

The implemented algorithm, principles of construction, and functioning, shown in Fig. 2 should be applied when forming software for the EIT device. The purpose of the experiment is to explore in more detail the graphics libraries specified in p. 2 to assess the feasibility of their application in developed software, taking into consideration previously stated criteria. To assess the performance of the graphics libraries, a number of experiments with the following conditions were conducted (Table 2).

Table 2

Parameters of the experiments on comparison of graphics libraries

Characteristic	Value
Number of visualized frames	10000
Programming language	C++, standard 2017
Processor	Intel Core i5-4460, 3.20 GHz
Graphics card	Intel HD Graphics 4600

Because the visualization process comes from the use of computing power either of a processor or a video card, RAM is not important for this study.

To determine an optimal graphics library, the P criterion, which is a function that combines visualization time and CPU loading, is introduced:

$$P = f(t_V, Q) \rightarrow \min, \tag{12}$$

where t_V is the total visualization time; Q is the CPU loading (%).

The results of the experiments on comparison of graphics libraries are shown in Table 3.

Parameter “Platform” was determined based on the operation systems, which are compatible with this library.

Parameter “Solution relevance: was determined based on the research into professional and scientific literature [10–13].

Table 3

Graphics libraries

Graphics library	Platform	Solution relevance	Time of model visualization without using the algorithm of neighboring finite elements $t_V(s)$	CPU loading Q (%)
GTK	Cross-platform	+	0.07	35
MFC	Windows	–	0.34	35
MFC+G-DIPLUS	Windows	–	1.12	35
MFC+OpenGL	Windows	+/-	0.05	4
Windows Forms+OpenGL	Windows	+	0.05	4

Parameter “Visualization time” was measured using the object «timer», implemented in the standard library C++. The time represented in the tables is the arithmetic mean for 10,000 visualized frames, which allows enhancing the objectivity of assessment of this parameter.

Parameter “CPU loading” was measured in percent of the maximum load using the resource monitor of the operating system Windows.

The null hypothesis, in this case, is the statement: “The time of model visualization does not depend on the use of a graphics library”. Let us take the significance level $\alpha=0.05$. The value, calculated by the Fischer-Snedecor criterion, is $F_{emp}=3.64$.

F_{cr} for the presented dataset of 10,000 frames takes the value $F_{cr}=1.01$. Since $F_{emp}>F_{cr}$, the null hypothesis is rejected, and one can conclude about the influence of the used graphics library on the time of model visualization.

The obtained results suggest that for medical visualization purposes, among the studied systems, it is optimal to use the OpenGL graphics library integrated into another graphics library implementing the user’s interface.

The resulting gain of 0.02 s due to the use of OpenGL makes it possible to display more frames per second, which is important in the case of real-time measurements. In addition, by transferring visualization to the GPU, the visualization speed increases and the CPU load decreases.

Experimental studies to assess the appropriateness of using the proposed algorithm of neighboring finite elements were conducted. The experiments with the model with the number of finite elements $n=4011$ were conducted.

In this study, the time of visualization using the finite elements algorithm t_{VF} is represented in the form of the following formula:

$$t_{VF} = t_R + t_O, \tag{13}$$

where t_R is the time of computation of the united finite elements (s); t_O is the time of the model visualization after the computation of the united finite elements (s).

Time t_{VF} is compared with the visualization time without using the finite elements algorithm t_V , obtained in Table 3.

The results of experiments over the implemented algorithm of neighboring finite elements showed that its use does not lead to a reduction in the total time spent (Table 4).

The obtained results show that the developed algorithm in all cases reduces the time of the model displaying on the screen. The time it took to compute the united finite elements is the same for all graphics libraries, because the used graphics library has no effect on this parameter. The total visualization time when using the algorithm in the case of applying graphics libraries, not involving the GPU, is smaller than the time to visualize the model without using the proposed algorithm. However, in the case of using the

Experimental results on the analysis of the algorithm of neighboring finite elements (Table 4) are explained by the fact that the time spent on uniting the finite elements is also quite a resource-intensive procedure. That is why its application gives benefits only in case of using graphics libraries that are less GPU resource-oriented.

EIT is a specialized field of tomography in which the results of the examination are reproduced in real-time, and consequently, the issues of rapid visualization of these results are very important. The algorithms

Table 4

Studying the algorithm of neighboring finite elements

Graphics libraries	Time of model visualization without using the algorithm of neighboring finite elements t_V (s)	Time of computation of united finite elements t_R (s)	Time of displaying model on the screen t_O (s)	Total visualization time using the algorithm of neighboring finite elements t_{VF} (s)
GTK	0.07	0.11	0.05	0.16
MFC	0.34	0.11	0.21	0.32
MFC+G-DIPLUS	1.12	0.11	0.88	0.99
MFC+OpenGL	0.05	0.11	0.03	0.14
Windows Forms+OpenGL	0.05	0.11	0.03	0.14

formed in this work make it possible to speed up the display of medical information on the screen and provide feedback with medical staff. The conducted study on comparison of graphics libraries is supported by experimental data and analyzes not only the time of direct display of finite elements on the screen but also the time of uniting the finite elements in case of using the proposed algorithm.

The merits of this study are its emphasis on dynamic visualization of the EIT examination. A series of software products for the EIT [20, 21] do not involve real-time reconstruction of the conductivity field in real-time. Unlike [3], this paper does not set the task of adapting the model to each individual patient, and

OpenGL graphics library, the proposed algorithm cumulatively requires more time than visualization without using it. Such results suggest that the developed algorithm can be used in systems that focus on CPU-oriented computation without using a GPU.

the principles outlined in [3] do not allow their application for dynamic visualization of the EIT. It can also be noted that the graphics libraries and their specifics for medical visualization were not compared in the reviewed papers [3–9].

The limitations of this study include the fact that a number of other visualization tools (VTK, SFML) were not studied. This is due to the fact that the study was conducted for medical equipment. That is why a series of graphics libraries rarely used in this area were not included in the research. In addition, only one hardware configuration to measure visualization time was considered in the framework of this study.

Further research may examine the development of the algorithm of neighboring finite elements. The possibility to avoid great time consumption during uniting finite elements may be ensured by the use of evolutionary algorithms, in particular, genetic algorithms to accelerate image visualization.

It is worth noting that the areas that are at the edges of this model and not taking part in the respiration process are not important for the EIT of lungs. It follows that these areas may not be redrawn every time one receives the data from the device but might be colored once when the measurement process starts. Since these areas are not important for ventilation and perfusion measurement, the indicators of field conductivity in them can be neglected.

8. Discussion of results of the study on comparison of graphics libraries

The proposed structure and algorithm of operation of the EIT channel software (Fig. 2, 3) was designed based on the requirements for speed and compatibility with medical technology. One of the key tasks of the EIT examination is to visualize the conductivity field, and the developed visualization algorithm (Fig. 5) makes it possible to implement two approaches to color selection for each finite element. Choosing one of these approaches leads to either higher speed of display, or higher image quality and depends on the needs of the conducted EIT examination. In any case, however, visualization of the model is a resource-intensive procedure and takes up most of the resources of a device when running an application. In this regard, we implemented the algorithm, the main idea of which is to unite the areas of the model having a similar color – the condition in formula (8), into one finite element – (9) and (10). This approach is explained by the fact that in the EIT examination, it is often important to separate key areas (lungs, heart), and heterogeneity within these areas is not so important.

The conducted experiments on comparison of the graphics libraries (Table 3), in which the OpenGL library showed the best result, are explained by the fact that this library is run at the expense of the GPU resources, which makes it possible to significantly reduce the load on the CPU.

9. Conclusions

1. We analyzed the structure of the EIT visualization software, allowing measurements of the differences in potentials of electrodes located on the patient’s body, computation of the conductivity field, and visualization of the tomographic section in real-time. The software op-

eration algorithm, enabling processing the measurement data obtained from several electrode belts, was developed. We developed the schematic representation of information units located in a software window, allowing outputting the processed measurement information in the software window and controlling the measurement process. The list of graphics libraries that can be used to visualize the results of electric impedance tomography was presented. The principles that are key principles for medical visualization were identified.

2. The algorithm of visualization of the results of conductivity field reconstruction was developed. This algorithm makes it possible to display on the screen the reconstructed conductivity field of the examined object. The algorithm implies two approaches to the computation of colors of finite elements. The color-scheme approach involves using a pre-generated color matrix and comparing the values of the conductivity field with that matrix. The second approach involves using dynamic determining the desired color, which includes red, green, and blue components. The advantage of the first approach is the high speed of color selection, of the second approach – the accuracy of determining the necessary color.

3. The algorithm of neighboring finite elements, which allows reducing the time spent on the visualization of the model by uniting neighboring elements with a similar color in one polygon, was proposed. At the first step of this algorithm, the matrix of the neighborhood of finite elements was plotted. Then, every time when one computes the values of the conductivity field, the neighboring elements of the conductivity field are checked for matching their color at a given accuracy. In the case of matching, the

neighboring elements are united. This makes it possible to reduce the number of polygons that need to be visualized, which in turn reduces the time taken to display the model on the screen.

4. Experiments on comparison of graphics libraries, their performance for the tasks of medical visualization were carried out. It was established that among the studied libraries, the shortest time was demonstrated by the OpenGL graphics library (0.05 s for visualization of a single frame). The obtained gain of 0.02 s (compared to GTK) makes it possible to display more frames per second, which is important in the case of real-time measurements. In addition, due to transferring visualization to the GPU, the visualization speed increases, and the CPU load decreases. The proposed algorithm of neighboring finite elements makes it possible to reduce the time of information outputting on the screen (from 0.05 s to 0.03 s for the OpenGL library), but at the same time increases the time of determining which finite elements should be united (this time is 0.11 s). In this case, applying libraries that mostly use CPU resources, this algorithm makes it possible to reduce the total time of visualization of the conductivity field.

Acknowledgments

This study is carried out within the framework of the federal target program “Research and development in priority areas of development of Russian scientific and technological complex for 2014–2020” with the financial support from the Ministry of Science and Higher Education (agreement No. 05.607.21.0305). Unique identifier of agreement RFMEFI60719X0305.

References

1. Pekker, Ya. S., Brazovskiy, K. S., Usov, V. N. (2004). *Elektroimpedansnaya tomografiya*. Tomsk: NTL, 192.
2. Draeger medical. Technical Data for PulmoVista 500 (2011).
3. Joldes, G., Bourantas, G., Zwick, B., Chowdhury, H., Wittek, A., Agrawal, S. et. al. (2019). Suite of meshless algorithms for accurate computation of soft tissue deformation for surgical simulation. *Medical Image Analysis*, 56, 152–171. doi: <https://doi.org/10.1016/j.media.2019.06.004>
4. Brown, A. C., Armstrong, G. S. J., Benda, J., Clarke, D. D. A., Wragg, J., Hamilton, K. R. et. al. (2020). RMT: R-matrix with time-dependence. Solving the semi-relativistic, time-dependent Schrödinger equation for general, multielectron atoms and molecules in intense, ultrashort, arbitrarily polarized laser pulses. *Computer Physics Communications*, 250, 107062. doi: <https://doi.org/10.1016/j.cpc.2019.107062>
5. Nikolić, D., Milošević, Ž., Saveljić, I., Filipović, N. (2016). Development of the software tool for generation and visualization of the finite element head model with bone conduction sounds. *AIP Conference Proceedings*, 1703. doi: <https://doi.org/10.1063/1.4939372>
6. Farhang, S., Foruzan, A. H., Chen, Y.-W. (2016). A real-time stable volumetric Mass-Spring Model based on a multi-scale mesh representation. 2016 23rd Iranian Conference on Biomedical Engineering and 2016 1st International Iranian Conference on Biomedical Engineering (ICBME). doi: <https://doi.org/10.1109/icbme.2016.7890950>
7. Zhao, J., Guoqiang, Y., Hui, L. (2017). Optimization of 3D Reconstruction Technique for CT Image and Analysis of the Derived Model. 22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). doi: <https://doi.org/10.1109/cse-euc.2017.176>
8. Otani, N., Dang, D., Beam, C., Mohammadi, F., Wentz, B., M Kamrul Hasan, S. et. al. (2019). Toward Quantification and Visualization of Active Stress Waves for Myocardial Biomechanical Function Assessment. 2019 Computing in Cardiology Conference (CinC). doi: <https://doi.org/10.22489/cinc.2019.425>
9. Courtecuisse, H., Jiang, Z., Mayeur, O., Witz, J. F., Lecomte-Grosbras, P., Cosson, M. et. al. (2020). Three-dimensional physics-based registration of pelvic system using 2D dynamic magnetic resonance imaging slices. *Strain*, 56 (3). doi: <https://doi.org/10.1111/str.12339>

10. Pozrikidis, C. (2007). Introduction to C++ Programming and Graphics. Springer, 372. doi: <https://doi.org/10.1007/978-0-387-68993-7>
11. Romero, J., Bisson, M., Fatica, M., Bernaschi, M. (2020). High performance implementations of the 2D Ising model on GPUs. Computer Physics Communications, 256, 107473. doi: <https://doi.org/10.1016/j.cpc.2020.107473>
12. Chotisarn, N., Merino, L., Zheng, X., Lonapalawong, S., Zhang, T., Xu, M., Chen, W. (2020). A systematic literature review of modern software visualization. Journal of Visualization, 23 (4), 539–558. doi: <https://doi.org/10.1007/s12650-020-00647-w>
13. Dehal, R. S., Munjal, C., Ansari, A. A., Kushwaha, A. S. (2018). GPU Computing Revolution: CUDA. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). doi: <https://doi.org/10.1109/icacccn.2018.8748495>
14. Prilozheniya MFC dlya rabochego stola. Available at: <https://docs.microsoft.com/ru-ru/cpp/mfc/mfc-desktop-applications?view=vs-2019>
15. The GTK Project. Available at: <https://www.gtk.org/>
16. OpenGL - The Industry Standard for high performance graphics. Available at: <https://www.opengl.org/>
17. Windows Forms. Available at: <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/windows-forms-overview>
18. Liang, C., Jiang, N., Yu, Y. W. (2011). Optimization and Implementation of the Surface Rendering Algorithm. Advanced Materials Research, 341-342, 152–157. doi: <https://doi.org/10.4028/www.scientific.net/amr.341-342.152>
19. Lu, L., Chen, C., Cheng, W. (2007). Medical image visualization using true 3D display technology. 2007 IEEE/ICME International Conference on Complex Medical Engineering. doi: <https://doi.org/10.1109/iccme.2007.4381873>
20. Liu, B., Yang, B., Xu, C., Xia, J., Dai, M., Ji, Z. Et. al. (2018). pyEIT: A python based framework for Electrical Impedance Tomography. SoftwareX, 7, 304–308. doi: <https://doi.org/10.1016/j.softx.2018.09.005>
21. EIDORS: Electrical Impedance Tomography and Diffuse Optical Tomography Reconstruction Software. Available at: <http://eidors3d.sourceforge.net>