AUTOMATYKA • 2009 • Tom 13 • Zeszyt 3

Tomasz Jaworski*, Radosław Wajman*

Graphical User Interface for Building the Spatial Definition of the Electrodes in 3D Electrical Capacitance Tomography

1. Introduction

The Electrical Capacitance Tomography (or ECT) [1, 3, 4, 6] is often used in industrial environment to obtain a permittivity distribution of materials inside the sensor. Such materials have to be dielectrics in order to change electric capacitance between sensor's electrodes. If materials inside the sensor are in two types only, it is possible to calculate physical quantity of both of them, for instance, a relationship between gas and liquid inside a pipeline.

To obtain a permittivity distribution, a specialized software with advanced image reconstruction algorithms must be used [5]. Moreover, the ECT tomography can be used to calculate flow rate of material inside a pipe or as a data acquisition system for more sophisticated applications, like automated control of technological processes, for instance in pneumatic transport lines. However, the most important feature of electrical tomography is the non-invasive nature of measurement process. Both sensor and electronics can gather data without any contact with medium inside the sensor.

The one of the most important challenge for the task of the image reconstruction algorithms for ECT is to solve the forward problem [2, 3, 7]. Basically, it is a simulation of the electrical field inside the ECT sensor. This task has two goals. One is to calculate the sensitivity matrix used for linearization of the image reconstruction process. The second one is to calculate the capacitance error for the current reconstructed image which in turn needs to be minimized during the next iteration. In order to solve the forward problem there is a need to build a computer model of the ECT sensor as accurate as possible. This is the difficult and usually time-consuming task. Normally, for generated three-dimensional tetrahedronal mesh there is a need to add the 3D ECT electrodes layout. In case of 3D ECT the shape and the position of the electrodes are irregular and the mesh has to be adjust to the electrodes layout and usually in consequence of this the shape of the tetrahedrons is destroyed. One of the tetrahedron's wall becomes incomparable to the rest of walls. It becomes thin and very elongated.

* Katedra Informatyki Stosowanej, Politechnika Łódzka

Unfortunately, there is no application which make possible to perform this tasks with mouse in any Graphical User Interface (GUI). Till this time it had to be done by writing a big piece of code in Matlab or in other programming language to have a non-universal algorithm specialized for one type of sensor. Therefore, the aim of presented work is to create a GUI application and methods useful in steps of ECT sensor development process, like:

– the development of the new spatial definitions of the electrodes in the simulated ECT sensor's geometry generated by an external application,
– the development of the sensors' layouts based on generated geometries or mask images for physical sensor implementations.

The application called **ECT Layout Builder** developed in this work maintains the speed and comfort of work regardless of either physical dimensions of the capacitance sensor or number of elements in the geometry mesh. Furthermore, this software ensures the high conformity of the built sensor with its computer model what is most important for the quality of the image reconstruction process.

2. The electrodes layout overview

A layout can be considered as a round virtual overlay placed on the vertical surface of a cylindrical sensor [1]. Its purpose is to hold and precisely arrange rings of electrodes and shield on the sensor's surface. Therefore, layout is a rectangular area with dimensions derived from the sensor's geometry and "wrapped" around the sensor.

Developed software helps in the design of a layout by providing a set of tools and real-time calculations of distances and dimensions of electrodes, gaps and shield. The software also takes into account the circular nature of the sensor's geometry and "unwraps" the 3D surface of any sensor (cylindrical or cuboids) on 2D screen to provide intuitive editing methods to the user.

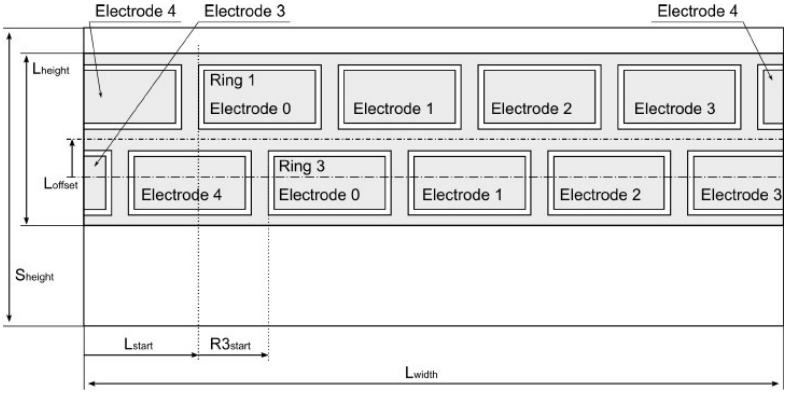

Fig. 1. Layout with specified all its dimensions

Figure 1 shows every dimension used when checking and calculating layout's parameters. Moreover, every of these dimensions can be modified by the user as a parameter of a ring or layout and they stand for as follows: L_{width} is a width value of the layout, L_{height} is a height value of the layout, S_{height} is a height value of the sensor/height of the geometry, L_{offset} is a vertical offset value of the layout, measured from sensor's horizontal middle line to the layout horizontal middle line, L_{start} is a horizontal offset value of the layout (common for every ring on the layout) and finally $R3_{start}$ is as example of individual horizontal offset for ring 3. It is not a standard parameter of a layout and it was placed here only for a better view. Considering the rectangular nature of the layout, its dimensions can be defined as follows:

$$\text{Width of the layout: } L_{width} = 2\pi S_{radius} \tag{1}$$

$$\text{Height of the layout: } L_{height} = S_{height} \tag{2}$$

Values L_{width} and L_{height} can be overridden by the user's own values but, by default, program suggests adjusting layout dimensions to sensor dimensions every time the new geometry is loaded. In other words, there is no need to use of sensor's geometry mesh to build a layout and in this case, the only condition to start a new layout is to setup the L_{width} and L_{height} values.

When the geometry of the sensor is available, values S_{radius} and S_{height} can be defined as follows:

$$S_{height} = \max\left(N_{z,i}\right) - \min\left(N_{z,i}\right) \tag{3}$$

$$S_{radius} = \frac{1}{2}\left[\max\left(N_{x,i}\right) - \min\left(N_{x,i}\right)\right] = \frac{1}{2}\left[\max\left(N_{y,i}\right) - \min\left(N_{y,i}\right)\right] \tag{4}$$

where $N_{x,i}$, $N_{y,i}$ and $N_{z,i}$ are X, Y, Z coordinates of i-node in sensor's geometry mesh and i is the node index and i = 1 .. number nodes.

2.1. Layout positioning

The layout can be positioned freely on surface plane with use of both L_{offset} and L_{start} parameters that are responsible of affecting respectively X and Y position on the plane. The maximal and minimal values of each of these parameters can be calculated as follows:

Vertical placement of the layout:

$$L_{offset_{MAX}} = \frac{1}{2}S_{height} - \frac{1}{2}L_{height}, \quad L_{offset_{MIN}} = -\left(\frac{1}{2}S_{height} - \frac{1}{2}L_{height}\right) \tag{5}$$

Horizontal placement of the layout:

$$L_{start_{MAX}} = 2\pi S_{radius}, \quad L_{start_{MIN}} = 0 \tag{6}$$

Please note, that the vertical position of the layout is based on the middle of the geometry. In spite of this, user can build a layout without using geometry but in such case, the values of both $L_{start_{MAX}}$ and $L_{start_{MIN}}$ are equal to zero.

2.2. Horizontal shield ring

The horizontal shield ring is one of two ring types that can be placed on a layout. The purpose of this ring is to provide customizable horizontal shield bar across perimeter of the sensor. Figure 2 shows horizontal shield ring ("*Ring n*+1") with one parameter.

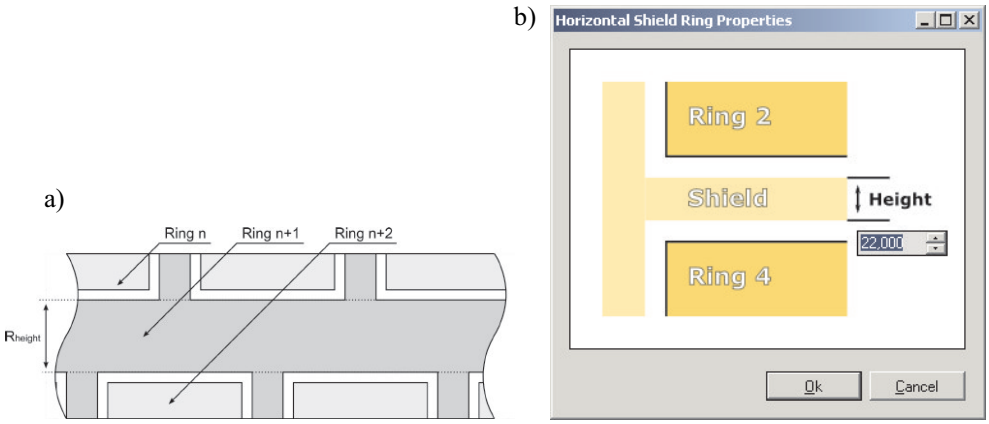

Fig. 2. Horizontal Shield Ring (a) and the 'Horizontal Shield Ring Properties' dialog window (b)

where R_{height} is a height value of the ring. In addition, minimal value of width for this ring type is always equal to zero. In this figure is also depicted the dialog window of the program for altering the parameter mentioned above.

2.3. Rectangular electrode ring

The rectangular electrode ring is the second ring type that can be placed on a layout. This ring type was designed to allow easy and precise placement and arrangement of electrodes on the layout. User has to specify number of electrodes, their dimensions and gaps between electrodes and shield.

The overview of this ring type is presented in Figure 3 with detailed information about dimensions and the suitable dialog window for changing this features is shown in Figure 4.

The G_{top} is a height value of the top gap, G_{bottom} is a height value of the bottom gap, G_{left} is a width value of the left gap, G_{right} is a width value of the right gap, E_{count} is a number of electrodes on the ring, E_{height} is a height value of the electrode, E_{width} is a width value of the electrode, E_{shield} is a width value (automatically calculated) of vertical shield bar be-

tween electrodes, R_{start} is a horizontal start offset of the ring measured from L_{start} (horizontal offset value of the layout). The R_{start} value is individual for each ring and allows to manually "rotate" the ring around the sensor. L_{start} is a horizontal offset value of the layout.

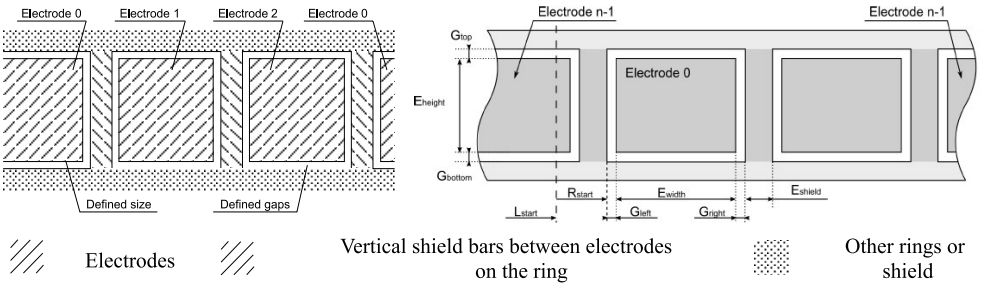

Fig. 3. The rectangular electrode ring – first view and its characteristics dimensions

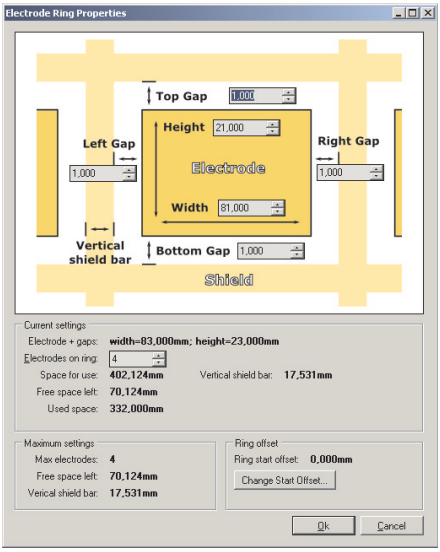

Fig. 4. The 'Electrode Ring Properties' dialog window

As it is shown in Figure 3, ring dimensions can be obtained from the following equations:

$$\text{Height of the ring: } R_{height} = G_{top} + E_{height} + G_{bottom} \tag{7}$$

Minimal width of the ring:
(Based on assumption that there is no vertical shield bars between the electrodes)

$$R_{width_{MIN}} = E_{count}\left(G_{left} + E_{width} + G_{right}\right) \tag{8}$$

Free space left after all electrodes are placed:

$$E_{space} = L_{width} - R_{width_{MIN}} \tag{9}$$

Width of every vertical shield bar between electrodes
(Number of the bars is equal to the number of electrodes):

$$E_{shield} = \frac{L_{width} - R_{width_{MIN}}}{E_{count}} \tag{10}$$

Maximal value of ring's start offset:

$$R_{start_{MAX}} = G_{left} + E_{width} + G_{right} + E_{shield} \tag{11}$$

Due to the periodic nature of this type of ring, values higher then $R_{start_{MAX}}$ would not have any effect.

The developed software allows to edit all these parameters and automatically recalculates them to show information like:

- maximum number of electrodes for the given dimensions and gaps,
- width of every vertical shield bar,
- total width of vertical shield bars,
- space left (on perimeter) for additional electrodes.

Changing any of the above parameters results in immediate update of Layout Editor view and Surface Grid Editor view.

3. Geometry transformation methods

Editing three-dimensional objects on a two-dimensional computer monitor is a quite difficult work to do, therefore it often requires agility and practice. To simplify and to make it more effective, specialized software has been developed, such as 3D Studio Max or Maya, which provide hundreds of functions, tools, many projections and views. In general, those applications are completely oriented on editing the spatial objects.

The software developed in this work also operates on data that represents three-dimensional objects like sensor's geometry or spatial distribution of the electrodes. However, objective of this application is not to edit whole sensor's geometry but only its surface. Taking into account i.e. the cylindrical shape of the sensor and the fact that only sensor's surface is used, it is decided to simplify editing process by moving it from three-dimensional to two-dimensional environment. To achieve this, the application has to unwrap the surface before using it and wrap it back to save possible changes made to the geometry. Both unwrapping and wrapping the surface algorithms are described in following sub-sections.

3.1. Unwrapping the surface

The purpose of the unwrapping algorithm is to extract nodes that are placed on the surface. Figure 5 shows how the surface is extracted from the geometry.

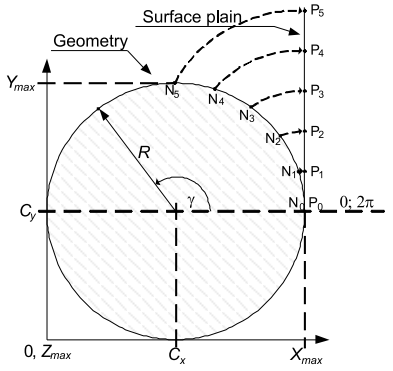

Fig. 5. Top view on the sensor's geometry

The C_x and C_y are the coordinates of the centre point of the geometry, R is the radius of the geometry, γ is an angle with marked rotation direction, $X_{\max}$, $Y_{\max}$, $Z_{\max}$ are maximal values calculated from the nodes vector, $N_0..N_5$ are the nodes in a three-dimensional space and finally $P_0..P_5$ are the surface points placed on two-dimensional surface. Nodes, after they are read from file can be considered as a matrix in the following form:

$$\mathbf{N} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}_{n,3} \tag{12}$$

where n is the total number of nodes in the geometry. The **N** matrix can be also described as a $\mathbf{n}_i$ vector with three components:

$$\mathbf{n}_i = (x : \mathbf{N}_{i,1}; y : \mathbf{N}_{i,2}; z : \mathbf{N}_{i,3}) \tag{13}$$

where $\mathbf{n}_i$ is the i-node.

3.2. Surface points extraction

A surface point is a node, for which R is a distance between the node and the geometry center point what can be described as follows:

$$\mathbf{P}^{3D} = \left\{ \mathbf{n}_i : \left| d_{Cxy}(\mathbf{n}_i) - R \right| \le \varepsilon \right\} \tag{14}$$

where $\mathbf{P}^{3D}$ is the set of surface points, but still located in three-dimensional space, d_{Cxy} is a function of distance with one point fixed on geometry center point and ε is empirically chosen accuracy value.

The function of distance used in equation (14) is defined as follows:

$$d_{Cxy}(q) = \sqrt{\left(C_x - q_x\right)^2 + \left(C_y - q_y\right)^2} \tag{15}$$

The center point of the geometry:

$$C_x = \frac{1}{2}\left(\max_i(\mathbf{n}_{i,x}) + \min_i(\mathbf{n}_{i,x}) \right) \tag{16}$$

$$C_y = \frac{1}{2}\left(\max_i(\mathbf{n}_{i,y}) + \min_i(\mathbf{n}_{i,y}) \right) \tag{17}$$

The accuracy value has been set to $\varepsilon = 1 \cdot 10^{-4}$ mm. This is 0.1 of the ECT Layout Builder's resolution which has been fixed on 1 μm.

3.3. The 2D to 3D transformation

The points in vector $\mathbf{P}^{3D}$ have to be transformed in order get plain 2D surface. For this reason, the following equation has been used:

$$\mathbf{P}^{2D}_{i=1..\left|\mathbf{P}^{3D}\right|} = \left(x : f_{unwrap}\left(\mathbf{P}^{3D}_i\right) L_{width};\ y : \mathbf{P}^{3D}_{i,z} \right) \tag{18}$$

where x and y are the coordinates of the surface point, $\mathbf{P}^{2D}$ is the vector of surface points after unwrapping transformation. This vector is the main data source for rendering routines, L_{width} is the layout width value and f_{unwrap} is unwrapping function, which is defined as follows:

$$f_{unwrap}(p) = \begin{cases} 1 - \dfrac{1}{2\pi} f_{angle}(p) & \text{when } \mathrm{p}_y < 0 \\ \dfrac{1}{2\pi} f_{angle}(p) & \text{otherwise} \end{cases} \tag{19}$$

Function f_{unwrap} uses a helper function f_{angle} that returns angle in range $\langle 0; \pi \rangle$:

$$\gamma = f_{angle}(p) = \operatorname{acos}\left(\frac{R\left(p_x - C_x\right)}{R\left\|p\right\|} \right) = \operatorname{acos}\left(\frac{p_x - C_x}{\sqrt{\left(p_x - C_x\right)^2 + \left(p_y - Cy\right)^2}} \right) \tag{20}$$

Values γ and R are shown in Figure 9. At this point, vector $\mathbf{P}^{2D}$ contains XY locations of each surface node, called from now the Surface Points. Values from this vector are used directly to render the screen.

3.4. Wrapping the surface back

Wrapping back the surface transformation is used for writing altered nodes to disk. In order to do this, all surface points have to be transformed back to three-dimensional space. This operation can be done with the following two equations:

$$\text{For X-axis: } \mathbf{P}'^{3D}_{i=1..\left|\mathbf{P}^{3D}\right|,x} = C_y + R\cos\left(2\pi\frac{\mathbf{P}^{2D}_{i,x}}{L_{width}}\right) \tag{21}$$

$$\text{For Y-axis: } \mathbf{P}'^{3D}_{i=1..\left|\mathbf{P}^{3D}\right|,y} = C_x + R\sin\left(2\pi\frac{\mathbf{P}^{2D}_{i,x}}{L_{width}}\right) \tag{22}$$

To obtain the full three-dimensional information an equation for Z-axis is needed, which is a simple assignment, as follows:

$$\mathbf{P}'^{3D}_{i=1..\left|\mathbf{P}^{3D}\right|,z} = \mathbf{P}^{3D}_{i,z} \tag{23}$$

where $\mathbf{P}'^{3D}$ is the vector with transformed surface points back to nodes (to three-dimensional space).

$\mathbf{P}'^{3D}$ vector contains only those nodes, which were obtained in equation (14). Therefore, to save all nodes to a file, $\mathbf{P}'^{3D}$ vector has to be combined with $\mathbf{n}_i$ vector by replacing nodes in $\mathbf{n}_i$ with corresponding nodes from $\mathbf{P}'^{3D}$.

If $\mathbf{n}_i$, $\mathbf{P}^{3D}$ and $\mathbf{P}'^{3D}$ are treated as sets, the following equations will define the replacing operation:

$$\mathbf{n}' = \left(\mathbf{n} \setminus \mathbf{P}^{3D}\right) \cup \mathbf{P}'^{3D} \tag{24}$$

The $\mathbf{n}'$ vector (or set) can be now saved to file for future use.

4. The GUI elements

4.1. The Layout editor

The **Layout Editor,** also called *Layout View*, is a one of two editors and it is shown by default after the ECT Layout Builder program starts. Layout editor is the main tool for managing the layout, its parameters and rings inside of it. Moreover, the editor also allows user to save and load layout definitions from disk. Figure 6 shows the Layout Editor with several rings already added

Using this editor it is possible to set up the new electrodes layout, to load and save the layout, to alter its features and to change its content like adding new and removing existing ring, rearranging the rings order on the layout and changing the Vertical Offset and the Start Offset of the layout. This editor provides also the tool for generation of the mask image.

This function automatically generates the mask images as a simple bitmap file in different formats chosen by the user. These images are high-resolution versions of user-defined layout. Therefore, they consist of horizontal and vertical shield bars, electrodes, gaps and in addition, of technological margins. User can alter some aspects of the output image, what is described in the following text. The output images can be used for etching process to obtain physical versions of the layout as a copper laminate, which can be used to create tubular sensors for electrical capacitance tomography. Figure 7 shows the mask of the layout prepared according to the one of the ECT sensors designed and manufactured in Tomography Laboratory of Computer Engineering Department.

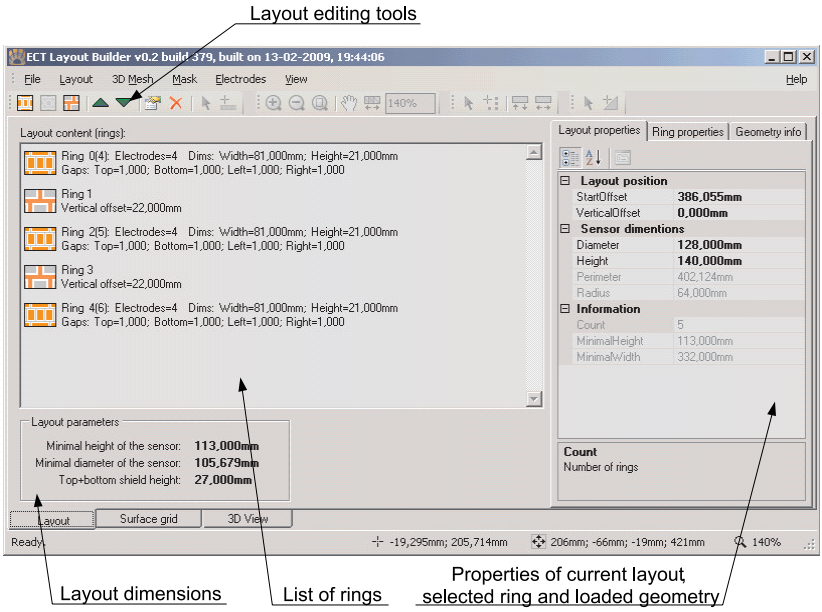

Fig. 6. Layout Editor with sample rings

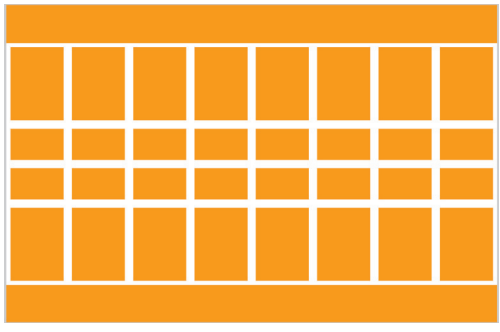

Fig. 7. Sample image mask of the 3D ECT sensor layout

4.2. The Surface Grid editor

The **Surface Grid Editor** is a CAD-like editor, used to rearrange the surface points on the mesh to create desired electrodes definition based on the layout. Those adjustments are usually a short-distance corrections of some points in such a way that the surface triangles based on those points and covered by layout's electrodes[1] could create shapes similar to those electrodes. Only triangles that have all three points covered by the same layout's electrode can be used in a definition. Those shapes, combined together represent the definition of the electrodes.

In the current version of the application, the Surface Editor allows to insert new surface points as well as to move existing one on the surface around the sensor. Figure 8 shows the Surface Grid Editor window.

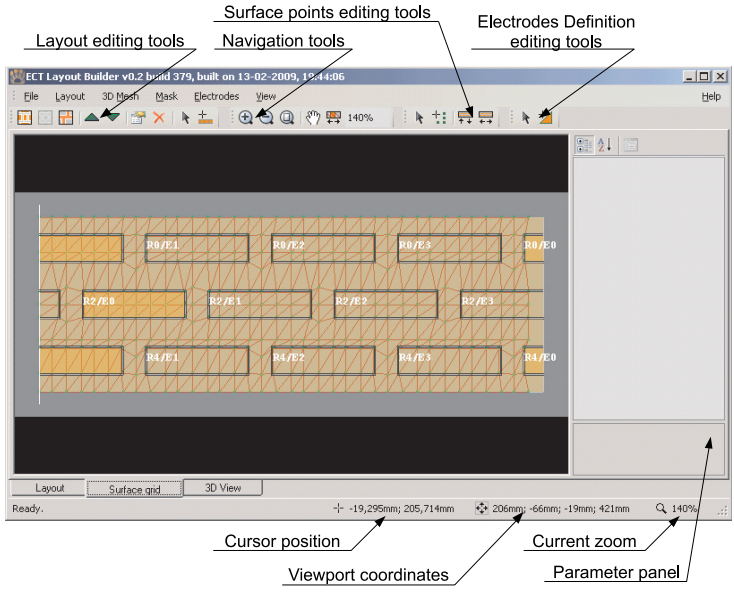

Fig. 8. Surface Grid Editor with sample mesh and layout

Surface Grid Editor consists of the following tools:

- **Mesh loading, unloading and saving tools** allow load 3D mesh generated by other applications and save modified mesh for further processing by other applications.
- **Layout editing tools** allow to adjust content of the layout from the editor. Layout altering commands available in this window are the same as for Layout Editor. Editor also provides additional tool for selecting a ring with mouse.

[1] Rectangular area defined on a sensor's surface by a layout, described by position and dimension

– **Navigation tools** used to zoom in the surface in order to change their position more accurately, zoom out to see more points, pan the viewport window around and rotate viewport around the sensor.
– **Surface points editing tools** allow to adjust positions of selected points with use of mouse or keyboard as an input. In addition, two tool buttons allow to change layout start and vertical offset with mouse pointer.
– **Surface points adding tools** allow to add the new point on the surface of the sensor and to divide the existing surface tetrahedrons into five smaller tetrahedrons as is shown in Figure 9 below with red dashed lines.

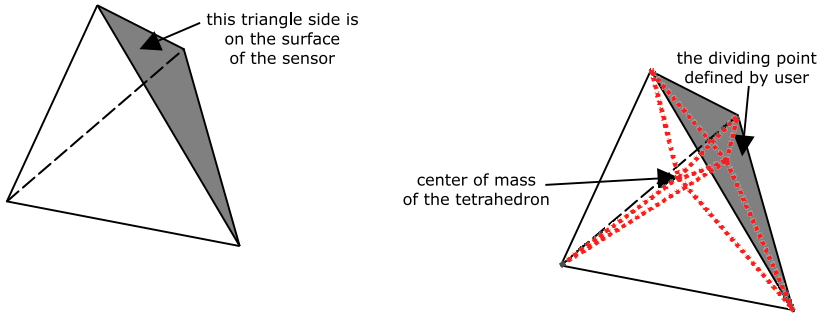

Fig. 9. The way of dividing the surface tetrahedron after adding one surface point

– **Electrodes Definition tools** that allow to select a surface triangle and modify its assignment to a ring or to an electrode.
– **Indicator bar** that shows current mouse cursor position, viewport window coordinates and zoom level.

4.3. The 3D model view

Based on the nature of input data, the ECT Layout Builder application provides the functionality to preview a sensor's geometry in 3D space. The model view functionality has been developed especially to allow user to preview his work, at least approximated, in 3D environment. Both the sensor's geometry and the electrodes distribution are, by nature, three-dimensional, therefore they can be easily shown with use of OpenGL.

To access the 3D view, user has to select third tab, named *3D View*, from the bottom left corner of the Builder's main window. As a result, the window as in Figure 10 appears.

In this view the currently displayed number of details can be altered by setting the states of four available view modifiers like:

– base and axes i.e. X, Y, Z-axis and gray square as the base for a geometry,
– all surface points as white dots,
– surface triangles as green lines,
– surface triangles marked as a part of a electrodes definition, which are drawn as triangles filled with orange color.

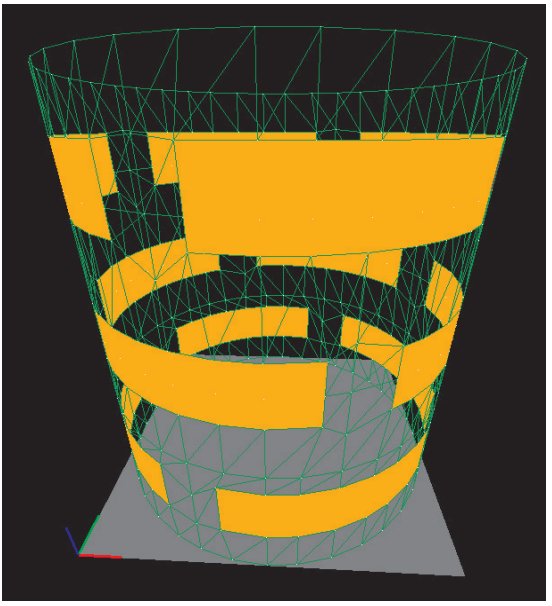

Fig. 10. 3D Model View and the view modifiers i.e. Base + Points + Triangles + Definition

5. Conclusions

The ECT Layout Builder application was designed and described in this paper. It allows to create and edit electrodes layout and its mask image as stand-alone products, without any relationship with the sensor's geometry. However, when a sensor's geometry is loaded, the layout becomes a base and a template for modifications of surface points' positions and further, for generating the spatial electrodes definition. The application is fully compatible with any file formats for geometry data, like for instance, NETGEN [1] or WinRECO [5] files.

The two methods: for transforming three-dimensional sensor's surface into two-dimensional rectangular plain area (called unwrapping process) and for transforming two-dimensional surface back to three-dimensional space was described as a part of this work. Bringing the editing problem from 3D space down to 2D area allows to use easy, convenient and intuitive methods for editing the set of spatial nodes located on the sensor's surface.

Used technologies, like C# for interacting with user, C++ for data preprocessing, numerical calculations and OpenGL for surface renderings allowed to create a GUI software convenient to use. Thanks to designed transform methods even very large geometries, which reach level of 400 MB data (and more than 10 million of simplexes), are preprocessed in less than 30 seconds and produces no perceptible delays during the work with the application.

Literature

[1] Banasiak R., *Algorytmy wizualizacji 3D w przemysłowych systemach elektrycznej tomografii procesowej*. Politechnika Łódzka, Łódź, 2007 (Rozprawa doktorska).

[2] Romanowski A., Grudzień K., *Zestawienie metod rekonstrukcji obrazów i modelowania przestrzennego 2D dla tomografii procesowej*. Automatyka (półrocznik AGH), t. 9, z. 3, 2005, 621–630.

[3] Soleimani M., *Three-dimensional electrical capacitance tomography imaging, Insight*. Non-Destructive Testing and Condition Monitoring, vol. 48, No. 10, 2006, 613–617.

[4] Wajman R., Banasiak R., Mazurkiewicz L., Dyakowski D., Sankowski D., *Spatial imaging with 3D capacitance measurements*. Meas. Sci. Technol., 17, No. 8, 2006, 2113–2118.

[5] Wajman R., *Nowa metoda rekonstrukcji obrazów dla potrzeb pojemnościowej tomografii procesowej*. Politechnika Łódzka, Łódź, 2006 (Rozprawa doktorska).

[6] Warsito W., Marashdeh Q., Fan L.S., *Electrical capacitance volume tomography*. IEEE Sensors Journal, vol. 7, Issue: 3–4, 2007, 525–535.

[7] Yang W.Q., PenG L., *Review of image reconstruction algorithms for electrical capacitance tomography, Part 1.* Principles, proc International Symposium on Process Tomography in Poland, Wroclaw, 2002, 123–132.