

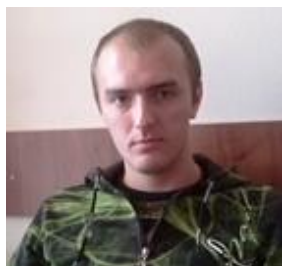
УДК 612.089.67:004.38

## СИСТЕМА ПОСТРОЕНИЯ ГЕОМЕТРИЧЕСКИХ МОДЕЛЕЙ ЛИЦЕВЫХ ИМПЛАНТОВ



**В.С. Осипович**

Доцент кафедры инженерной психологии и эргономики БГУИР, кандидат технических наук, доцент



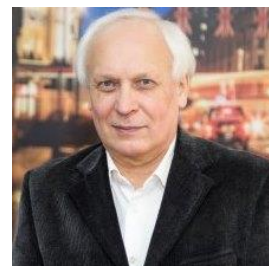
**А.Ю. Николаев**

Ассистент кафедры инженерной психологии и эргономики БГУИР, магистр техники и технологий наук, аспирант



**А.Л. Раднёнок**

Ассистент кафедры инженерной психологии и эргономики БГУИР, магистр технических наук, аспирант



**К.Д. Яшин**

Заведующий кафедрой инженерной психологии и эргономики БГУИР, кандидат технических наук, доцент

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь  
E-mail: seth22@yandex.ru

**Аннотация.** Реализована возможность на основе комплекта выходных файлов рентгеновского компьютерного томографа получения векторного файла контура повреждения костей для изготовления индивидуального импланта.

**Ключевые слова:** трехмерная модель, лицевой имплант, векторный файл, развертка поверхности трехмерной модели

**Введение.** Благодаря своей высокой точности, производительности и универсальности решаемых задач информационные технологии нашли широкое применение в медицине и, в частности, в лицевой хирургии. Цифровые технологии используются на всех этапах лечения. Существуют системы автоматизированного заполнения и ведения различных форм медицинской документации. В этих системах помимо автоматизации работы с документами присутствует функция моделирования на экране конкретной клинической ситуации и предлагаемого плана лечения стоматологических пациентов. Компьютерная обработка графической информации позволяет быстро обследовать пациента и показать его результаты как самому пациенту, так и другим специалистам. В настоящее время разработаны разнообразные внутриротовые цифровые фото- и видеокамеры. Такие приборы легко подключаются к персональному компьютеру и просты в использовании. Для рентгенологического обследования все чаще используются компьютерные радиовизиографы. Новые технологии позволяют минимизировать вредное воздействие рентгеновских лучей и получить более точную информацию. Таким образом возникает необходимость в разработке все более новых программных средств, которые бы смогли улучшить и автоматизировать некоторые процессы, которые выполняются врачами повседневно.

Научный и практический интерес представляет разработка системы для автоматизации построения геометрических моделей лицевых имплантов. Это позволяет врачам больше сконцентрироваться на проведении операций и облегчить создание индивидуальных имплантов для пациентов.

**Результаты разработки.** Для построения геометрических моделей проведем компьютерную томографию черепа. Она определяет качество стереолитографических моделей, а они

в свою очередь – соответствие индивидуальных имплантатов анатомии человека. По данным компьютерной томографии построим трехмерные реконструкции костных структур и мягких тканей. После этого, в приложении используем алгоритм, приведенный

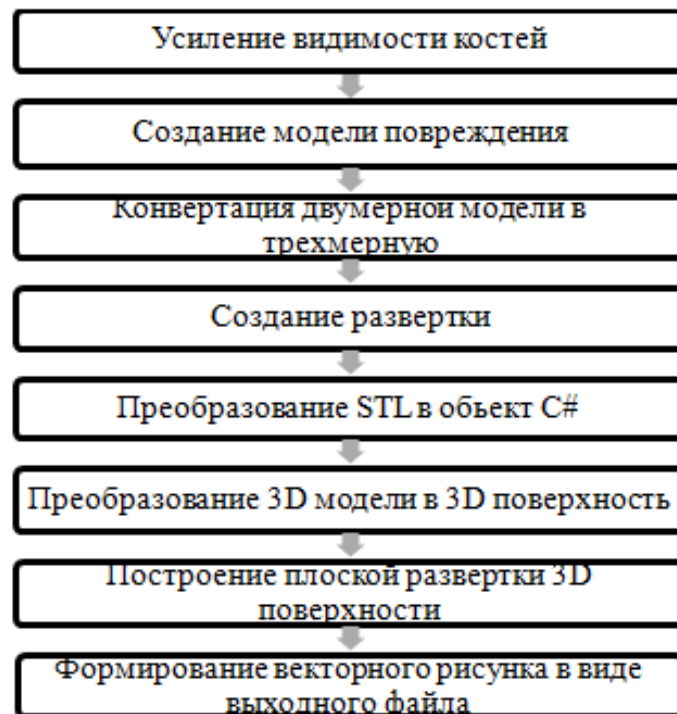


Рисунок 1. Алгоритм построения развертки

Для достижения цели разработаем алгоритм построения развертки из трехмерной модели. За счет автоматизации процесса построения лицевых имплантов возрастает скорость их создания, а также качество. Распечатанная на 3D принтере модель повреждения помогает наглядно изучить моделирование имплантов и методы лечения с их помощью.

Для усиления видимости костей применяется алгоритм добавления пикселей в исходные файлы рентгеновского компьютерного томографа (DICOM файлов). При этом, они того же оттенка серого цвета, так как при проведении томографии используются только оттенки серого и белого цвета для обозначения местоположения костей и мягких тканей.

Создание модели повреждения проводится после предварительной обработки DICOM файла, в результате получается STL файл, содержащий информацию о расположении вершин треугольников, а также их нормалей, которые вместе составляют видимую модель повреждения черепа человека [1].

Для конвертации двумерной модели в трехмерную производится поиск видимой границы будущего импланта. Из набора треугольников, представленных в STL файле выделяются крайние по двум из осей координат, после этого по формуле (1) находятся точки, лежащие выше прямой, образованной соединением двух соседних точек в плоскости, после нахождения, алгоритм повторяется, пока не будут найдены все точки, образующие контур модели импланта.

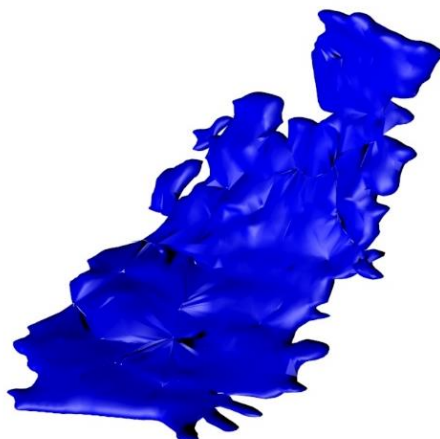


Рисунок 2. Трехмерная модель повреждения

$$\frac{x - x_1}{x_2 - x_1} - \frac{y - y_1}{y_2 - y_1}, \quad (1)$$

где:  $x_1, y_1$  и  $x_2, y_2$  – координаты точек, образующих прямую;  $x, y$  – координаты проверяемой точки на плоскости [2].

Если результат меньше нуля, то точка находится выше прямой и подходит для продолжения алгоритма поиска.

Затем заполним пространство внутри контура, для этого выбирается треугольник и методом сравнения соединений вершин, проверяется каждый треугольник на принадлежность нужной плоскости. Алгоритм повторяется до тех пор, пока не будет достигнут контур.

Преобразование STL в объект происходит средствами сторонней библиотеки IxMilia.Stl, которая свободно распространяется и имеет открытый исходный код. Она позволяет изменять, создавать, добавлять и удалять элементы из трехмерной модели STL из программного кода.

Полученную 3D модель в виде объекта C# разрезаем по одной из плоскостей с интервалом 0,5 мм. Для этого находим все треугольники, лежащие в плоскости разреза и находим все пересечения треугольников в заданной плоскости. После алгоритм повторяем до тех пор, пока не будет достигнут конец модели. Например, имеем произвольный треугольник в пространстве, обозначим его вершины – ABC. Вершины треугольника имеют координаты A ( $x_A: y_A: z_A$ ), B ( $x_B: y_B: z_B$ ) и C ( $x_C: y_C: z_C$ ) [3]. Режем треугольник плоскостью параллельной плоскости XY с шагом  $a$ . Координата  $z$  точек пересечения плоскости со сторонами треугольника ABC рассчитываем следующим образом.

$$z_i = z_{i-1} + a, \quad (2)$$

где:  $z$  – координата плоскости;  $a$  – шаг продвижения по модели.

Если плоскость пересекает сторону AC треугольника в точке F ( $x_F; y_F; z_F$ ), то координаты  $x_F$  и  $y_F$  этой точки рассчитываем по следующим формулам:

$$x_F = \frac{z_F - (z_A - \frac{x_A(z_C - z_A)}{x_C - x_A})}{\frac{z_C - z_A}{x_C - x_A}}, \quad (3)$$

$$y_F = \frac{z_F - (z_A - \frac{y_A(z_C - z_A)}{y_C - y_A})}{\frac{z_C - z_A}{y_C - y_A}}, \quad (4)$$

Координаты точек пересечения с другими сторонами треугольника рассчитываем аналогичным образом. Для расчёта берём координаты другой пары точек, ограничивающие другую сторону треугольника [4, 5].

При этом, если рассчитывая координаты точки пересечения со стороной, получим одну из координат искомой точки по значению больше, чем координаты вершин стороны треугольника, это значит, что плоскость не пересекает рассматриваемую сторону.

После нахождения всех точек пересечения плоскостей, отсекаем верхнюю часть среза. Для этого находим среднюю линию среза. По методу скользящего среднего (5), используя формулу на всех интервалах поочередно, построим динамически изменяющуюся линию, которая разделяет верхнюю часть среза от нижней.

$$SMA = \frac{\sum_{i=1}^n P_i}{n}, \quad (5)$$

где:  $P_i$  – точка на заданном интервале;  $n$  – количество точек.

Для того, чтобы отсечь верхнюю часть, используем формулу (1). В итоге получаем набор точек, необходимых для получения развертки. Применив формулу (6), найдем длину образовавшейся ломаной линии.

$$A = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (6)$$

где:  $x$  и  $y$  – координаты точек на плоскости.

Зная длину кривой и начальные координаты, которые были известны из трехмерной модели, нарисуем развертку. Для этого каждую линию располагаем по координатам исходной модели и шагом, с которым производились срезы [6].

Таким образом будет получена развертка нужной нам трехмерной модели.

Для формирования векторного рисунка используем встроенную XML библиотеку, которая позволяет создавать SVG файлы, необходимые для решения задачи. При построении линии используем тег `line`, в котором указываем толщину линии, равную шагу выполнения срезов, начальную точку и конечную  $x$  и  $y$ .

В результате проведенных операций, получаем развертку из DICOM файла [7].



Рисунок 3. Развертка 3D модели повреждения

Таким образом разработана система построения геометрических моделей лицевых имплантов на основе выходных файлов рентгеновского компьютерного томографа.

#### **Список литературы**

- [1] Ли Дж. Трёхмерная графика и анимация. — 2-е изд. — М.: Вильямс, 2002. — 640 с.
- [2] Херн Д. Компьютерная графика и стандарт OpenGL. — 3-е изд. — М., 2005. — 1168 с.
- [3] Энджел. Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL. — 2-е изд. — М.: Вильямс, 2001. — 592 с.
- [4] Снук, Г. 3D-ландшафты в реальном времени на C++ и DirectX 9. — 2-е изд. — М.: Кудиц-пресс, 2007. — 368 с.
- [5] Иванов, В. П., Батраков А. С. Трёхмерная компьютерная графика / Под ред. Г. М. Полищука. — М.: Радио и связь, 1995. — 224 с.
- [6] Мэтью Мак-Дональд. WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов, 4-е издание. — М.: «Вильямс», 2013. — 1024 с.
- [7] Мэтью Мак-Дональд. WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C# 2010 для профессионалов. — М.: «Вильямс», 2011. — 1024 с.

### **FACIAL IMPLANTS GEOMETRIC MODELS BUILDING SYSTEM**

**V.S. OSIPOVICH, PhD**  
Associate professor of  
department of  
Human Engineering  
and Ergonomics

**A. NIKOLAEV**  
Engineer of Human  
Engineering  
and Ergonomics,  
Master of Technical  
Science

**A. RADNIONOK**  
Professor assistant of  
Human Engineering  
and Ergonomics De-  
partment of BSUIR,  
Master of Technical  
Science

**K.D. YASHIN, PhD**  
Head of the Depart-  
ment of Engineering  
Psychology and Ergo-  
nomics BSUIR, Asso-  
ciate Professor

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus  
E-mail: seth22@yandex.ru

**Abstract.** The possibility of obtaining a vector file of bone damage contour for the manufacture of an individual implant based on a set of output files x-ray computed tomography was realized

**Key words:** three-dimensional model, facial implant, vector file, 3D surface sweep.