Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра системного программирования

Моделирование установки имплантатов на воксельной модели лица

Курсовая работа студента 444 группы
Фасхитдинова Рамиля Равильевича

Научный руководитель Петров А.Г.

Санкт-Петербург
2014

Содержание

Введение. Постановка задачи.

Вследствие прогресса в технологии сканирования и доступности сканеров, в современном мире наблюдается потребность в обработке трехмерных моделей. В частности, данная работа имеет свое применение в отрасли пластической хирургии.

Цель данной курсовой работы состоит в разработке инструмента для моделирования внешности пациента после установки лицевого имплантата.

Подробнее, к инструменту предъявляются следующие требования:

1. Преобразование полигональной модели в воксельную.
2. Добавление к произвольной области модели заданного объема.
3. Описание физической зависимости в воксельном представлении модели.
4. Преобразование воксельной модели в полигональную.
5. Приведение полученной полигональной модели к стандартному формату, т.е. соблюдение топологической связности, гладкости и правильной триангуляции полигональной сетки.
6. Наложение текстуры деформируемой области на деформированную.

Обзор существующих решений.

На данный момент пластические хирурги используют следующие инструменты для решения подобного рода задач:

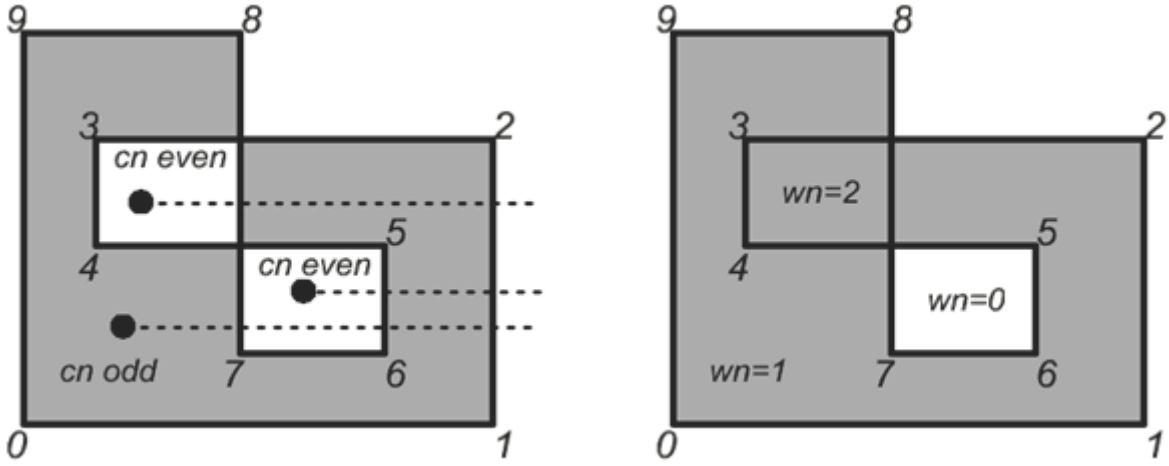
- Программы для воксельного скульптурирования, такие как Sculptris и 3DCoat. Данные программы основаны на воксельном представлении модели с поддержкой рисования текстур и динамической тесселяции, позволяющей детально проработать дизайн модели.
  Недостатком этих инструментов в контексте моделирования пластических операций является невозможность производить вычисления.
- Математические методы вычислений, более известные как направление вычислительной математики computational surgery. Данные методы предоставляют высокую степень точности вычислений, однако для ее достижения накладываются значительные требования на входные данные. Вследствие этого они сложны в применении к моделированию внешности на основе одних лишь трехмерных снимков пациента. Еще одним их недостатком является большое время рассчета.

Таким образом цель данной работы состоит в разработке инструмента, который был бы также удобен в использовании, как программы воксельного скульптурирования, и позволял бы производить необходимые вычисления в реальном времени.

Теоретическая основа курсовой работы.

Описание метода моделирования.

Так как требуется вычислять объем поверхностей, для моделирования будет использовано воксельное представление трехмерной модели, значительно упрощающее процесс вычисления по сравнению с полигональной моделью. Процесс поверхностной вокселизации и метод произведения объемных вычислений описан в курсовой работе “Моделирование и параметризация изменения внешности на 3D модели лица”.
Однако для моделирования установки имплантатов необходима также вокселизация внутренней полости полигональной модели, которая производится после поверхностной. Для этого рассматриваются сечения воксельной решетки одной из координатных плоскостей, критерием принадлежности полости является нечетность точек пересечения:



Для моделирования физической зависимости между деформируемыми тканями на полученную модель будет наложена модель масс с пружинками, т.е. каждый воксель имеет массу и считается связанным с соседними вокселми пружинами. Таким образом зависимость будет задаваться матрицей масс и коэффициентов жесткости пружин.
Следующим этапом метода является обратное преобразование воксельной модели в полигональную. Проведя обзор алгоритмов аппроксимации поверхностей, заданных в виде трехмерных скалярных полей, среди алгоритмов Канейро, MT6, Скалы и Марширующих кубов был выбран последний, как генерирующий наименьшее количество треугольников, в среднем 2 треугольника на куб.
Тем не менее, для ускорения дальнейшей обработки и визуализации количество треугольников должно быть уменьшено в рамках допустимой погрешности в объеме модели.
Для сглаживания полученной полигональной сетки использовался алгоритм Таубина, сохраняющий объем сглаживаемой поверхности.

Алгоритм марширующих кубов.

Алгоритм марширующих кубов состоит из двух стадий:

1. Поиск ячеек воксельной решетки, пересекаемых искомой поверхностью
2. Аппроксимация поверхности в найденных ячейках треугольниками.

Ввиду применения объемной вокселизации, т.е. заполнения внутренних полостей модели вокселями, первый этап сводится к поиску граничных заполненных вокселей.

Рассмотрим варианты аппроксимации куба треугольниками. Каждая его вершина может принадлежать объему, ограниченному искомой поверхностью, либо находиться вне его. Таким образом получаем 2^8 = 256 вариантов. Таким образом можно сопоставить каждому граничному вокселю 8-ми битовый индекс, задающий конфигурацию расположения треугольника. После чего остается только вычислить координаты вершины на ребрах куба через линейную интерполяцию соседних вершин.

Однако заметим, что среди этих вариантов есть одинаковые с точки зрения генерируемых полигонов:

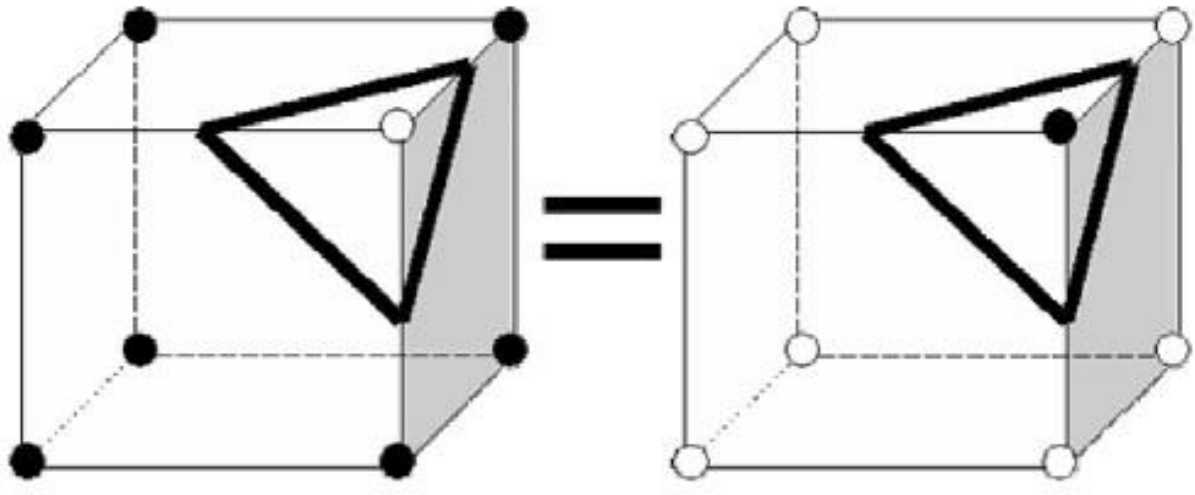
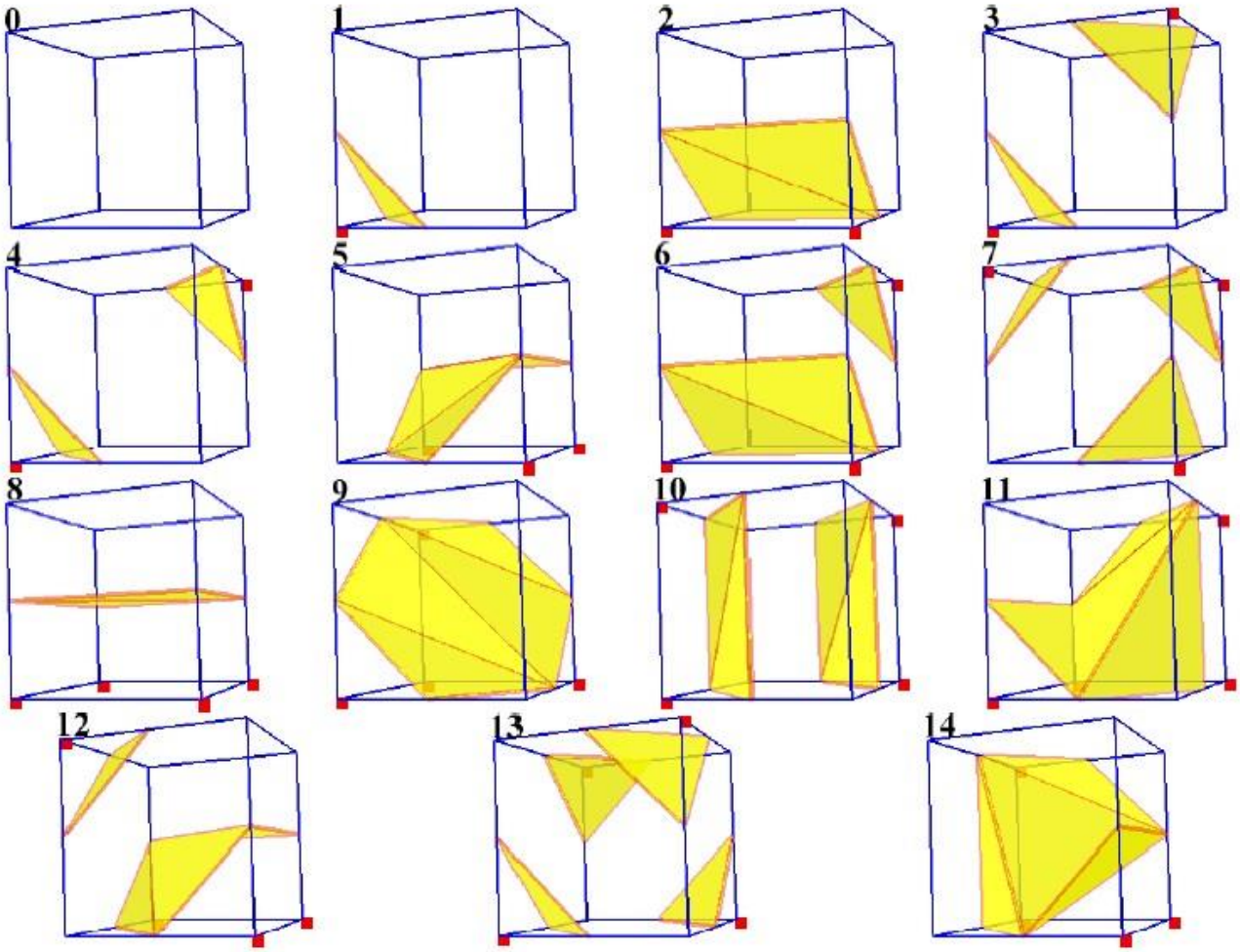


Рис. Повторяющиеся варианты полигонизации (белые вершины – вне объема, черные – внутри объема)

Учитывая получающиеся друг из друга симметрией и вращением расстановки треугольников, 256 вариантов сокращаются до 15 основных:



Однако такой алгоритм не гарантирует топологической корректности полученной сетки, так как при выборе положения треугольников в каждом из оставшихся случаев существует неоднозначность:

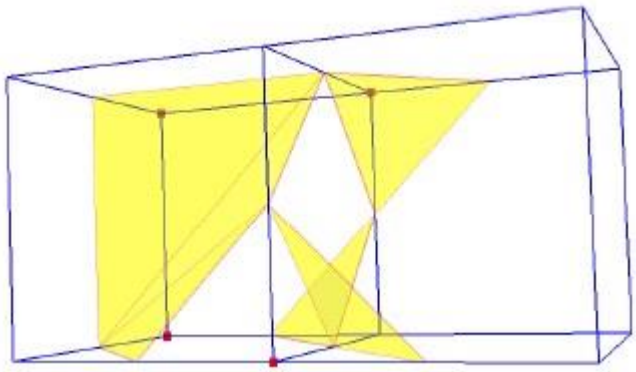


Рис. Вариант нарушения топологической связности, пара конфигураций 12 и 3

То есть в худшем случае полученная модель может состоять из отдельных треугольников. Помимо нарушения связности полигонов такая неоднозначность приводит к возникновению сцепок близко прилегающих друг к другу поверхностей:

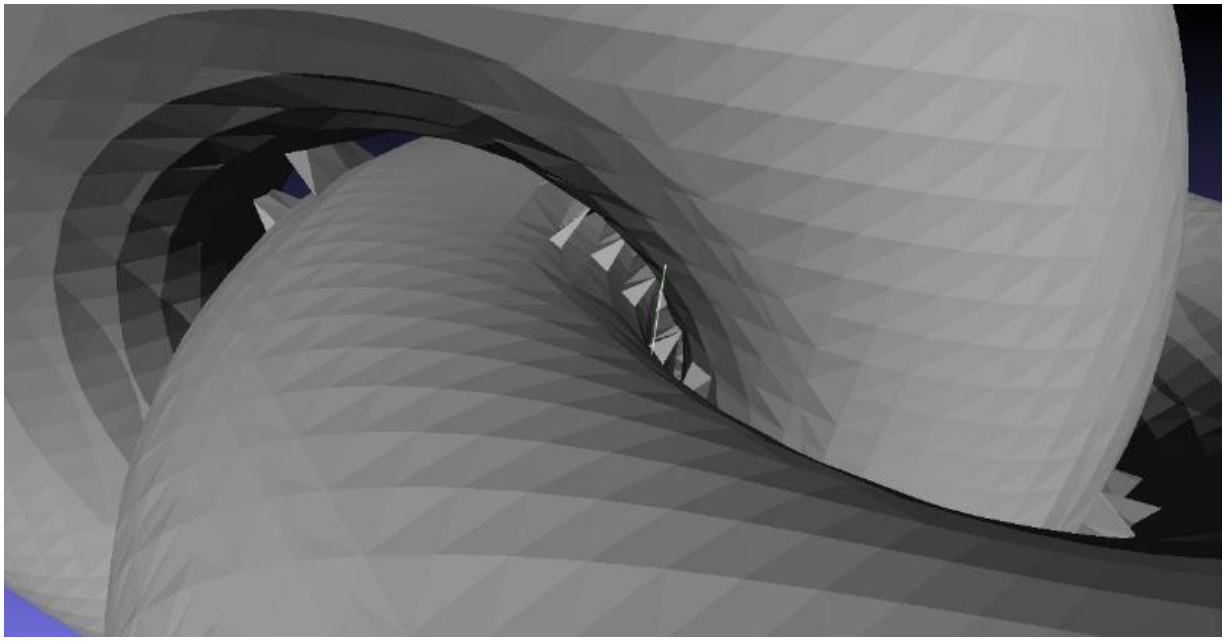


Рис. Алгоритм марширующих кубов. Сцепки.

Для разрешения неоднозначности вводятся дополнительные конфигурации и таблицы соответсвия между ними для построения треугольников, описанные в статье Е.В.Черняева “Marching cubes 33: construction of topologically correct isosurfaces”.

Применение описанных в статье таблиц гарантирует топологическую связность:

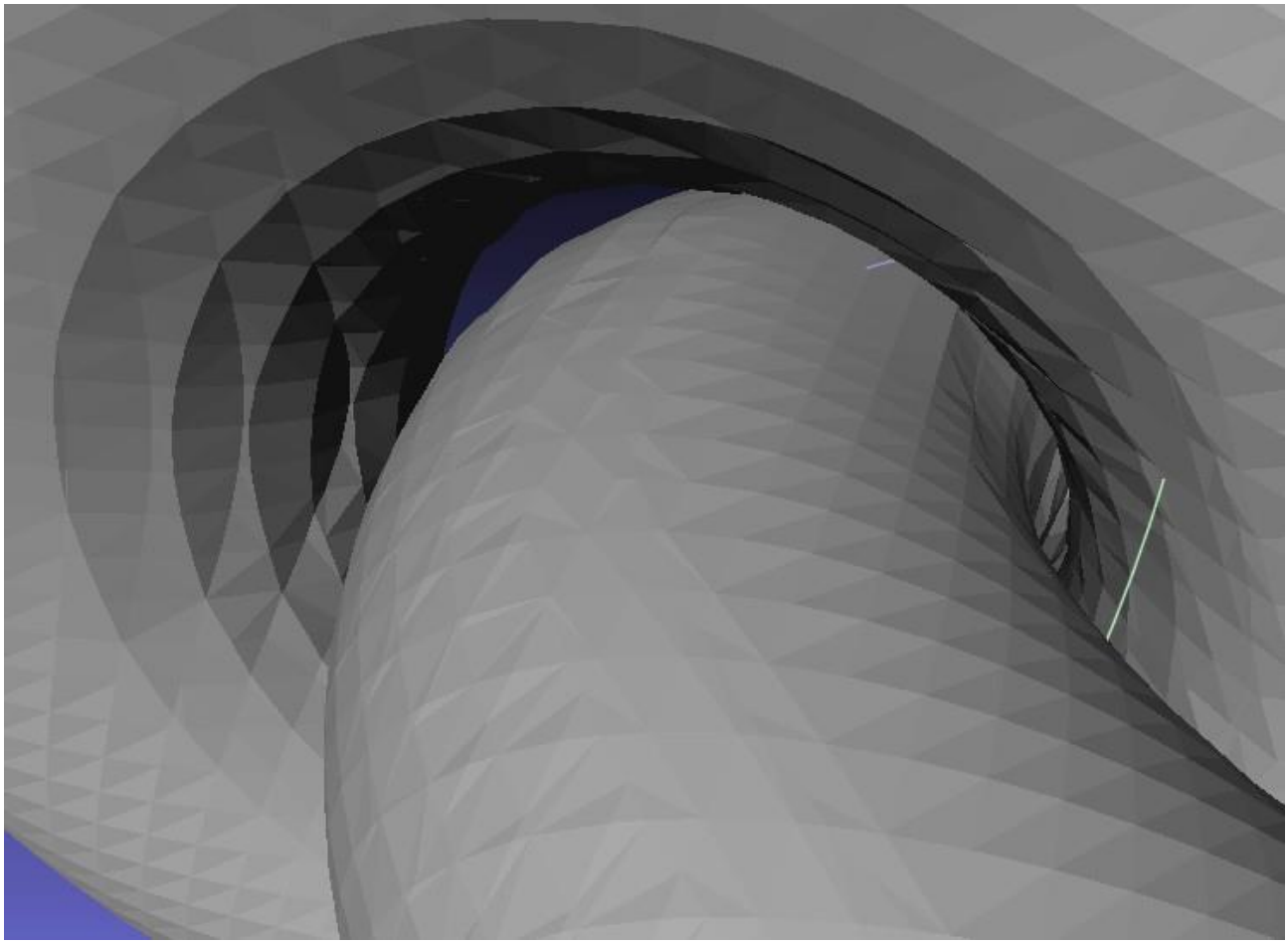
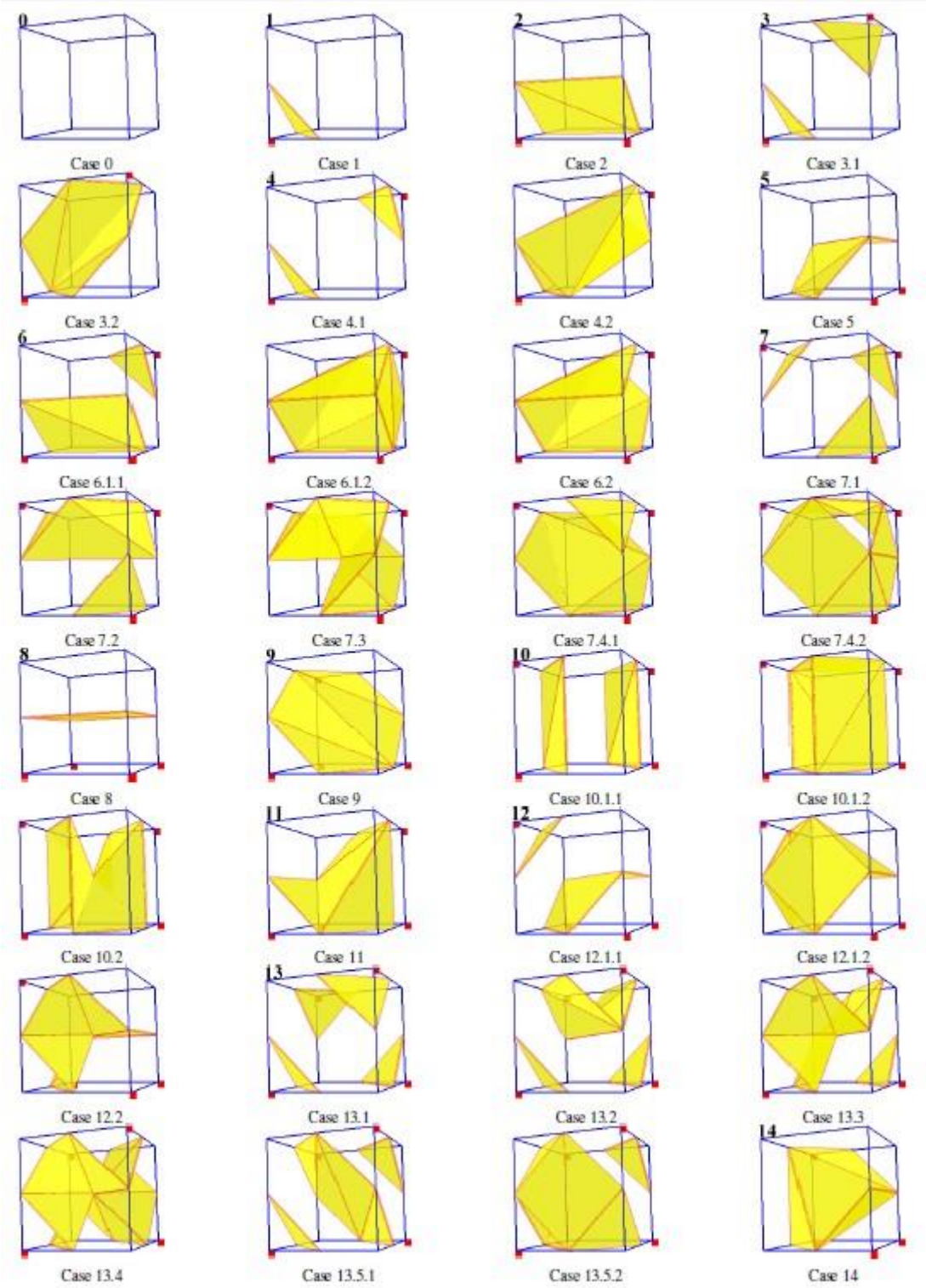


Рис. Алгоритм марширующих кубов. Модификация Черняева



Расширенная таблица вариантов марширующих кубов Черняева

Децимация.

Уменьшение количества полигонов модели сводится к стиранию лишних ребер полигонов и объединению двух вешин в одну. Удаляемое ребро выбирается исходя из используемой метрики. В контексте поставленной задачи необходимо, чтобы потери объема были минимальны.

Рассмотрим принцип метрики Quadric Error Metric, которая для каждого треугольника составляет плоскость N * V + d = 0, где N=(a,b,c) – нормализованный вектор нормали к плоскости, и определяет функцию квадрата расстояния от вершины до плоскости Q(v) = vAv + 2Bv + B, где A = N*N, B = d * N, C = d^2, и стоимость каждого ребра, соединяющего вершины s и t вычисляется по формуле: Cost(s,t) = (Q(s) + Q(t)) * t, где t – вершина, значение Q в которой больше, т.е. Q(t) > Q(s).
Таким образом редактируются сначала те ребра, которые максимально удаляют вершину от соседних полигонов.

Предлагаемая метрика основывается на QEM, учитывая также площадь треугольника, т.к. изменения треугольников с меньшей площадью повлекут меньшие изменения объема. Оператор Q задается матрицей 4x4, и считается для каждой вершины модели.
Ниже приведен листинг вычисления функции стоимости вершины:

```
result[0] = v.x * Q[0][0] + v.y * Q1][0] +
                                    v.z * Q[2][0] + 1 * Q[3][0];
result[1] = v.x * Q[0][1] + v.y * Q[1][1] +
                                    v.z * Q[2][1] + 1 * Q[3][1];
result[2] = v.x * Q[0][2] + v.y * Q[1][2] +
                                    v.z * Q[2][2] + 1 * Q[3][2];
result[3] = v.x * Q[0][3] + v.y * Q[1][3] +
                                    v.z * Q[2][3] + 1 * Q[3][3];


cost = result[0] * v.x + result[1] * v.y +
                          result[2] * v.z + result[3] * 1;
cost /= triArea;
```

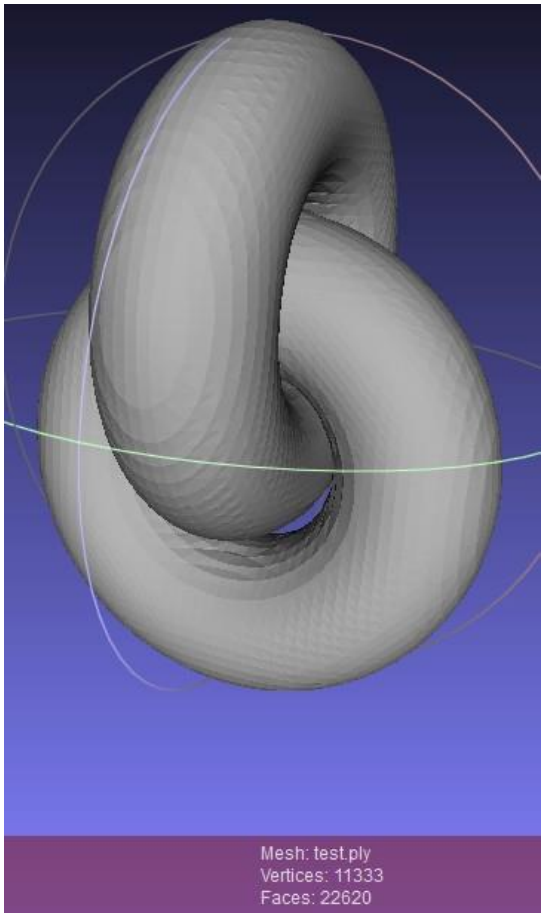


Рис. Исходная модель 22620 треугольников

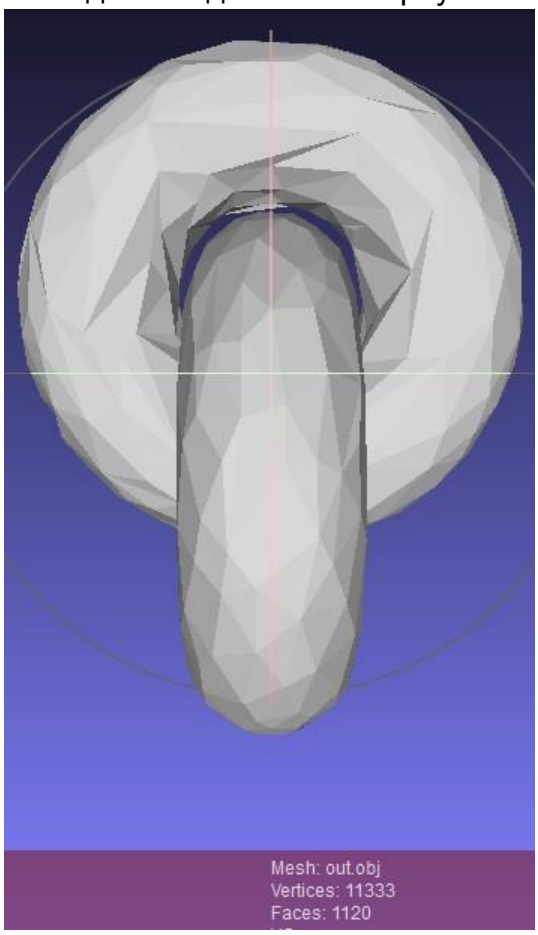


Рис. Модель после децимации 1120 треугольников

Сглаживание.

Так как алгоритм марширующих кубов строит довольно точное приближение исходной воксельной формы, она выглядит ребристой, и ее необходимо сглаживать.

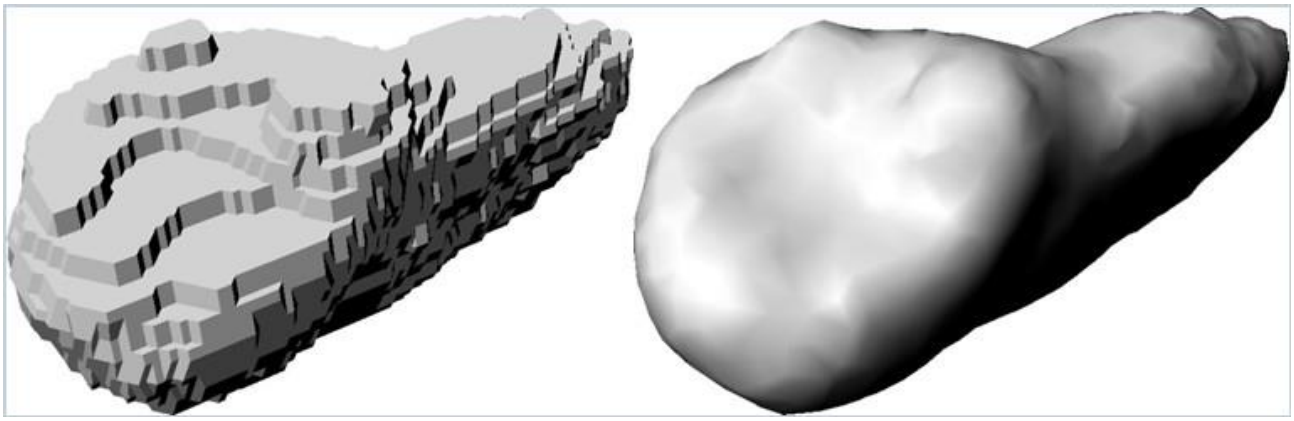
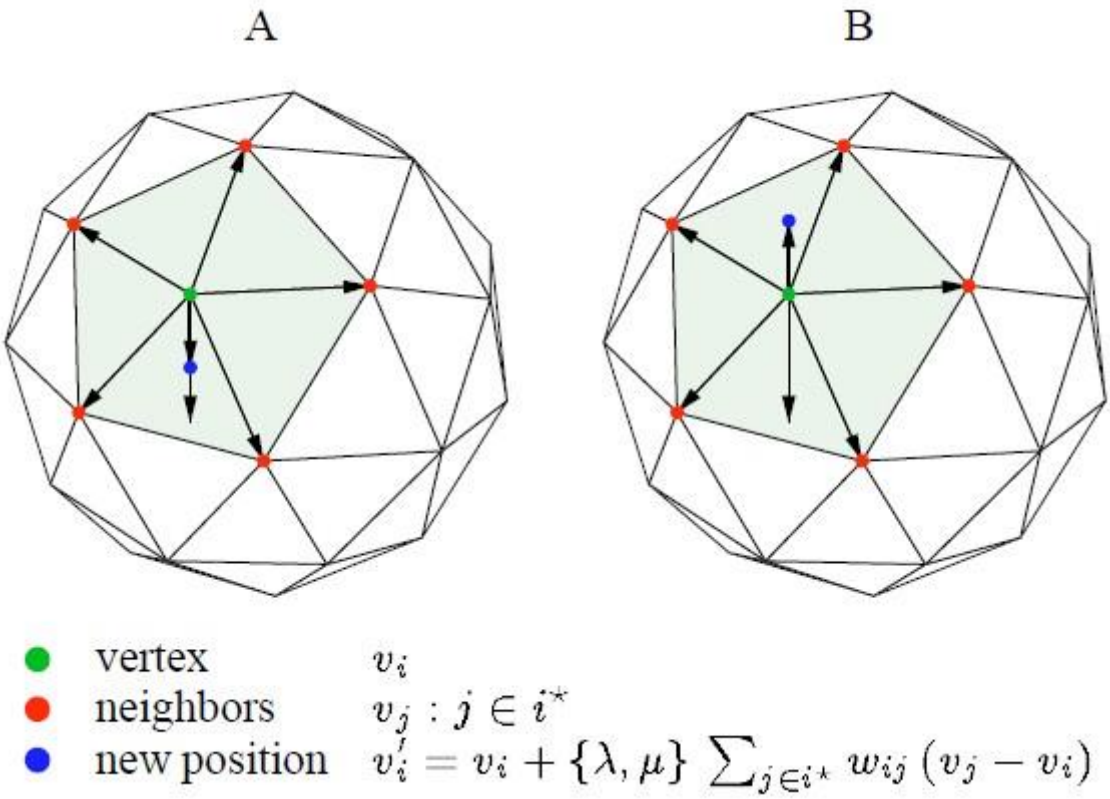


Рис. Результат работы марширующих кубов и желаемый сглаженный результат.

Сглаживание состоит в изменении положения вершин модели, без изменения полигональной сетки. В алгоритме Таубина используются следующие преобразования:



При этом $0 < \lambda < 1$, $\mu < 0$, $|\mu| > \lambda$.

Процесс сглаживания представляет собой вычисление смещений всех вершин, полученных из формулы показанной на рисунке сначала с положительным коэффициентом λ, а затем с отрицательным коэффициентом μ.

В качестве весовой функции взято число, обратное количеству соседних вершин: Wij = 1 / i*. Так как при построении полигонов в алгоритме марширующих кубов у вершин на ребрах смежных вокселей больше соседей, и эти вершины должны сдвигаться на меньшее расстояние, чем вершины лежащие на граничных ребрах, которые как раз и создают визуальную ступенчатость модели.

Наложение текстуры.

Для наложения текстуры на деформированную область применяется проективная UV развертка. А именно каждая вершина проецируется на исходную поверхность в направлении, обратном направлению сдвига при деформации. Однако при таком наивном подходе текстура накладывается нереалистично, вследствие того, что угол между нормалью некоторых треугольников и направлением сдвига значительно отличается от 0, из-за чего текстура накладывается неравномерно, и отчетливо виден переход между гранями треугольников.

Рис. Разрыв в текстуре

Для их устранения при наложении текстуры проектируются не сами вершины полученной области, а копии данных треугольников, которые повернуты перпендикулярно направлению сдвига.

Для этого вычисляется среднее арифметическое координат вершин треугольника, и покоординатно вычитается из вершин треугольника. Таким образом копия треугольника оказывается близко к началу координат, и ее поворот незначительно сместит координаты от начальных.

Так как размер треугольников генерируемых маршрующими кубами зависит от размера вокселя, который в свою очередь выбирается довольно малым, то для большего захвата текстуры применятся масштабирование.

Рис. Наложение текстуры без разрывов

Реализация.

Проект, частью которого является данная работа, основан на графическом движке с открытым исходным кодом Ogre3D. Метод выделения области реализован с помощью raycast-а Ogre3D. Проектирование координат при наложении текстуры реализовано с помощью кватернионов Ogre3D. Использовалась реализация алгоритма марширующих кубов Левинера, однако ввиду больших требований к памяти, для хранения вокселей вместо обычного массива использовался ассоциативный массив map.

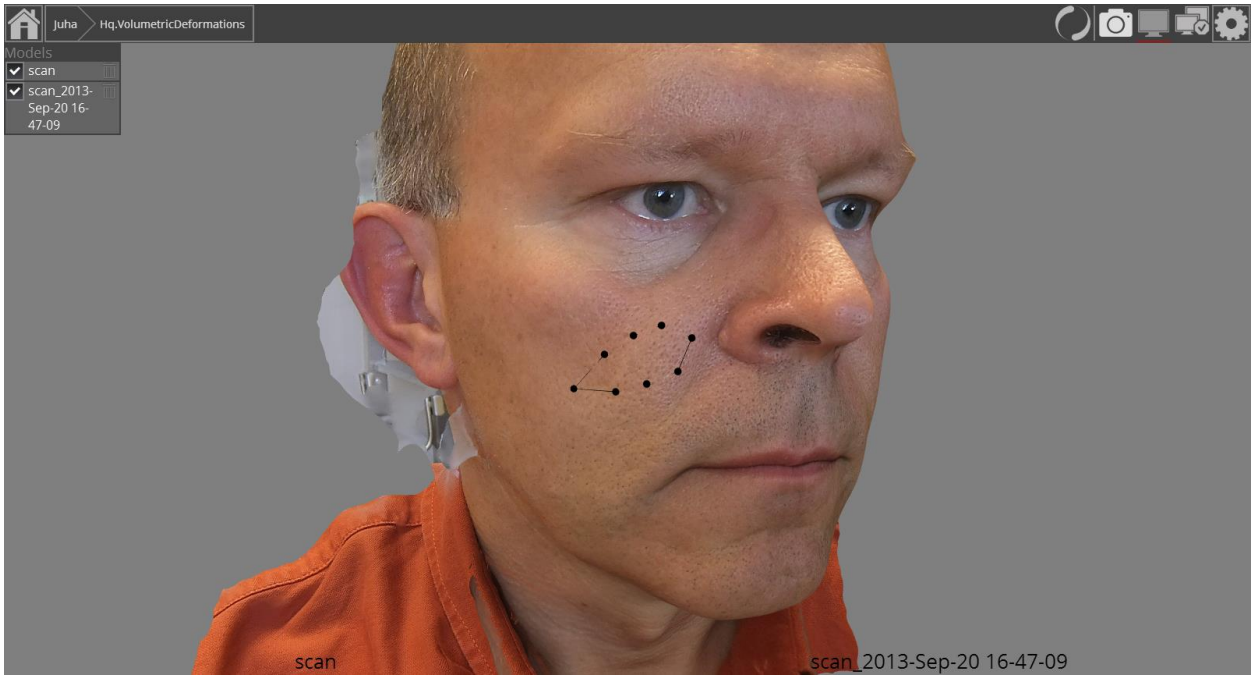
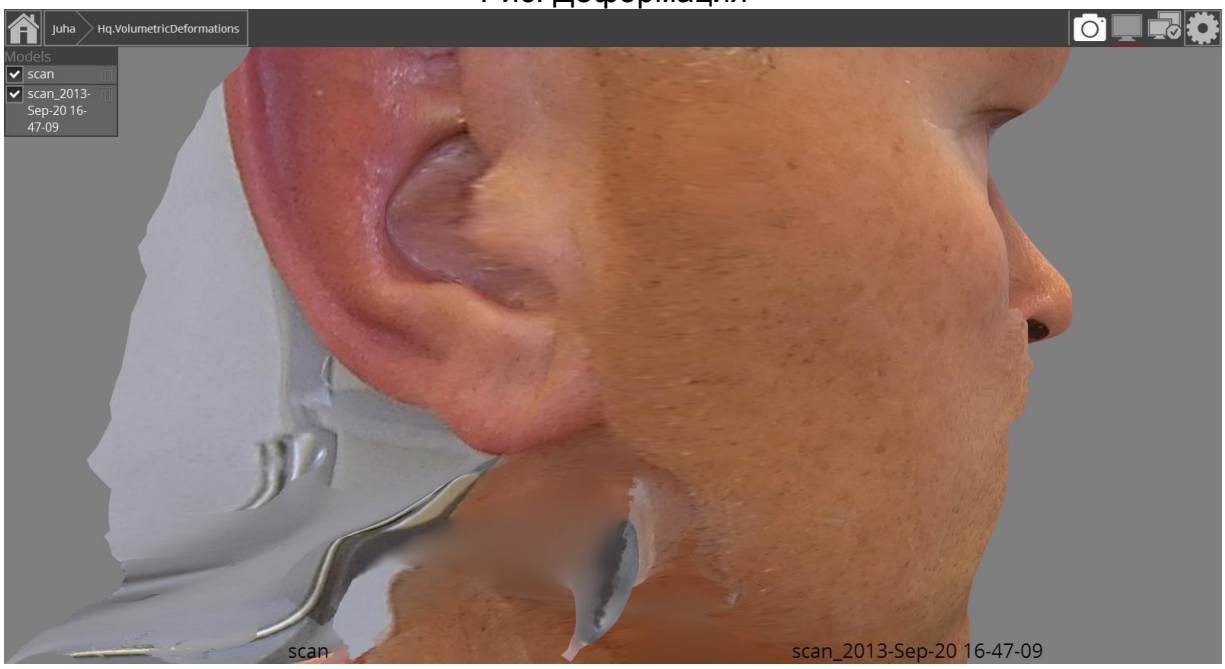


Рис. Выделение области

Рис. Деформация



Результаты

Поставленная задача решена, требуемая функциональность реализована:

1. Преобразование полигональной модели в воксельную.
2. Добавление к произвольной области модели заданного объема.
3. Описание физической зависимости в воксельном представлении модели.
4. Преобразование воксельной модели в полигональную.
5. Приведение полученной полигональной модели к стандартному формату, т.е. соблюдение топологической связности, гладкости и правильной триангуляции полигональной сетки.
6. Наложение текстуры деформируемой области на деформированную.

Написанная библиотека будет использована для моделирования пластических операций с учетом объемных параметров моделей.

Ссылки

[1] Rolf Michael Koch “Methods for Physics Based Facial Surgery Prediction”ftp://ftp.inf.ethz.ch/pub/publications/diss/th13912.pdf

[2] Samuel Hans Martin Roth “Bernstein–Bézier Representations for Facial Surgery Simulation” ftp://ftp.inf.ethz.ch/pub/publications/diss/th14531.pdf

[3] Efficient implementation of Marching Cubes’ cases with topological guarantees http://zeus.mat.puc-rio.br/tomlew/pdfs/marching_cubes_jgt.pdf

[4] Ogre3D, open-source graphics rendering engine http://www.ogre3d.org/

[5] Gabriel Taubin A Signal Processing Approach To Fair Surface Design http://www-evasion.imag.fr/people/Franck.Hetroy/Teaching/Geo3D/Articles/taubin1995.pdf

[6] A New Method of Mesh Simplification Algorithm Based on QEM http://scialert.net/fulltext/?doi=itj.2010.391.394&org=11

[7] Чернышенко А.Ю. Технология построения адаптируемых многогранных сеток и численное решение эллиптических уравнений 2-го порядка в трехмерных областях и на поверхностях http://www.math.uh.edu/~molshan/ftp/pub/Chernyshenko_thesis.pdf

[8] Chernyaev E. Marching Cubes 33: Construction of Topologically Correct Isosurfaces // Tech. Rep. CN/95-17. CERN, 1995.

[9] Конечно-элементный анализ задач биоимпендансной диагностики http://dodo.inm.ras.ru/research/_media/bioimpedance/vyc0733.pdf