

**Санкт-Петербургский Государственный Университет
Математико-механический факультет**

Кафедра системного программирования

Моделирование и параметризация изменения внешности на 3D модели лица

Курсовая работа студента 344 группы
Фасхитдинова Рамиля Равильевича

Научный руководитель

Петров А.Г.

Санкт-Петербург
2013

Содержание

Введение	3
Постановка задачи	2
Обзор существующих решений	3
Теоретическая основа курсовой работы	3
Перенос текстуры	3
Преобразование полигональной модели в воксельную	5
Параметризация	6
Тесселяция	7
Реализация	8
Результаты	9
Заключение	9
Ссылки	10

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 2 инструментов для моделирования пластических операций:

- Перенос участков кожи
- Добавление геля в подкожные слои

Подробнее, моя задача состояла в реализации следующей функциональности:

1. Перемещение текстуры по поверхности трехмерной модели
2. Геометрическое моделирование увеличения области трехмерной модели при добавлении жира или геля

Обзор существующих решений.

На данный момент существуют следующие методы моделирования объемных деформаций:

- Параметрическое моделирование, примером которого является Free Form Deformation, а также сплайны. При использовании данных подходов возникают трудности при необходимости точно задавать объем, который должен получиться у изменяемой области.
- Перенос облака точек по заданной функции. Аналогично параметрическому моделированию, изменение объема контролировать неудобно.
- Воксельное представление трехмерной модели и деформация, при использовании которой удобно считать объем и тесселировать модель. Из недостатков можно отметить большие, по сравнению с предыдущими, ресурсные затраты на хранение и обработку модели.

В данной работе будет описан метод моделирования, позволяющий воспользоваться описанными преимуществами воксельного представления модели, минимизируя ее недостатки. А именно, воксельное представление будет использовано только для деформируемой области, как промежуточное, с целью вычисления значений функции переноса точек полигональной модели.

Теоретическая основа курсовой работы.

Перенос текстуры.

Задача состоит в поиске соответствия между точками на трехмерной полигональной модели и на текстуре после UV развертки.

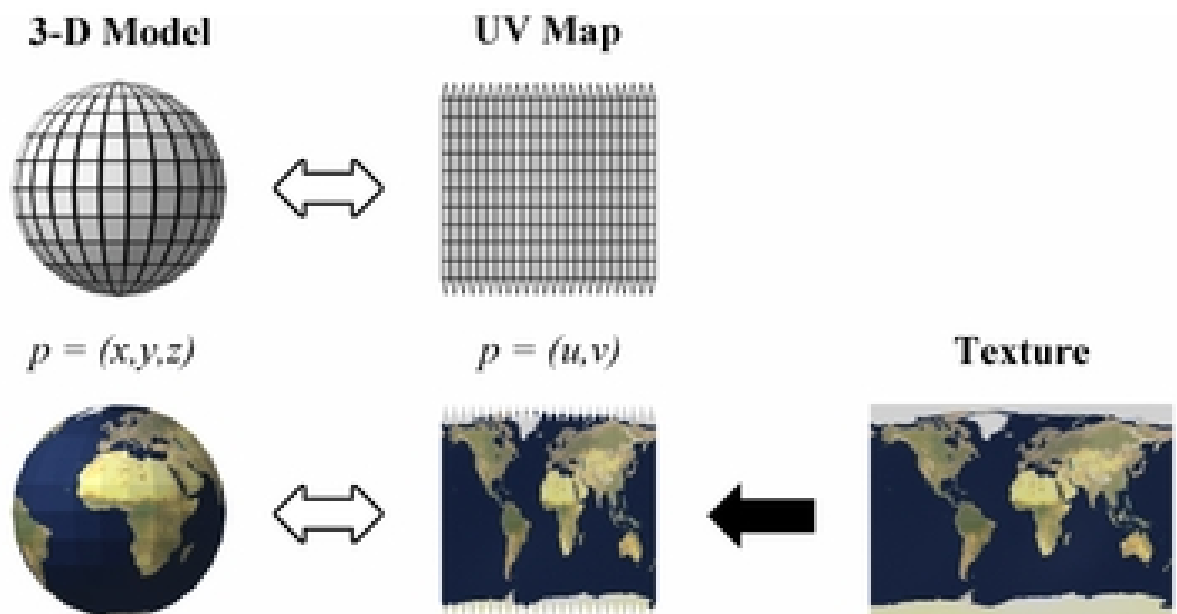


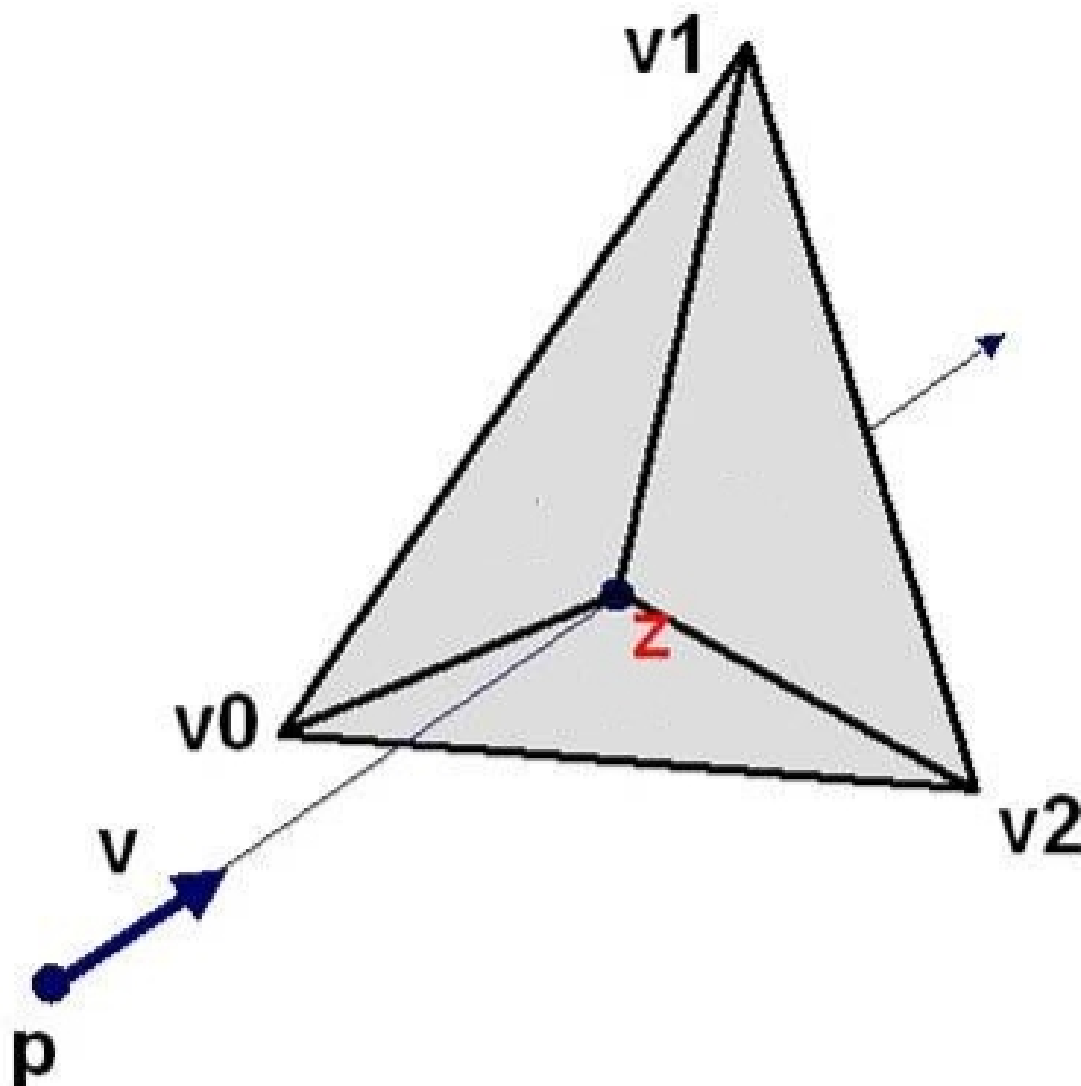
Рис. UV развертка

Текстуры обычно представляют из себя квадратное изображение, UV координаты

обозначают какая именно точка текстуры соответствует вершине данного полигона. Для конкретизации, все используемые модели имеют треугольную сетку, то есть данные полигоны это треугольники. Значения U и V обычно изменяются от 0 до 1.

Таким образом необходим метод, позволяющий определить координаты точки внутри треугольника, не зависящий от конкретной реализации UV – преобразования. Так как UV развертка в пределах одного треугольника является аффинным преобразованием, то необходимы аффинно-независимые координаты.

Примером таких координат являются барицентрические координаты, которые для некоторой точки пространства относительно заданного полигона могут быть определены как массы, которыми должны обладать вершины полигона, чтобы данная точка была центром масс этого полигона.

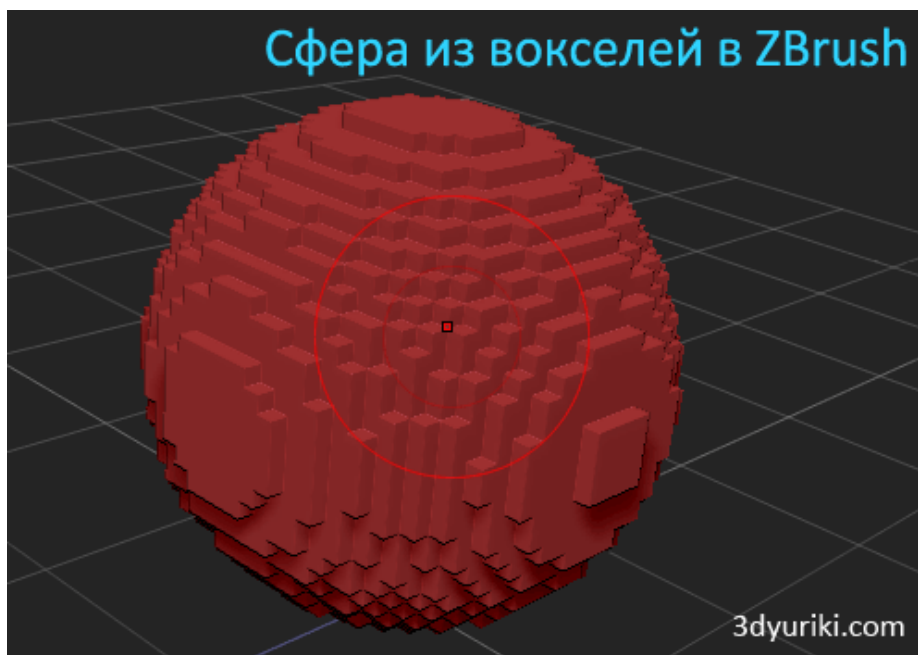


В данном случае, мы имеем трехмерные координаты вершин $V0(x_1, y_1, z_1)$, $V1(x_2, y_2, z_2)$, $V3(x_3, y_3, z_3)$, $Z(x, y, z)$. Барицентрические координаты точки $Z(m_1, m_2, m_3)$ находятся из следующей системы:

$$\begin{aligned} m_1 + m_2 + m_3 &= 1, \\ m_1x_1 + m_2x_2 + m_3x_3 &= x, \\ m_1y_1 + m_2y_2 + m_3y_3 &= y, \\ m_1z_1 + m_2z_2 + m_3z_3 &= z. \end{aligned}$$

Преобразование полигональной модели в воксельную.

Так как моделирование вливания геля в подкожные слои требует вычисления объема, для него лучше подходит воксельная модель, представляющая из себя аппроксимацию исходного объекта кубами. Тогда минимальной единицей изменения объема будет объем одного вокселя, зависящий от воксельного разрешения. Таким образом, исходя из величины объема, который необходимо влить будет подбираться воксельное разрешение так, чтобы погрешность удовлетворяла необходимой точности моделирования.



Алгоритм преобразования полигональной модели в воксельную состоит в следующем: после того как определено воксельное разрешение N , вершины исходной модели нормализуются таким образом, чтобы модель была вписана в куб размера $N \times N \times N$ с левой нижней вершиной в начале координат, называемый воксельной решеткой. Далее для каждого ребра треугольной сетки определяются элементы воксельной решетки, которые оно пересекает, и эти воксели считаются заполненными.

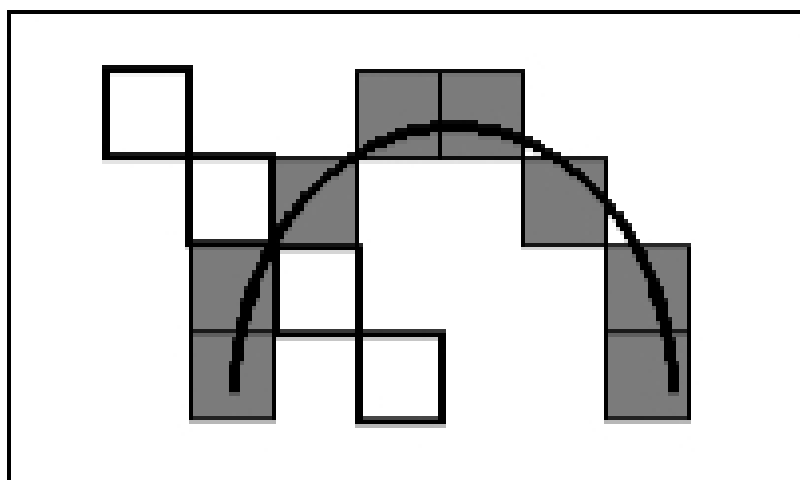
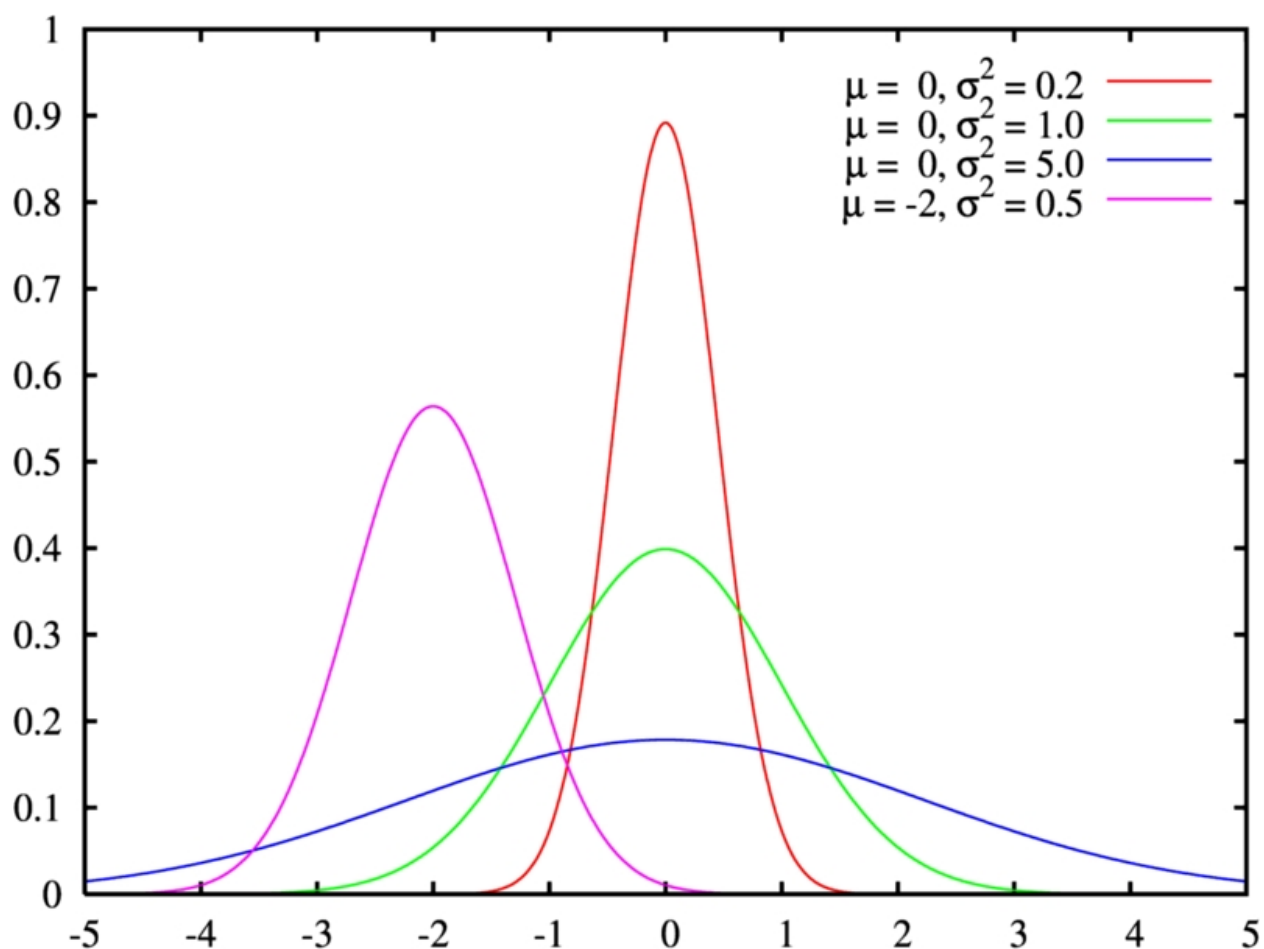


Рис. Иллюстрация работы алгоритма на плоскости

Параметризация.

Для определения формы деформированной поверхности необходима функция, задающая смещение для каждого вокселя. Для данной задачи использовано нормальное распределение Гаусса $f(x)$, с математическим ожиданием $\mu = 0$ и стандартным отклонением σ .

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$



Вследствие специфики прикладных задач, деформируемые области имеют выпуклую границу, близкую к окружности. Поэтому, вычислив средний радиус области R , функцию смещения вокселя можно задать как функцию от расстояния ρ между данным вокселем и центром области. А для того, чтобы полученная в результате поверхность была гладкой, в качестве аргумента для функции $f(x)$ берется $\rho * (L / R)$, где L – длина прямой на поверхности от центра области до граничной точки.

Так как площадь нормального распределения равна 1, объем деформированной области может быть определен как $V_0 = \pi * R^2$. Соответственно, чтобы получить требуемый объем V необходимо домножить $f(\rho * (L / R))$ на коэффициент $k = V/V_0$

Остается определить значение параметра σ , согласно правилу трех σ 99,7% объема будет приходиться на область $[-3\sigma; 3\sigma]$, следовательно σ можно

положить равным $R/3$.

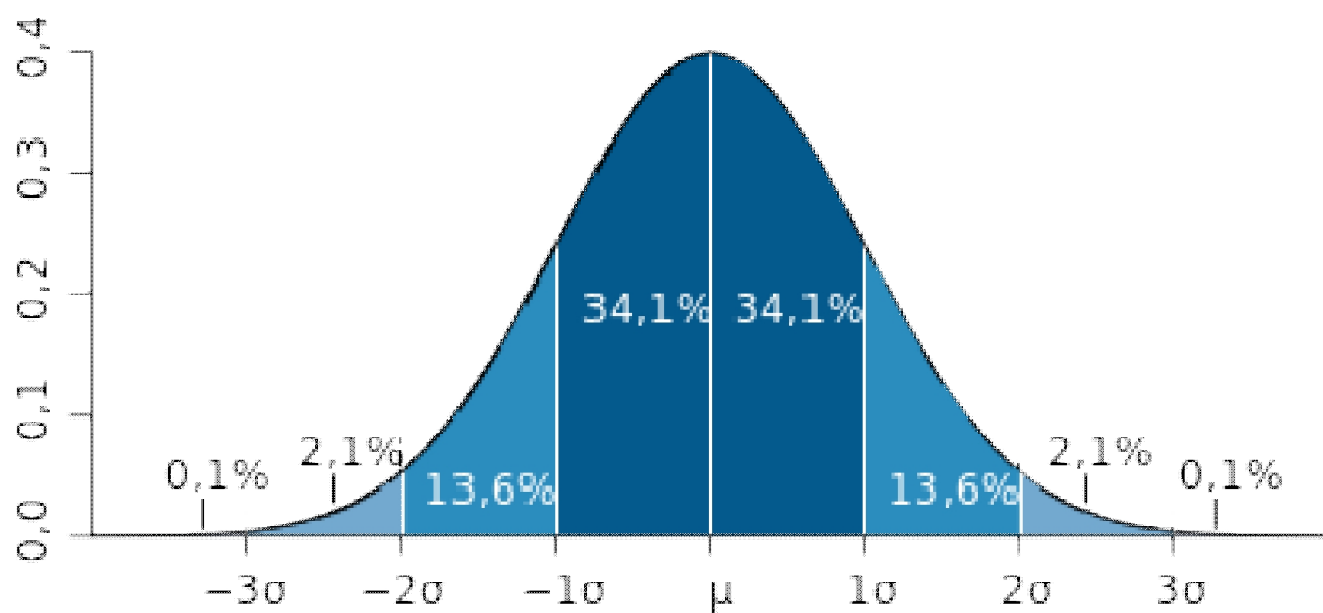


Рис. Правило 3 σ

Тесселяция.

Для того чтобы полученная после сдвига поверхность была более гладкой, деформируемую область можно тесселировать, т. е. повысить детализацию с помощью увеличения количества полигонов исходной области.

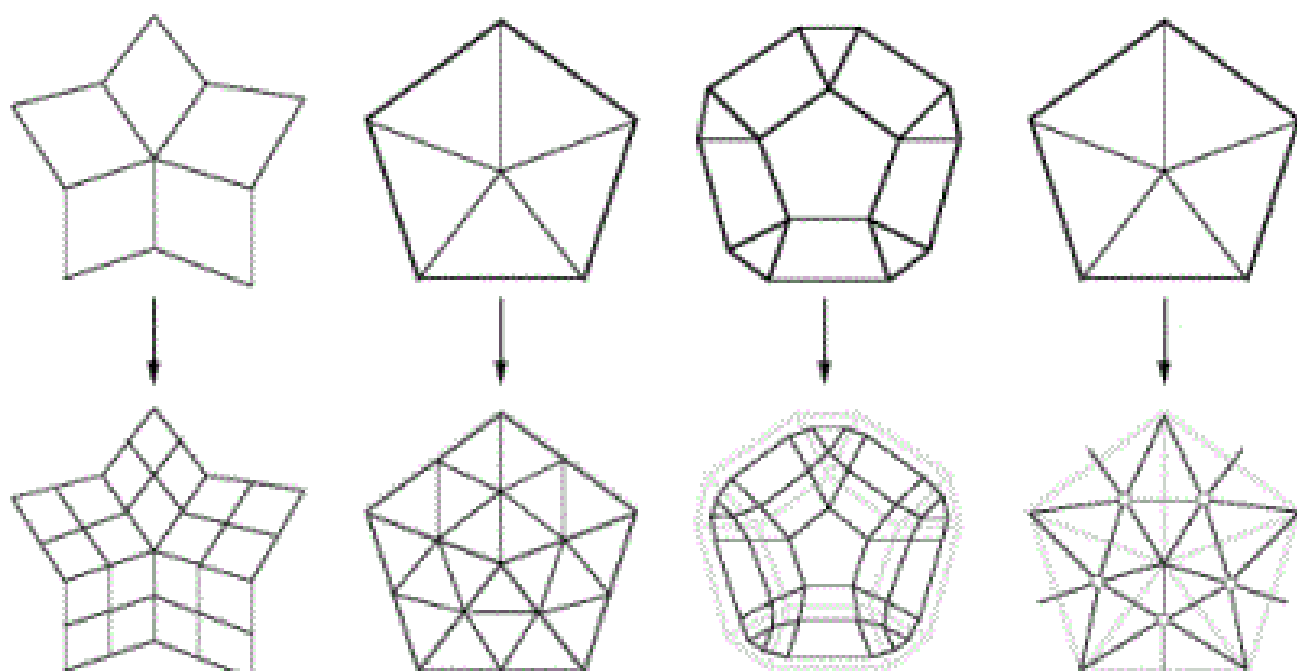


Рис. Примеры методов тесселяции из библиотеки CGAL

Реализация

Проект, частью которого является данная работа, основан на графическом движке с открытым исходным кодом Ogre3D. Метод, позволяющий определять соответствие между точками на модели и на текстуре, описанный в теоретической части, был написан на основе метода `gaucast` из Ogre3D. Для работы с текстурой использовались битовые маски OpenCV. Также с помощью метода библиотеки OpenCV область, из которой была перенесена текстура закрашивается под цвет окружающей ее текстуры.

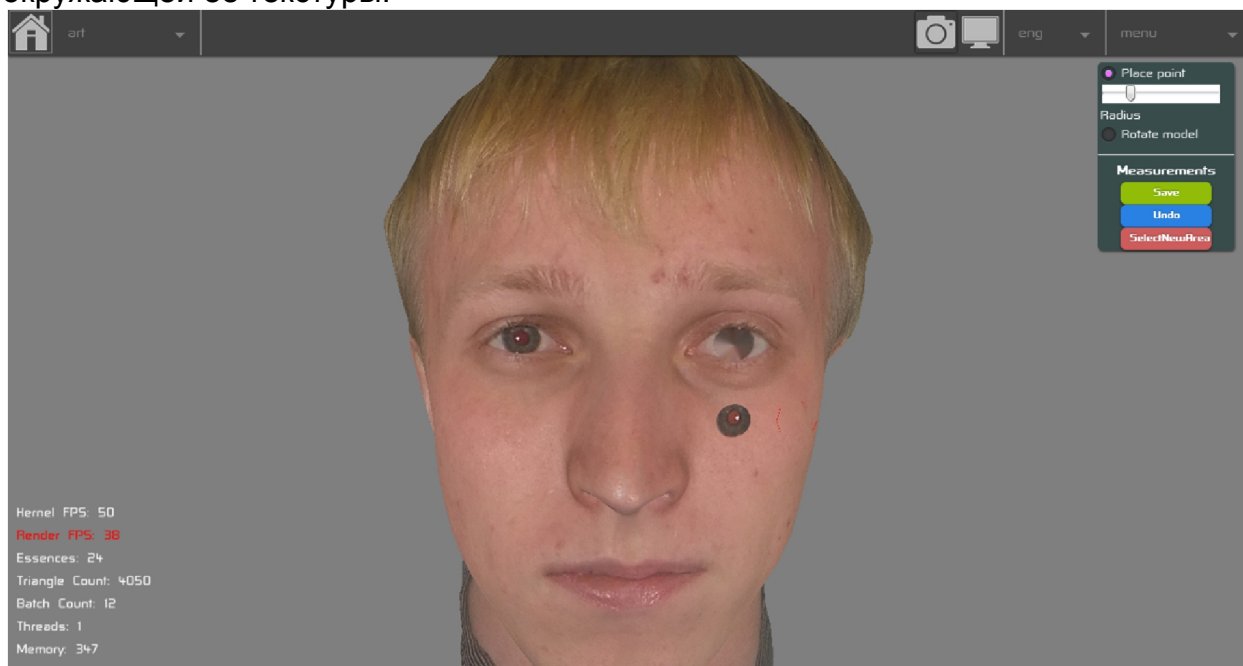


Рис. Скриншот переноса текстуры

Метод смещения вокселей реализован с помощью паттерна стратегия, что позволит применять различные виды распределений, в зависимости от того, какая часть тела деформируется.

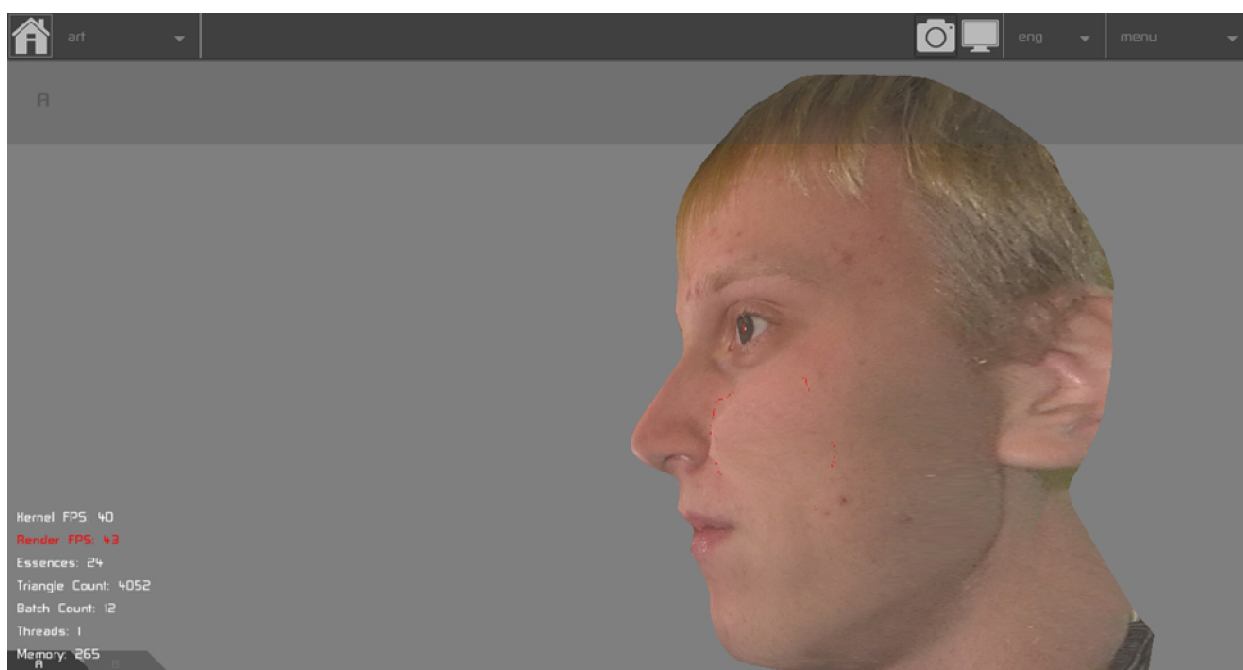


Рис. Скриншот добавления геля

Результаты

Реализованы модули, для:

1. Перемещения текстуры по поверхности трехмерной модели
2. Геометрического моделирования увеличения области трехмерной модели при добавлении жира или геля

Заключение

Поставленная задача решена, планируется внедрение и демонстрация. Планируется, по результатам демонстрации, настройка параметров и оптимизация расчетов, так как реализованные в ходе курсовой работы методы объемной деформации не обладают желаемым соотношением точности и скорости выполнения. В дальнейшем предполагается пойти по пути улучшения качества использованных алгоритмов. Существует два способа улучшения – использовать параллельные вычисления, которые хорошо применимы к воксельной модели, и оптимизировать методы тесселяции деформируемой области.

Ссылки

[1] 3D Free-form Deformation

<http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/ffdeform.html>

[2] David Eberly “Game Physics”

[3] PolyVox, library for the storage and processing of volumetric (voxel-based) environments

<http://www.volumesoffun.com/polyvox-documentation/>

[4] Ogre3D, open-source graphics rendering engine

<http://www.ogre3d.org/>

[5] OpenCV (Open Source Computer Vision), library of programming functions for real time computer vision.

<http://opencv.org/>

[6] CGAL subdivision (Computational Geometry Algorithms Library), geometric algorithms in the form of a C++ library

http://www.cgal.org/Manual/latest/doc_html/cgal_manual/Subdivision_method_3/Chapter_main.html

[7] Elmar Langetepe, Gabriel Zachmann “Geometric Data Structures for Computer Graphics”

[8] Fundamentals of Voxelization

<http://www.cs.sunysb.edu/labs/projects/volume/Papers/Voxel/index.html>