Санкт-Петербургский государственный университет

Фундаментальная информатика и информационные технологии

Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Монькин Александр Александрович

Название

Модуль расчета деформации трехмерной модели мягкого тела для компьютерной системы планирования хирургических операций

Магистерская диссертация

Научный руководитель:
д.ф.-м.н., профессор Терехов А. Н.

Рецензент:
инженер-программист Николаев С. Н.

Санкт-Петербург

2016

SAINT-PETERSBURG STATE UNIVERSITY

Fundamental Computer Science and Information Technologies

Software of Computers, Complexes and Networks

Monkin Alexandr

Title

Calculation module of 3d soft body model deformation for surgery planning computer system

Master's Thesis

Scientific supervisor:
professor Terekhov A. N.

Reviewer:
software engineer Nikolaev S. N.

Saint-Petersburg

2016

Оглавление

1 Введение

Компьютерные системы широко применяют при проведении хирургических операций. Пациента сканируют, восстанавливают трехмерную модель интересующей области, по модели изучают свойства области и моделируют возможные результаты операции. Для подготовки к сложным операциям моделируемый объект печатают на принтере и изучают его в действительном масштабе. В случае пластических операций возможные результаты демонстрируют пациенту и уточняют желаемый результат.

Распространение фото техники в клиниках делает актуальным восстановление, визуализацию и обработку трехмерных моделей пациентов. Современные компьютеры позволяют проводить симуляцию операции с достаточной точностью непосредственно во время консультации. Для ускорения вычислений разрабатывают специальные модели, структуры данных, распараллеливают вычисления, переносят вычисления на графические ускорители.

Целью пластических операций является устранение дефектов и восстановление функций тканей и органов, улучшение внешнего вида пациента. С целью уменьшения асимметрии лица проводят трансплантацию жировой ткани в заданные области. При планировании операции требуется определить разницу между областями и объем, который необходимо заполнить. Демонстрация возможных результатов этой операции пациенту может быть использована при принятии решения о проведении операции.

В компьютерной системе планирования хирургических операций Phoenixcas 3D Viewer [1] встроена возможность восстановления поверхностной трехмерной модели пациента, визуализация, выполнение замеров, разные

способы сравнения трехмерных моделей и другое. В целях расширения функциональности в систему планирования добавляются возможности симуляции результатов различных операций. В статье описывается разработка библиотеки физических расчетов деформации мягкого тела, разработка модуля планирования операции по трансплантации жировой ткани в области лица по трехмерной модели поверхности лица, апробация модуля.

2 Постановка задачи

Целью работы является добавление в систему планирования возможности симуляции результатов операции по трансплантации жировой ткани в области лица по трехмерной модели поверхности лица. Для этого необходимо выполнить следующие задачи:

- Реализовать физическую модель деформации мягкого тела
- Разработать модуль симуляции результатов операции
- Провести апробацию модуля

От модуля требуется явное выделение последовательности шагов симуляции, время расчетов не должно превышать 30 секунд.

3 Обзор

3.1 Методы физического моделирования

Подходы, используемые для физического моделирования, описаны, например в [2]. Объект представлен позицией *x* (координатами в пространстве) и скоростью *v* в момент времени *t*. Чтобы определить положение объекта в следующий момент времени (*t* + *h*), сначала складывают внешние и внутренние силы, влияющие на объект, затем вычисляют ускорение по второму закону Ньютона. Потом обновляют скорость и позицию объекта, используя численный метод решения систем обыкновенных дифференциальных уравнений.

Алгоритм можно записать так:

1. compute forces F
2. compute accelerations a = F / m
3. v += a * h
4. x += v * h
5. t += h

Для обновления позиции и скорости можно использовать метод Эйлера (точность O(h)). Более стабилен метод Midpoint (точность $O(h^2)$), в котором используется не текущая скорость, а пересчитанная на середине участка. Эти методы входят в класс методов Рунге-Кутты. Широко распространен метод Рунге-Кутты четвертого порядка. В методе используют значение производной в начале участка и три оценки на участке. Методы, использующие текущую скорость и положение, называют явными (explicit methods). Методы, в которых для обновления на текущем шаге используют значения в конце шага, называют неявными (implicit methods). Например, в методе predictor-corrector делают шаг

вычислений явным методом, получают новую скорость и пересчитывают этот шаг с новой скоростью. В методах Верле (точность $O(h^2)$) используют информацию о текущем положении объекта и предыдущем, не записывая скорость явно. Разработаны модификации, в которых скорость явно присутствует. В методе symplectic Euler сначала обновляют скорость, затем позицию. Также этот метод называют semi-implicit.

Обновление позиции и скорости в алгоритме методом Эйлера

$$x_{i+1} = x_i + v_i * h$$

$$v_{i+1} = v_i + a_i * h$$

Velocity Verlet

$$v_{i+0.5} = v_i + a_i * h / 2$$

$$x_{i+1} = x_i + v_{i+0.5} * h$$

$$v_{i+1} = v_{i+0.5} + a_{i+1} * h / 2$$

Метод Верле

$$x_{i+1} = 2 * x_i - x_{i-1} + a * h^2$$

Symplectic Euler

$$v_{i+1} = v_i + a_i * h$$

$$x_{i+1} = x_i + v_{i+1} * h$$

Чтобы обработать пересечения объектов и введенные на объекты ограничения, пересчитывают силы, действующие на объекты. Такие методы называют force based. Можно напрямую изменять ускорения или скорости объектов, эти подходы называют penalty methods и impulse methods. Также можно обрабатывать пересечения объектов и ограничения на них, изменяя текущие положения объектов. Это называют projection methods. В [3] применяют метод Верле, в цикле симуляции после обновления позиций обрабатывают пересечения объектов, сразу же меняя позиции так, чтобы пересечений не было. Используя метод symplectic Euler при impulse based подходе обработки столкновений объектов пересечения обрабатывают после обновления скоростей (перед обновлением позиций).

Среди проблем моделирования важно отметить скорость работы и стабильность метода. Стабильность можно увеличить, уменьшив шаг симуляции. При этом время работы алгоритма возрастет. Время работы также зависит от числа моделируемых объектов, особенно от проверки пересечений. Для ускорения проверки пересечений используют схемы распараллеливания, специальные структуры данных (деревья, воксели), графические ускорители.

3.2 Моделирование мягких тел

Подходы к моделированию мягких тел описаны, например в [4]. Их делят по типу решетки, используемой для представления объекта: решетка определена на объекте (методы Лагранжа) или решетка определена на пространстве (методы Эйлера). В первом случае объект состоит из множества частиц с собственными позициями и скоростями, перемещение частиц определяет движение и форму объекта, меняя связи между частицами можно задавать свойства объекта. К таким методам относятся, например система Масс-с-пружинами (Mass-Spring System), Finite Element Method (FEM), Smoothed Particle Hydrodynamics (SPH). В методах, где решетка определена на пространстве, объект представлен теми частями пространства, в которых он находится. Например, в воксельной модели трехмерное пространство разделено на кубы, объект в нем — множество кубов, которые объект пересекает. Изменение занимаемых частей пространства задает движение и форму объекта, свойства объекта вычисляются из свойств занимаемых частей. Эти методы используют, например для симуляции движения жидкости и газа в замкнутом пространстве.

В методе Масс-с-пружинами объект представлен множеством частиц и пружин. Каждой частице заданы позиция, скорость и масса. Сумма масс частиц определяет массу объекта. Между парами частиц установлены пружины,

сохраняющие расстояние между частицами. Каждой пружине заданы начальная длина и жесткость. Пружины могут быть установлены как силы, действующие на частицы, или, в других подходах, как ограничения на парах частиц.

Свойства объекта задаются расположением и жесткостью пружин. С помощью пружин, построенных по поверхности объекта, сохраняют форму объекта, если их недостаточно, добавляют пружины (и частицы) внутри объекта. Сохраняют объем замкнутого объекта, например, добавляя новую частицу — центр масс — и новые пружины, проведенные к новой частице. При изменении объема по этим пружинам передаются силы с целью восстановить объем [5]. Для задания определенных свойств, например растяжения ткани, фиксируют расположение пружин (например квадратной топологией) и изменяют жесткости пружин в соответствии с коэффициентом растяжения. В [6] описана модификация модели масс-с-пружинами для моделирования растяжения тканей: нелинейной упругости и коэффициента Пуассона.

Для моделирования поведения мягкого тела вводят дополнительные силы, влияющие на частицы. В [7] и [8] мягкое тело моделируют системой масс-с-пружинами. Частицы и пружины описывают поверхность моделируемого объекта. На объект воздействует сила гравитации. На каждую частицу действует сила внутреннего давления: направление соответствует нормали (n') вершины поверхности, давление вычисляется так: $P = V^{-1} n R T$, где V — объем тела, n — моль, R — универсальная газовая постоянная, T — температура, сила вычисляется так: $F = n' P A$, где A — площадь поверхности (соответствующая вершине).

В FEM объект разбит на элементы. Вершины на границе соседних элементов общие. Задаются граничные условия, например положения вершин поверхности

объекта после деформации. Значения в узлах (положения вершин) неизвестны. Значения в узлах вычисляются по заданным свойствах деформации элементов.

В Эйлеровых методах пространство разбито на части, ячейки. Объект представлен множеством ячеек, которые он занимает в пространстве. Свойства объекта распределены по ячейкам пространства. Например для симуляции воды ячейкам задают некоторые начальные значения скорости и давления, изменения этих значений во времени описывают уравнениями Навье-Стокса. В методе SPH объект представлен частицами, которые могут находиться где угодно в пространстве. Каждая частица обладает скоростью и массой. Свойства объекта распределены по частицам. Значение свойства в частице вычисляется с учетом влияния соседних частиц, взвешенной суммой, используя функцию ядра (smoothing kernel).

В [9] описано применение метода SPH для симуляции течения крови в сосуде. В статье сосуд представлен тетраэдальной сеткой (tetrahedral mesh), жидкость в сосуде — множеством частиц, расположенных внутри тетраэдальной сетки. Для симуляции деформации сосуда на поверхности сетки установлено множество частиц, вступающих во взаимодействие с частицами, используемыми для симуляции течения жидкости.

Из перечисленных FEM является самой точной моделью, требует знания всех характеристик объекта, требует больше вычислений. Для симуляции методом SPH требуется обработка пересечения большого числа частиц, требует много вычислительных ресурсов. Система Масс-с-пружинами проста в реализации, вычислительных ресурсов требует меньше других, достаточно точная в поставленной задаче.

3.3 Инструменты

Среди инструментов физического моделирования можно выделить ANSYS [10], ABAQUS [11], ADINA [12]. Это коммерческие продукты, симуляция основана на FEM, применяются в проектировании сложной техники, автомобилей, судов, самолетов, для анализа прочности конструкций, вязкоупругих материалов, теплопереноса.

FEBio [13] (Finite Element for Biomechanics) специально разработан под задачи биомеханики, с открытым исходным кодом. В SOFA [14] (Simulation Open Framework Architecture) поддерживаются симуляция твердых тел, модель масс-с-пружинами, FEM, SPH, разные методы обработки столкновений, разные схемы интегрирования. В OpenTissue [15] реализованы деформируемые объекты (FEM, Particle System), обработка пересечений, визуализация, сегментация.

В Canfield [16] представлены коммерческие решения для восстановления трехмерных моделей, визуализации, проведения замеров, измерения разницы объемов, анализа состояния кожи, симуляции результатов различных операций на лице и теле. В Crisalix [17] представлено веб-приложение для симуляции результатов различных операций на лице и теле, платное.

3.4 Position Based Dynamics

Для симуляции динамических систем широко распространены подходы, в которых для выполнения наложенных на объект ограничений и обработки пересечений используют силы. Применяют подходы, в которых для тех же целей напрямую изменяют скорости объектов. В [18] представлен подход,

основанный на непосредственном изменении позиций. В статье описано, как обрабатывать пересечения объектов и ограничения (constraints) общего вида.

Цикл симуляции выглядит следующим образом:

(1) **forall** vertices i
(2) initialize $x_i = x_i^0$, $v_i = v_i^0$, $w_i = 1/m_i$
(3) **endfor**
(4) **loop**
(5) **forall** vertices i **do** $v_i \leftarrow v_i + \Delta t\, w_i\, f_{ext}(x_i)$
(6) dampVelocities(v_1, …, v_N)
(7) **forall** vertices i **do** $p_i \leftarrow x_i + \Delta t\, v_i$
(8) **forall** vertices i **do** generateCollisionConstraints($x_i \rightarrow p_i$)
(9) **loop** solverIterations **times**
(10) projectConstraints(C_1, ..., $C_{M+Mcoll}$, p_1, …, p_N)
(11) **endloop**
(12) **forall** vertices i
(13) $v_i \leftarrow (p_i - x_i) / \Delta t$
(14) $x_i \leftarrow p_i$
(15) **endfor**
(16) velocityUpdate(v_1, …, v_N)
(17) **endloop**

x_i – позиции частиц, v_i – скорости частиц, m_i – массы частиц, $w_i = 1 / m_i$.

В цикле симуляции сначала обновляют скорости частиц, затем в p_i заносят новые позиции, вычисленные методом Эйлера. Позиции p_i меняют так, чтобы выполнить заданные ограничения. Стабильность не зависит от размера шага итерации (dt). Затем позиции p_i копируют в x_i и обновляют скорости частиц v_i.

Объект представляется множеством частиц и ограничений на них. Например, если ткань задана полигональной сеткой, то ее можно представить частицами, соответствующими вершинам, и построенными по ребрам сетки

ограничениями, сохраняющими расстояния между парами частиц. Дополнительно на такое представление ткани можно наложить ограничения, сохраняющие углы между смежными полигонами. Замкнутый объект, поверхность которого задана полигональной сеткой, можно представить как ткань, на все частицы которой наложено ограничение на сохранение объема. Твердые тела можно представлять множеством частиц, заполняющих объем тела, на которых наложено ограничение на сохранение формы.

В [18] авторы описывают обработку ограничений общего вида. Чтобы выполнить заданное на позициях ограничение $C(p_1, \ldots, p_n)$ необходимо для каждой позиции применить поправку:

$$\Delta p_i = - s\, w_i \nabla p_i\, C(p_1, \ldots, p_n),$$

$$\text{где } s = C(p_1, \ldots, p_n) / \sum_j w_j |\nabla p_j\, C(p_1, \ldots, p_n)|^2$$

Например, ограничение на длину пружины (d), установленной на частицы p_1 и p_2, выглядит так:

$$C(p_1, p_2) = |p_1 - p_2| - d$$

Затем надо взять производную C по p_1 и p_2:

$$\nabla p_1\, C(p_1, p_2) = (p_1 - p_2) / |p_1 - p_2|$$

$$\nabla p_2\, C(p_1, p_2) = - (p_1 - p_2) / |p_1 - p_2|$$

При этом $s = (|p1 - p2| - d) / (w1 + w2)$

В итоге для восстановления расстояния между частицами надо применить к ним следующие поправки:

$$\Delta p_1 = - n * (|p_1 - p_2| - d) * w_1 / (w_1 + w_2)$$

$$\Delta p_2 = + n * (|p_1 - p_2| - d) * w_2 / (w_1 + w_2)$$

$$\text{где } n = (p_1 - p_2) / |p_1 - p_2|$$

Для сохранения объема трехмерной замкнутой модели можно применить следующее ограничение:

$$C(p_{1,}\dots,p_N)=\left(\frac{1}{6}\sum_{i=1}^{n_{triangles}}(p_{t_1^i}\times p_{t_2^i})\cdot p_{t_3^i}\right)-k_{pressure}V_0$$

и градиент

$$\nabla_{p_i}C=\sum_{j:t_1^j=i}(p_{t_2^j}\times p_{t_3^j})+\sum_{j:t_2^j=i}(p_{t_3^j}\times p_{t_1^j})+\sum_{j:t_3^j=i}(p_{t_1^j}\times p_{t_2^j})$$

Обзор способов симуляции различных объектов, основанных на позициях, приведен в [19], там описаны симуляция ткани, жидкости, деформируемых объектов, применение разных способов затухания (damping), схемы распараллеливания. В [20] выложена открытая библиотека, в основе которой подход, основанный на позициях.

4 Библиотека физических расчетов деформации мягкого тела

4.1 Архитектура

В качестве основы физической модели деформации мягкого тела выбран подход, описанный в [18]. Для настройки и отладки физической модели реализована библиотека алгоритмов на языке C++. Визуализация симуляции осуществляется средствами графического движка Ogre3d. В реализации используются структуры данных этого движка. Структура библиотеки представлена на рис. 1.

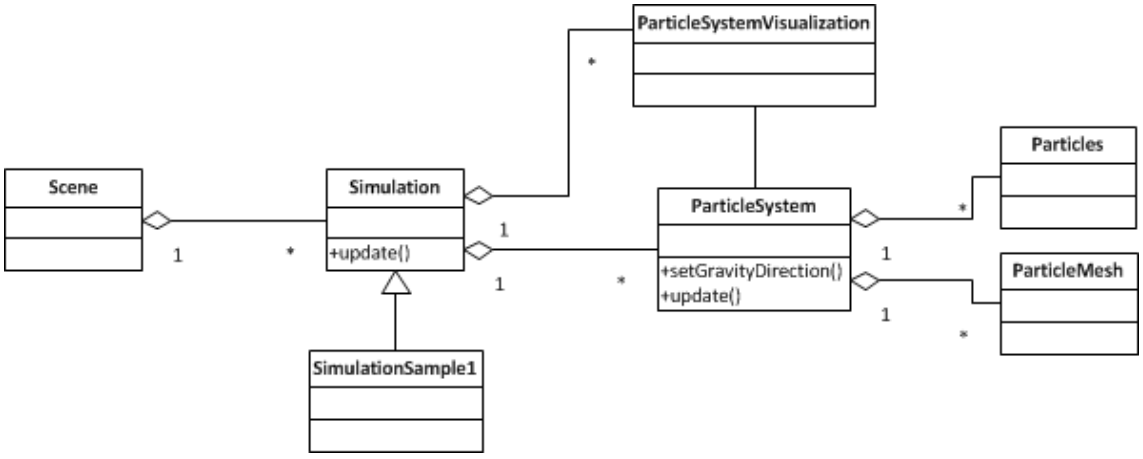

Рисунок 1: Диаграмма классов библиотеки физических расчетов

Главным объектом в структуре классов является класс Scene. Класс объявлен наследником класса Ogre::FrameListener и подписан на соответствующие события, есть возможность использовать шаг симуляции, согласованный с обновлением кадров на экране. Также класс подписан на события мыши и клавиатуры. В классе Scene находится указатель на класс Simulation, в котором непосредственно выполняется заданная симуляции. В классе Simulation

хранятся физические модели, графические модели, объекты, осуществляющие взаимодействие с пользователем, вспомогательные данные. В реализации используются классы — наследники класса Simulation, в которых прописаны собственные сценарии симуляции. Такая структура позволяет запускать разные примеры симуляции без необходимости перезапускать систему целиком: достаточно заменить активный класс-наследник Simulation на новый. Физическая модель реализована в классе ParticleSystem.

В классе ParticleSystem хранятся указатели на объекты классов ParticleMesh и Particles. В классе ParticleMesh хранится физическое представление трехмерной полигональной модели объекта. Физические представления разнесены в собственные объекты класса ParticleMesh с целью упрощения различения физических представлений разных полигональных моделей и упрощения доступа к позициям частиц. В классе Particles хранятся позиции и скорости частиц. Радиус и масса одинаковы для всех хранящихся частиц. Частицы вынесены в отдельный класс, чтобы упростить визуализацию частиц и обработку пересечений частиц. Для ускорения поиска пересечения частиц используется воксельное представление. На каждом шаге симуляции в классе ParticleSystem для каждого объекта класса ParticleMesh и объекта класса Particles сначала вычисляются новые временные позиции частиц, затем обрабатываются ограничения, затем обновляются позиции и скорости. Для обработки пересечения частиц из объекта класса Particles с полигонами трехмерной модели из класса ParticleMesh в классе ParticleSystem добавляется обработка пересечений частиц с полигонами, которая действует перед вызовом обработки ограничений этих классов.

4.2 Представление объекта

Трехмерная модель поверхности объекта представляется полигональной сеткой, записываемой как множество вершин и множество индексов треугольников (полигонов), ссылающихся на вершины. Физическая модель объекта использует вершины полигональной сетки для определения частиц, по ребрам полигонов задаются пружины — ограничения на расстояния между парами частиц. Масса каждой частицы вычисляется по массе, заданной объекту, умноженной на коэффициент — отношение суммы площадей полигонов, содержащих частицу, на суммарную площадь всех полигонов (площадь поверхности объекта). Объем замкнутой полигональной модели — сумма объемов тетраэдров, взятых со знаками плюс или минус (по каждому полигону строится тетраэдр: три вершины тетраэдра принадлежат треугольнику, четвертая вершина у всех общая, знак зависит от направления нормали треугольника относительно общей точки).

4.3 Используемые ограничения и силы

Для моделирования поведения мягкого тела в классе ParticleMesh выполняется обработка ограничений: сохранение расстояний между парами частиц (пружины) и сохранение объема замкнутого тела. Обработки пружин достаточно для сохранения формы объекта. Для сохранения объема ограничение наложено на все частицы физической модели и задано исходное значение объема. Действительное значение объема в данный момент времени пересчитывается по тем же полигонам. Для симуляции введения в тело дополнительного объема заданное исходное значение сохраняемого объема изменяется на вводимую величину, ограничение на сохранение объема сохраняет измененную величину. Дополнительно в физическую модель введено гидростатическое давление для имитации поведения тела, наполненного

жидкостью. Сила, влияющая на частицу, записывается так: $F = P\,A$, при этом $P = ro\,g\,h$, где ro — плотность жидкости, g — ускорение свободного падения, h — высота столба жидкости, A — площадь дна сосуда. В реализации A соответствует сумме площадей полигонов (содержащих частицу) деленной на три, h — длина отрезка, проведенного от частицы до ближайшей точки на полигонах поверхности по направлению, обратному направлению ускорения свободного падения, $g = 9.81$ м/с^2, ro — плотность жидкости, в поставленной задаче подходит от 600 до 1000 кг/м^3, направление силы совпадает с направлением ускорения свободного падения. При плотной сетке полигональной модели можно упростить реализацию: наклоном полигонов, содержащих частицу, можно пренебречь, для ускорения определения высоты столба жидкости достаточно найти полигон, на который проектируется позиция частицы, и взять ближайшую к частице вершину полигона, погрешностью можно пренебречь.

4.4 Особенности реализации

В алгоритме при первом обновлении скоростей применяется ускорение свободного падения. В реализации затухание (damping) заменено на коэффициент от 0 до 1, применяемый при последнем обновлении скоростей в итерации алгоритма. Для начальной стабилизации состояний физических моделей перед симуляцией несколько раз выполняется обработка ограничений без обновления скоростей, как показано в [21]. Симуляция останавливается при уменьшении ниже заданного порога изменения суммы абсолютных значений позиций по всем координатам. Шаг симуляции задан константой. Единицы измерения — килограмм, метр.

4.5 Примеры

Ниже представлены примеры работы библиотеки физических расчетов.

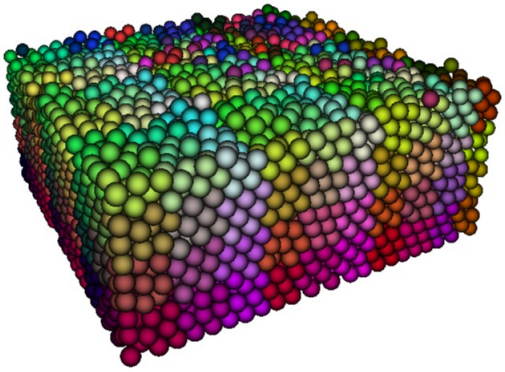

Рисунок 2: Обработка столкновений (8000 частиц)

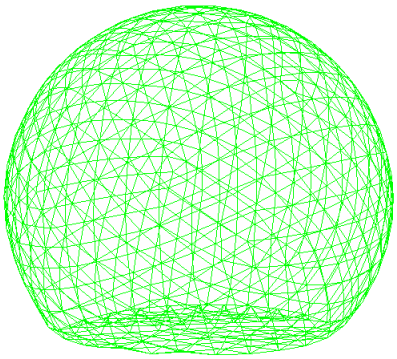

Рисунок 3: Сохранение объема,давление (шар)

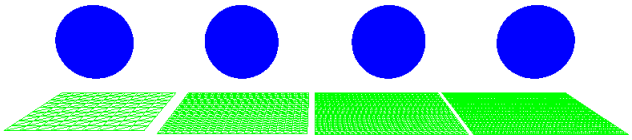

Рисунок 4: Пересечение ткани и частиц (1)

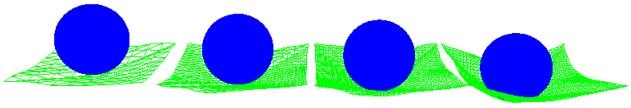

Рисунок 5: Пересечение ткани и частиц (2)

На рис. 2 изображено 8000 частиц (радиус 0.4, масса 0.286) в замкнутом пространстве, действует сила гравитации, производится обработка столкновений частиц, симуляция происходит в реальном времени (для ускорения поиска пересечений используются воксели, частицы из одного вокселя обозначены одним цветом). На рис. 3 изображен результат симуляции

падения шара (объем 0.00412) на плоскость, обрабатывается ограничение на сохранение объема, действуют гравитация, гидростатическое давление. Шар состоит из 642 вершин и 1280 полигонов, радиус 1, в начале симуляции сохраняется объем 0.00415, при действующей гравитации (когда шар находится на плоскости) сохраняемый объем немного уменьшается, при действующем гидростатическом давлении сохраняется объем, равный 0.0041. На рис. 4 изображены модели ткани с разной плотностью сетки и 4 шара над ними (масса ткани равна 10, распределена по частицам, радиус шара — 4, объем — 8.18, масса — 81, всего 3000 вершин и 5608 полигонов), на рис. 5 результат симуляции с действующей гравитацией, обработкой пружин и пересечений частиц с полигонами.

5 Модуль симуляции операции в системе планирования

5.1 О системе

Архитектуру системы планирования можно поделить на три уровня: уровень модулей, уровень общих компонент и уровень сторонних библиотек. Архитектура системы представлена на рис. 6. Каждый модуль первого уровня независим от других модулей и выполняет определенную задачу. Модули разделены на логику модуля, которая реализуется на C++, и пользовательский интерфейс, который реализуется на HTML и Javascript. На уровне общих компонент хранятся алгоритмы и структуры данных, доступные всем модулям. Уровень сторонних библиотек закрыт от прямого доступа для модулей. В систему встроена визуализация, работа с вводом и выводом данных, логирование, работа с потоками. Для разработки модулей системы используется Visual Studio 2010, проект находится под контролем версий в системе Mercurial.

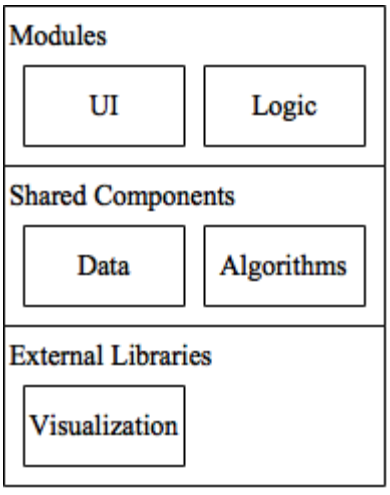

Рисунок 6: Архитектура системы

5.2 О модуле

Модуль в системе реализован как динамически подключаемая библиотека. Из библиотеки физических расчетов в модуле используется класс ParticleSystem. Этот класс обернут вспомогательным классом, в котором находятся данные, необходимые для симуляции (вершины и индексы трехмерной модели выделенной области и замкнутой области, индексы неподвижных вершин, исходные значения площади выделенной области, объема замкнутой области). При необходимости класс симуляции пересоздается с теми же данными, но другими параметрами симуляции, независимо от подготовленных данных.

С целью упрощения взаимодействия с пользователем моделирование операции разбито на последовательность шагов: выбор модели пациента, выделение области для симуляции, задание параметров симуляции, расчет и демонстрация результатов. Чтобы ускорить вычисления, физическая симуляция вынесена в отдельный поток.

5.3 Шаги симуляции

Для симуляции операции в модуле выделено четыре шага. На первом шаге необходимо выбрать трехмерную модель пациента, на втором шаге выделить маркерами область, на которой будет проводиться симуляция. При переходе со второго шага на третий запускается отдельный поток, в котором происходит построение замкнутой трехмерной модели по выделенной области. На третьем шаге необходимо задать параметры симуляции — величину объема в $см^3$, которую требуется ввести в выделенную область. При переходе с третьего на четвертый шаг запускается новый поток, в котором выполняются расчеты симуляции. По окончании расчетов к существующим моделям добавляется новая — копия выбранной на первом шаге модели с примененными результатами симуляции. С новой моделью сохраняются маркеры разметки

области, значения площади выделенной области и объема построенной замкнутой модели до симуляции и после, параметры симуляции, заданные на третьем шаге.

Для построения замкнутой трехмерной модели используются полигоны поверхностной модели, выделенные маркерами на втором шаге. В модуле реализовано два способа построения замкнутой трехмерной модели для симуляции. В первом способе добавляется новая вершина — центр масс. Для новых полигонов (треугольников) одна из вершин — центр масс, другие взяты на ребрах границы выделенной области. Во втором способе замкнутая область состоит из двух наборов вершин и полигонов: первый — вершины и полигоны выделенной области, второй — вершины и развернутые полигоны выделенной области. Объем такой замкнутой модели равен нулю. В зависимости от выбранного способа будут по-разному вычисляться силы, введенные для имитации тела, заполненного жидкостью: для частицы эта сила зависит от расстояния до ближайшей точки по направлению, обратному направлению ускорения свободного падения.

5.4 Особенности реализации

С целью организации взаимодействия с потоком, в котором производится процесс симуляции, в модуле хранится переменная с текущим состоянием процесса вычислений. На рис. 7 изображена диаграмма состояний процесса симуляции. Процесс симуляции может находиться в следующих состояниях: исходное состояние NULL — нет данных, READY — готов к началу симуляции, данные есть, PROCESS — происходят вычисления в новом потоке, COMPLETE — вычисления завершены успешно, необходимо сохранить результаты симуляции и перейти в состояние READY, RESET — вычисления прерваны, необходимо перейти в состояние READY без сохранения результатов

симуляции, DELETE — вычисления отменены, необходимо перейти в состояние NULL без сохранения результатов симуляции. Процесс симуляции находится в состояниях NULL и READY только в главном потоке, в состоянии PROCESS — при переходе из главного потока в новый и при выполнении расчетов в новом потоке, в состояниях COMPLETE, RESET и DELETE при переходе из нового потока в главный. Различение этих состояний необходимо для определения действий, выполняемых по окончании вычислений.

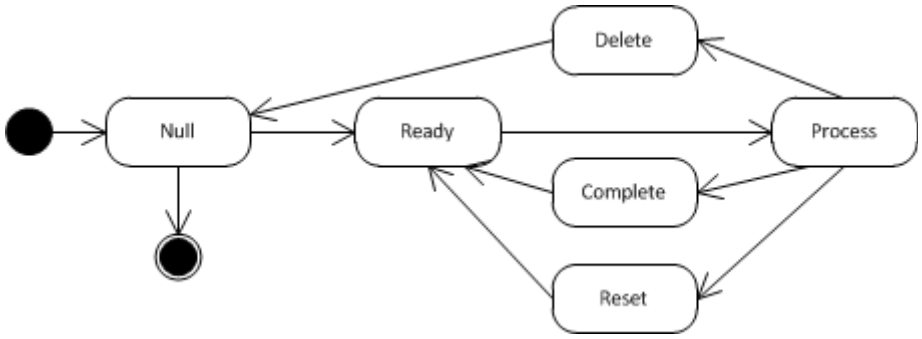

Рисунок 7: Диаграмма состояний процесса симуляции

Трехмерные модели объектов в системе планирования измеряются в дециметрах. Для проведения симуляции модели переводятся из дециметров в метры. Алгоритмы симуляции библиотеки физических расчетов в этом случае не требуют изменений. Результаты симуляции переводятся обратно в дециметры.

6 Апробация

С целью проверки работоспособности модуля симуляции результатов операции по трансплантации жировой ткани использовались трехмерные модели поверхности лица пациента до операции трансплантации жировой ткани и после операции. Модели получены с помощью пассивной фотограмметрии. Модели выравнены и масштабированы по краям глаз. Для симуляции операции в модуле выделены области слева и справа, как показано на рис. 8. На рис. 9 показаны трехмерная модель до операции (слева), после операции (посередине) и результат симуляции операции (справа). Разница объемов между трехмерными моделями, вычисленная (в другом модуле) по выделенной левой области, как показано на рис. 10, равна 12 см^3, по правой — 13 см^3. Для симуляции задан вводимый объем для левой области — 12 см^3, для правой области — 13 см^3.

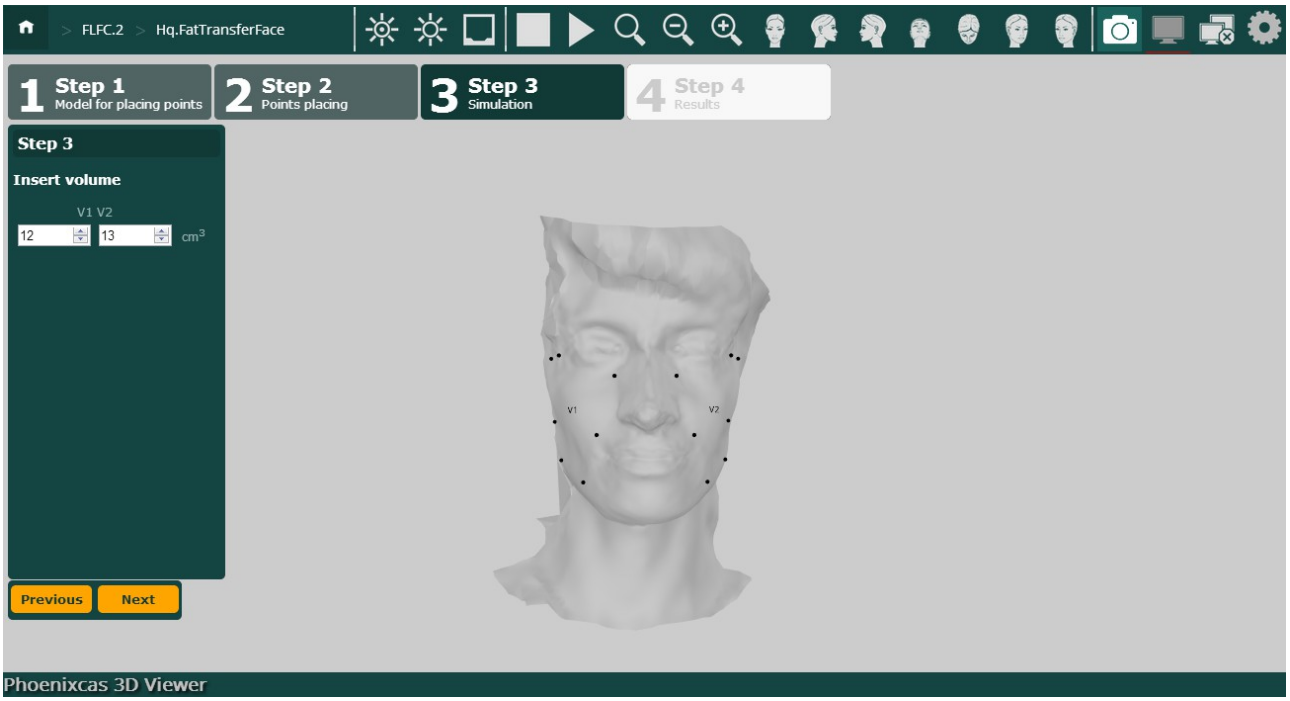

Рисунок 8: Третий шаг симуляции

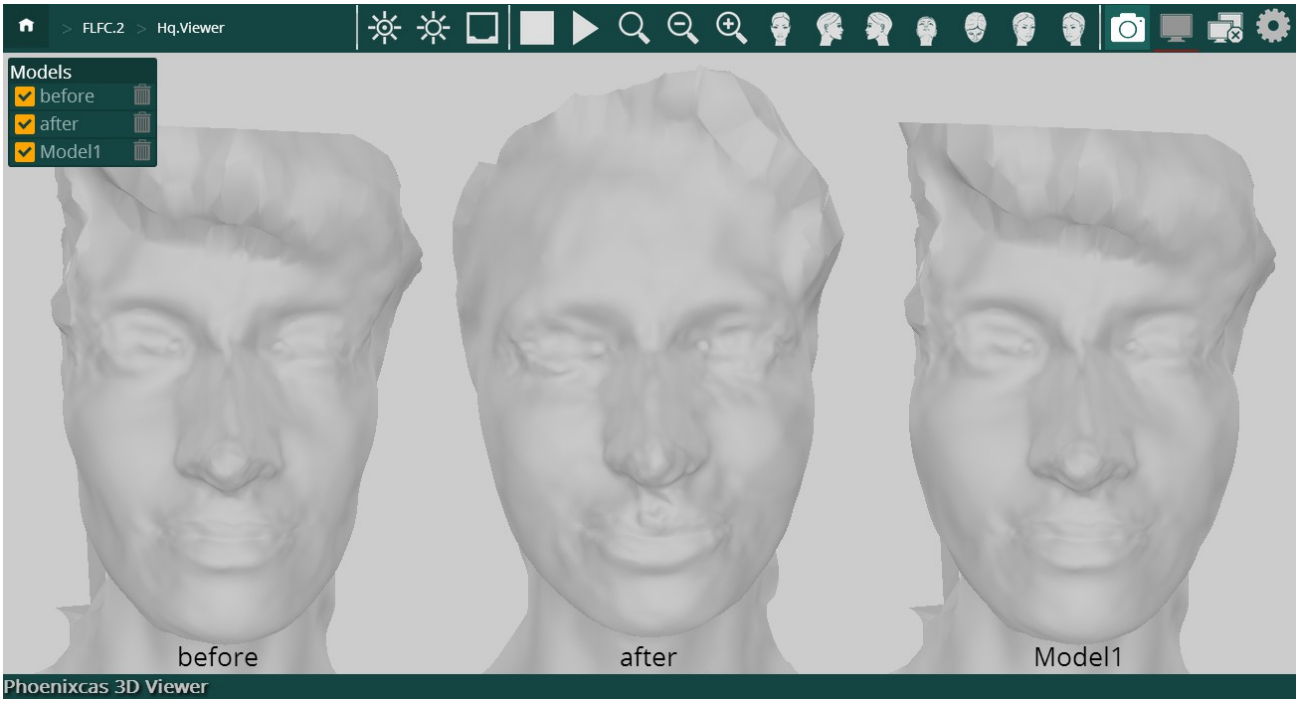

Рисунок 9: Модели до операции (before), после (after), результат симуляции (Model1)

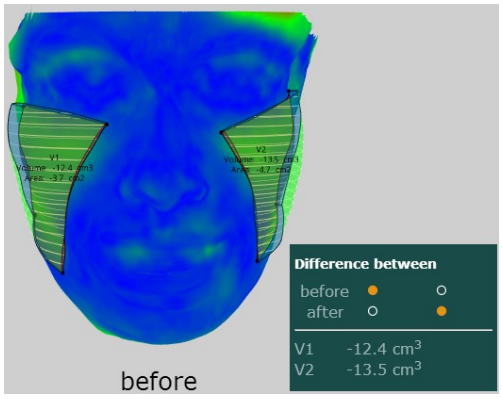

Рисунок 10: Разница объемов моделей before и after

Разница объемов между моделями before и Model1 в выделенной области слева равна около +13 см3, в другой области равна около +15 см3. Разница объемов между моделями after и Model1 в выделенной области слева равна около -1.0 см3, справа — около -1.1см3. На погрешность влияет выравненность моделей after и before и качество моделей. Выравнивание моделей затруднено неровностью восстановленных моделей, снимки для моделей получены в разных условиях.

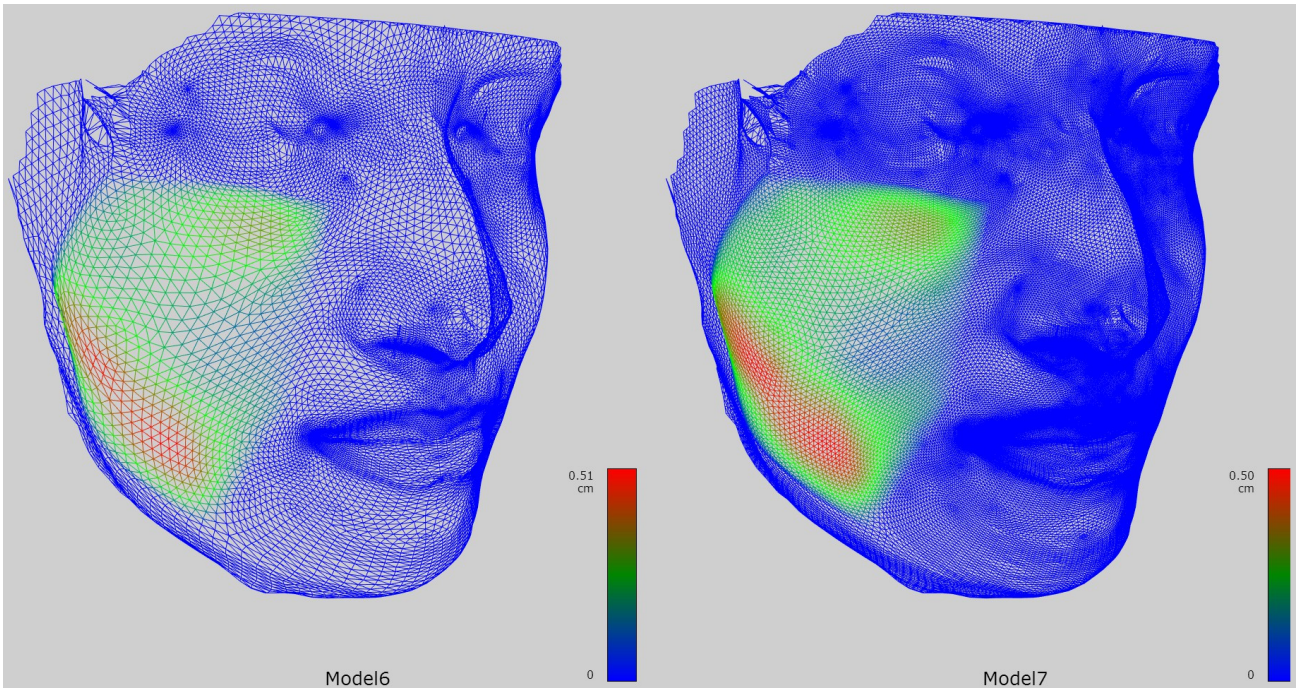

Рисунок 11: Симуляция на моделях с сеткой разной плотности

На рис. 11 приведен пример работы модуля на трехмерных моделях лица с разной плотностью сетки. Слева изображен результат симуляции, выполненный на выделенной области, содержащей 2375 вершин и 2093 полигона, справа — 8417 вершин и 7859 полигонов. Вторая модель — копия первой с разбитыми треугольниками (каждый треугольник разбит на четыре), области выделены по одному набору маркеров, в симуляции использованы пружины и сохранение объема, введено по 10 $см^3$, видимых различий нет, по метрике Хаусдорфа результаты моделирования совпадают (с незначительной погрешностью). При использовании гравитации область на более плотной трехмерной модели немного сдвигается. Симуляция на модели слева занимает примерно 5 секунд (около 300 итераций). Для более плотной модели необходимо большее время для стабилизации (дольше 15 секунд, более 600 итераций).

7 Результаты

В систему планирования добавлены возможности симуляции результатов операции по трансплантации жировой ткани в области лица по трехмерной модели поверхности лица. В ходе работы выполнены следующие задачи:

- Реализована физическая модель деформации мягкого тела
- Разработан модуль симуляции результатов операции
- Проведена апробация модуля

8 Список литературы

1: Петров А. Г., Анализ и манипулирование геометрией формы 3d скана человека, 2010
2: Eberly D. H., Game Physics, 2004
3: Jakobsen T., Advanced character physics, 2001
4: Nealen A. et al., Physically based deformable models in computer graphics, 2005
5: Vassilev I. T., Spanlang B., A mass-spring model for real time deformable solids, 2002
6: Николаев С. Н., Нелинейная система масс-с-пружинками для моделирования больших деформаций мягких тканей, 2013
7: Matyka M., Ollila M., Pressure model of soft body simulation, 2003
8: Mesit J., Guha R., Chaudhry S., 3d soft body simulation using mass-spring system with internal pressure force and simplified implicit integration, 2007
9: Müller M., Schirm S., Teschner M., Interactive blood simulation for virtual surgery based on smoothed particle hydrodynamics, 2003
10: ANSYS, www.ansys.com
11: ABAQUS, www.3ds.com/products-services/simulia/products/abaqus/
12: ADINA, www.adina.com
13: FEBio, febio.org
14: SOFA, www.sofa-framework.org
15: Erleben K., Sporring J., Dohlmann H., Opentissue-an open source toolkit for physics-based animation, 2005
16: Canfield, canfieldsci.com
17: Crisalix, www.crisalix.com
18: Müller M. et al., Position Based Dynamics, 2006
19: Bender J. et al., A Survey on Position-Based Simulation Methods in Computer Graphics, 2014
20: Position Based Dynamics library, github.com/janbender/PositionBasedDynamics
21: Macklin M. et al., Unified particle physics for real-time applications, 2014