

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

Система верификационного тестирования в задаче построения трехмерных
моделей и мягких тканей

Бакалаврская работа

Монькин Сергей Александрович

Зав. кафедрой:

Д.ф.-м.н. профессор Терехов А.Н.

Научный руководитель:

Аспирант Петров А.Г.

Рецензент:

Ст. преподаватель Антипов И.Г.

Санкт-Петербург

2013

SAINT-PETERSBURG STATE UNIVERSITY

Mathematics&Mechanics Faculty

Chair of Software Engineering

Verification testing system in the task of constructing three-dimensional models
and soft tissues

Bachelor's Thesis

Monkin Sergei

Head of department:

Professor Terekhov A.N.

Scientific advisor:

Graduate Assistant Petrov A.G.

Reviewer:

Senior Lecturer Antipov I.G.

Saint-Petersburg

2013

Оглавление

Введение.....	2
Постановка задачи.....	4
Верификационное тестирование	5
Сравнение трехмерных моделей.....	5
Построение трехмерных моделей.....	7
Реализация.....	9
Система верификационного тестирования.....	9
Сравнение трехмерных моделей	11
Поточечное сравнение.....	11
Сравнение дескрипторов.....	13
Построение трехмерных моделей	15
Глобальная деформация	16
Локальная деформация.....	18
Тестирование	20
Заключение	24
Ссылки.....	25

Введение

С увеличением сложности разрабатываемого программного обеспечения и с расширением сфер применения программных средств увеличивается число рисков, связанных с работой систем, зависящих от этих программных средств. С целью обеспечения качества разрабатываемого программного обеспечения вводится множество мер в процессы разработки, что позволяет добиться высокого качества программного обеспечения, но увеличивает время и стоимость разработки. Одной из таких мер является тестирование, заключающееся в различных видах исследования программного обеспечения.

Проверка корректности работы систем является важным этапом в разработке программного обеспечения. Среди подходов к тестированию выделяются статическое и динамическое тестирования. Отдельно выделяются функциональное, модульное, верификационное виды тестирования, тестирование производительности и другие.

Верификационное тестирование представляет собой проведение ряда запусков системы с целью определения корректности ее работы. Верификационное тестирование отличается от других видов тестирования тем, что производится тестирование всей системы на корректность выдаваемого ею результата. Система верификационного тестирования автоматизирует проверку корректности работы системы.

Система верификационного тестирования реализуется в рамках общей задачи проверки корректности работы алгоритмов, моделирующих поведение мягких тканей. Для подготовки системы верификационного тестирования необходимо определить интерфейсы входных и выходных данных проектов, реализующих алгоритмы, а также определить алгоритмы сравнения выходных данных. Для проверки работоспособности системы верификационного тестирования решается частная задача, заключающаяся в

проверке корректности работы алгоритмов, реализующих построение трехмерных моделей.

Для построения трехмерной модели объекта могут быть использованы различные подходы. Существует множество средств для ручного создания трехмерных моделей любой формы и сложности (Blender [1]), но в различных задачах, где требуется наличие большого числа однотипных моделей, такой подход является достаточно медленным. При определенных требованиях к модели можно воспользоваться средствами автоматического построения трехмерных моделей, основанных на восстановлении моделей по серии снимков объекта (123D Catch [2]), с помощью устройств лазерного сканирования, устройств сканирования на основе структурированного света (DAVID Laserscanner [3]).

Постановка задачи

Реализовать систему верификационного тестирования для автоматизации проверки корректности реализации и сравнения алгоритмов. Для этого определить интерфейсы входных и выходных данных для запуска и сравнения результатов работы алгоритмов.

Для проверки работоспособности системы верификационного тестирования провести тестирование работы алгоритма, реализующего фотограмметрический подход к построению трехмерных моделей. Реализовать построение трехмерных моделей на основе параметрического подхода. Сравнить результаты работы алгоритмов, реализующих фотограмметрический и параметрический подходы к построению трехмерных моделей.

1. Реализовать систему верификационного тестирования
 - a. Реализовать модули сборки проектов, запуска, тестирования
 - b. Определить интерфейсы взаимодействия системы с тестируемыми проектами
2. Реализовать алгоритмы сравнения трехмерных моделей
3. Провести верификационное тестирование алгоритмов, реализующих фотограмметрический и параметрический подходы к построению трехмерных моделей
 - a. Реализовать систему построения трехмерных моделей на основе параметрического подхода
 - b. Провести тестирование алгоритмов

Верификационное тестирование

Верификационное тестирование [4] относится к видам тестирования, целью которых является проверка работы тестируемой системы в целом, корректности выполнения основных функций. Тестирование представляет собой проведение ряда проверок системы на определение корректности результата ее работы. Система запускается на определенных входных данных, корректность работы определяется соответствием выдаваемого системой результата ожидаемому результату. Определение корректности получаемого результата можно определить различными способами. Среди них можно выделить сравнение результата с подготовленными данными, сравнение результата с результатом работы другой системы, выполняющей те же действия. В первом случае подготавливается множество тестовых данных и множество соответствующих тестовым ожидаемых данных. Тестирование системы происходит сравнением результатов работы системы с ожидаемыми данными. Во втором случае тестирование системы происходит сравнением выдаваемого результата работы данной системы с результатом работы другой системы.

Система верификационного тестирования автоматизирует проверку корректности работы системы для ее многократного проведения.

Сравнение трехмерных моделей

Распространенным способом представления трехмерных моделей является представление полигональными структурами (полигональными сетками). Полигональная структура задается множеством вершин (точек в трехмерном пространстве), ребер между парами вершин, граней (полигонов), образованных тремя или более вершинами. Сравнить трехмерные модели, представленные полигональной структурой, можно различными способами.

При этом важно учитывать, что модели могут отличаться положением в пространстве, иметь разное количество вершин, полигонов.

Одним из подходов к сравнению трехмерных моделей является поточечное сравнение. Сравнение пары моделей происходит в два этапа: на первом происходит сопоставление в пространстве одной модели к другой, на втором происходит поточечное сравнение. Сопоставление трехмерных моделей в пространстве происходит преобразованием положения в пространстве одной модели относительно другой. Преобразование заключается в растяжении, повороте и переносе одной модели таким образом, чтобы ее положение в пространстве как можно менее отличалось от положения в пространстве другой модели.

Поточечное сравнение происходит следующим образом: для каждой вершины первой модели в определенном радиусе по вершинам второй модели, попавшим в этот радиус, вычисляется средняя точка; затем вычисляется расстояние от вершины первой модели до средней точки. По множеству таким образом вычисленных для каждой вершины первой модели расстояний определяется сходство моделей.

Среди подходов к сравнению трехмерных моделей выделяется подход, основанный на сравнении дескрипторов формы [5]. Сравнение происходит следующим образом: для каждой сравниваемой модели вычисляется множество точек на поверхности модели, по ним с помощью функций формы вычисляется дескриптор формы, строятся гистограммы, сравнение моделей происходит сравнением гистограмм. Сравнение гистограмм происходит различными способами. Например, как сравнение функций распределения [6], используя евклидово расстояние.

Точность сравнения зависит от количеств построенных точек на поверхностях моделей, детальности гистограмм.

Построение трехмерных моделей

Автоматизированное построение трехмерных моделей применяется в различных областях, таких как археология, медицина. При выборе способа построения трехмерных моделей одним из важнейших факторов является точность восстанавливаемой модели относительно моделируемого объекта и поставленной задачи.

Среди подходов к автоматизированному построению трехмерных моделей выделяются фотограмметрический и параметрический (на основе деформации порождающей модели) [7].

При фотограмметрическом подходе создается трехмерная сцена по серии фотографий (например, [8]). Фотографии представляют собой последовательность снимков сцены, сделанных с разных ракурсов при условии, что сцена в последовательности снимков изменяется незначительно. Некоторые алгоритмы используют две или одну фотографии. На фотографиях детектируются характерные точки, вычисляются их положения в пространстве сцены. Затем происходит сопоставление (*matching*) идентифицированных точек. По этим точкам строится поверхность, вычисляются текстурные координаты для наложения текстуры на поверхность. Сцена может быть также представлена облаком точек, в этом случае вершинам могут быть назначены цвета. Качество восстановления трехмерной модели зависит от качества используемых изображений (шум, искажения), сложности восстанавливаемого объекта (отражающие поверхности, мелкие детали).

Второй из выделенных способов автоматизированного построения трехмерных моделей основан на использовании шаблонного объекта (например, [9], [10]). По фотографиям вычисляются точки, по которым деформируется шаблонный объект, принимая форму моделируемого объекта. Точки выделяются вручную (возможна некоторая автоматизация выделения

точек на основе обработки изображений). По ним вычисляются новые положения вершин шаблонного объекта в зависимости от их удаленности от этих точек. По точкам также вычисляются текстурные координаты для наложения текстуры. Качество восстановления трехмерной модели зависит от используемого шаблонного объекта, количества выделенных точек.

Реализация

Система верификационного тестирования

Система верификационного тестирования реализована как часть системы непрерывной интеграции Jenkins [11]. Jenkins является бесплатной системой, позволяет осуществлять сборку проектов, производить различные виды тестирования. Для него разработано множество различных дополнений, расширяющих его функциональные возможности и позволяющих работать с различными системами, возможно использование собственных дополнений.

Система верификационного тестирования разделена на три модуля (рис. 1): модуль сборки (CompileModule), модуль запуска (RunModule), модуль тестирования (TestingModule). Модуль сборки производит сборку проектов, отправляет в модуль тестирования исполняемые файлы (1), передает управление модулю запуска (2). Модуль запуска запускает тестирование (3), получает результаты тестирования от модуля тестирования (4).

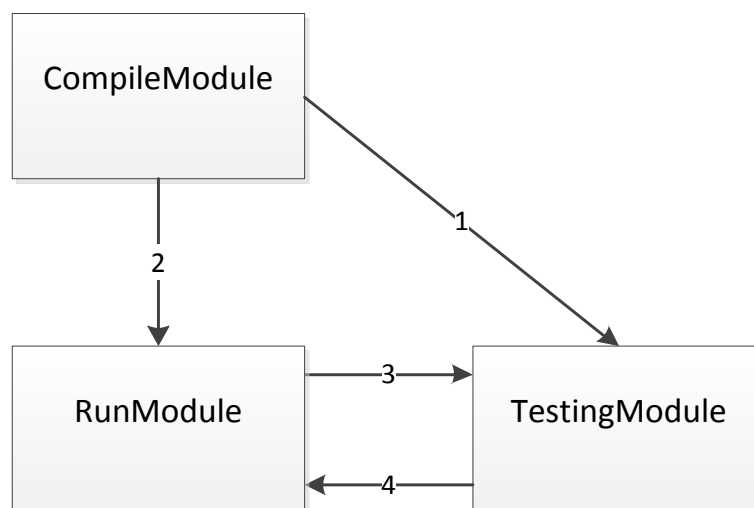


Рис. 1. Архитектура системы верификационного тестирования

Взаимодействие модулей с тестируемыми проектами описаны в конфигурационных файлах, которые содержат параметры сборки проектов,

план тестирования, параметры запуска проектов. Выходными данными проектов являются трехмерные модели. Сборка начинается только при наличии конфигурационного файла.

Ниже (пример 1) представлен пример конфигурационного файла (файл XML [12]), описывающего план тестирования. В файле записаны планы тестирования (“compare”), пути к конфигурационным файлам тестируемых проектов (“project”) относительно конфигурационного файла плана тестирования, метрики сравнения (“metric”). В данном примере сравниваются проекты “project1” с “project2” вычислением сходства площадей поверхностей моделей и “project3” с “project4” поточечным сравнением моделей.

```
<configuration>
  <compare>
    <project>project1\project1.xml</project>
    <project>project2\project2.xml</project>
    <metric>Square</metric>
  </compare>

  <compare>
    <project>project3\project3.xml</project>
    <project>project4\project4.xml</project>
    <metric>Match</metric>
  </compare>
</configuration>
```

Пример 1. Пример плана тестирования

Модуль сборки производит сборку тестируемых проектов по параметрам, записанным в конфигурационном файле. Затем составляется конфигурационный файл, содержащий параметры запуска и тестирования. После сборки исполняемые файлы проектов и составленный конфигурационный файл отправляются в модуль тестирования, передается

управление модулю запуска. Осуществляется запуск модуля тестирования, получение и хранение результатов тестирования.

Модуль тестирования производит тестирование программ, полученных от модуля сборки. По конфигурационному файлу, содержащему параметры запуска и тестирования, производится запуск программ для получения выходных данных и тестирование полученных выходных данных.

Сравнение трехмерных моделей

Поточечное сравнение

Поточечное сравнение двух трехмерных моделей происходит в два этапа: сопоставление моделей и определение их сходства. На первом этапе происходит преобразование положения в пространстве одной модели в соответствии с положением в пространстве второй модели. Преобразование происходит с помощью алгоритма ICP [13] следующим образом:

1. Выделить ближайшие вершины между двумя моделями.
2. Вычислить параметры преобразования по ближайшим вершинам.
3. Применить преобразование к вершинам первой модели.
4. Вычислить ошибку между первой моделью и ближайшими вершинами.
5. Повторять шаги 1-4, пока уменьшается ошибка.

Выделение ближайших вершин между двумя моделями происходит следующим образом: для каждой точки первой модели вычисляется ближайшая точка второй модели таким образом, чтобы одной вершине первой модели соответствовала одна вершина второй модели (если вершине второй модели соответствует несколько вершин первой модели, среди них выбирается ближайшая). Затем по выделенным вершинам происходит

вычисление преобразований растяжения, поворота, переноса. Вычисление происходит способом, описанным для преобразования положения шаблонной модели в пространстве по контрольным точкам при построении трехмерных моделей на основе параметрического подхода [9]. К первой модели применяются вычисленные преобразования, ошибка вычисляется между вершинами преобразованной модели (V) и ближайшими к ней вершинами второй модели (C) по формуле (1). Шаги алгоритма повторяются, пока уменьшается ошибка.

$$err(V, C) = \sum_i \|V_i - C_i\|^2, \quad (1)$$

где $\|V_i - C_i\|^2$ вычисляется как евклидово расстояние между двумя вершинами V_i и C_i .

В результате изменяется положение первой модели в пространстве таким образом, что оно наименее отличается от положения в пространстве второй модели (рис. 2).

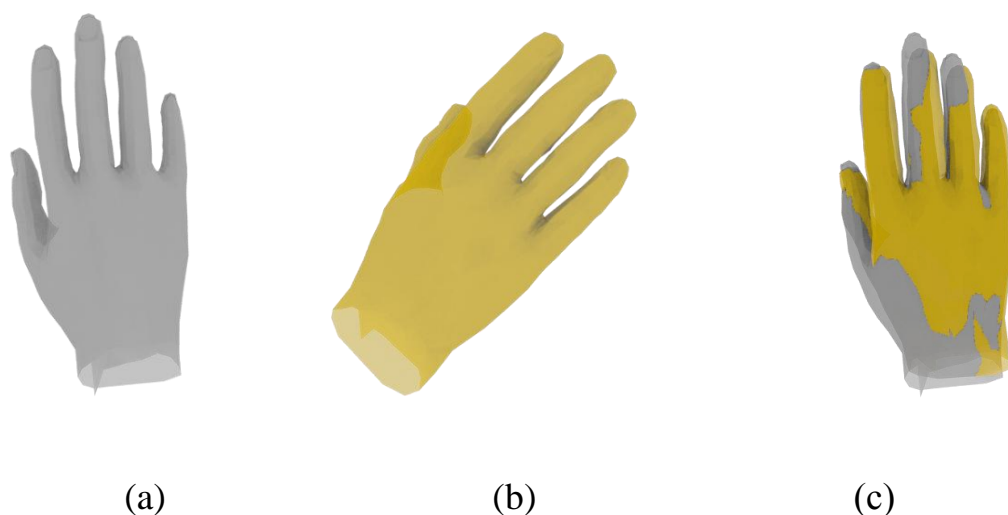


Рис. 2. Пример двух трехмерных моделей, отличающихся положением в пространстве (a, b) и сопоставления с помощью алгоритма ICP (c)

На втором этапе происходит поточечное сравнение моделей. Сравнение трехмерных моделей происходит вычислением меры сходства в каждой вершине первой модели. Вычисление происходит следующим образом: выделяется вершина первой модели; для нее в определенном радиусе выделяется множество вершин второй модели; по выделенным вершинам вычисляется средняя точка; затем вычисляется расстояние от вершины первой модели до средней точки. Расстоянием определяется сходство моделей в данной вершине. Шаги повторяются для каждой точки первой модели.

Ниже (рис. 3) представлен пример результата поточечного сравнения моделей, изображенных на рис. 2. Здесь показано, как первая модель отличается от второй: зеленым выделены участки максимального совпадения, красным – превышение порогового значения.

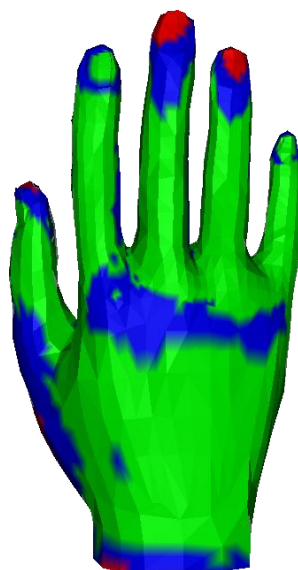


Рис. 3. Пример результата поточечного сравнения трехмерных моделей
(зеленым выделены участки максимального совпадения)

Сравнение дескрипторов

Сравнение трехмерных моделей сравнением дескрипторов происходит следующим образом (по [5]): для каждой сравниваемой модели вычисляется

множество точек на поверхности модели, по ним вычисляется дескриптор формы, гистограммы распределений (рис. 4). Сравнение моделей заключается в сравнении этих гистограмм.

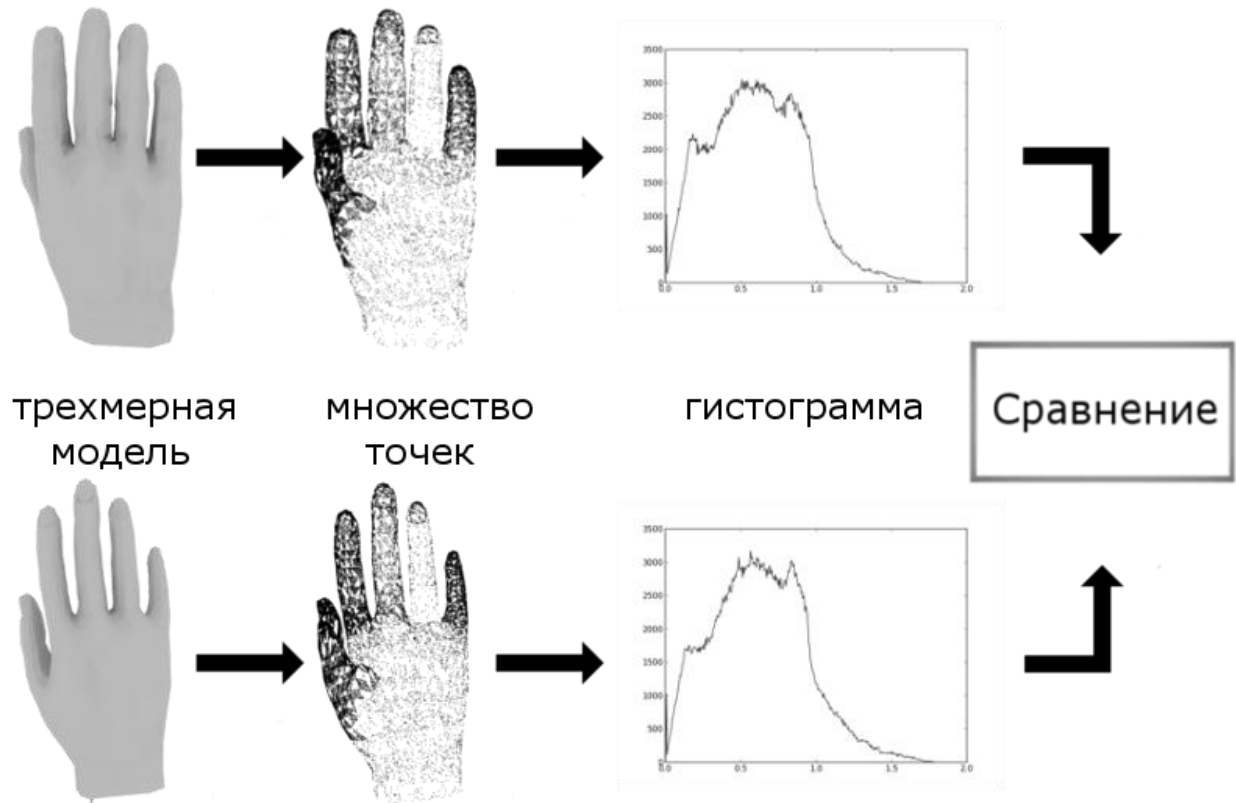


Рис. 4. Сравнение трехмерных моделей

Множество точек на поверхности модели вычисляется следующим образом:

1. Для каждого полигона по его площади вычисляется количество точек (n), которое будет построено для данного полигона.
2. Для каждого полигона (треугольника) вычисляются n произвольных точек на его поверхности. Точки (p) вычисляются по следующей формуле [5]:

$$p = (1 - \sqrt{r_1}) * A + \sqrt{r_1} * (1 - r_2) * B + \sqrt{r_1} r_2 * C, \quad (2)$$

где A , B , и C – вершины треугольника, r_1 и r_2 – произвольные числа от 0 до 1.

Дескриптор формы вычисляется по построенным точкам с помощью функций формы. Выбраны следующие функции: $D2$, для каждой пары точек вычисляющая евклидово расстояние между ними, $D2a$ [5], для каждой пары полигонов модели вычисляющая отношение полигона с меньшей площадью поверхности к полигону с большей площадью поверхности, $D3$, вычисляющая для каждой трех точек квадратный корень площади поверхности образованного ими треугольника [14].

По вычисленным с помощью функций формы значениям строятся гистограммы распределения. Гистограммы строятся следующим образом: по горизонтали – интервалы принимаемых функцией значений, по вертикали – количества значений, вычисленных по функции от построенных точек. Сравнение гистограмм осуществляется с помощью расстояния Евклида:

$$S(f, g) = \sum_i \|f_i - g_i\|, \quad (3)$$

где f , g – гистограммы двух сравниваемых моделей.

Построение трехмерных моделей

Построение трехмерных моделей на основе параметрического подхода реализовано на основе статьи [9]. Используется шаблонная модель, контрольные точки шаблонной модели, контрольные точки, вычисленные по фотографиям. Шаблонная модель деформируется в соответствии с контрольными точками фотографий. Контрольные точки шаблонной модели используются для преобразования положения модели в пространстве и для вычисления ошибки, используемой для определения остановки работы алгоритма.

Перед построением моделей по фотографиям с помощью графического интерфейса выделяются контрольные точки.

Построение происходит в два этапа: глобальная деформация, локальная деформация. На первом этапе (глобальная деформация) происходит преобразование положения шаблонной модели в пространстве в соответствии с выделенными контрольными точками. Преобразование основано на вычислении сингулярного разложения матрицы (singular value decomposition). На втором этапе (локальная деформация) происходит деформация шаблонной модели с использованием радиальных базисных функций в соответствии с выделенными контрольными точками.

Глобальная деформация

На этапе глобальной деформации происходит преобразование шаблонной модели контрольными точками модели в соответствии с контрольными точками, вычисленными по фотографиям (преобразования растяжения, поворота и переноса). Преобразование положения шаблонной модели происходит таким образом, чтобы как можно ближе сопоставить контрольные точки модели с контрольными точками, вычисленными по фотографиям. Этот этап является подготовительным перед этапом деформации шаблонной модели.

Для преобразования положения шаблонной модели по контрольным точкам вычисляются коэффициент растяжения, матрица поворота и вектор переноса. Затем они применяются к шаблонной модели (к каждой вершине) и контрольным точкам модели, изменяя их положения в пространстве. Применение происходит по следующей формуле:

$$v_t = S * v * R + T, \quad (4)$$

где v_t – преобразованная вершина, v – вершина шаблонной модели, S – коэффициент растяжения, R – матрица поворота, T – вектор переноса.

Преобразование происходит, пока минимизируется ошибка между контрольными точками модели и контрольными точками, вычисленными по фотографиям. Ошибка вычисляется по формуле (1).

Алгоритм преобразования шаблонной модели контрольными точками модели (P_t) и контрольными точками, вычисленными по фотографиям (P_c), выглядит следующим образом:

1. Вычислить для P_t и P_c координаты центров ($mean$) в пространстве ($c_0 = mean(P_c)$ для P_c).
2. Переместить P_t и P_c в начало координат ($P_{c0} = P_c - c_0$ для P_c).
3. Вычислить нормы P_t и P_c ($\|P_{c0}\|^2$ для P_c).
4. Нормализовать P_t и P_c ($P_{c0} = P_c / \|P_{c0}\|^2$ для P_c).
5. Вычислить матрицу A : $A = P_{c0}^t * P_{t0}$.
6. Вычислить сингулярное разложение матрицы A : $L, D, M = SVD(A)$.
7. Вычислить матрицу поворота R : $R = M * L^t$.
8. Вычислить коэффициент растяжения S : $S = \sum diagonal(D) * \frac{\|P_c\|^2}{\|P_t\|^2}$.
9. Вычислить вектор переноса T : $T = mean(P_{c0}) - S * mean(P_{t0}) * R$.
10. Преобразовать контрольные точки по формуле (4).
11. Преобразовать шаблонную модель по формуле (4).

Локальная деформация

На этапе локальной деформации происходит преобразование шаблонной модели радиальными базисными функциями в соответствии с контрольными точками, вычисленными по фотографиям. К каждой вершине шаблонной модели применяется формула (5), изменяющая положение вершины в пространстве, где R – радиальная базисная функция, w – матрица весов, v – вершина шаблонной модели, p_i – контрольная точка, вычисленная по фотографиям, N – число контрольных точек.

$$f(v) = \sum_{i=1}^N w_i * R(\|v - p_i\|) \quad (5)$$

Формула применяется отдельно для каждой координаты вершины:

$$f(v_j) = \sum_{i=1}^N w_i^j * R(\|v - p_i\|) \quad (6)$$

Матрица весов w представляет собой три вектора, вычисляемых по формуле (7) для каждой координаты x , y , z , где w^j – вычисляемый вектор весов (j – координаты x , y , z), H^{-1} – обратная матрица к матрице H , вычисляемой по формуле (8) от вектора контрольных точек, вычисленных по фотографиям, p^j – вектор контрольных точек, вычисленных по фотографиям (j – координаты x , y , z).

$$w^j = H^{-1} * p^j \quad (7)$$

$$H = \begin{bmatrix} R(\|p_1 - p_1\|) & \dots & R(\|p_1 - p_N\|) \\ \dots & \dots & \dots \\ R(\|p_N - p_1\|) & \dots & R(\|p_N - p_N\|) \end{bmatrix} \quad (8)$$

Выбор радиальной базисной функции влияет на деформацию шаблонной модели. Для реализации используется следующая функция (ε – константа):

$$R(d) = \frac{1}{\sqrt{1+(\varepsilon d)^2}} \quad (9)$$

Локальная деформация происходит, пока минимизируется ошибка (1) между контрольными точками модели и контрольными точками, вычисленными по фотографиям.

Тестирование

Произведено тестирование проектов, реализующих построение трехмерных моделей фотограмметрическим и параметрическим способами в системе верификационного тестирования.

Был подготовлен конфигурационный файл (пример 2) с описанием плана тестирования. В данном примере записаны пути к конфигурационным файлам тестируемых проектов, метрики сравнения (“Match” – поточечное сравнение моделей, “D2” – сравнение дескрипторов моделей по функции “D2”).

```
<configuration>
  <compare>
    <project>Photogrammetric\photogrammetric.xml</project>
    <project>Parametric\parametric.xml</project>
    <metric>Match</metric>
    <metric>D2</metric>
  </compare>
</configuration>
```

Пример 2. Конфигурационный файл плана тестирования проектов, реализующих построение трехмерных моделей фотограмметрическим и параметрическим способами

Для каждого проекта были подготовлены конфигурационные файлы, содержащие параметры их сборки и запуска. Проекты были подготовлены таким образом, чтобы можно было запустить их из системы верификационного тестирования, не настраивая ничего вручную.

Ниже представлены трехмерные модели, построенные фотограмметрическим (рис. 5) и параметрическим (рис. 6) способами. Трехмерная модель, построенная фотограмметрическим способом, состоит из 20 тысяч вершин.

Для построения трехмерной модели параметрическим способом использована модель, состоящая из 3.5 тысяч вершин.



Рис. 5. Трехмерная модель, построенная фотограмметрическим способом

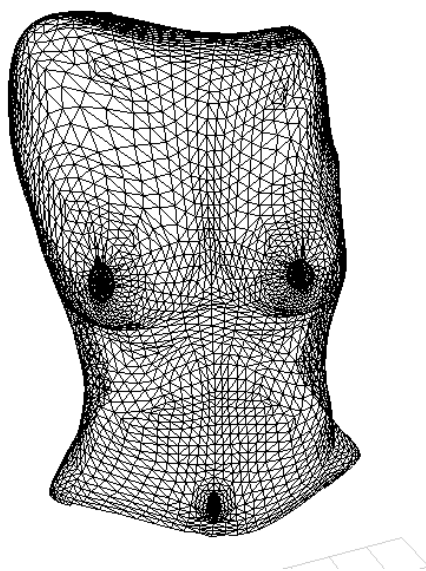


Рис. 6. Трехмерная модель, построенная параметрическим способом

Ниже (рис. 7) представлен результат поточечного сравнения трехмерных моделей, построенных фотограмметрическим и параметрическим способами. Определялось сходство по вершинам модели, построенной фотограмметрическим способом. Для радиуса, по которому вычислялись расстояния между вершинами моделей, выделены участки совпадения моделей: расстояние между точками от 0 до $1/3$ радиуса (зеленый цвет), расстояние между точками от $1/3$ до $2/3$ радиуса (синий цвет), расстояние между точками, превышающее $2/3$ радиуса (красный цвет).

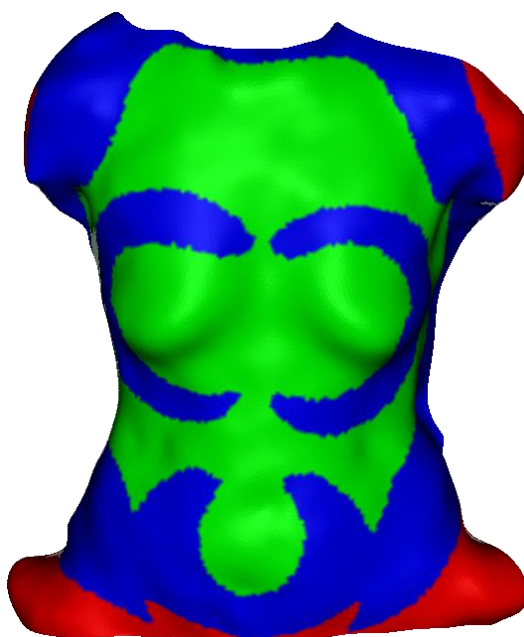


Рис. 7. Поточечное сравнение трехмерных моделей

Ниже (рис. 8) представлены гистограммы, построенные с помощью функций “D2” по трехмерным моделям, построенным фотограмметрическим и параметрическим способами.

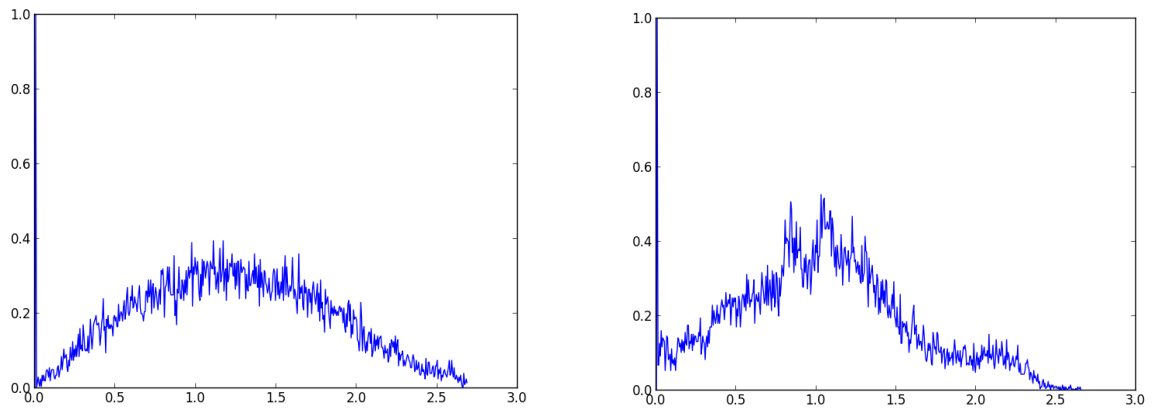


Рис. 8. Гистограммы, вычисленные по трехмерным моделям, построенным фотограмметрическим (слева) и параметрическим (справа) способами

Заключение

Реализована система верификационного тестирования: определена архитектура системы, реализованы модули сборки проектов, запуска, тестирования, определены интерфейсы входных, выходных данных. Для проверки работоспособности системы проведено тестирование алгоритмов построения трехмерных моделей: реализованы алгоритмы сравнения трехмерных моделей, реализована система построения трехмерных моделей на основе параметрического подхода, проведено тестирование.

1. Реализована система верификационного тестирования
2. Реализованы алгоритмы сравнения трехмерных моделей
3. Проведено верификационное тестирование алгоритмов, реализующих фотограмметрический и параметрический подходы к построению трехмерных моделей

Ссылки

- [1] Blender. <http://www.blender.org>.
- [2] 123D Catch. <http://www.123dapp.com>.
- [3] DAVID Laserscanner. <http://www.david-laserscanner.com>.
- [4] Кулямин В.В. Методы верификации программного обеспечения. 2008.
- [5] Levi C. Monteverde, Conrado R. Ruiz Jr., Zhiyong Huang. A Shape Distribution for Comparing 3D Models. 2007.
- [6] Грудинин С.Н. Сравнение трехмерных объектов. Критерии оценки сходства. 2011.
- [7] R. Szeliski. <http://szeliski.org/Book>.
- [8] D. Bradley, T. Boubekeur, W. Heidrich. Accurate multi-view reconstruction using robust binocular stereo and surface meshing. 2008.
- [9] Youngjun Kim, Kunwoo Lee, Wontae Kim. Image-based 3d torso body modeling. 2008.
- [10] W. Lee and N. Magnenat-Thalmann. Fast head modeling for animation. 2000.
- [11] Jenkins. <http://www.jenkins-ci.org>.
- [12] XML. <http://www.w3.org/XML>.
- [13] Michael Wild. Recent Development of the Iterative Closest Point (ICP) Algorithm. 2010.
- [14] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin. Matching 3D Models with Shape Distributions. 2001.