

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ

Математико-механический факультет  
Кафедра системного программирования

Павлов Сергей Николаевич

Регистрация трехмерных моделей с помощью дескрипторов  
формы

Бакалаврская работа

Допущена к защите.

Зав. кафедрой:

д.ф.-м.н., профессор Терехов А. Н.

Научный руководитель:

к.ф.-м.н., доцент Вахитов А. Т.

Рецензент:

ген. директор ООО “Биомоделирование” Петров А. Г.

Санкт-Петербург

2014

SAINT-PETERSBURG STATE UNIVERSITY

Mathematics&Mechanics Faculty

Software Engineering Department

Pavlov Sergey

3D model registration with shape descriptors

Bachelor's Thesis

Admitted for defence.

Head of department:

Professor Terekhov A. N.

Scientific advisor:

Associate Prof. Vakhitov A.T.

Reviewer:

Petrov A. G.

Saint-Petersburg

2014

# Оглавление

<b>Введение.....</b>	<b>4</b>
<b>Постановка задачи .....</b>	<b>11</b>
<b>Теоретическое обоснование.....</b>	<b>13</b>
Трехмерные модели.....	13
Метод 4-Point Congruent Sets для совмещения моделей .....	15
Альтернативные методы регистрации и совмещения моделей .....	17
Point Cloud Library.....	20
<b>Требования к реализации алгоритма.....</b>	<b>22</b>
<b>Особенности реализации .....</b>	<b>23</b>
<b>Апробация .....</b>	<b>29</b>
<b>Заключение.....</b>	<b>31</b>
<b>Литература .....</b>	<b>33</b>

# Введение

Все больше и больше растет роль биомоделирования в нашей жизни. Биомоделирование – это построение компьютерной модели внутреннего органа, части тела или человека целиком. В медицине существуют задачи планирования сложных реконструктивных хирургических операций, диагностики сердечной системы, пересадки органов и других задач, справиться с которыми человеку непросто, поскольку подобные задачи требуют опыта проведения таких процедур и точных способов планирования и прогнозирования результата лечения. Виртуальные модели, получаемые в процессе биомоделирования, должны точно соответствовать реальному объекту. Так как, не смотря на то, что компьютер будет руководствоваться виртуальной моделью, все действия будут проводиться над живым человеком. Одним из этапов биомоделирования может являться совмещение трехмерных моделей из разных источников: фотоснимков, томографических снимков, или совмещение снимков до и после операции.

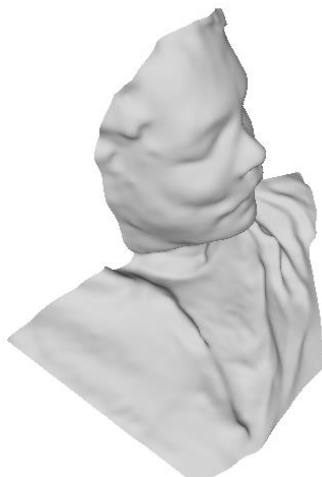


Рис. 1а

Фотографическая модель из стерео снимков

Традиционно, моделью называют упрощенное представление реального объекта. В течение всей работы мы будем иметь дело с трехмерными (объемными) моделями человека или моделями внутренних органов и частей тела. Такие модели будем называть анатомическими. Одним из способов получения таких моделей является компьютерная томография (далее КТ), которая позволяет послойно получить модель, содержащую не только внешнее (поверхностное) строение объекта, но и внутреннюю структуру. Одним из примеров, используемых в данной работе, является фотографическая модель лица, построенная по стерео снимкам (рис. 1а), и КТ модель головы мужчины (рис. 1б).

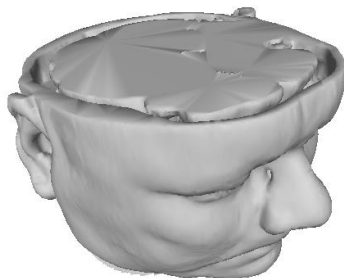


Рис. 16

КТ модель головы

Часто для получения более полной трехмерной картины появляется необходимость совместить модели, чтобы они сложились в один новый объект, который имеет особенности и первой и второй модели. В данной работе под совмещением будем понимать следующие шаги:

- регистрация соответствий между моделями,
- вычисление необходимых преобразований для совмещения моделей,
- применение преобразований к одной из моделей.

Регистрация соответствий (или просто регистрация моделей) – это вычисление особенностей тех точек или областей, которые принадлежат обоим моделям. С помощью особенностей можно определить, какие из точек одной модели соответствуют точкам другой модели. Зная попарные соответствия, можно вычислить преобразование для совмещения поверхностей. В данной работе будем оперировать только евклидовыми преобразованиями – это линейные преобразования, не изменяющие расстояния между точками. Операция масштабирования не рассматривается, поскольку считается, что модели находятся в одном масштабе

по построению. Последний шаг – это для каждой точки модели применить матрицу преобразования.

Для регистрации моделей существует несколько различных методов, о которых подробнее будет рассказано в главе “Теоретическое обоснование” в пункте “Альтернативные методы регистрации и совмещения моделей”. Одним из более распространенных алгоритмов является Iterative Closest Point (ICP) [6]. Он совмещает модели путем пошагового приближения к ближайшим точкам. Такой метод не подходит для анатомических моделей, так как внутренние точки КТ модели будут сильной помехой. Но он позволяет совместить модели после работы какого-либо другого метода, использующего поиск соответствий между моделями. Среди них существуют методы, основанные на извлечении особенной информации о точках: информация о цвете [9] (интенсивности [10]) точки или о направлении нормали [11]. Еще можно выделить такие дескрипторы особенностей: инварианты моментов [12], инварианты сферических гармоник [7] и дескрипторы интегральных объемов [8]. Недостаток этих методов – это чувствительность к шумам, поэтому их применение для анатомических моделей также невозможно. Шумы могут являться внутренностями модели, как в КТ моделях, или другими посторонними элементами модели. Также к шумам можно отнести любые различия между моделями, например: различия до и после операции.

В статье [1] авторы приводят алгоритм регистрации моделей, который, по их словам, основан на новой оригинальной технологии: извлечение всех множеств из 4 точек, лежащих в одной плоскости (4-Point Congruent Sets (4PCS)). Этот алгоритм регистрации тоже можно отнести к методам, основанным на особенностях точек, но в данном случае они исследуются не на основе точек в окрестности, а на основе их положения на всей модели. При этом можно

получить информацию о форме поверхности модели.

Для совмещения двух моделей  $P$  и  $Q$  в статье предлагают:

- найти в  $P$  4 точки, которые лежат в одной плоскости – это базис  $B$ ;
- найти в  $Q$  всевозможные множества из 4 точек, которые будут соответствовать базису  $B$ , и сохранить их в  $U$ ;
- для каждой пары  $(B, U[i])$  найти преобразование, переводящее  $B$  в  $U[i]$ ;
- среди всех таких преобразований найти лучшее;
- применить преобразование ко всей модели  $P$ .



Рис. 2а

Пример работы демо-версии оригинального алгоритма 4PCS. Слева до совмещения поверхностей. Справа после совмещения

Данный алгоритм позволяет аппроксимировать одну из моделей плоскостью, полученной на первом шаге при поиске базиса. А так как большинство задач опирается на то, что выполняется совмещение КТ моделей, то такая аппроксимация упростит эту операцию. Помимо этого, алгоритм устойчив к



шумам, что дает преимущество перед другими.

Существует авторская реализация этого алгоритма [5], совмещающая, например, модели чашки (рис. 2а), но у этой демо-версии совмещение анатомических КТ моделей не получается так хорошо, как хотелось бы (рис. 2б). Это связано с большим объемом обрабатываемых данных. Также влияет ограниченность изменяемых параметров, предложенных в демо-версии алгоритма. Из них на работу влияют следующие два:  $\delta$ ,  $\text{overlap}$ .

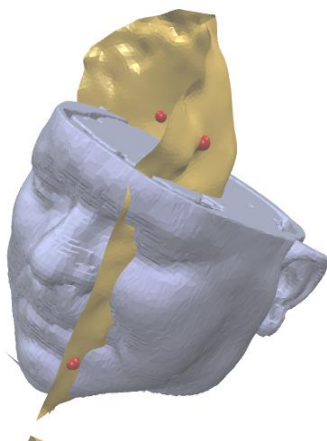


Рис. 2б

Изображен пример работы демо-версии алгоритма 4PCS, реализованного авторами статьи [1]. Использованы КТ модель головы и модель лица

Дельта ( $\delta$ ) – пороговое значение, которое задает точность совмещения и которое изменяется от 0.1 до 1. Если две точки разных моделей лежат меньше, чем на расстоянии, зависящем от  $\delta$ , то такие точки считаются хорошо совмещенными. Параметр  $\text{overlap}$  отвечает тому, как сильно одна модель покрывает другую. То есть это та область точек, которая есть и на одной, и на второй модели. Параметр

задается в процентах: 0.1 – 1 (10% - 100%). Значение overlap влияет на расстояние между точками в базисе. При уменьшении параметра overlap демо-программа работает все медленнее, так как находит много четверок из Q, которые соответствуют базису. При значениях overlap меньших 0.5, уже не удалось получить каких-либо результатов. В рамках одной из задач от компании Биомоделирование для оптимизации работы алгоритма с анатомическими моделями было решено реализовать алгоритм заново с модификациями, которые позволят улучшить совмещение интересных нас моделей.

Сложности также возникают на этапе проверки точности совмещения. Для этого можно использовать области на модели, полученные с помощью маркеров, выставляемых человеком, или можно использовать информацию о том, насколько сильно модели накладываются друг на друга. Но для этого необходимо использовать достаточно маленькие пороговые значения для остановки алгоритма совмещения, которые зависят от масштаба моделей и их начального местоположения в пространстве относительно друг друга.

Реализации некоторых алгоритмов совмещения и регистрации моделей есть в Point Cloud Library (PCL) [4] – это автономный сильно масштабируемый проект по обработке 2D/3D моделей. В одноименной библиотеке реализованы алгоритмы по фильтрации, регистрации, сегментации, визуализации облаков точек (point clouds). Среди них есть и алгоритм ICP, и алгоритм на основе особенностей точек PFH. На сайте [4] в разделе “документация” представлены хорошие, легко понимаемые примеры. Данная библиотека используется и для других задач в компании Биомоделирование, поэтому было решено, что реализация модифицированного алгоритма будет выполняться с помощью библиотеки PCL для дальнейшей интеграции его в проекты.

# Постановка задачи

Целью работы является регистрация трехмерных анатомических моделей с последующим совмещением этих моделей, которое будет выполняться модифицированным алгоритмом 4PCS. Также необходимо разработать среду для совмещения моделей, собрав в одну библиотеку методы для регистрации моделей и оценки точности совмещения. Реализация всех задач выполняется с помощью библиотеки PCL. Для достижения этой цели были поставлены следующие задачи.

- Реализовать алгоритм 4PCS с помощью библиотеки PCL.
- Сравнить работу алгоритма 4PCS с PFH и ICP методами.
- Модифицировать алгоритм для работы с анатомическими моделями.
- Разработать среду для совмещения трехмерных моделей: собрать методы для регистрации и оценки точности совмещения в библиотеку.
- Подать заявку на добавление в библиотеку PCL в раздел методов регистрации.
- Проверить работу модифицированного 4PCS алгоритма на паре трехмерных моделей: поверхностная трехмерная модель и трехмерная модель из КТ снимков.

Необходимо осуществить работу именно на исходных анатомических моделях, не применяя никаких алгоритмов по удалению шумов. При удалении шумов может быть повреждена область, необходимая для осуществления

действий, для которых и осуществлялось совмещение. Также в дальнейшем планируется работа над совмещением анатомических моделей в реальном времени с помощью дополненной реальности (рис. 3). Поэтому в данной работе необходимо создать базу для дальнейших исследований в области биомоделирования в реальном времени с помощью элементов дополненной реальности.

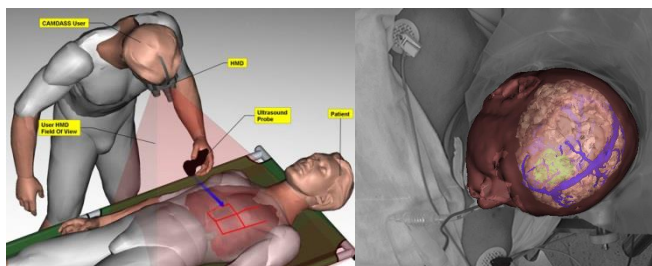


Рис. 3

Использование дополненной реальности в медицине [16][17]

# Теоретическое обоснование

## Трехмерные модели

Моделью называют упрощенное представление реального объекта. В течение всей работы мы будем иметь дело с трехмерными (объемными) моделями. Такие модели располагаются в трехмерном евклидовом пространстве, в котором имеется три пространственные координаты:  $x$ ,  $y$ ,  $z$ .

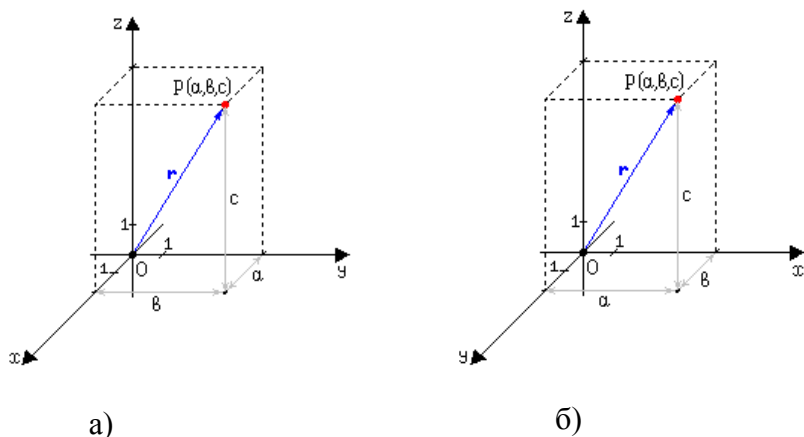


Рис. 4

а) изображение правой декартовой системы координат;

б) изображение левой декартовой системы координат.[3]

Выше на рисунке изображены координатные системы, задающиеся направлениями для соответствующих осей, которые начинаются в общей точке  $O$  – начале координат. Такие системы подразделяют на правые (рис. 4а) и левые (рис.4б).

Правая система задается по правилу правой руки (ось  $X$  – вектор большого пальца, ось  $Y$  – вектор указательного пальца, ось  $Z$  – вектор среднего пальца, расположенного перпендикулярно плоскости  $OXY$ ) и изображается следующим способом: ось  $OX$  направлена на наблюдателя, ось  $OY$  – направо, ось  $OZ$  – вверх. Левая декартова система координат используется реже, ее нельзя получить из правой системы координат с помощью только поворотов. Мы работаем только с правыми системами по построению трехмерных моделей.

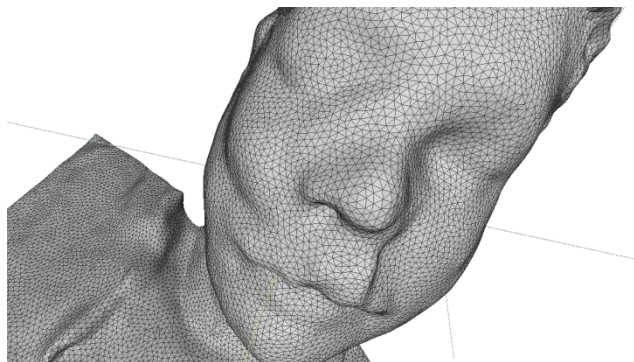


Рис. 5a

Изображение полигональной сетки для модели, полученной с помощью стерео снимков

Имеется несколько способов представления трехмерных моделей. КТ модели сохраняются в виде полигональных сеток. Производится обработка модели для перевода из воксельного представления в полигональное. Полигональная модель означает, что точки объединены в полигоны. В нашем случае все точки (поверхность и внутренность модели) соединяются в треугольники. Выше и ниже представлены изображения с полигональными сетками для фотографической

модели по стерео снимкам (рис. 5а) и для КТ модели (рис. 5б).

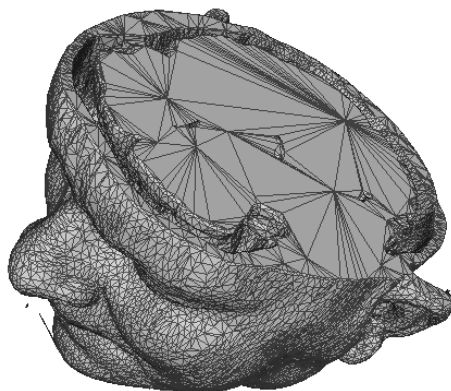


Рис. 5б

Изображение полигональной сетки для КТ модели

### ***Метод 4-Point Congruent Sets для совмещения моделей***

Рассмотрим каждый шаг алгоритма 4PCS. Необходимо совместить две модели Р и Q. В нашем случае моделью Р будет фотографическая модель лица, а фиксированной моделью Q – КТ модель головы.

1. Поиск базисного множества (базиса) из 4 точек в модели Р. Условием хорошего базиса является копланарность всех 4 точек, то есть они должны лежать в одной плоскости. Точки выбираются случайно. Построение базиса было реализовано в течение работы. Он основан на подсчете детерминанта матрицы, который позволяет сказать, лежат ли точки в одной плоскости или нет. Описание этого метода

указано в главе “Особенности реализации”

2. Поиск множеств в модели  $Q$ , соответствующих базису. На данном этапе выполняется построение двух множеств  $R1$  и  $R2$  из вершин  $Q$ . Элементы этих множеств есть пары вершин, соответствующие парам вершин из базиса. Геометрический смысл этих пар – это прямые. Пересекая прямые из  $R1$  и  $R2$ , получаем точку пересечения прямых. Существует 4 варианта расположения этой точки пересечения (рис. 6). Поэтому для каждой пары из  $R1$  и  $R2$  вычисляются 4 возможных точки пересечения. Далее остается только совместить эти два множества. Если точка пересечения из  $R2$  совпадает с точкой пересечения из  $R1$ , значит, эти 4 точки образуют множество, соответствующее базису. Так получается множество кортежей из 4 элементов, с которыми будет осуществляться дальнейшая работа.

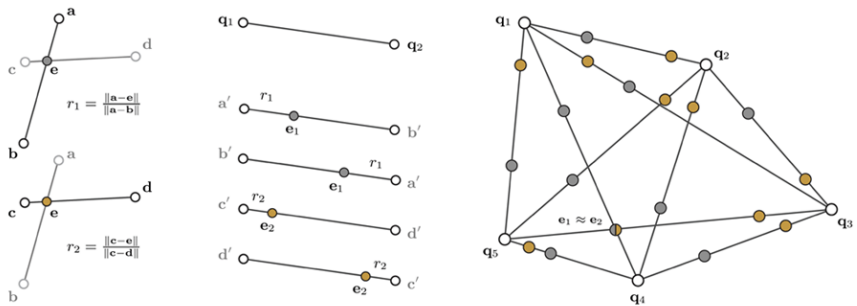


Рис. 6

Слева изображен базис (a,b,c,d), имеющий точку пересечения e. По центру представлены все 4 варианта расположения этой точки пересечения.

Справа представлен пример: вершины ( $q_1, q_4, q_5, q_3$ ) соответствуют базису (a,b,c,d). [1]



3. Для каждой четверки из множества кортежей считается преобразование, позволяющее совместить базис и эту четверку. Преобразование – это матрица поворота и вектор сдвига. Эксперименты показали, что point-to-point методы вычисления матриц поворота (Singular Value Decomposition (SVD) [6] и подсчет преобразования с помощью ICP, реализованные в PCL) дают плохой результат совмещения плоскостей, полученных из 4 точек. Point-to-point методы позволяют преобразовать точки к точкам в то время, как мы работаем с плоскостями. Для вычисления точного преобразования был реализован plane-to-plane метод, позволяющий посчитать матрицу поворота для плоскостей и вектор сдвига между моделями с помощью статьи [2]. В главе “Особенности реализации” подробно описан данный метод.

4. Проверить, как текущая итерация совмещает две модели. Необходимо посчитать ошибку совмещения моделей с помощью Largest Common Pointset (LCP) [1] – это облако точек, которое принадлежит обеим моделям и задает соответствия между точками моделей, так что можно проверить насколько каждая точка из одной модели совпала со своей парой из другой модели. Значение параметра overlap для демо-программы задает размер этой области относительно всей модели в процентном соотношении.

В главе “Введение” были приведены примеры работы демо-версии для моделей чашек и анатомических моделей (рис.2а и 2б).

## ***Альтернативные методы регистрации и совмещения моделей***

Помимо 4PCS существуют и другие способы совмещения и регистрации моделей, о которых и пойдет речь в этой главе. Одним из самых известных методов является ICP.

## **Алгоритм Iterative Closest Point (ICP) для совмещения поверхностей**

Данный алгоритм [6] выполняет итеративное приближение одной модели к другой, основываясь на принципе “ближайшего соседа”. Поэтому, чем ближе модели лежат друг к другу, тем лучше будет результат.

Общий алгоритм ICP выглядит следующим образом:

- для каждой точки модели P (source cloud) вычисляется ближайшая к ней точка в модели Q (reference cloud);
- вычисляется линейное преобразование (поворот и сдвиг), которое лучшим образом выровняет найденные пары точек;
- преобразование применяется ко всей модели P;
- если необходимая точность достигнута, то алгоритм останавливается, иначе возвращаемся к первому пункту.

Этот алгоритм широко известен, существуют различные модификации на некоторых шагах алгоритма, позволяющие улучшить совмещение, например: использовать для совмещения подмножества точек, а не модели целиком, или использовать весовые функции [6]. В PCL имеется своя реализация ICP с хорошим руководством по применению, которое находится в разделе документации на сайте [4].

Ниже на рис. 7а изображен пример работы ICP на анатомических моделях. Видно, что результат нельзя считать даже близким к ожидаемому.

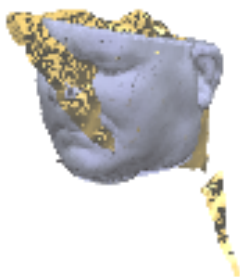


Рис. 7а

Результат работы алгоритма ICP без указания соответствий.

В статьях [7][8][9][10] предложены алгоритмы регистрации моделей на основе цветовых характеристик (цвет и/или интенсивность точки), нормалей точек и на основе вычисления пространственных характеристик (дескрипторов сферических гармоник и интегральных объемов). Для некоторых из них используются гистограммы особенных точек.

## **Point Feature Histograms (PFH) дескрипторы.**

Метод, основанный на PFH дескрипторах, используется для регистрации соответствий между моделями. Он основан на вычислении особенностей в окрестности специфических точек. Под особенностями понимаются, например, нормали и кривизна поверхностей. На основе полученных данных строятся гистограммы, позволяющие выбрать среди всех точек те, которые похожи друг на друга. Помимо обычного алгоритма PFH [13], существует быстрый алгоритм: Fast PFH (FPPH) [14].

Методы PFH и FPPH реализованы в библиотеке PCL.

Данный алгоритм не подходит для совмещения анатомических моделей в связи с тем, что очень чувствителен к шумам, и точки имеют сильные различия гистограмм в областях, где была проведена операция. Ниже на рис. 76 приведен пример работы регистрации с помощью PFH.

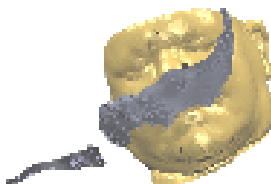


Рис. 76

Результат работы на основе регистрации нормалей с помощью PFH.

## ***Point Cloud Library***

Реализация модифицированного алгоритма 4PCS выполняется с использованием библиотеки PCL. В ней содержатся алгоритмы для обработки трехмерных моделей в полигональном представлении.

Список структур данных, позволяющих осуществлять обработку информации о моделях:

- `pcl::PointXYZ` – кортеж, содержащий координаты точки в трехмерном пространстве. Помимо него есть и другие представления точек (`PointXY`, `PointXYZHSV`, `PointXYZI`, `PointXYZINormal`, `PointXYZL`, `PointXYZRGB`, `PointXYZRGBA`, `PointXYZRGBL`, `PointXYZRGBNormal`). В библиотеке PCL реализованы различные и удобные методы для работы с точками;

- `pcl::PointCloud<PointType>` - вектор из элементов типа `PointType`. Также имеет множество удобных методов для работы с облаками точек;
- `pcl::PolygonMesh` – структура, позволяющая сохранять не только точки модели, но также и информацию о полигонах, по которым строится гладкая поверхность модели;
- `pcl::KdTreeFLANN<PointType>` - дерево для быстрого поиска элементов типа `PointType`.

Среди методов можно выделить:

- `pcl::fromPCLPointCloud2` – метод позволяет преобразовать облако точек из `pcl::PolygonMesh` в `pcl::PointCloud`, то есть получить из полигонального представления облачное;
- `pcl::toPCLPointCloud2` – метод делает обратное преобразование из `pcl::PointCloud` в `pcl::PolygonMesh`;

В библиотеке реализована векторная алгебра, позволяющая легко работать с матрицами и векторами.

## Требования к реализации алгоритма

Так как оригинальный алгоритм 4PCS не позволяет совмещать анатомические трехмерные модели, полученные из КТ снимков, то для новой реализации необходимо сформулировать требования, которые позволят достигнуть желаемого результата.

По сравнению с демо-версией, представленной авторами статьи [1], наша реализация должна выполняться с большим диапазоном значений входных параметров: значение параметра дельта ( $\delta$ ), изначально лежащие в промежутке  $[0.1, 1]$ , теперь могут быть и меньше 0.1. Это должно повлиять на точность совмещения моделей. Кроме этого, необходимо выполнять фильтрацию множества 4-элементных кортежей, получаемых на втором этапе алгоритма. Это необходимо для ускорения работы алгоритма, чтобы он не зависал при маленьких значениях  $\delta$  и  $\text{overlap}$ .

Алгоритм будет использоваться врачами, поэтому возникает необходимость максимально упростить их работу. Поэтому ручное выделение особых областей, имеющих на анатомических моделях (глаза, нос и др.), использовать не рекомендуется. Необходимо реализовать автоматический подбор параметров в зависимости от входных моделей.

Одним из главных требований является то, что алгоритм должен работать с исходными КТ моделями, в которых не удалены такие шумы, как внутренности объекта. Это необходимо для дальнейшего применения алгоритма в совмещении моделей в реальном времени.

## Особенности реализации

Перед реализацией алгоритма 4PCS с помощью PCL, была разработана библиотека RegistrationLib, которая теперь используется в компании Биомоделирование для совмещения моделей до и после операции. Она содержит в себе функциональность, которая позволяет быстро и просто работать с трехмерными моделями, методы по предобработке моделей, а также алгоритмы по совмещению трехмерных моделей. Ниже на рис. 8а представлена диаграмма класса Reg4PCS, созданного для алгоритма 4PCS, а также указана библиотека RegistrationLib с реализованными методами. Среди функций, постоянно использующихся в течение работы, есть следующие: чтение трехмерных полигональных моделей из файла с расширением .obj, а также и сохранение моделей в этом же формате. Формат записи вершин и полигонов в файл стандартный:

$v \ X \ Y \ Z$  — для вершины  $(X, Y, Z)$

$f \ v1 \ v2 \ v3$  — для полигона с вершинами  $v1, v2, v3$

Для предобработки моделей были реализованы методы по прореживанию (downsampling) моделей, по масштабированию (scaling) моделей с помощью маркеров, по подсчету точности совмещения (подсчету ошибки между моделями) при помощи маркеров. Во всех случаях маркеры — это набор точек, взятых вручную при помощи каких-либо средств, позволяющих сделать это. Маркеры передаются в качестве параметров, как и сами модели (в таком же формате, но без указания полигонов).

В качестве алгоритмов совмещения были реализованы методы, позволяющие быстро запускать алгоритмы ICP и PFH, уже реализованные в PCL. Для реализации совмещения при помощи дескрипторов PFH, был использован метод Sample

Consensus Initial Alignment, который также реализован в PCL. Он позволяет вычислять соответствия на основе переданных ему гистограмм особенностей точек (PFH). Кроме этого был также реализован алгоритм совмещения при помощи маркеров. Методы с маркерами позволили хорошо совмещать трехмерные анатомические модели, но далее встала задача совмещения моделей без указания дополнительной информации.

Алгоритм 4PCS позволяет регистрировать соответствия между двумя моделями P и Q на основе поиска плоскостей, полученных из 4 точек. Как известно, для построения достаточно и 3 точек, но в данном алгоритме 4 точки используются для нахождения соответствий. Модифицированный алгоритм, реализованный в данной работе, принимает 2 или 4 параметра. Во-первых, это модели P и Q, которые необходимо совместить. В нашем случае модель Q зафиксирована и в результате не изменяет своего положения, в то время как к модели P будет применено преобразование, полученное на 3 этапе алгоритма. Наша реализация для P и Q использует тип данных `pcl::PolygonMesh::Ptr`. Чтобы получить модель данного типа, достаточно воспользоваться реализованной функцией для чтения полигональной модели из файла.

Следующие опциональные параметры - это delta и overlap. Дельта (delta) – указывает на то расстояние между точками, которое считается приемлемым для окончания алгоритма. Параметр overlap – означает размер Largest Common Pointset (LCP) в процентах относительно всей модели. От этого параметра зависит расстояние между точками в базисе, которое используется в течение всего алгоритма. Также это значение используется при подсчете точности совмещения. Если параметры delta и/или overlap не указываются, то алгоритм автоматически определяет их. В качестве значения



для параметра *delta* устанавливается минимальное из расстояний между всеми парами точек:

$$\textit{delta} = \min_{p_1, p_2 \in P} \textit{pcl} :: \textit{euclidianDistance}(p_1, p_2)$$

Для параметра *overlap* было выбрано значение 0.8. Мы предположили, что в большинстве случаев совмещаемые модели будут совпадать на 80%, так как это будут модели одного и того же объекта, но с небольшими изменениями (в случае, например, операции).

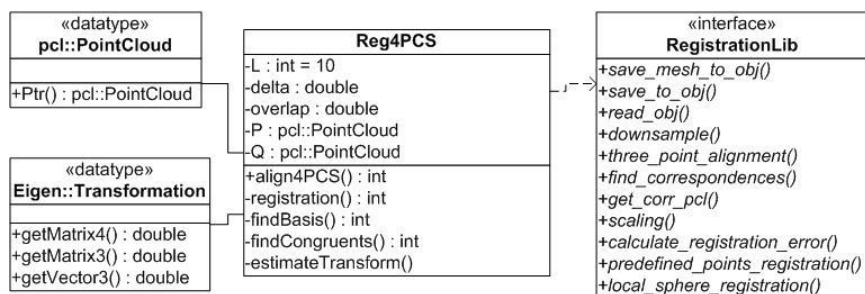


Рис. 8а

Реализованный класс для алгоритма 4PCS с указанием основных типов, используемых при обработке, а также реализованная библиотека `RegistrationLib`.

Алгоритм 4PCS был реализован согласно статье [1]. Для того, чтобы данный алгоритм опубликовали в библиотеке PCL, необходимо было реализовать его в виде класса (рис.8а). Структура, использованных методов, изображена на рис. 8б. На первом шаге алгоритма необходимо вычислить базис из 4 точек, которые лежат в одной точке. Для этого был реализован метод `get4BasisPoint`. В качестве параметров принимает модель `P` типа `pcl::PolygonMesh::Ptr` и число с плавающей точкой (`float`) `dist` – расстояние между точками в базисе. Первая точка базиса выбирается из `P` случайно без каких-либо

ограничений. Вторая точка выбирается случайно из множества в окрестности указанного размера вокруг первой точки. При этом не рассматриваются точки, лежащие на расстоянии меньше, чем указанное в качестве параметра `dist`. Также находятся 3 и 4 точка. Но для четвертой проверяется условие копланарности точек. Для этого вычисляется значение детерминанта матрицы, имеющей в качестве столбцов координаты найденных четырех точек. Если это значение равно нулю, то эти четыре точки лежат в одной плоскости. Это идеальный вариант, но на практике такое редко встречается. Поэтому мы считаем, что 4 точки лежат в одной плоскости, если значение детерминанта их матрицы не превосходит значения  $10^{-3}$ . Метод возвращает базис `B` типа `pcl::PointCloud::Ptr`.

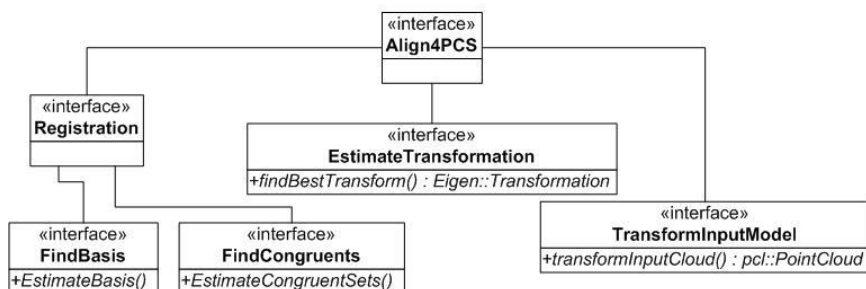


Рис. 86

Структура используемых в классе `Reg4PCS` методов для совмещения моделей.

Для второго этапа алгоритма – поиска соответствующих базису кортежей в модели `Q`, был реализован метод `findCongruent`. Метод принимает следующие параметры: базис `B` типа `pcl::PointCloud::Ptr` и модель `Q` типа `pcl::PolygonMesh`, а также параметр `delta` типа `float`. В результате возвращается `U` – множество кортежей, которые

соответствуют базису.

Далее необходимо найти преобразование для совмещения базиса и кортежа из  $U$ . В течение работы сначала пытались использовать уже реализованные методы поиска преобразований. Реализованный в PCL метод ICP позволяет получить матрицу преобразования, используемую в процессе совмещения. Пытались применить ICP для двух множеств из 4 точек каждое. Но результат получался неудовлетворительный. Кроме ICP в PCL также есть класс для вычисления преобразования с помощью SVD (singular value decomposition). Но он также не показал хорошего результата. После изучения этих методов стало ясно, что они используют поточечное (point-to-point) совмещение, но так как наш алгоритм совмещает плоскости, было решено использовать именно plane-to-plane преобразование. Для этого был реализован метод, описанный в статье [2]. Он позволяет поэтапно совместить две плоскости: сначала повернуть две плоскости так, чтоб они совпали, имея одну и ту же нормаль, а далее совместить точки, на основе которых и были построены эти плоскости (в нашем случае 4 точки базиса и 4 точки кортежа из  $U$ ). Для реализации двух последовательных преобразований в статье использовались кватернионы. В PCL есть класс для работы с кватернионами, поэтому удалось точно реализовать алгоритм, предложенный в статье [2] для плоскостей. Метод `estimateTransformMatrix` принимает на вход базис и кортеж из  $U$ . На выходе получаем матрицу поворота  $T$ .

Далее преобразование применяется к модели  $P$ . В качестве оценки точности совмещения мы считаем, какое количество точек модели  $P$  имеют в окрестности  $\delta$  точку из  $Q$ . Таким образом, при маленьких значениях  $\delta$ , можно утверждать, что модели хорошо совместились, если набралось количество точек, соответствующее значению `overlap`. Этот процесс

повторяется в течение указанного числа итераций, по умолчанию равного 10.

В качестве модификации алгоритма было решено использовать не четыре, а только два возможных положения для точки пересечения отрезков. А также подобраны достаточно маленькие значения внутренних параметров метода `findCongruent`, которые влияли на отбор пар точек, соответствующих точкам базиса. Также это было сделано для устранения ошибки распределения памяти (`bad allocation`), которая возникала в связи с появлением слишком большого числа таких пар точек.

## Апробация

Одной из задач работы было предложить модификации алгоритма 4PCS, которые позволят совмещать не только поверхностные модели (рис.9а), но и модели, полученные с помощью КТ снимков. Однако, как можно заметить на рис. 9б, эту задачу выполнить не удалось. Предложенные модификации не позволили улучшить алгоритм для использования его с КТ моделями. Тем не менее, теперь этот алгоритм можно интегрировать в проекты компании Биомоделирование. А также продолжить изучение его особенностей и поиск модификаций для улучшения.



Рис. 9а

Пример работы реализованного алгоритма 4PCS с моделями, полученными с помощью лазерного сканера Артек.

Если присмотреться к результату работы с поверхностными моделями (рис. 9а), то можно заметить неточности совмещения в области носа, но их можно поправить,

применив алгоритм ICP. Поэтому такая точность совмещения считается удовлетворительной.

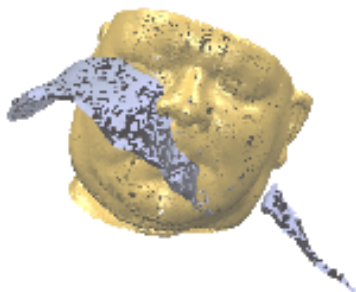


Рис. 96

Пример работы реализованного модифицированного алгоритма 4PCS с моделями, полученными с помощью КТ снимков.

## Заключение

В рамках данной работы были получены результаты, приведенные ниже.

- Реализован модифицированный алгоритм 4PCS на языке C++ с помощью библиотеки PCL.
- В качестве модификаций предложено использовать только два из возможных положения точек пересечения, используемых для поиска соответствий на втором этапе алгоритма. А также использовать меньшие значения параметров, используемых в процессе обработки, например, значения параметра  $\delta$  меньшие, чем 0.1.
- Реализации алгоритмов 4PCS, PFH, FPFH, ICP на основе библиотеки PCL включены в библиотеку RegistrationLib компании Биомоделирование. Библиотека используется для совмещения трехмерных моделей до и после операции.
- Создана экспериментальная среда на основе библиотеки PCL для оценки точности совмещения, позволяющая запускать несколько алгоритмов регистрации моделей в одинаковых условиях.
- Проведен анализ работы алгоритма на двух парах трехмерных анатомических моделей: поверхностных моделях, полученных с помощью лазерного сканера Артек, и поверхностной модели из стерео снимков и модели, полученной из томографических данных.
- Подан запрос на включение реализации алгоритма 4PCS в модуль Registration библиотеки PCL.

В дальнейшем планируются изучение алгоритма 4PCS еще

на более детальном уровне для определения модификаций, которые позволят совмещать и КТ модели. Это необходимо для дальнейшего использования алгоритма при совмещении моделей в реальном времени с помощью дополненной реальности. Для этого будет необходимо осуществить поддержку выравнивания для динамических моделей, которые будут изменяться в течение времени. Необходима стабильность работы такого алгоритма, чтобы не было “прыжков” модели в разные стороны. Работа над библиотекой RegistrationLib не будет приостановлена, в нее постепенно будут включаться другие алгоритмы совмещения, а также и другая функциональность, позволяющая обрабатывать трехмерные модели.



# Литература

[1] Dror Aiger, Niloy J. Mitra, Daniel Cohen-Or. 4-Points Congruent Sets for Robust Pairwise Surface Registration. <http://www.esiee.fr/~aigerd/a85-aiger.pdf>.

[2] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions, <http://www.stanford.edu/class/cs273/refs/Absolute-OPT.pdf>.

[3] Изображения правой и левой декартовых систем координат. <http://algotlist.manual.ru/maths/geom/coord.php>.

[4] Сайт проекта Point Cloud Library, <http://pointclouds.org/>.

[5] Изображения и полный список статей метода 4PCS, [http://graphics.stanford.edu/~niloy/research/fpcs/fpcs\\_sig\\_08.html](http://graphics.stanford.edu/~niloy/research/fpcs/fpcs_sig_08.html).

[6] Iterative Closest Point algorithm, <http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/slides/17-icp.pdf>.

[7] G. Burel and H. H'enoq, "Three-dimensional invariants and their application to object recognition," Signal Process., vol. 45, no. 1, pp. 1–22, 1995

[8] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust Global Registration," in Proc. Symp. Geom. Processing, 2005.

[9] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3-D data," in NRC '97: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, 1997.

[10] D. Akca, "Matching of 3D surfaces and their intensities," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 62, no. 2, pp. 112–121, June 2007.

[11] K.-H. Bae and D. D. Lichti, "Automated registration of

unorganized point clouds from terrestrial laser scanners,” in International Archives of Photogrammetry and Remote Sensing (IAPRS), 2004, pp. 222–227.

[12] F. Sadjadi and E. Hall, “Three-Dimensional Moment Invariants,” PAMI, vol. 2, no. 2, pp. 127–136, 1980.

[13] Aligning Point Cloud Views using Persistent Feature Histograms, Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, Michael Beetz

[14] Fast Point Feature Histograms (FPFH) for 3D Registration, Radu Bogdan Rusu, Nico Blodow, Michael Beetz

[15] Registration, Jochen Sprickerhof, <http://www.pointclouds.org/assets/icra2012/registration.pdf>

[16] Изображение с использованием дополненной реальности в медицине, [http://www.cs.rochester.edu/~brown/Images/brain\\_ar.gif](http://www.cs.rochester.edu/~brown/Images/brain_ar.gif)

[17] Изображение с использованием дополненной реальности в медицине, <http://scitechdaily.com/researchers-at-esa-develop-augmented-reality-headset-for-medical-diagnosis/>