МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент
Начальник управления
информатизации образования
ФГБОУ ВО «ЧелГУ»

_____ А.В. Рубцов

“___”_____ 2018 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,
д.ф.-м.н., профессор

_____ Л.Б. Соколинский

“___”_____ 2018 г.

РАЗРАБОТКА ПРОГРАММНОЙ СИСТЕМЫ ДЛЯ СОВМЕЩЕНИЯ ДАННЫХ ТРЕХМЕРНОГО СКАНИРОВАНИЯ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2018.115-015.ВКР

Научный руководитель

Преподаватель кафедры СП

_____ М.С. Тимченко
Автор работы,
студент группы КЭ-401

_____ А.А. Митцева
Ученый секретарь
(нормоконтролер)

_____ О.Н. Иванова

“___”_____ 2018 г.

Челябинск-2018

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ 6
ВВЕДЕНИЕ 7
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 10
1.1. Анализ существующих алгоритмов для реализации проекта 10
1.1.1. Базовый алгоритм ICP 10
1.1.2. Модификации алгоритма ICP 12
1.2. Анализ аналогичных проектов 19
Вывод по главе 1 21
2. ТРЕБОВАНИЯ К СИСТЕМЕ 22
2.1. Функциональные требования 22
2.2. Нефункциональные требования 22
2.3. Варианты использования системы 23
Вывод по главе 2 24
3. АРХИТЕКТУРА СИСТЕМЫ 25
3.1. Реализация предметной области 25
3.2. Компоненты системы 26
3.3. Представление архитектуры системы 28
Вывод по главе 3 30
4. РЕАЛИЗАЦИЯ СИСТЕМЫ 31
4.1. Средства реализации 31
4.2. Структура системы 31
4.3. Реализация модулей системы 33
4.3.1. Реализация модуля совмещения облаков 33
4.3.2. Реализация интерфейса 40
Вывод по главе 4 41
5. ТЕСТИРОВАНИЕ СИСТЕМЫ 42
5.1. Тестирование предложенного алгоритма на синтетических данных 42
5.2. Тестирование устойчивости предложенного алгоритма 44
5.3. Сравнительное тестирование алгоритма ICP и его модификаций 46

Вывод по главе 5 50
ЗАКЛЮЧЕНИЕ 51
СПИСОК ЛИТЕРАТУРЫ 52

ГЛОССАРИЙ

Трехмерная модель – математическое представление объекта в трехмерной среде [19].

3D-сканер – устройство, анализирующее физический объект или пространство и на основе полученных данных создающее его трехмерную модель [30].

Облако точек – набор вершин в трехмерной системе координат, полученный с использованием лазерного 3D-сканирования или других технологий [28].

Регистрация облаков точек – процесс совмещения нескольких облаков точек одного объекта в единую систему координат [20].

Итеративный алгоритм ближайших точек (Iterative Closest Points, ICP) – алгоритм, использующийся для сведения к минимуму разницы между двумя облаками точек [30].

Особая точка – это такая точка, изображение, которой можно устойчиво отличать от изображений всех соседних с ней точек [22].

Детектор – это метод извлечения особых точек из изображение [22].

Дескриптор – идентификатор особой точки, выделяющий ее из остального множества особых точек [22].

BA изображение (Bearing Angle image) – это изображение в оттенках серого цвета, в котором уровень цвета пикселя равен углу между двумя соответствующими соседними точками [9].

ВВЕДЕНИЕ

Актуальность темы

Многие сферы человеческой жизни не мыслимы без трехмерных цифровых моделей объектов. Существует два подхода создания моделей: «с нуля» при помощи пакетов компьютерного моделирования и при помощи 3D-сканеров с последующим совмещением нескольких частей одной модели. Первый способ является трудоемким и подходит далеко не во всех случаях. Второй способ является альтернативой, потому что трехмерные сканеры позволяют зарегистрировать любую форму объекта за короткое время и с высокой точностью. В настоящее время 3D-модели, полученные при помощи сканирования, находят широкое применение в самых разных областях.

Большой интерес уделяется 3D-моделям человеческого тела [24, 29], т.к. они могут широко применяться в кинематографе, приложениях виртуальной реальности, видеоиграх. 3D-моделирование человека упрощает создание анимационных фильмов, т.к. предоставляет все требуемые пропорции и позволяет накладывать нарисованный образ на актера. Также 3D-моделирование используется в медицине, позволяя устанавливать диагноз и создавать различные трехмерные модели внутренних органов и костей человека, в последующем используя их во время операции. Таким образом, появляется возможность под контролем навигационной системы сопоставить результат выполнения операции по конкретному клиническому примеру и сравнить с прототипом.

Задача регистрации облаков актуальна в области построения трехмерного окружающего пространства для определения местоположения роботов и планирования их оптимального пути [21, 23]. При помощи сканера строится 3D-модель части сцены (локальная реконструкция), полученной из определенной точки. Затем полная модель создается посредством объединения локальных реконструкций. Таким образом, получается трехмерная

модель всего окружающего пространства робота, в котором уже рассчитывается оптимальный путь.

3D-сканирование и 3D-моделирование также применяются в музейном деле и археологии для точного восстановления и реконструкции скульптур и памятников, в дизайне для получения формы объекта и его последующей обработки, в ортодонтии для качественного моделирования объектов небольшого размера, в геологии для исследования оползней [25].

В любой области применения принцип построения трехмерных моделей при помощи 3D-сканирования одинаков и заключается в совмещение отсканированных частей объекта в одной системе координат. Таким образом, задача совмещения трехмерных облаков точек актуальна на сегодняшний день, т.к. позволяет получать полноценные 3D-модели как объектов большого размера со сложной формой, состоящей из множества выпуклостей и впадин, так и трехмерные модели объектов простой формы небольшого размера.

Цель и задачи работы

Целью данной работы является разработка системы совмещения данных трехмерного сканирования.

Для достижения поставленной цели необходимо решить следующие задачи:

1) ознакомиться с теоретическим основами совмещения данных трехмерного сканирования;

2) проанализировать существующие решения, смежные проекты и выбрать алгоритм для реализации поставленной цели;

3) спроектировать архитектуру системы;

4) реализовать систему совмещения данных трехмерного сканирования;

5) провести тестирование реализованной системы совмещения данных трехмерного сканирования.

Структура и объем работы

Работа состоит из введения, пяти разделов, заключения и библиографии. Объем работы составляет 55 страниц, объем библиографии – 32 источника.

В первой главе, «Анализ предметной области», проведен обзор существующих алгоритмов совмещения данных трехмерного сканирования и смежных проектов.

Во второй главе, «Требования к системе», описаны функциональные и нефункциональные требования к системе и варианты использования системы.

В третьей главе, «Архитектура системы», описано проектирование системы. Подробно рассмотрена общая архитектура системы и ее компоненты.

В четвертой главе, «Реализация системы», описаны используемые инструменты реализации и представлены детали реализации системы.

В пятой главе, «Тестирование системы», представлены результаты тестирования предложенного алгоритма и проведено сравнительное тестирование с другими алгоритмами.

В заключении сделаны выводы о проделанной работе.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Одной из основных проблем 3D-моделирования является проблема совмещения отсканированных снимков в единую систему координат. Снимки представляют собой облака точек одного объекта, полученных с различных ракурсов. Регистрация снимков включает в себя нахождение взаимного расположения и ориентации одного изображения сцены относительно другого, и в последующем наиболее точное совмещение перекрывающихся областей точек. Существуют различные способы регистрации моделей, которые рассмотрим подробно в этой главе.

1.1. Анализ существующих алгоритмов для реализации проекта

1.1.1. Базовый алгоритм ICP

Наиболее известным методом регистрации облаков точек в трехмерном пространстве является «итерационный алгоритм ближайших точек» (Interactive Closest Points, ICP) [8]. ICP представляет собой алгоритм, выполняющий преобразования, такие как перемещение и вращение, необходимые для сведения к минимуму расстояния между точками, принадлежащим двум необработанным фреймам сканирования.

Целью алгоритма ICP является нахождение матрицы вращения R и вектора переноса T, которые выравнивают облака точек $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_n\}$, и для которых ошибки между преобразованным первым облаком точек и ближайшими точками второго облака являются минимальными [31]. Входными данными алгоритма являются два облака точек, первичная оценка трансформации и критерий для остановки итераций. Выходными данными являются матрица вращения и вектор переноса.

Пусть имеются два облака точек X и Y, которые описывают трехмерный объект, полученный с различных точек зрения при помощи 3D-сканера. Для перевода любой точки из облака X в систему координат облака Y используется ортогональное геометрическое преобразование [17]:

$$y_i = Rx_i + T,$$

где: T – вектор переноса;

R – матрица поворота;

x_i – произвольная точка из облака точек X;

y_i – произвольная точка из облака точек Y.

Базовый алгоритм ICP состоит из следующих шагов [30].

1. Для каждой точки из облака X определяется пара из облака Y, используя критерий поиска ближайшего соседа согласно некоторой функции близости.

2. Осуществляется преобразование путем смещения и поворота одного из облаков точек, а также расчет и оценка среднеквадратичного расстояния между парами точек:

$$E(R,T) = \frac{1}{N}\sum_{i=0}^{N} ||x_i - Ry_i - T||^2,$$

$$N = \sum_{i=1}^{N_X}\sum_{j=1}^{N_Y} w_{ij},$$

где: x_i, y_i – соответствующие точки облаков X и Y;

N_X, N_Y – количество точек в облаках X и Y;

w_{ij} – веса пар точек; если x_i ближайшая точка к y_i, тогда $w_{ij} = 1$, иначе $w_{ij} = 0$.

3. Шаги повторяются до тех пор, пока среднеквадратичное расстояние не станет меньше некоторого значения ε.

Обобщенная схема алгоритма представлена на рисунке 1.

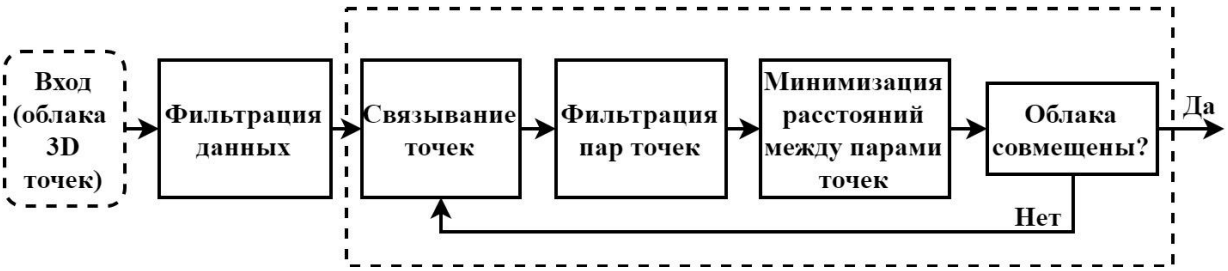


Рис. 1. Схема алгоритма ICP

Сложность алгоритма зависит от времени определения ближайших точек и времени нахождения параметров преобразования, поэтому в худшем случае она составляет $O(N_X N_Y)$ для N точек из облаков X и Y, т.е. $O(N^2)$ при $N_X \approx N_Y$.

Базовый алгоритм ICP имеет ряд недостатков [32]:

1) необходимо задавать такие начальные приближения R и T для первой итерации, чтобы предотвратить попадание алгоритма в локальные экстремумы, иначе итерация не сможет сойтись к правильному результату;

2) совмещаемые облака точек должны иметь общую область перекрытия;

3) на каждой итерации необходимо выполнять трудоемкую операцию поиска ближайшей точки.

1.1.2. Модификации алгоритма ICP

В связи с тем, что базовый алгоритм ICP имеет недостатки, было создано множество модификаций данного алгоритма, позволяющих улучшить тем или иным способом базовый ICP. Для каждого шага алгоритма существует несколько стратегий [10, 18].

1. Отбор точек, используемых для решения:

– использование всех точек;

– случайная выборка;

– поиск особых точек.

Самым эффективным является поиск особых точек, т.к. совмещение на основе особых точек позволяет значительно сократить время выполнения алгоритма.

2. Метрика сопоставления [26]:

– метрика «точка – точка»;

– метрика «точка – плоскость».

Для эффективного поиска ближайшей точки можно использовать модель k-d деревья [4] или метод воксель-разбивки [1].

3. Вес пар:

– одинаковый вес пар для всех точек;

– присвоение более низких весов парам с большим расстоянием между точками.

4. Отбрасывание пар:

– отбрасывание пар больше заданного значения;

– отбрасывание $n\%$ наихудших пар согласно выбранной метрике;

– отбрасывание пар, расстояние которых до точки больше, чем $n * \sigma$.

5. Минимизация средней квадратичной ошибки:

– метод, основанный на сингулярном расположении матриц (SVD) [2] – это разложение матрицы в виде:

$$M = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T,$$

где: $U, V \in R^3$ унитарны;

$\sigma_1 \geq \sigma_2 \geq \sigma_3$ – сингулярные значения матрицы M.

– метод на основе кватернионов – это система, образующая векторное пространство размерностью 4×4, в котором координаты x, y, z – компоненты трехмерного вектора, w – скаляр.

Далее рассмотрим подробное описание некоторых модификаций базового ICP, позволяющих устранить недостатки алгоритма.

Алгоритм Generalized-ICP [15, 16] также является модификацией базового алгоритма ICP. Данный алгоритм использует вероятностную структуру для определения функции ошибки. Суть алгоритма заключается в следующем.

В вероятностной модели предполагаем существование базового набора точек $\hat{A} = \{\hat{a}_i\}$ и $\hat{B} = \{\hat{b}_i\}$, который генерируется из облаков A и B в соответствии с $a_i \sim N(\hat{a}_i, C_i^A)$ и $b_i \sim N(\hat{b}_i, C_i^B)$, где $\{C_i^A\}$ и $\{C_i^B\}$ – ковариационные матрицы, связанные с точками.

Разница между точками a_i и b_i определяется, как $d_i = b_i - Ta_i$. Учитывая, что a_i и b_i взяты из независимых Гауссовых распределений, d_i можно записать в виде:

$$d_i \sim N(0, C_i^B + T^* C_i^A (T^*)^T).$$

Используя метод максимального правдоподобия (MLE), получается следующая формула:

$$T = argmin \sum_i d_i^{(T)^T} (C_i^B + TC_i^A T^T)^{-1} d_i^{(T)}.$$

Таким образом, Generalized-ICP отбирает точки из сплошных и гладких участков и совмещает эти участки поверхностей, содержащих данные точки, что в результате улучшает производительность и повышает точность выполнения алгоритма ICP.

Алгоритм на основе ориентационных гистограмм является одной из модификаций базового алгоритма и позволяет решить задачу без использования начальных приближений [1].

Оценка угловой ориентации состоит из двух шагов.

1. Получение ориентационных гистограмм H_1 и H_2 для облаков точек X и Y соответственно.

2. Поиск максимума корреляционной функции $C(H_1, H_2)$ для всех возможных значений матрицы поворота R по формуле:

$$C(H_1, H_2) = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} H_1(i,j)H_2(i,j)}{\sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} H_1(i,j)^2 \sum_{i=1}^{N}\sum_{j=1}^{M} H_2(i,j)^2}},$$

где: H_1, H_2 – ориентационные гистограммы размера $M \times N$.

Значение матрицы поворота $R_{C_{max}}$, соответствующее максимуму корреляционной функции, является искомой угловой ориентацией.

Оценка вектора сдвига состоит из трех шагов.

1. К облаку точек Y применяется найденная матрица поворота $R_{C_{max}}$.

2. Строится бинарное воксельное представление V_1 и $V_2{}'$ для облаков точек X и Y' с помощью процедуры трехмерной дискретизации.

3. Оценивается вектор сдвига между системами координат как максимум корреляционной функции.

Достоинством данной модификации базового алгоритма ICP является эффективность алгоритма даже с большими начальными углами ($> 100^{\circ}$) между входными облаками точек.

Еще одной из модификаций алгоритма ICP, позволяющей не устанавливать начальные приближения, является *алгоритм ICP с использованием геометрических признаков облаков точек (GF-ICP)* [5].

Алгоритм GF-ICP включает следующие шаги.

1. Найти k ближайших соседних точек для каждой точки из второго облака, рассчитать геометрические характеристики для каждой точки.

2. Выбрать начальные точки $\{p_i\}$, такие что $p_i \in X, Y$ облакам и удовлетворяют условиям:

– $\omega(p_i) > \tau_{gf}$, где ω – геометрическая функция, τ_{gf} – порог геометрической функции;

– $\left|H\left(p_i^X\right) - H\left(p_j^Y\right)\right| \leq \tau_{curvature}$, где H – кривизна поверхности, $\tau_{curvature}$ – порог кривизны поверхности;

– $\left|\omega_\alpha\left(p_i^X\right) - \omega_\alpha\left(p_j^Y\right)\right| \leq \tau_{normal}$, где ω_α – нормальный угол, τ_{normal} – порог нормального угла.

3. Вычислить начальные приближения R^1 и T^1 и получить преобразование $X^1 = R^1X + T^1$, где верхний индекс 1 представляет собой первую итерацию.

4. Вычислить ошибку сопоставления точек по формуле:

$$d^k = \sum_{i=1}^{N} ||X_i^k - Y_j^k||^2 + H\left(X_i^k, Y_j^k\right) + \omega_\alpha\left(X_i^k, Y_j^k\right),$$

где: X_i^k, Y_j^k – ближайшие соседние точки из двух облаков;

k – количество итераций.

5. В соответствии с новым соотношением получаем приближения R^k и T^k, и затем X^k после преобразования.

6. Повторять шаг 4 до тех пор, пока ошибка сопоставления точек не будет меньше порога τ_d или максимальному количеству итераций.

Алгоритм GF-ICP не требует задания первоначальных приближений матрицы поворота и вектора сдвига, выполняет совмещение облаков за меньшее количество итераций по сравнению с базовым. Также данный алгоритм обладает небольшой устойчивостью к шуму.

Для уменьшения вычислительной сложности и сохранения качества выполнения алгоритма предлагается *использование манхэттенской метрики* вместо евклидовой для измерения расстояния между точками [10].

Евклидово расстояние между двумя точками вычисляется по формуле:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}.$$

Манхэттенская метрика между двумя точками вычисляется по формуле:

$$d(x, y) = \sum_{i=1}^{n}|x_i - y_i|.$$

Данная модификация обладает лучшей производительностью и помехоустойчивостью, затрачивает меньшую вычислительную стоимость и меньшее время выполнения по сравнению с базовым алгоритмом, сохраняя при этом аналогичное качество регистрации.

Еще одной модификацией алгоритма ICP является алгоритм с *использованием глобальных контрольных точек* [2].

Введем глобальные контрольные точки, рассчитанные как средние значения: $\overrightarrow{x_G}$ для X и $\overrightarrow{y_G}$ для Y. Для каждой точки вычисляется евклидово расстояние между данной и контрольной точками:

$$\begin{cases} d_i^x = ||\overrightarrow{x_i} - \overrightarrow{x_G}||_2, i = 1,2, \dots, N_x \\ d_i^y = ||\overrightarrow{y_i} - \overrightarrow{y_G}||_2, i = 1,2, \dots, N_y \end{cases}.$$

Поскольку исходные позиции и новое расстояние являются разными измерениями, мы используем w для определения веса. Таким образом, получаем уравнение:

$$\mathrm{c}_k(i) = argmin(\left\|\left(R_{k-1}\overrightarrow{x_i} + \overrightarrow{t_{k-1}}\right) - \overrightarrow{y_{c(i)}}\right\|_2^2 + w(d_i^x - d_{c(i)}^y)^2),$$

где: $c(i) \in \{1,2, \dots N_Y\},\ i = 1,2, \dots, N_x$.

Для минимизации среднеквадратичной ошибки будем использовать разложение матрицы по сингулярным значениям (SVD):

$$H = \sum_{i=1}^{N_x}(\overrightarrow{x_i} - \overrightarrow{x_G})\,(\overrightarrow{y_{c(i)}} - \overrightarrow{y_G})^T,$$

где $H = U\Sigma V^T$.

Следовательно, матрица вращения вычисляется следующим образом:

$$R_k = VU^T,$$

а вектор сдвига представлен как:

$$\overrightarrow{t_k} = \overrightarrow{y_G} - R_k\overrightarrow{x_G}.$$

Данная модификация не требует задания начальных приближений для матрицы поворота и вектора сдвига, тем самым позволяя избежать попадания в локальный минимум, затрачивает меньшее количество итераций на совмещение облаков точек и выполняется быстрее по сравнению с базовым алгоритмом.

Для улучшения базового ICP предлагается использовать особые точки, полученные на основе *изображений углов поворота (Bearing Angle image)* [9]. BA изображение – это изображение в оттенках серого цвета, в котором уровень цвета пикселя равен углу между двумя соответствующими соседними точками. Угловое значение BA_{ij} точки PC_{ij} определено по формуле:

$$BA_{ij} = arccos\frac{p_{ij} - p_{i-1j-1}\cos d\varphi}{p_{ij}^2 + p_{i-1j-1}^2 - 2p_{ij}p_{i-1j-1}\cos d\varphi},$$

где: p_{ij} – значение j точки из i отсканированного слоя;

p_{i-1j-1} – значение $(j-1)$ точки из $(i-1)$ отсканированного слоя;

$d\varphi$ – соответствующий угловой шаг.

Таким образом, предложенный метод трансформирует облака точек в BA изображения и затем, используя детекторы и дескрипторы, находятся соответствующие пары пикселей между двумя изображениями. Затем отбираются наилучшие по минимальному расстоянию между дескрипторами. Полученные особые точки преобразуются в трехмерные точки, и на их основе выполняется базовый алгоритм.

Данная модификация позволяет сократить время выполнения алгоритма ICP, при этом сохраняя высокую точность совмещения облаков. Также метод на основе BA изображений позволяет совмещать облака точек с большим углом поворота относительно друг друга.

Для того, чтобы получить особые точки на ВА изображениях, необходимо использовать детекторы и дескрипторы ключевых точек. Детекторов и дескрипторов ключевых точек существует большое количество, поэтому рассмотрим только самые эффективные из них.

SIFT (Scale-Invariant Feature Transform) – алгоритм выявления и описания локальных особенностей на изображениях [3, 6, 7]. Алгоритм SIFT включает в себя два этапа: определение точек интереса и построение дескрипторов окрестностей данных точек. Сначала строятся пирамиды гауссианов и разностей гауссианов. Затем определяются точки, являющиеся локальными экстремумами разности гауссианов, они и будут особыми. На следующем шаге вычисляется ориентация ключевой точки, строится взвешенная гистограмма градиентов в окрестности этой точки, выбирается направление соответствующее максимальной компоненте гистограммы (m_{max}), и точке присваиваются все направления, которые больше, чем $0.8 * m_{max}$.

Затем определяются дескрипторы на гауссиане, максимально приближенном по масштабу к особой точке, и на основе градиентов в некоторой области особой точки. Каждому градиенту в окне дескриптора соответствует гистограмма дескриптора, построенная на основе расстояний до градиента по горизонтали, вертикали и в гистограмме. Таким образом, дескриптор ключевой точки состоит из всех полученных гистограмм.

SURF (Speeded Up Robust Features) – алгоритм, основанный на тех же принципах, что и SIFT [3, 6, 7]. Определение ключевых точек основано на использовании матрицы Гессе. Детерминант матрицы Гессе достигает экстремума в точках максимального изменения градиента яркости. Таким образом, используя детектор SURF определяются ключевые точки, в которых достигается максимум детерминанта матрицы Гессе. Затем у каждой ключевой точки определяется ориентация при помощи фильтра Хаара.

После этого формируются дескрипторы особых точек. Сначала область вокруг ключевой точки делится на 16 квадратов. Для каждого квад-

рата вычисляется значение частных производных с использованием вейвлетов Хаара. В качестве дескриптора используются суммы частных производных внутри каждого квадрата. Размерность итогового дескриптора равна 64.

ORB (Oriented FAST and Rotated BRIEF) – алгоритм, основанный на комбинации детектора FAST и дескриптора BRIEF с некоторыми изменениями [3, 14]. Выделение особых точек основано на масштабной гауссовой пирамиде изображений. С помощью детектора FAST определяются экстремумы функции яркости следующим образом. Для каждой точки рассматривается окружность некоторого радиуса, и в ней подсчитываются точки, имеющие значение яркости больше или меньше, чем центр окружности. Если таких точек больше, чем 0,75 от их количества, тогда центр окружности считается особой точкой. Точки не берутся в расчет, если они лежат на границах объекта.

Ориентация особой точки задается вектором, направленного из данной точки в сторону центра масс окрестности особой точки. Для сопоставления дескриптора используется квадратное окно, построенное относительно центра особой точки и согласованное с ее ориентацией. В данном окне выбираются пары точек и сравниваются между собой по значению яркости. Если яркость первой точки выше, то дескриптору присваивается 1, иначе 0. Таким образом, формируются дескрипторы точек.

1.2. Анализ аналогичных проектов

В настоящее время существует несколько программных пакетов, в которых реализована функция совмещения трехмерных облаков точек. Рассмотрим их подробнее.

Geomagic Studio – программный пакет, предоставляющий широкие возможности для создания поверхностных моделей по оцифрованным данным [12]. Данный пакет поддерживает взаимодействие со всеми сканерами и приборами трехмерной оцифровки. Программа позволяет объединять несколько облаков точек в одно, используя различные инструменты. Функция

обработки облаков точек включает в себя очистку данных от «шумов», прореживание и упорядочивание точек с учетом кривизны поверхности, заполнение отверстий. Результаты работы программы представлены на рисунке 2.

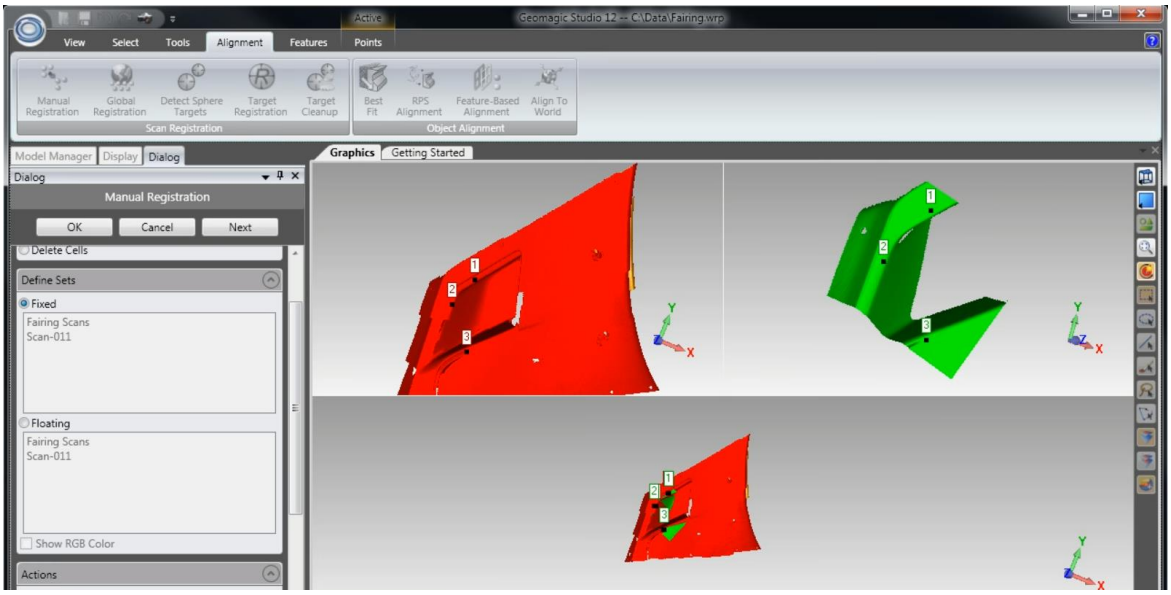


Рис. 2. Совмещение двух облаков точек в программе Geomagic Studio

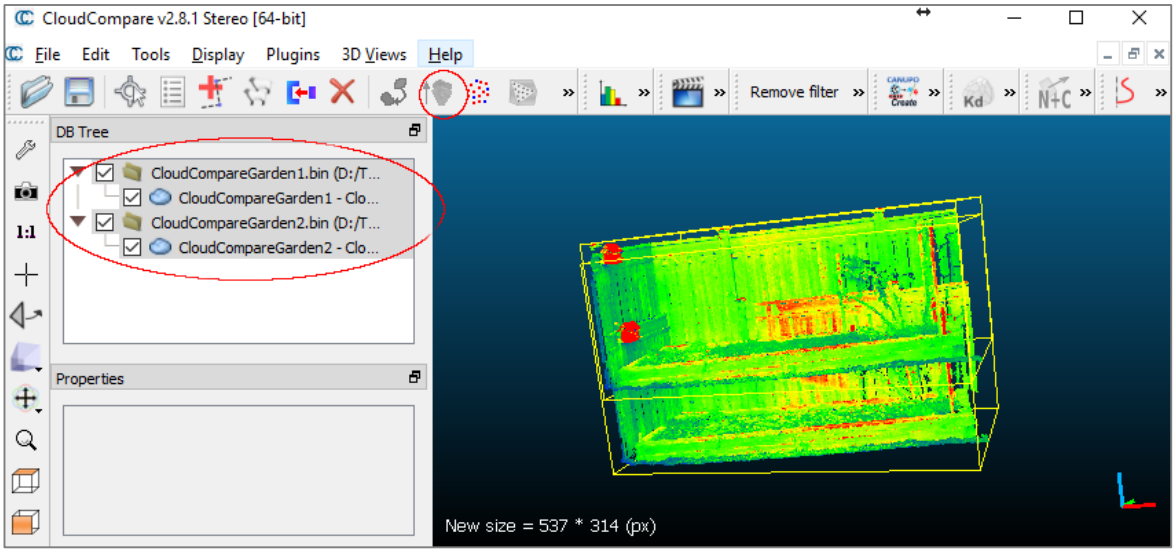


Рис. 3. Совмещение двух облаков точек в программе CloudCompare

CloudCompare – бесплатное программное обеспечение для обработки трехмерных облаков точек, сеток с треугольными ячейками и калиброванных изображений [11]. Программа предоставляет набор инструментов для воспроизведения и редактирования облаков точек таких, как проецирование, регистрация, сегментация, дистанционные расчеты, статистика, расчет

геометрических характеристик. Результаты совмещения двух облаков точек представлены на рисунке 3.

Российский разработчик ПО VR Concept разработал модуль, совмещающий в виртуальной реальности 3D-модель с облаком точек, полученным в результате 3D-сканирования объекта [27]. Данный модуль позволяет сразу после сканирования увидеть полученное облако точек в системе виртуальной реальности, изучить его со всех сторон и совместить с исходной 3D-моделью для анализа различных отклонений.

Вывод по главе 1

Анализ существующих модификаций алгоритма регистрации облаков точек показал, что совмещения трехмерных облаков является актуальной задачей на сегодняшний день, т.к. до сих пор не создана идеальная модификация алгоритма ICP. В ходе анализа алгоритмов было принято решение использовать несколько модификаций, которые бы значительно улучшили базовый ICP: использование BA изображений, Манхэттенской метрики и модификации, основанной на глобальных контрольных точках.

Проведенный обзор аналогичных проектов показал, что на данный момент существует несколько ПО, позволяющих совмещать облака точек. Однако, чтобы получить хорошую 3D-модель в данных программах, требуется вручную редактировать облака, которые затем будут автоматически совмещены. Это говорит о том, что задача регистрации облаков точек полностью не решена.

2. ТРЕБОВАНИЯ К СИСТЕМЕ

Система совмещения данных трехмерного сканирования будет представлять собой приложение, позволяющее совмещать исходные облака точек, полученных из снимков с помощью камеры Kinect, в единое облако точек, с целью получения 3D модели окружающего пространства или объекта. Пользователь должен иметь возможность загружать одновременно несколько снимков и получать на выходе единую модель.

2.1. Функциональные требования

На основании анализа предметной области были определены следующие функциональные требования к проектируемой системе.

1. Система должна предоставлять пользователю интерфейс для выбора снимков.
2. Система должна преобразовывать входные снимки в трехмерные облака точек.
3. Система должна совмещать трехмерные облака точек в единое облако.
4. Система должна визуализировать облака точек в единой системе координат до выполнения совмещения и после совмещения.
5. Система должна отображать время совмещения трехмерных облаков точек.
6. Система должна рассчитать ошибку совмещения облаков.

2.2. Нефункциональные требования

На основе функциональных требований и анализа предметной области были определены следующие нефункциональные требования.

1. Система должна быть написана на языке C++.
2. Система должна работать со снимками в формате *.png, полученных с помощью камеры Kinect, которые включают в себя RGB снимки и соответствующие им снимки глубины.

3. Система должна визуализировать облака в разных цветах, чтобы пользователь мог наглядно отличать одно облако от другого.

4. Система должна отображать время совмещения облаков точек в секундах.

2.3. Варианты использования системы

На основе списка требований, предъявляемых к проектируемой системе, были разработаны варианты использования, представленные на рисунке 4.

Рис. 4. Диаграмма вариантов использования системы совмещения данных трехмерного сканирования

С системой взаимодействует один актер – пользователь, использующий данную систему для автоматизированного совмещения трехмерных облаков точек. Данный актер может реализовать следующие варианты использования.

1. «Выбрать RGB снимки и снимки глубины» – выбрать снимки, полученные с помощью камеры Kinect.

2. «Получить трехмерные облака точек до выполнения алгоритма и после» – может просмотреть и проанализировать расположение облаков точек до совмещения и после совмещения, оценить визуально качество совмещения облаков.

Вывод по главе 2

В процессе анализа требований был определен основной актер системы, варианты использования системы, сформулированы основные функциональные и нефункциональные требования, предъявляемые к разрабатываемой системе.

3. АРХИТЕКТУРА СИСТЕМЫ

Как уже отмечалось раннее, базовый алгоритм имеет недостатки, поэтому были созданы различные модификации данного алгоритма, но многие из них предназначены для облаков точек с небольшим смещением. Однако, данные алгоритмы не позволяют совмещать облака точек с частичной областью перекрытия. Эта проблема на сегодняшний день особенна актуальна для моделирования окружающего пространства. Таким образом, возникает следующая задача: создать модификацию алгоритма ICP, позволяющую совмещать трехмерные облака точек с частичной областью перекрытия за короткий промежуток времени, при этом не уступая в точности совмещения другим алгоритмам на основе ICP. В данной работе предлагается модификация алгоритма ICP с использованием BA изображений, Манхэттенской метрики и глобальных контрольных точек, направленная на решение этой задачи.

3.1. Реализация предметной области

На вход подаются два облака точек: динамическое и статическое. В ходе выполнения алгоритма динамическое облако преобразовывается к системе координат статического облака. Таким образом, получив на вход трехмерные облака точек, необходимо выбрать такие пары точек, которые бы позволили наиболее точно совместить облака. В результате такие точки можно назвать особыми или ключевыми. Для поиска особых точек предложено использовать BA изображения. Следовательно, для каждого облака точек строится BA изображение по следующей формуле:

$$BA_{ij} = \arccos\left(\frac{a+b-c}{2a^2b^2} * \frac{180}{\pi}\right),$$

где: $a = \left\|p_{ij}\right\|_2$, $b = \left\|p_{ij} - p_{ij+1}\right\|_2$, $c = \left\|p_{ij+1}\right\|_2$, p_{ij}, p_{ij+1} – соседние точки в облаке.

Затем на основе полученных BA изображений предложено выполнить поиск особых точек с помощью детектора точек. Между полученными особыми точками необходимо установить соответствия, чтобы точка из

первого изображения наиболее точно соответствовала точке из второго изображения. Для этого используется дескриптор точек. Поскольку особые точки, полученные на ВА изображениях являются двумерными, они преобразовываются в трехмерные с использованием соответствующих облаков. Таким образом, получив пары трехмерных особых точек, решается проблема фильтрации точек, на основе которых будет происходить совмещение облаков точек.

После этого особые точки подаются на вход алгоритму ICP с использованием глобальных контрольных точек. Глобальные контрольные точки вычисляются как средние значения всех особых точек в облаках. Затем необходимо сместить облака особых точек путем вычитания координат глобальной контрольной точки. На основе полученных данных рассчитываются матрица поворота и вектор сдвига с использованием сингулярного разложения матриц (SVD). На следующем шаге происходит обновление динамического облака в соответствии с рассчитанными матрицей поворота и вектором сдвига.

Для расчета ошибки между облаками предложено использовать Манхэттенскую метрику вместо Евклидовой, т.к. данная метрика имеет меньшую вычислительную сложность и, следовательно, алгоритм будет выполняться за меньшее время.

3.2. Компоненты системы

Прежде чем, приступить к реализации системы необходимо описать ее структуру компонентов. Основные компоненты системы и их зависимости представлены на рисунке 5.

Система представляет собой клиент-серверное приложение. Клиент включает в себя интерфейс пользователя, обеспечивающий взаимодействие пользователя с системой, сервер включает в себя модуль совмещения облаков, отвечающий за совмещение данных трехмерного сканирования.

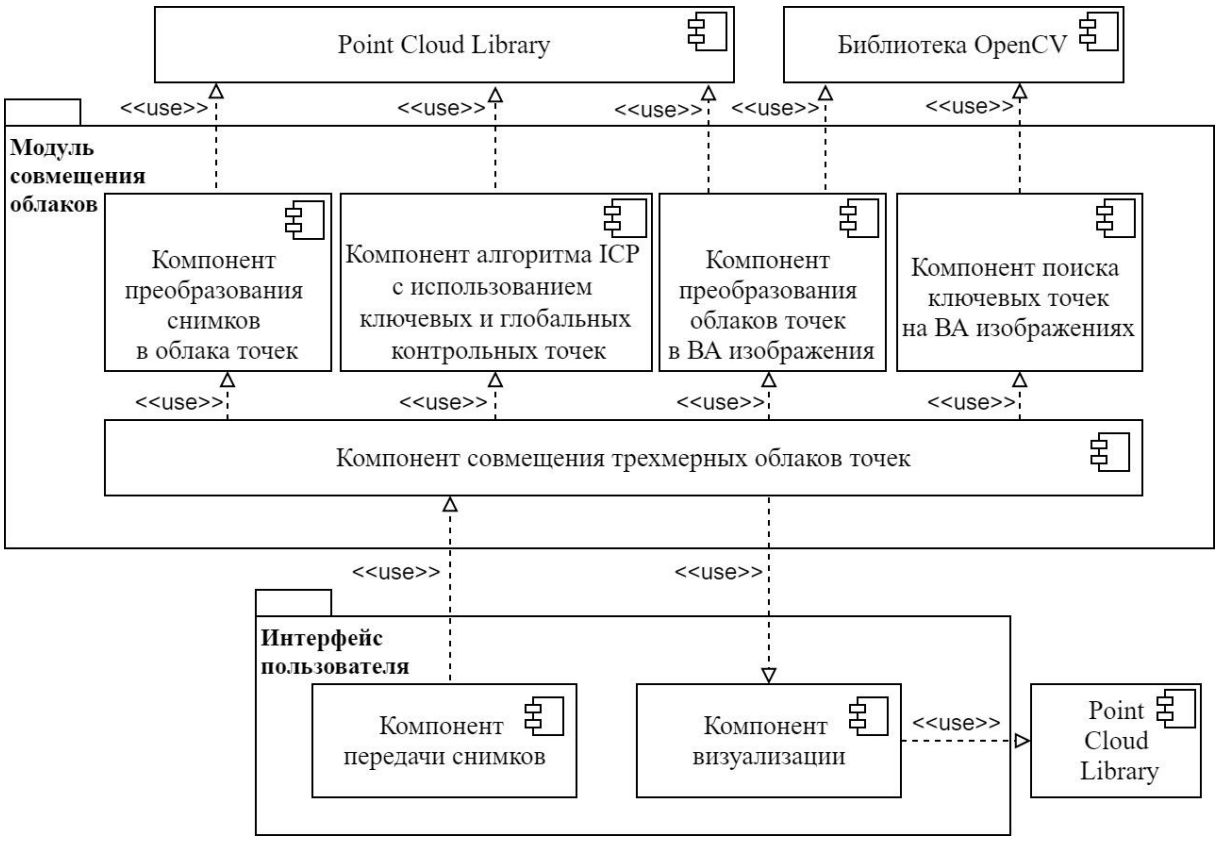


Рис. 5**.** Диаграмма компонентов системы совмещения данных трехмерного сканирования

Компонент совмещения трехмерных облаков точек реализует совмещение облаков с использованием BA изображений, глобальных контрольных точек и Манхэттенской метрики.

Компонент преобразования снимков в облака точек реализует перевод снимков, полученных с помощью камеры Kinect, в трехмерные облака точек.

Компонент преобразования облаков точек в BA изображения реализует перевод облаков точек в изображения в оттенках серого цвета, в котором уровень цвета пикселя равен углу между двумя соответствующими соседними точками.

Компонент поиска ключевых точек на BA изображениях реализует поиск точек интереса в изображениях, полученных на основе облаков точек.

Компонент алгоритма ICP с использованием ключевых и глобальных контрольных точек реализует совмещение облаков точек при помощи алгоритма ICP с использованием глобальных контрольных точек на основе ключевых точек, полученных на раннее выполненных шагах.

Компонент передачи снимков предоставляет пользователю возможность выбора RGB снимков и снимков глубины, на основе которых будут получены трехмерные облака точек и совмещены.

Компонент визуализации предоставляет пользователю возможность просматривать в единой системе координат облака точек до совмещения и после и оценивать визуально качество совмещения.

Система использует две библиотеки: Point Cloud Library (PCL) и OpenCV. Библиотека PCL предоставляет удобные структуры для работы с облаками точек и средства визуализации, позволяющие отображать облака точек в 3D пространстве с различных ракурсов. Библиотека OpenCV предоставляет средства для работы с изображениями, а также детекторами и дескрипторами особых точек.

3.3. Представление архитектуры системы

На рисунке 6 представлена диаграмма деятельности, отражающая пошаговое выполнение предложенного алгоритма ICP. На первом шаге пользователь подает снимки Kinect, затем снимки преобразуются в облака точек, которые в последующем преобразуются в BA изображения. После этого осуществляется поиск особых точек на BA изображениях, и на основе полученных точек происходит совмещения облаков с использованием алгоритма ICP на основе глобальных контрольных точек и Манхэттенской метрики. В результате пользователь получает облака точек в одной системе координат до выполнения алгоритма совмещения и после, которые он может визуально рассмотреть со всех сторон и проанализировать качество совмещения.

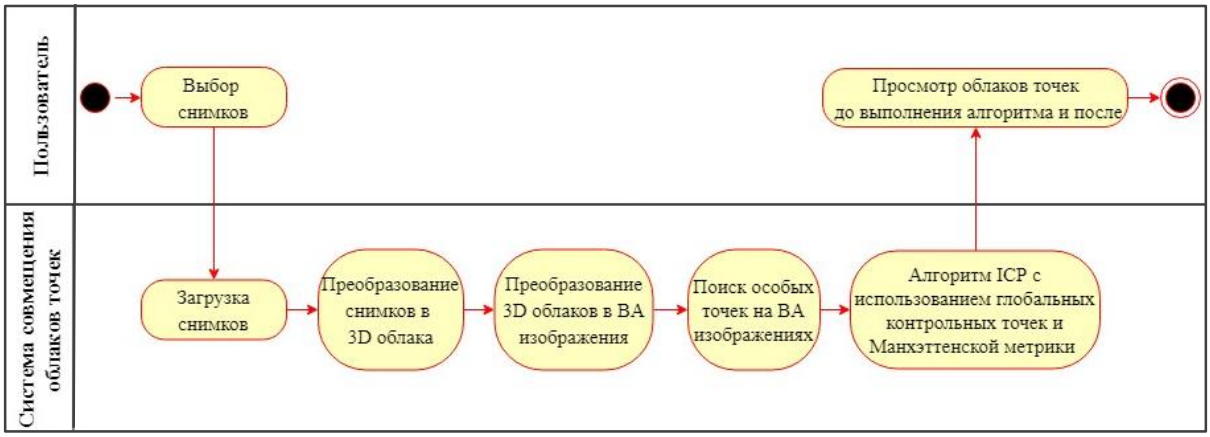


Рис. 6. Диаграмма деятельности системы совмещения данных трехмерного сканирования

Более подробный разбор модификации алгоритма ICP с использованием BA изображений, глобальных контрольных точек и Манхэттенской метрики представлен в листинге 1. Входными данными алгоритма являются снимки Kinect, точность eps и порог threshold для остановки итераций в случае, если разность между ошибкой на предыдущей итерации и следующей итерации меньше данной точности, либо ошибка на выполненной итерации меньше заданного порога. На выходе получаются совмещенные облака точек. Алгоритм предназначен для совмещения нескольких облаков точек, поэтому для выполнения этой задачи, предложено сначала совмещать попарно каждое облако с другим, а затем они объединяются все в одну систему координат. Таким образом, можно получить трехмерную модель сфотографированного с помощью камеры Kinect пространства или предмета.

Листинг 1. Псевдокод предложенной модификации алгоритма ICP

```
МОДИФИКАЦИЯ АЛГОРИТА ICP
ВХОДНЫЕ ДАННЫЕ: Снимки, полученные с помощью Kinect, точность eps, порог threshold
ВЫХОДНЫЕ ДАННЫЕ: Совмещенные облака точек
НАЧАЛО
      для каждого снимка Kinect
            Преобразовать в облако точек
      для каждого облака точек
            Преобразовать в BA изображение
      для каждого BA изображения
            Применить детектор для поиска особых точек
      для каждой пары BA изображений, облака которых нужно совместить
            Назначить статическое и динамическое облака
```

```
            Применить дескриптор, чтобы найти соответствия между особыми
            точками из статического и динамических облаков
            Перенести найденные особые точки на облака
            Рассчитать глобальную контрольную точку статического облака
            error <- 0
            пока не конец итераций
                  Рассчитать глобальную контрольную точку динамического
                  облака
                  Рассчитать newError на основе Манхэттенской метрики
                  между облаками
                  Вычислить глобальные контрольные точки на основе особых
                  точек в обоих облаках
                  Сместить облака особых точек путем вычитания координат
                  глобальной контрольной точки
                  Рассчитать на основе полученных данных R и T
                  Обновить динамическое облако на основе R и T
                  если |newError – error|< eps или newError < threshold
                        то возврат динамическое облако
                  error <- newError
            возврат динамическое облако
            Объединить динамическое и статическое облака в единую систему
            координат
      Объединить все преобразованные облака в одну систему координат
КОНЕЦ
```

Вывод по главе 3

В соответствии с требованиями к системе была спроектирована архитектура системы совмещения данных трехмерного сканирования. На основе реализации предметной области были подробно рассмотрены компоненты системы и соответствующее им представление архитектуры системы.

4. РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1. Средства реализации

Для реализации системы совмещения данных трехмерного сканирования использовался язык C++, т.к. существующие библиотеки по обработке изображений и работе с облаками точек реализованы на данном языке. В качестве среды разработки программного обеспечения была выбрана Microsoft Visual Studio Community 2017.

Для работы с облаками точек использовалась библиотека Point Cloud Library. PCL содержит множество различных алгоритмов, включая фильтрацию, реконструкцию поверхности, регистрацию, сегментацию, визуализацию.

Для работы с изображениями, детекторами и дескрипторами точек использовалась библиотека OpenCV. OpenCV предназначена для обработки изображений и содержит алгоритмы компьютерного зрения и численные алгоритмы.

Для сингулярного разложения матриц использовалась библиотека Eigen. Она представляет собой библиотеку линейной алгебры и предназначена для векторно-матричных вычислений и связанных с ними операций.

Для сохранения BA изображений в формате *.png, использовалась библиотека LodePNG. Она предназначена для декодирования и кодирования изображений в формат PNG.

4.2. Структура системы

Система совмещения данных трехмерного сканирования состоит из 6 классов, которые представлены на рисунке 7.

Класс *Interface* предоставляет пользователю удобный интерфейс для выбора снимков, облака которых должны быть совмещены, и для просмотра облаков в единой системе координат до совмещения и после. Также интерфейс отображает время совмещения облаков и ошибку совмещения.

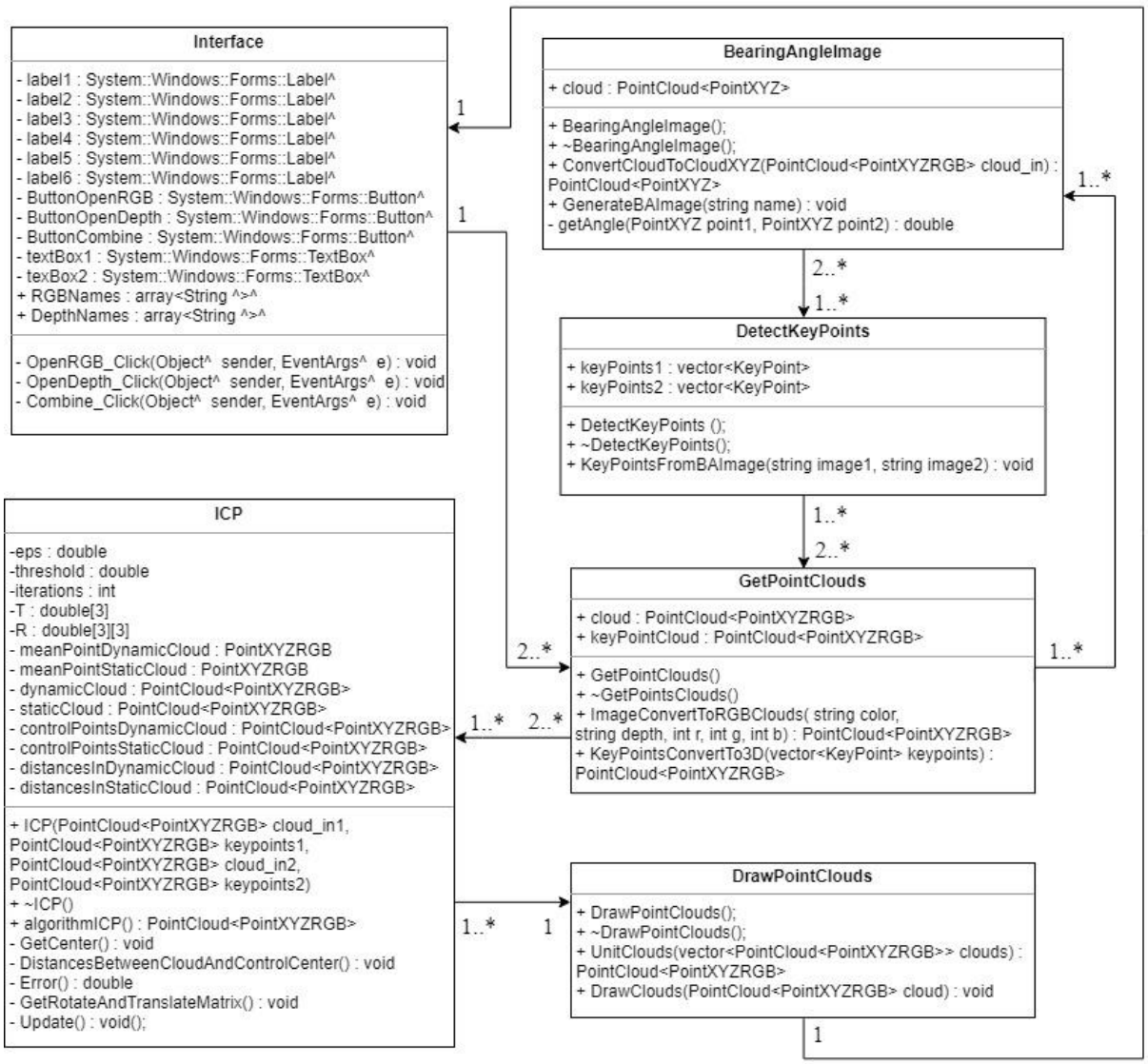


Рис. 7. Диаграмма классов

Класс *GetPointClouds* преобразует входные снимки в трехмерные облака точек и преобразует двумерные ключевые точки в трехмерные.

Класс *BearingAngleImage* переводит цветные RGB облака точек в простые бесцветные трехмерные облака, которые содержат только координаты вершин, и на основе этих облаков генерирует BA изображения.

Класс *DetectKeyPoints* определяет на BA изображениях ключевые точки.

Класс *ICP* на основе ключевых точек с использованием Манхэттенской метрики и глобальных контрольных точек преобразовывает одно облако к системе координат второго, тем самым совмещая их.

Класс *DrawPointClouds* позволяет объединять несколько облаков в одно и также обеспечивает визуализацию трехмерных облаков точек до совмещения и после.

4.3. Реализация модулей системы

Система совмещения данных трехмерного сканирования основана на клиент-серверной архитектуре. На сервере реализован модуль совмещения облаков точек, а на клиенте интерфейс пользователя. Клиент и сервер написаны с использованием сокетов TCP. Для создания сокетов используются IP-адреса и номера портов TCP локальной машины и той, с которой устанавливается связь. Сервер поддерживает непрерывную работу, тем самым позволяя клиенту подключаться несколько раз. Таким образом, пользователь имеет возможность много раз подавать на вход снимки, и на выходе получать совмещенные облака, время выполнения алгоритма и ошибку совмещения.

4.3.1. Реализация модуля совмещения облаков

Согласно предложенному алгоритму в главе 3, на первом шаге снимки Kinect преобразуются в трехмерные облака точек с помощью алгоритма, представленного в листинге 2. При преобразование снимков в облака точек используются внутренние параметры камеры Kinect. Реализацию рассмотрим на примере снимков 180 и 200 из коллекции «Office», представленных на рисунке 8 справа и слева.

Рис. 8. Снимки 180 и 200 из коллекции «Office»

Листинг 2. Преобразование снимка Kinect в облако точек

```
PointXYZRGB pt;
for (int y = 0; y < rgb_image.rows; ++y) {
      for (int x = 0; x < rgb_image.cols; ++x) {
            if (depth_image.at<unsigned short>(y, x) != 0)
            {
                  pt.z = depth_image.at<unsigned short>(y, x) / 1000.0;
                  pt.x = (x - cx)*pt.z / f;
                  pt.y = (y - cy)*pt.z / f;
                  pt.r = r;
                  pt.g = g;
                  pt.b = b;
            }
            else
            {
                  pt.z = 0;
                  pt.x = 0;
                  pt.y = 0;
                  pt.r = 0;
                  pt.g = 0;
                  pt.b = 0;
            }
            cloud->points.push_back(pt);
      }
}
```

На втором шаге трехмерные облака точек преобразуются в BA изображения. Реализация преобразования облака точек в BA изображение представлено в листинге 3.

Листинг 3. Преобразование облака точек в BA изображение

```
for (y = 0; y < height - 1; y++)
{
      for (x = 0; x < width; x++)
      {
            theta = getAngle(cloud.at(x, y), cloud.at(x, y + 1));
            gray = theta;
            r = gray;
            g = gray;
            b = gray;
            image[4 * width * y + 4 * x + 0] = 0;
            image[4 * width * y + 4 * x + 1] = b;
            image[4 * width * y + 4 * x + 2] = g;
            image[4 * width * y + 4 * x + 3] = r;
      }
}
```

На третьем шаге на каждом BA изображение осуществляется поиск ключевых точек с помощью детектора. В первой главе описано 3 самых эффективных детектора: SIFT, SURF и ORB. Из них был выбран тот, который наилучшим образом находит ключевые точки на изображениях, т.е. такие

точки, которые лежат только на контурах объектов. Тестирование детекторов представлено на рисунках 9–11.

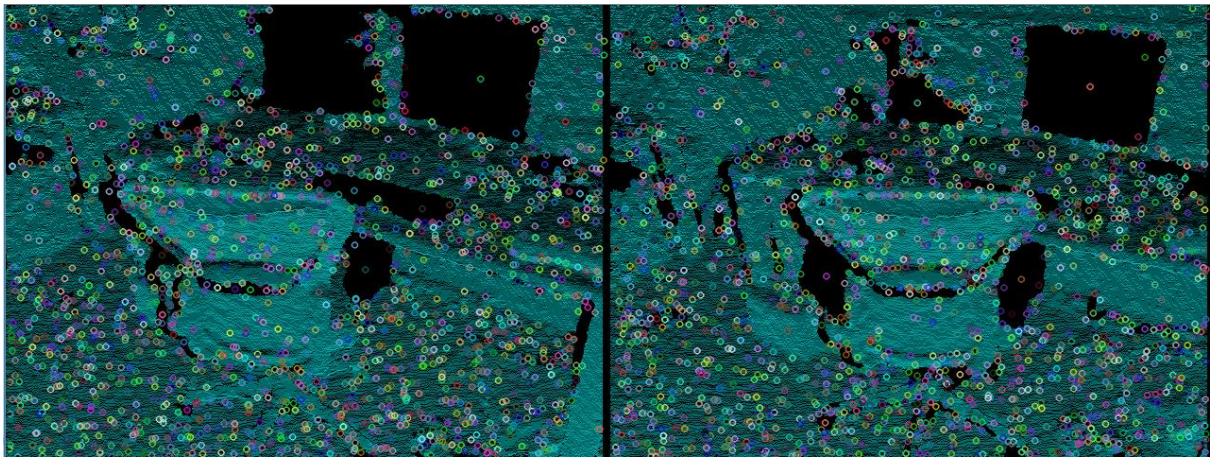


Рис. 9. Детектор ключевых точек SIFT на BA изображениях

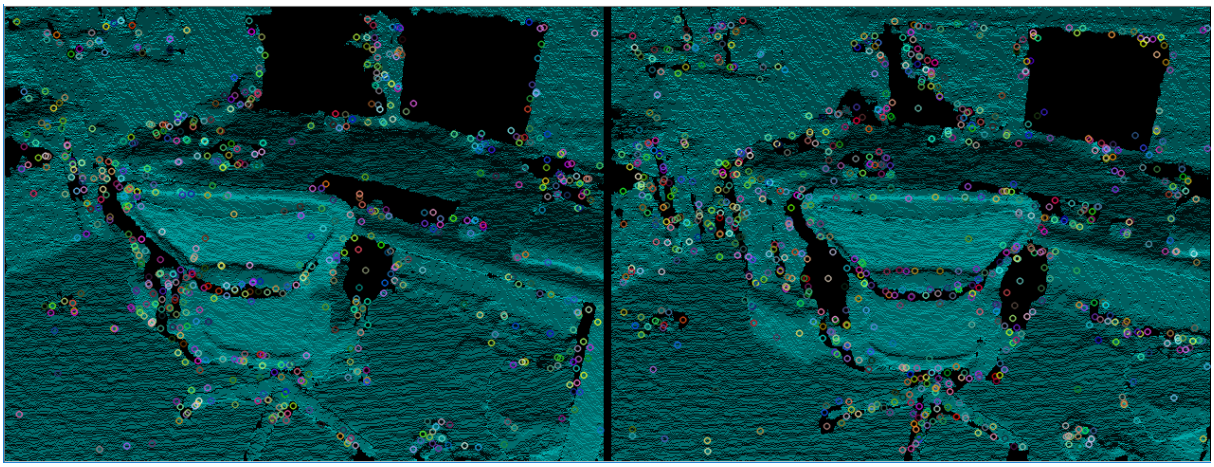


Рис. 10. Детектор ключевых точек SURF на BA изображениях

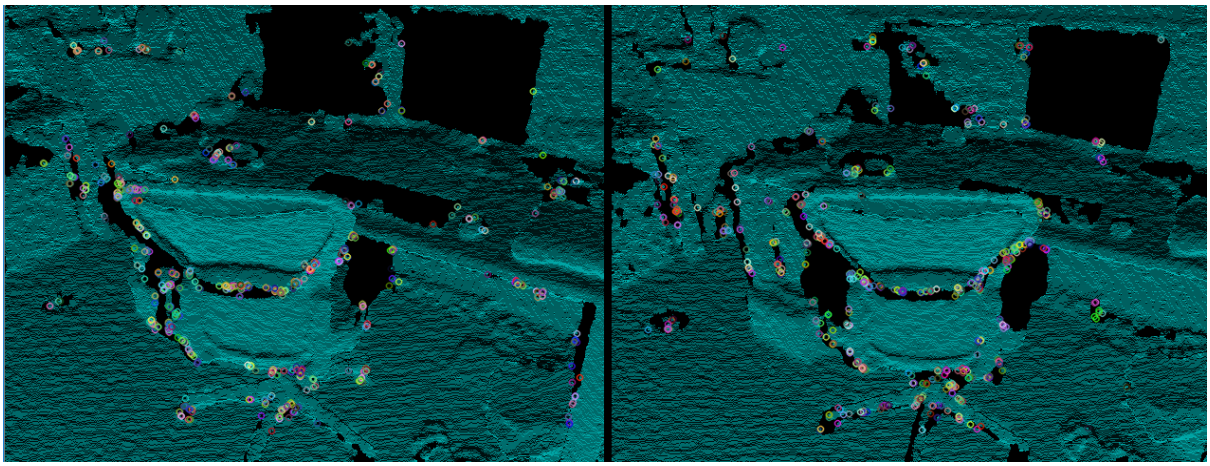


Рис. 11. Детектор ключевых точек ORB на BA изображениях

Визуально видно, что детектор SIFT находит много лишних ключевых точек, детектор SURF уже лучше определяет ключевых точки, однако все-равно присутствуют выбросы, и, наконец, детектор ORB находит такие ключевые точки, которые лежат только на границах объекта. Таким образом, для выявления ключевых точек был выбран детектор ORB.

На четвертом шаге необходимо найти такие точки на обоих изображениях, чтобы ключевая точка из первого изображения соответствовала такой же ключевой точке со второго изображения. Для этого используется дескриптор особых точек. В первой главе описаны три дескриптора с одноименными названиями детекторов: SIFT, SURF и ORB. Тестирование дескрипторов представлено на рисунках 12–14. Визуально видно, что дескриптор SIFT лучше находит соответствия ключевых точек, т.к. выбросов значительно меньше, чем у дескрипторов SURF и ORB.

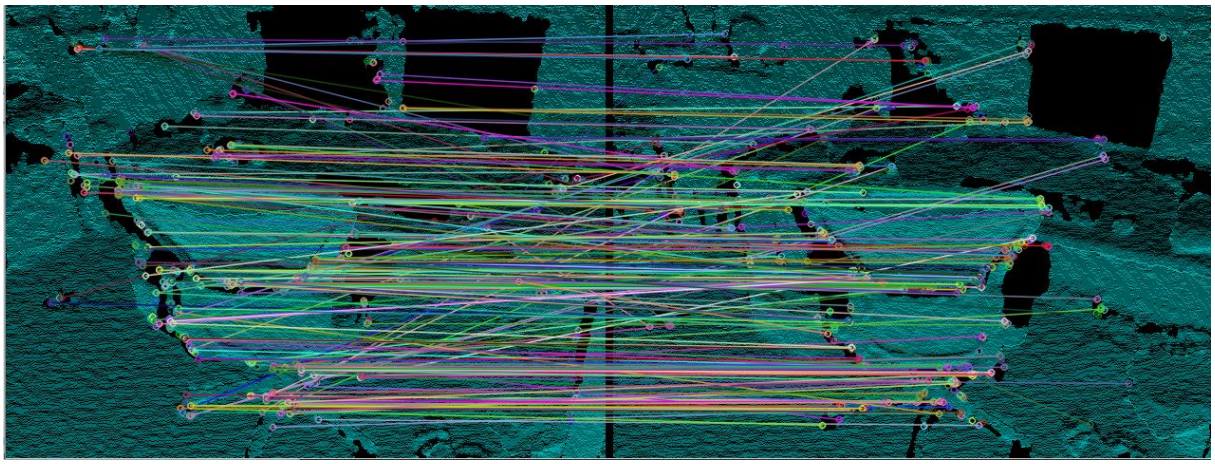


Рис. 12. Соотношение особых точек с помощью дескриптора SIFT

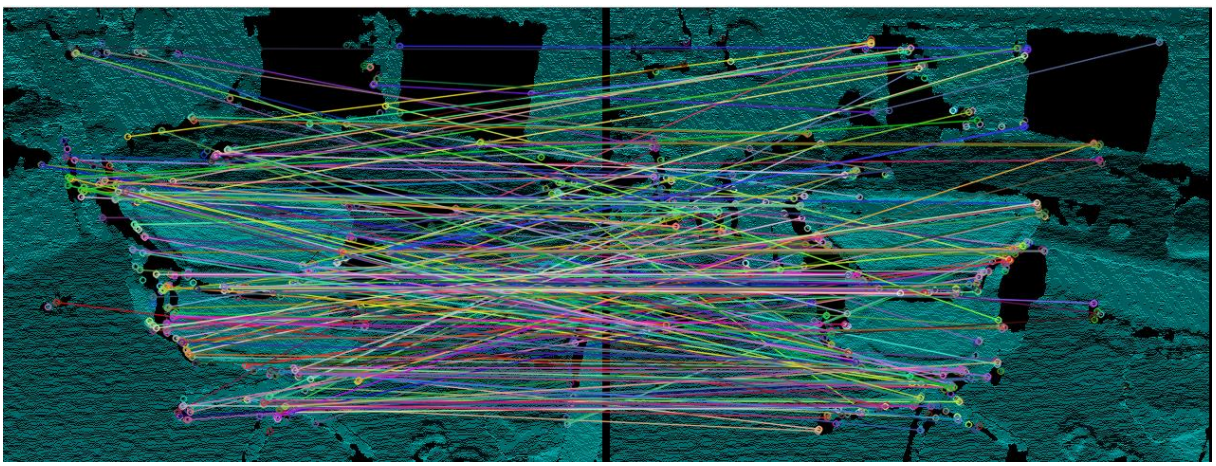


Рис. 13. Соотношение особых точек с помощью дескриптора SURF

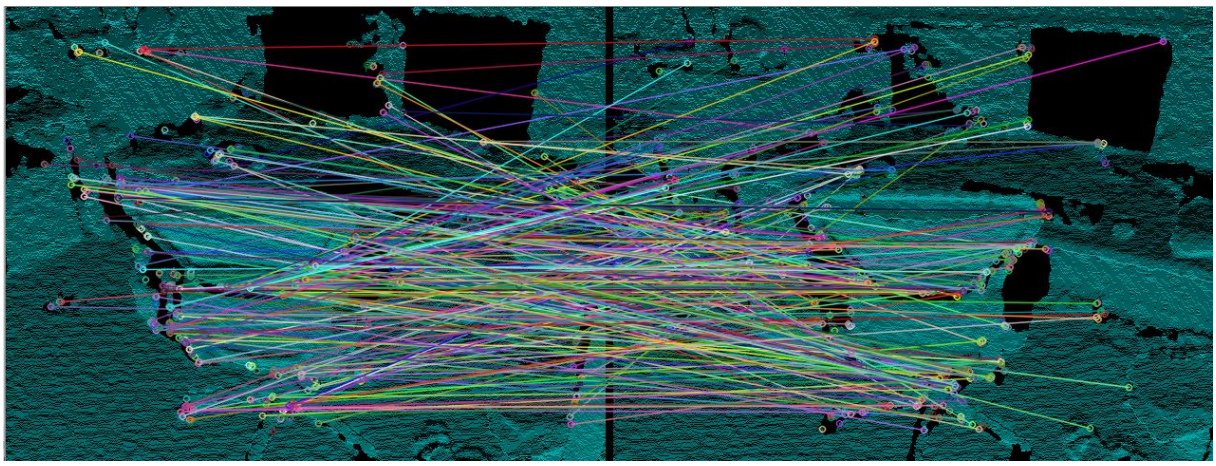


Рис. 14. Соотношение особых точек с помощью дескриптора ORB

На пятом шаге происходит фильтрация соответствий, чтобы получить только лучшие из них. Для этого сортируются пары ключевых точек по наименьшему расстоянию между дескрипторами, и из них выбирается только часть пар с наименьшим расстоянием. Затем выбирается наилучшая пара, у нее измеряется угол наклона и длина прямой, которая соединяет ключевые точки в паре. Из отобранных пар выбираются пары с примерно таким же наклоном и длиной прямой. В результате, находятся наилучшие соответствия пар ключевых точек, представленных на рисунке 15.

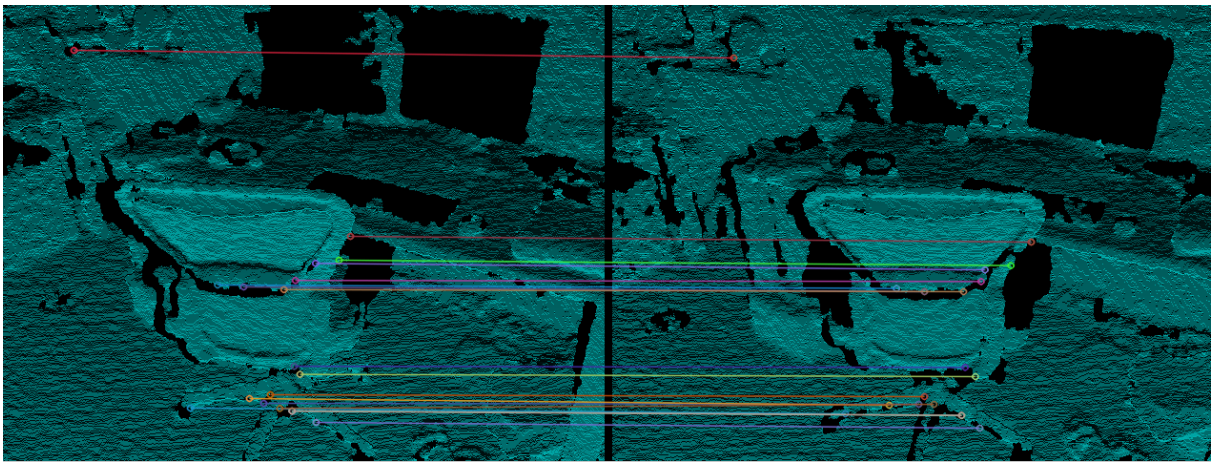


Рис. 15. Отобранные соответствия пар ключевых точек

На шестом шаге полученные двумерные ключевые точки преобразовываются в трехмерные. На рисунке 16 отмечены трехмерные ключевые точки, полученные в результате преобразования двумерных. Однако после

преобразования полученные трехмерные точки фильтруются, если хотя бы у одной точки из пары координаты равны 0, то такая пара отбрасывается. Также пары отбрасываются, если ключевые точки сильно отличаются по координате z, т.к. это означает, что точки лежат на разном уровне глубины.

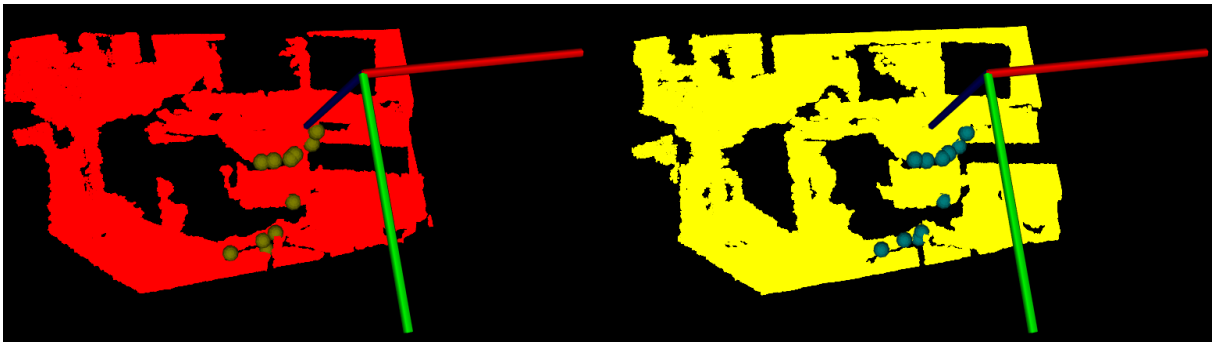


Рис. 16. Преобразование двумерной ключевой точки в трехмерную

На последнем шаге на основе ключевых точек выполняется алгоритм ICP с использованием глобальных контрольных точек и Манхэттенской метрики, представленный в листинге 4. В результате первое облако преобразуется к системе координат второго облака, таким образом, происходит совмещение.

Листинг 4. Алгоритм ICP на основе ключевых точек с использованием глобальных контрольных точек и Манхэттенской метрики

```
double err = 0;
GetCenter(meanPointStaticCloud, controlPointsStaticCloud);
for (int i = 0; i < iterations; i++)
{
	GetCenter(meanPointDynamicCloud, controlPointsDynamicCloud);
	newerr = Error();
	DistancesCloudAndControlCenter();
	GetRotateAndTranslate();
	Update();
	if (newerr < threshold or abs(newerr - err) < eps)
	{
		return dynamicCloud;
	}
	err = newerr;
}
return dynamicCloud;
```

Система совмещения данных трехмерного сканирования позволяет совмещать несколько облаков точек в одно облако. Совмещение происходит

постепенно. Сначала совмещаются первые два облака, и, как следствие, динамическое облако изменяет свои координаты. Затем преобразованное динамическое облако совмещается с третьим облаком. В результате третье облако также преобразовывает свои координаты в систему координат первого облака. Процесс продолжается до тех пор, пока все облака не преобразуются в систему координат первого облака через промежуточные облака точек. На последнем шаге все облака складываются, и, таким образом, формируется единая модель. Результат совмещения нескольких облаков точек представлен на рисунках 17–18.

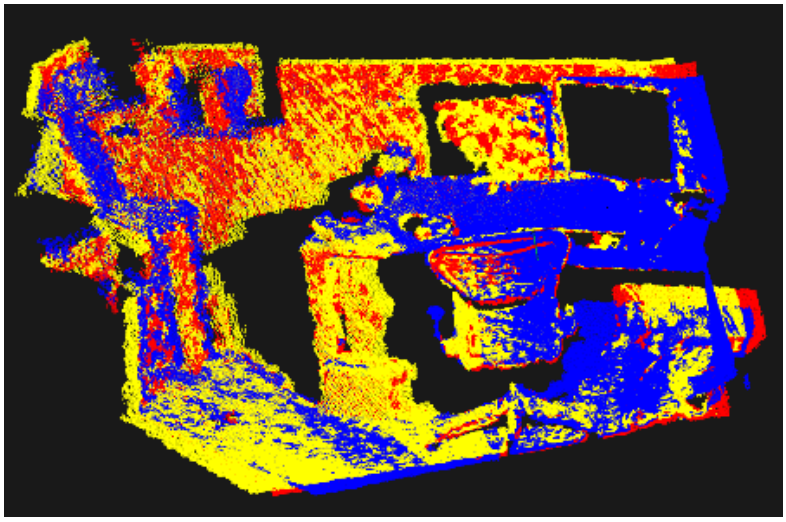


Рис. 17. Результат совмещения нескольких облаков точек сцены «Office»

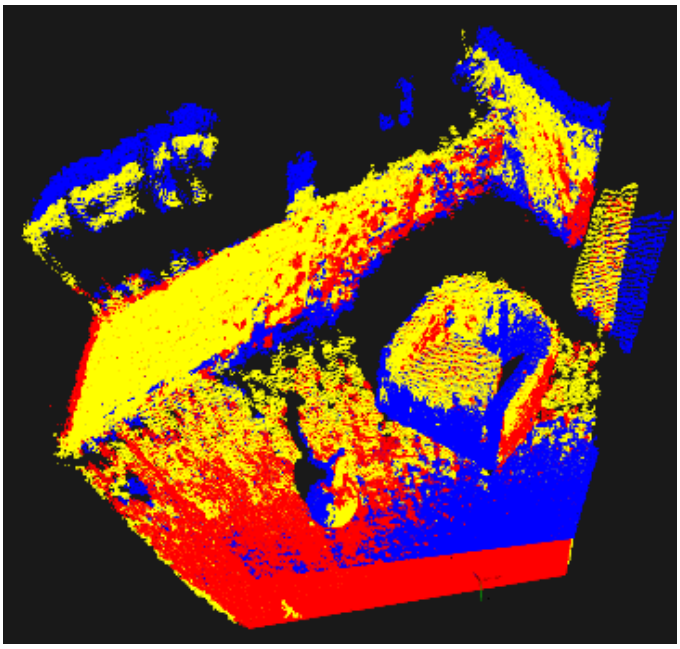


Рис. 18. Результат совмещения нескольких облаков точек сцены «Pumpkin»

4.3.2. Реализация интерфейса

Согласно функциональным требованиям система совмещения данных трехмерного сканирования предоставляет пользователю удобный интерфейс, состоящий из двух окон: окно ввода-вывода и окно визуализации облаков. Окно ввода-вывода позволяет загрузить снимки Kinect и предоставляет пользователю некоторую информацию в результате совмещения облаков, такую как время выполнения алгоритма и ошибку совмещения. Окно визуализации облаков позволяет пользователю просматривать облака точек до выполнения алгоритма и после выполнения алгоритма с любого ракурса. Интерфейс системы совмещения данных трехмерного сканирования представлен на рисунках 19–20.

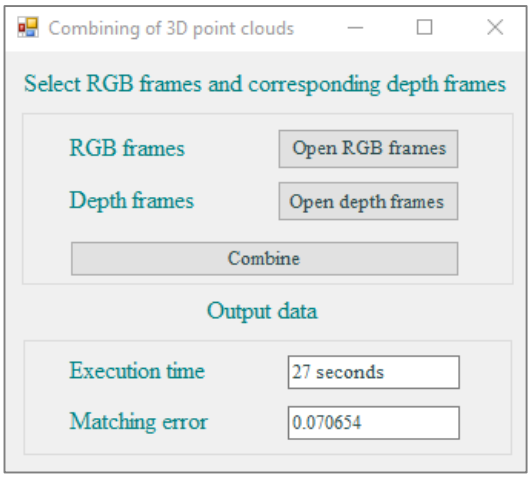


Рис. 19. Окно ввода-вывода

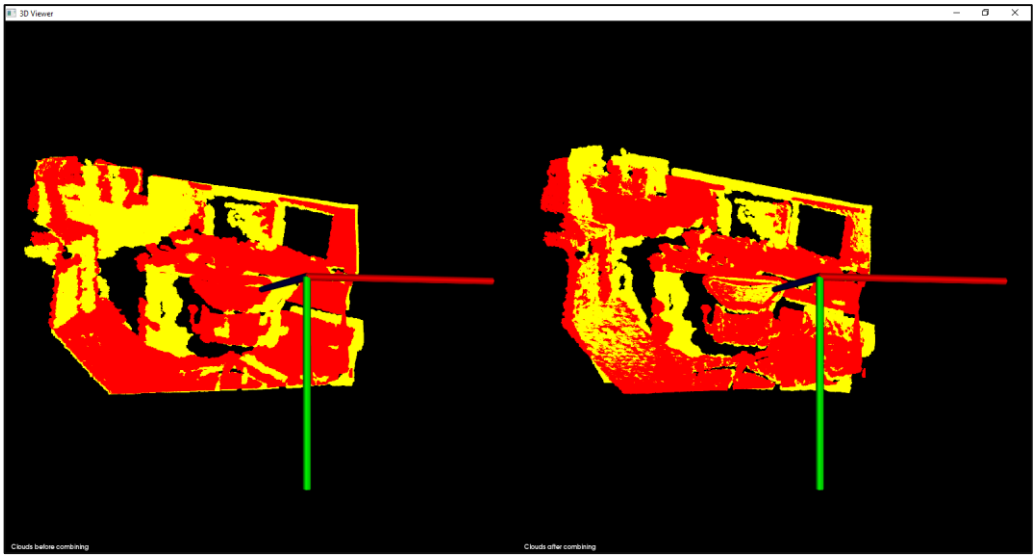


Рис. 20. Окно визуализации облаков

Вывод по главе 4

На основе разработанной архитектуры была реализована система совмещения данных трехмерного сканирования. Подробно рассмотрены средства реализации, структура системы и реализация модулей системы, включающая в себя, реализацию модуля совмещения облаков точек и реализацию интерфейса пользователя.

5. ТЕСТИРОВАНИЕ СИСТЕМЫ

Тестирование предложенной модификации алгоритма ICP было проведено в несколько этапов: тестирование алгоритма на основе синтетических и реальных данных с целью проверки адекватности результатов и сравнительное тестирование с целью определения достоинств и недостатков предложенного алгоритма в сравнение с существующими.

Для тестирования использовалась коллекция снимков RGB-D Dataset 7-Scenes от компании Microsoft [13], полученных с помощью камеры Kinect. Каждый снимок включает в себя файл с изображением глубины и файл с цветным изображением. Тестирование проводилось на ПК с Intel(R) Core(TM) i5-3230M CPU 2x2.60GHz и 6.0GB RAM.

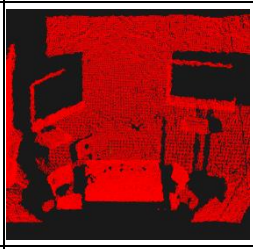
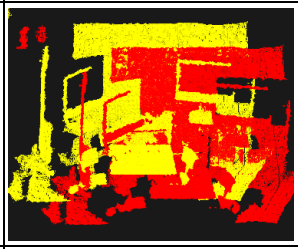
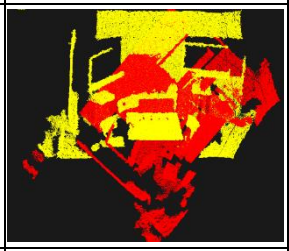
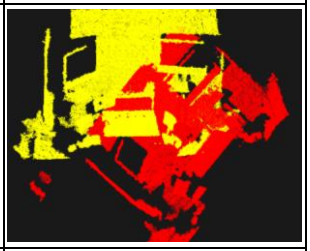
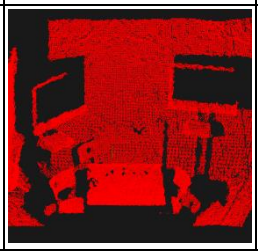
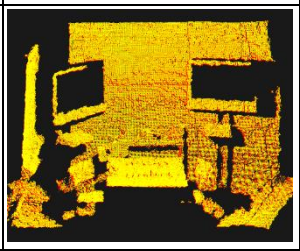
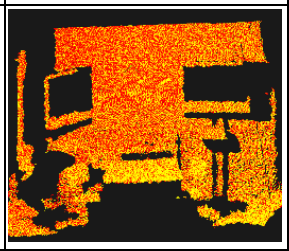
5.1. Тестирование предложенного алгоритма на синтетических данных

Для проверки корректности работы предложенной модификации алгоритма ICP проводилось тестирование следующим образом. Выбирался произвольный снимок из коллекции RGB-D Dataset 7-Scenes и конвертировался в облако точек. Затем формировалось второе облако, полученное в результате преобразования первого облака с помощью заданной матрицы поворота и вектора сдвига. На вход подавились два этих облака, и в результате совмещения сравнивались полученная матрица поворота R и полученный вектор сдвига T с заданными значениями. Рассматривалось 4 ситуации, представленных в таблице 1.

1. Идентичные облака точек.
2. Облака точек со сдвигом $T = \{0.5\ 0.3\ 0.1\}$.
3. Облака точек с поворотом $R = \{\alpha_X = 30°, \beta_Y = 15°, \gamma_Z = 20°\}$.
4. Облака точек со сдвигом T и поворотом R.

Проведенное тестирование показало, что предложенная модификация алгоритма ICP способна совмещать облака точек со 100 % областью перекрытия при любом сдвиге и повороте, т.к. ошибка между искомой и исходной матрицей поворота и сдвига является несущественной.

Табл. 1. Результаты тестирования предложенного алгоритма на основе синтетических данных

	Ситуация 1	Ситуация 2	Ситуация 3	Ситуация 4
Исходные R и T	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	1 0 0 0.5 0 1 0 0.3 0 0 1 0.1 0 0 0 1	0.65 -0.76 0.09 0 0.75 0.61 -0.24 0 0.13 0.22 0.97 0 0 0 0 1	0.65 -0.76 0.09 0.5 0.75 0.61 -0.24 0.3 0.13 0.22 0.97 0.1 0 0 0 1
Искомые R и T	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1	0.999 -0.001 0 0.502 0 0.999 0.001 0.299 0 -0.002 0.999 0.087 0 0 0 1	0.649 -0.756 0.089 0 0.751 0.612 -0.243 0 0.129 0.224 0.968 0 0 0 0 1	0.649 -0.756 0.089 0.499 0.751 0.612 -0.243 0.299 0.129 0.224 0.968 0.099 0 0 0 1
Ошибка расчета R и T	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0.001 0.001 0 0.002 0 0.001 -0.001 0.001 0 0.002 0.001 0.013 0 0 0 0	0.001 -0.004 0.001 0 -0.001 -0.002 0.003 0 0.001 -0.004 0.002 0 0 0 0 0	0.001 -0.004 0.001 0.001 -0.001 -0.002 0.003 0.001 0.001 -0.004 0.002 0.001 0 0 0 0
Облака точек до совмещения				
Значение метрики до совмещения	0	0.9019434	0.7221667	1.4332741
Облака точек после совмещения				
Значение метрики после совмещения	0	0.0050074	0.0003632	0.0043389

Было выявлено, что значение метрики до совмещения и после совмещения самые минимальные в третьей ситуации, т.е. когда происходило совмещения облаков, которые были повернуты относительно друг друга. Это

объясняется тем, что во время поворота точки, лежащие рядом с центром облака, практически не изменяют свои координаты, а изменяют сильно свои координаты только точки, лежащие на расстояние от центра. При совмещение облаков сдвинутых относительно друг друга, ошибка больше, чем в третьей ситуации, потому что все точки при сдвиге изменяют свои координаты, и чем больше сдвиг, тем больше ошибка до совмещения. В четвертой ситуации ошибка максимальная до совмещения, т.к. точки при повороте и сдвиге сильно изменили свои координаты.

5.2. Тестирование устойчивости предложенного алгоритма

Для оценки устойчивости предложенного алгоритма ICP были взяты снимки Kinect из коллекции «Pumpkin». Проводилось тестирование с различными группами снимков с целью определения устойчивости предложенного алгоритма к частичной области перекрытия. В таблицах 2–4 приведены результаты тестирования.

Табл. 2. Результаты совмещения снимков из коллекции «Pumpkin» 000 и 050

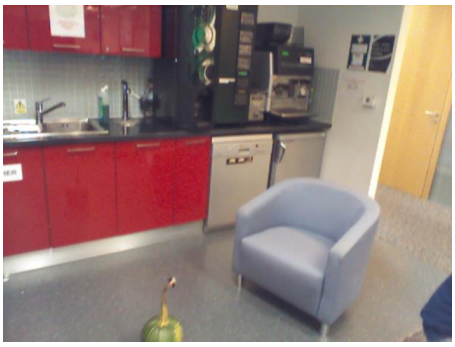
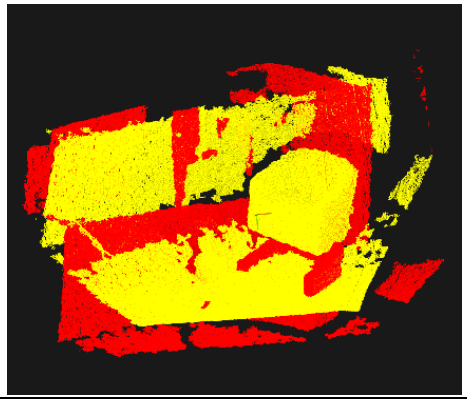
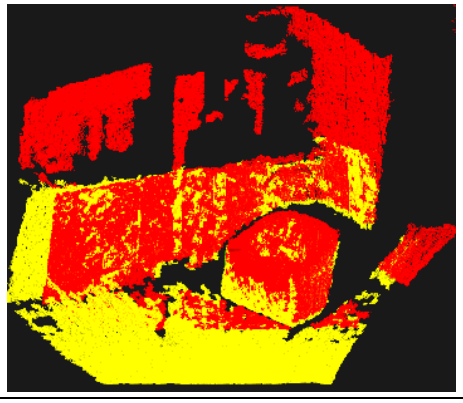
Исходные снимки		
Облака точек до совмещения и после		
Значение метрики до совмещения и после	1.2812542	0.1391239

Табл. 3. Результаты совмещения снимков из коллекции «Pumpkin» 000 и 080

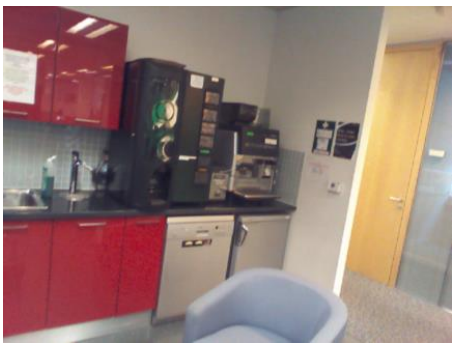
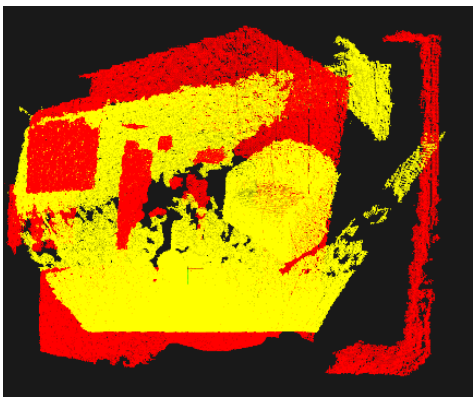
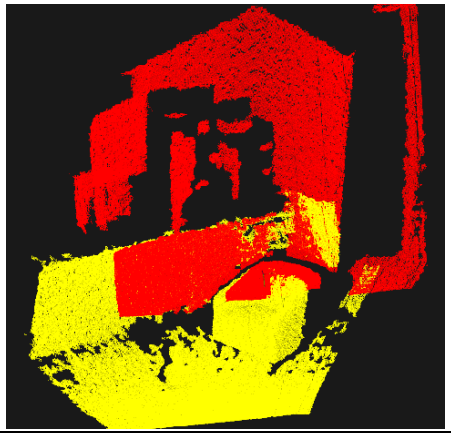
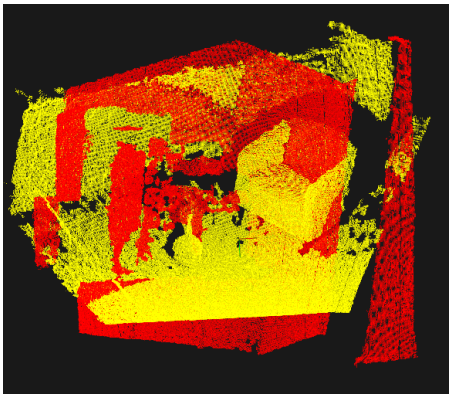
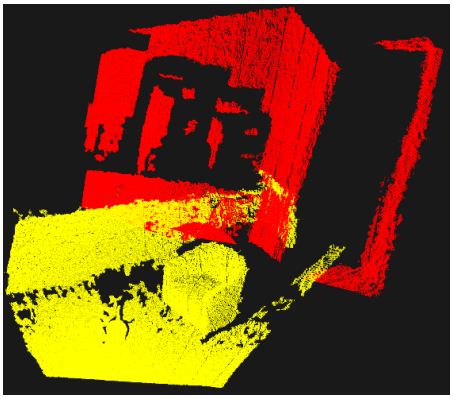
Исходные снимки		
Облака точек до совмещения и после		
Значение метрики до совмещения и после	2.4119668	0.5414112

Табл. 4. Результаты совмещения снимков из коллекции «Pumpkin» 000 и 100

Исходные снимки		
Облака точек до совмещения и после		

Окончание табл. 4

Значение метрики до совмещения и после	2.7280316	0.9280041

В ходе тестирования устойчивости предложенной модификации алгоритма ICP к частичной области перекрытия, было выявлено, что данный алгоритм способен с небольшой ошибкой совмещать снимки с областью перекрытия 50% и более. Однако при области перекрытия менее 50% точность совмещения уменьшается прямо пропорционально сокращению области перекрытия. Это связано с тем, что чем меньше область перекрытия, тем меньше особых точек будет найдено.

5.3. Сравнительное тестирование алгоритма ICP и его модификаций

Для оценки эффективности и производительности предложенного алгоритма проводилось сравнительное тестирование с базовым ICP и GICP по ошибке совмещения и по времени выполнения. Для сравнения использовалось три коллекции снимков от компании Microsoft: «Stairs», «RedKitchen», «Fire», представленных на рисунках 21–23 справа и слева.

Сравнение происходило следующим образом. Выбирался 0 снимок и совмещался с 1, 20, 50 и 80. Оценивались в каждом случае ошибка совмещения и время выполнения. На основе полученных данных были построены графики зависимостей ошибки совмещения и времени выполнения от номеров снимка.

Рис. 21. Снимки 00 и 80 из коллекции «Stairs»

Рис. 22. Снимки 00 и 80 из коллекции «RedKitchen»

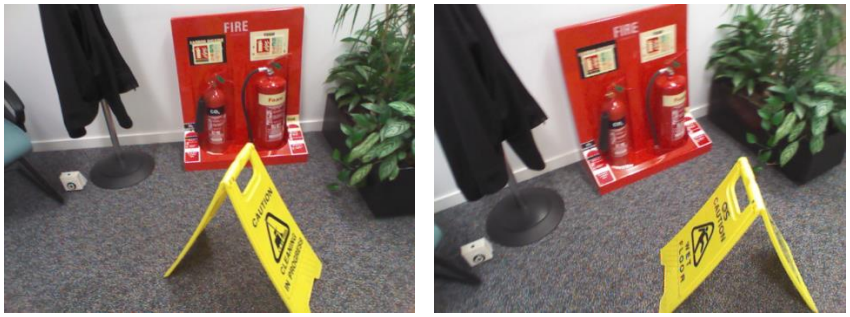


Рис. 23. Снимки 00 и 80 из коллекции «Fire»

На рисунках 24–26 представлены графики зависимости ошибки совмещения от номеров снимков в трех коллекциях. Предложенный алгоритм обозначен как MyICP. Графики показывают, что MyICP незначительно хуже совмещает облака точек с максимальной областью перекрытия в сравнение с ICP и GICP. Однако, при постепенном уменьшении области перекрытия MyICP начинает выигрывать по точности совмещения. Также ошибка совмещения варьируется в зависимости от коллекции на снимках с одинаковым интервалом. Это происходит потому, что сцены снимаются под разными углами и с разными смещением. Поэтому можно заметить, что на снимках с одинаковым интервалом в сцене «Stairs» область перекрытия меньше, а угол поворота больше, чем в других сценах. Отсюда следует, что и на графике, представленном на рисунке 24, ошибка совмещения во всех алгоритмах выше, чем на графиках, представленных на рисунках 25 и 26. Самая маленькая ошибка совмещения в коллекции «Fire», потому что смещение сильного не происходит, меняется только угол съемки.

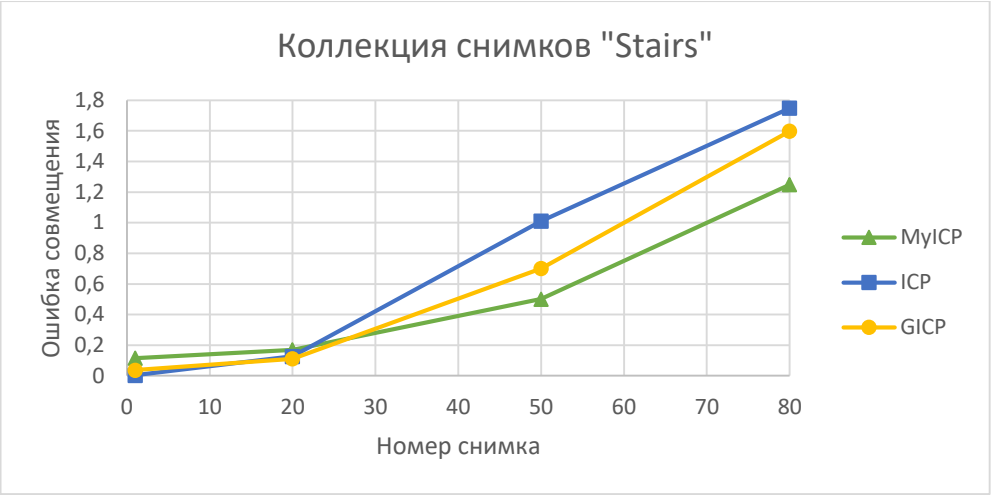


Рис. 24. График зависимости ошибки совмещения для коллекции «Stairs»

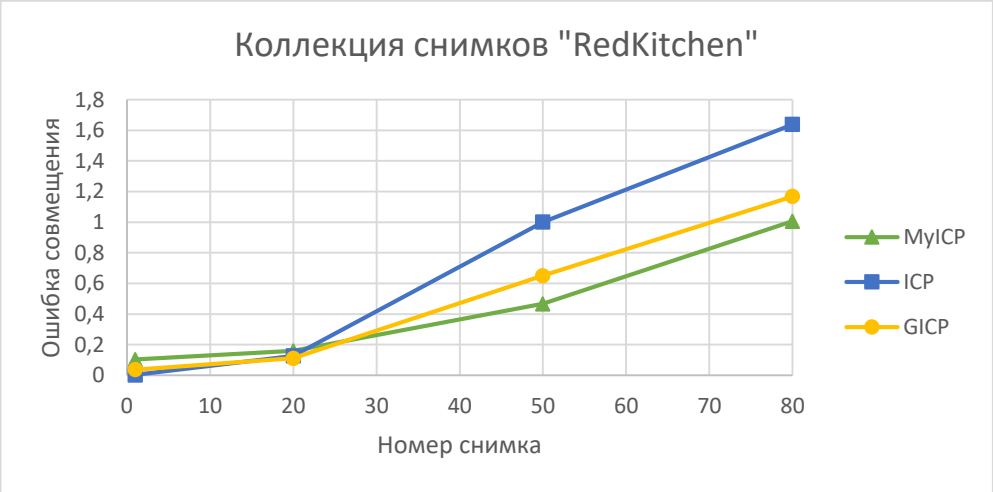


Рис. 25. График зависимости ошибки совмещения для коллекции «RedKitchen»

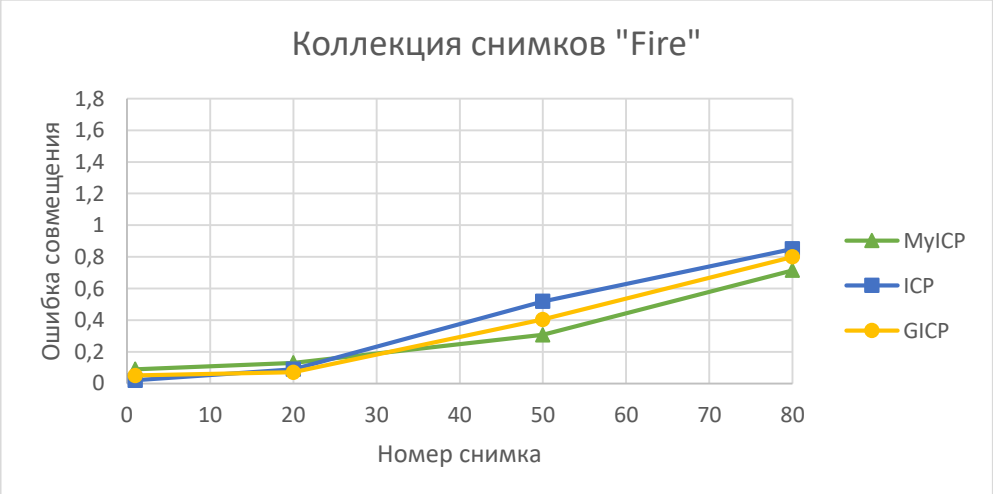


Рис. 26. График зависимости ошибки совмещения для коллекции «Fire»

На рисунках 27–29 представлены графики зависимости времени выполнения от номеров снимков в трех коллекциях.

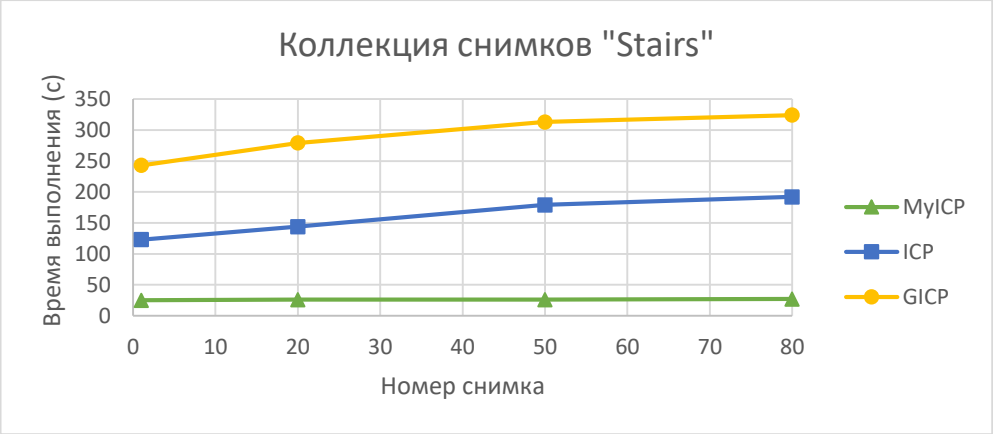


Рис. 27. График зависимости времени выполнения для коллекции «Stairs»

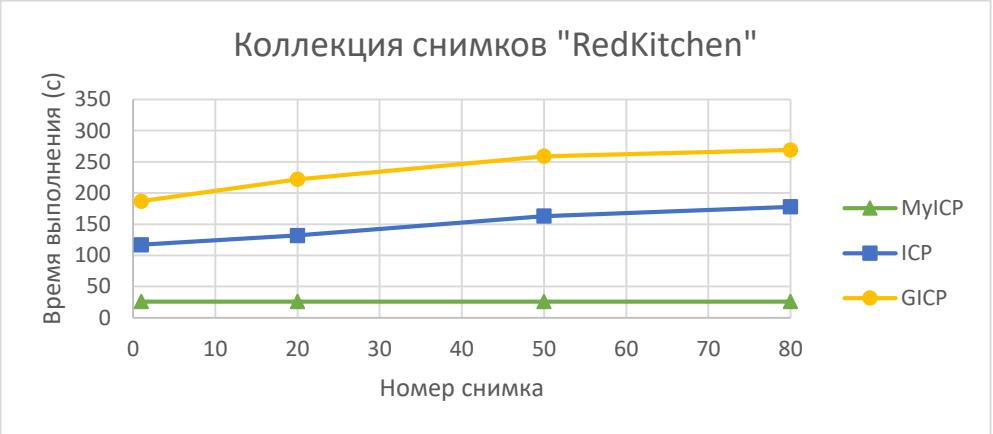


Рис. 28. График зависимости времени выполнения для коллекции «RedKitchen»

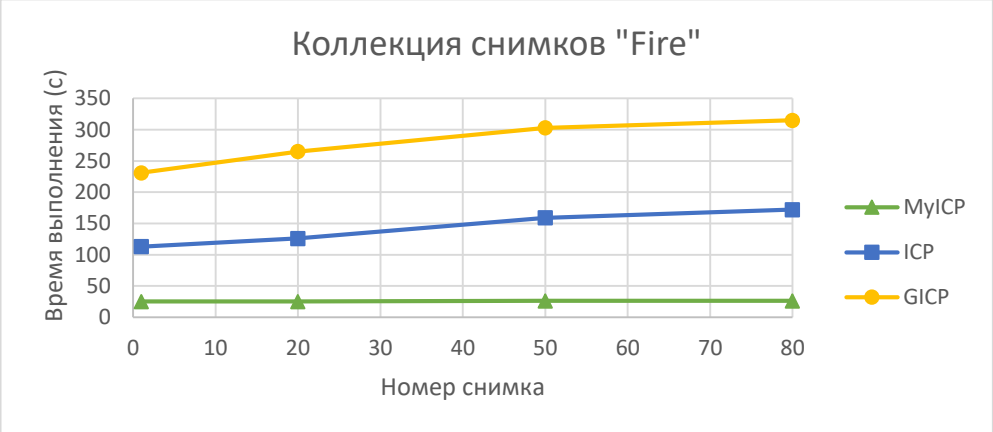


Рис. 29. График зависимости времени выполнения для коллекции «Fire»

Согласно графикам, время выполнения в предложенном алгоритме значительно меньше, чем у ICP и GICP. Это объясняется тем, что в MyICP совмещение происходит только на основе особых точек, а их обычно не больше 30, поэтому и времени на совмещение требуется меньше. GICP является самым долгим по времени выполнения, потому что затрачивается много времени на совмещение небольших участков поверхностей друг с другом. Однако, чем больше будет найдено гладких и плоских участков поверхностей, тем меньше потребуется времени на совмещение, т.к. будет выполнено меньшее количество итераций. Также стоит отметить, что время выполнения у ICP и GICP растет прямо пропорционально уменьшению области перекрытия.

Таким образом, сравнительно тестирование позволило выявить следующие достоинства и недостатки предложенной модификации ICP в сравнение с другими алгоритмами:

1) предложенный алгоритм согласно ошибке совмещения лучше совмещает облака точек с частичной областью перекрытия, и незначительно хуже при максимальной области перекрытия, чем ICP и GICP;

2) время выполнения предложенного алгоритма значительно меньше в сравнение с ICP и GICP при любой области перекрытия.

Вывод по главе 5

Было проведено три вида тестирований: тестирование предложенного алгоритма на основе синтетических данных, тестирование устойчивости алгоритма к частичной области перекрытия и сравнительное тестирование с базовым ICP и GICP. В ходе тестирования были определены достоинства и недостатки предложенной модификации алгоритма ICP, а также было определено в чем данная модификация выигрывает у существующих алгоритмов ICP, а в чем проигрывает.

ЗАКЛЮЧЕНИЕ

В рамках дипломного проекта была разработана программная система для совмещения данных трехмерного сканирования.

При этом были решены следующие задачи:

1) был рассмотрен базовый алгоритм ICP и его недостатки;

2) проведен обзор существующих модификаций алгоритма ICP и смежных проектов;

3) определены требования к системе совмещения данных трехмерного сканирования и разработаны варианты ее использования;

4) спроектирована архитектура системы совмещения данных трехмерного сканирования;

5) разработана система совмещения данных трехмерного сканирования на основе алгоритма ICP с использованием BA изображений, глобальных контрольных точек и Манхэттенской метрики;

6) проведено тестирование разработанной системы на синтетических, реальных данных и сравнительное тестирование с базовым алгоритмом ICP и GICP. Выявлено, что предложенный алгоритм является более эффективным и производительным на снимках с частичной областью перекрытия.

В ходе работы была выполнена одна апробация: доклад на тему «Автоматизированная система совмещения данных трехмерного сканирования на 71 научной студенческой конференции Южно-Уральского государственного университета 14 мая 2018 г.

СПИСОК ЛИТЕРАТУРЫ

1. Chibunichev A.G., Velizhev A.B. Automatic matching of terrestrial scan data using orientation histograms. // The international archives of the photogrammetry, remote sensing and spatial information sciences. – 2008. – Vol. 37. – P. 601–603.

2. Du S. Robust iterative closest point algorithm based on global reference point for rotation invariant registration. / S. Du, Y. Xu, T. Wan, H. Hu, S. Zhang, et al. // PloS One. – 2017. – Vol. 12(11). – P. 1–14.

3. Eruhimov V., Kogan A. Dependency of detectors and descriptors efficiency on image resolution for SIFT, SURF and ORB. // The 23rd International Conference on Computer Graphics and Vision. – 2013. – P. 32–35.

4. Gieseke F. Buffer k-d trees: processing massive nearest neighbor queries on GPUs. / Gieseke F., Heinermann J., Oancea C., Igel C. // Journal of Machine Learning Research. – 2014. – Vol. 32 (1). – P. 172–180.

5. He Y. An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features. / Y. He, B. Liang, J. Yang, S. Li, J. He. // Sensors. – 2017. – Vol. 17(8). – P. 1–16.

6. Isik S., Ozkan K. A Comparative Evaluation of Well-known Feature Detectors and Descriptors. // International Journal of Applied Mathematics, Electronics and Computers. – 2014. – No. 3(1). – P. 1–6.

7. Karami E., Prasad S., Shehata M. Image Matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. // In proceedings of the 2015 newfoundland electrical and computer engineering conference. – 2015. – P.1–5.

8. Kumara W.G.C. 3D models construction from RGM video stream. / W.G.C. Kumara, K.S. Timothy, S.J. Wu, A.S. Klimenko, S.V. Klimenko. // Proceedings of the international scientific conference CRT2015. – 2016. – P.59–66.

9. Lin C.C. A novel point cloud registration using 2D image features. / C.C. Lin, Y.C. Tai, J.J. Lee, Y.S. Chen. // Journal on Advances in Signal Processing. – 2017. – Vol. 5. – P.1–11.

10. Mora H. Computational analyses of distance operators for the iterative closest point algorithm. / H. Mora, J.M. Mora-Pascual, A. Garcia-Garcia, P. Martinez-Gonzalez. // PloS One. – 2016. – Vol. 11(10). – P. 1–19.

11. Official site CloudCompare. [Electronic resource] URL: http://www.cloudcompare.org/ (date of access: 13.03.2018).

12. Official site Geomagic Studio. [Electronic resource] URL: http://support1.geomagic.com/Support/5605/5668/en-US/Article/Folder/336/Geomagic-Studio (date of access: 10.03.2018).

13. Official site RGB-D Dataset 7-Scenes. [Electronic resource] URL: https://www.microsoft.com/en-us/research/project/rgb-d-dataset-7-scenes/ (date of access: 23.04.2018).

14. Rublee E. ORB: An efficient alternative to SIFT or SUFT. / E. Rublee, V. Rabaud, K. Konolige, G. Bradski. // ICCV '11 Proceedings of the 2011 International Conference on Computer Vision. – 2011. – P. 2564–2571.

15. Segal A., Haehnel D., Thrun S. Generalized-ICP. // Robotics: Science and Systems. – 2009. – P. 26–27.

16. Servos J., Waslander S. L. Multi-Channel Generalized-ICP: A robust framework for multi-channel scan registration. // Robotics and Autonomous systems. – 2017. – Vol. 87. – P. 247–257.

17. Tao L., Bui T., Hasegawa H. Global ray-casting range image registration. // IPSJ transactions on computer vision and applications. – 2017. – Vol. 9. – No. 1. - P.1–15.

18. Абрамов А.И., Абрамов И.В., Мазитов Т.А. Модификация алгоритма ICP путем внедрения коэффициента усиления для ускорения совмещения двумерных облаков точек. // Интеллектуальные системы в производстве. – 2016. – Вып. 2 (29). – С. 4–9.

19. Анатомия 3D-моделей. [Электронный ресурс] URL: http://www.soobshestva.ru/wiki/anathomy_of_3d_model/ (дата обращения: 15.02.2018).

20. Башков Е.А., Бабков В.С. Средства интерактивного виртуального и реального пространств на основе трехмерного сканирования объектов с использованием платформы Microsoft Kinect. // Известия ЮФУ. Технические науки. – 2013. – Вып. 5(142). – С. 51–56.

21. Бобков В.А. Восстановление траектории движения робота и реконструкция среды по изображениям. / В.А. Бобков, А.П. Кудряшов, С.В. Мельман, М.А. Морозов. // Вестник Дальневосточного отделения Российской академии наук. – 2016. – Вып. 4. – С. 60–69.

22. Детекторы углов. [Электронный ресурс] URL: https://habr.com/post/244541/ (дата обращения: 15.02.2018).

23. Караваев Ю.Л., Клековкин А.В., Лесин С.К. Мультисенсорная информационно-измерительная система мобильного робота для реализации движения в недетерминированной среде. // Интеллектуальные системы в производстве. – 2016. – Вып. 4(31). – С. 111–115.

24. Константинов В.М., Орлова Ю.А., Розалиев В.Л. Разработка 3D- модели тела человека с использованием MS Kinect. // Известия Волгоградского государственного технического университета. – 2015. – Вып. 6(163). – С.65–69.

25. Ляпишев К.М., Погорелов А.В., Шуляков Д.Ю. Исследование оползней с применением технологии наземного лазерного сканирования. // Геодезия, картография и маркшейдерия. – 2014. – С.26–32.

26. Маковецкий А.Ю. Точные решения вариационной задачи алгоритма ICP в классе аффинных преобразований. / А.Ю. Маковецкий, С.М. Воронин, Д.В. Тихоньких, М.Н. Алексеев. // Челябинский физико-математический журнал. – 2017. – Т. 2. – Вып. 3. – С. 282–294.

27. Официальный сайт VR Concept. [Электронный ресурс] URL: http://vrconcept.net/ (дата обращения: 15.03.2018).

28. Работа с облаками точек. [Электронный ресурс] URL: https://knowledge.autodesk.com/ru/support/autocad/learn-explore/caas/CloudHe

lp/cloudhelp/2016/RUS/AutoCAD-Core/files/GUID-C0C610D0-9784-4E87-A857-F17F1F7FEEBE-htm.html (дата обращения: 15.02.2018).

29. Стешенко А.С., Розалиев А.В., Орлова Ю.А. Автоматизация построения 3D-модели головы человека при помощи Microsoft Kinect. // Известия Волгоградского государственного технического университета. – 2016. – Вып. 3(182). – С.83–86.

30. Тлеубаев И. С. Аналитический обзор алгоритмов совмещения данных трехмерного сканирования. // Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. – 2016. – Т. 2. – С.231–232.

31. Тлеубаев И. С. Алгоритмы совмещения данных трехмерного сканирования. // Молодежь и современные информационные технологии: сборник трудов XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. – 2016. – Т. 2. – С. 186–187.

32. Цапко И.В., Омельянюк М.Ю. Совмещение трехмерных изображений, полученных в результате ручного лазерного сканирования. // Вестник науки Сибири. – 2014. – Вып. 4(14). – С. 112–116.