

Комбинированный алгоритм совмещения облаков точек

А.И. Ефимов¹, Т.Н. Крючкова¹, А.И. Ярославцева¹

¹ Рязанский государственный радиотехнический университет имени В.Ф. Уткина, ул.Гагарина, 59/1, Рязань, 390005, Россия

Аннотация

Вопросы совмещения трехмерных поверхностей и трехмерных облаков точек являются важными для решения целого ряда прикладных задач: трехмерной реконструкции подстилающей поверхности летательных аппаратов, построения трехмерных моделей объектов и ряда других. В докладе рассмотрены вопросы регистрации облаков точек, в частности алгоритм жесткой регистрации трехмерных облаков. Описана программная реализация итеративного алгоритма совмещения трехмерных облаков точек, проанализированы его основные преимущества и недостатки, а также предлагается модификация алгоритма ICP, включающая предварительное совмещение соответствующих частей облаков с применением подхода на основе трехмерных дескрипторов ключевых точек. Применение комбинированного алгоритма совмещения трехмерных облаков позволяет строить плотные трехмерные модели объектов на основе их отдельных разрозненных стереоснимков. Представлены точностные и временные оценки предложенных подходов.

Ключевые слова

3D-модель, трехмерные облака точек, алгоритм жесткой регистрации, алгоритм итеративной ближайшей точки, комбинированный алгоритм совмещения облаков точек.

Combined Algorithm for Combining Point Clouds

A.I. Efimov¹, T.N. Kryuchkova¹, A.I. Yaroslavtseva¹

¹ Ryazan State Radio Engineering University named after V.F.Utkin Ryazan, 390005, Russia

Abstract

The issues of combining three-dimensional surfaces and three-dimensional point clouds are important for solving a number of applied problems: three-dimensional reconstruction of the underlying surface of aircraft, construction of three-dimensional models of objects and a number of others. The report deals with the registration of point clouds, in particular, the algorithm of rigid registration of three-dimensional clouds. A software implementation of an iterative algorithm for combining three-dimensional point clouds is described, its main advantages and disadvantages are analyzed, and a modification of the ICP algorithm is proposed, including preliminary combination of the corresponding parts of the clouds using an approach based on three-dimensional descriptors of key points. The use of a combined algorithm for combining three-dimensional clouds allows you to build dense three-dimensional models of objects based on their separate disparate stereo images. The accuracy and time estimates of the proposed approaches are presented.

Keywords

3D models, three-dimensional point clouds, rigid registration algorithm, iterative nearest point algorithm, combined point cloud matching algorithm.

ГрафиКон 2023: 33-я Международная конференция по компьютерной графике и машинному зрению, 19-21 сентября 2023 г., Институт проблем управления им. В.А. Трапезникова Российской академии наук, г. Москва, Россия

EMAIL: lexie62rus@mail.ru (А.И. Ефимов); kriuchkovatiana2000@mail.ru (Т.Н. Крючкова); 79065445680@yandex.ru (А.И. Ярославцева)

ORCID: 0000-0002-4014-8718 (А.И. Ефимов); 0009-0004-1762-9689 (Т.Н. Крючкова); 0000-0001-7612-813X (А.И. Ярославцева)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

1. Введение

В настоящее время проблема совмещения трехмерных облаков точек является довольно распространенной, так как данные методы широко используются в создании 3D-моделей, что, в свою очередь, применяется в различных сферах человеческой жизни: в кинематографе, медицине, архитектуре, метрологии, компьютерном моделировании и так далее.

Совмещение трехмерных облаков точек обусловлено так называемой регистрацией облаков точек. Схема процесса регистрации представлена на рисунке 1.

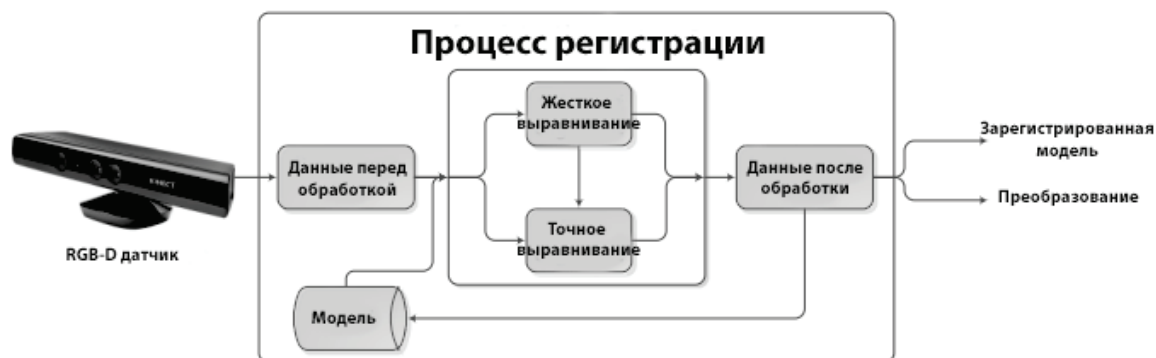


Рисунок 1– Процесс регистрации облаков точек

Существует большое количество алгоритмов для данной регистрации. Среди них особо выделяются следующие:

- 1) алгоритм жесткой регистрации;
- 2) алгоритм нежесткой регистрации.

Алгоритм жесткой регистрации построен на использовании жесткого преобразования, отображающего одни точки в другие. Данное преобразование относится к преобразованиям евклидова пространства. Оно сохраняет евклидово расстояние между каждой парой точек, и, таким образом, в ходе правильного преобразования объект должен сохранить свою форму. К жестким преобразованиям относятся сдвиги, повороты, отражения или любая их последовательность. Отражения иногда исключают из определения жесткого преобразования, так как требуется, чтобы преобразование также сохраняло одну сторону объекта в евклидовом пространстве (например, чтобы левая рука внезапно не стала правой рукой и т.д.).

Все жесткие преобразования являются примерами аффинных преобразований. Множество всех жестких преобразований представляет собой математическую группу, так называемую евклидовую группу, обозначаемую $E(n)$ для n -мерных евклидовых пространств. Множество собственных жестких преобразований называется специальной евклидовой группой, с обозначением $SE(n)$.

Жесткое преобразование формально определяется, как преобразование, которое при воздействии на произвольный вектор v в n -мерном евклидовом пространстве создает преобразованный вектор $T(v)$:

$$T(v) = Rv + t, \quad (1)$$

где $R^T = R^{-1}$ (т.е. R – ортогональное преобразование), а t – вектор, дающий перевод начала координат.

Правильное жесткое преобразование имеет, кроме того,

$$\det(R) = 1, \quad (2)$$

что означает, что R не создает отражения и, следовательно, представляет собой вращение (ортогональное преобразование с сохранением ориентации).

Пример использования алгоритма жесткой регистрации представлен на рисунке 2.

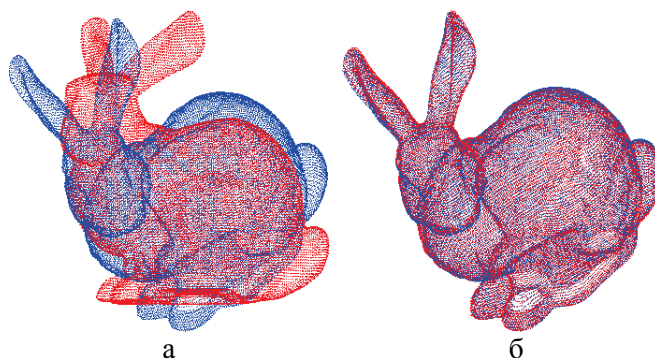


Рисунок 2– Пример использования алгоритма жесткой регистрации: а – облака точек до процедуры совмещения, б – результат совмещения облаков

В алгоритм жесткой регистрации входит такой метод, как «итеративная ближайшая точка» (ICP - IterativeClosestPoint) [1, 2].

2. Алгоритм ICP

Итеративная ближайшая точка – алгоритм, используемый для регистрации облаков точек. ICP часто используется для реконструкции 2D или 3D поверхностей из полученных разрозненных отсканированных изображений, для локализации роботов и достижения оптимального планирования движения (например, когда привод колес ненадежен из-за скользкой местности) и т.д. Его цель состоит в том, чтобы найти такое преобразование, которое минимизирует среднееквадратичное евклидово расстояние между соответствующими точками.

В алгоритме итеративной ближайшей точки (итеративной соответствующей точки) одно облако точек – эталон (цель) – остается фиксированным, в другое – исходное – преобразуется для наилучшего соответствия эталону. Алгоритм итеративно пересматривает преобразование (комбинацию переноса и поворота), необходимое для минимизации метрики ошибки, обычно расстояния от источника до облака опорных точек. ICP – один из широко используемых алгоритмов для совмещения трехмерных моделей с учетом исходного предположения о требуемом жестком преобразовании.

Алгоритм итеративной ближайшей точки отличается от таких алгоритмов, как алгоритм Кабша [3] и других решений ортогональной задачи Прокруста (задачи матричной аппроксимации в линейной алгебре) тем, что алгоритм Кабша требует соответствия между наборами точек в качестве входных данных, а итеративный алгоритм ближайшей точки рассматривает соответствие как переменную, которую необходимо подвергнуть оценке.

Входные данные: эталонное и исходное облака точек (сцена / модель), начальная оценка преобразования для выравнивания источника с эталоном (необязательное условие), критерии остановки итераций (достаточная точность).

Выход: уточненное преобразование.

Шаги алгоритма итеративной ближайшей точки:

1) для каждой точки (из всего набора вершин, обычно называемого плотным, или выборки пар вершин из каждой модели) в исходном облаке точек сопоставить ближайшую точку в облаке опорных точек (или выбранном наборе);

2) оценить комбинацию поворота и переноса, используя среднееквадратичный метод минимизации метрики расстояния от точки до точки, который лучше всего выравнивает каждую исходную точку с ее сопоставлением, найденным на предыдущем этапе. Данный шаг также может включать взвешивание точек и отбрасывание лишнего до выравнивания;

3) преобразовать исходные точки, используя полученное преобразование;

4) повторить шаги (повторно связать точки и т.д.).

Структура метода ICP представлена на рисунке 3.

Из рисунка 3 можно выделить, что один из наборов данных – модель, другой – сцена. ICP начинает работу с заданного начального преобразования, а затем продолжает итерацию в два последовательных шага. Сначала точки сцены обрабатываются с использованием текущего

преобразования. После этого пары соответствий вычисляются с использованием точек сцены и модели. В конце каждой итерации и с использованием информации о паре соответствий вычисляется преобразование, которое наилучшим образом показывает соответствия.

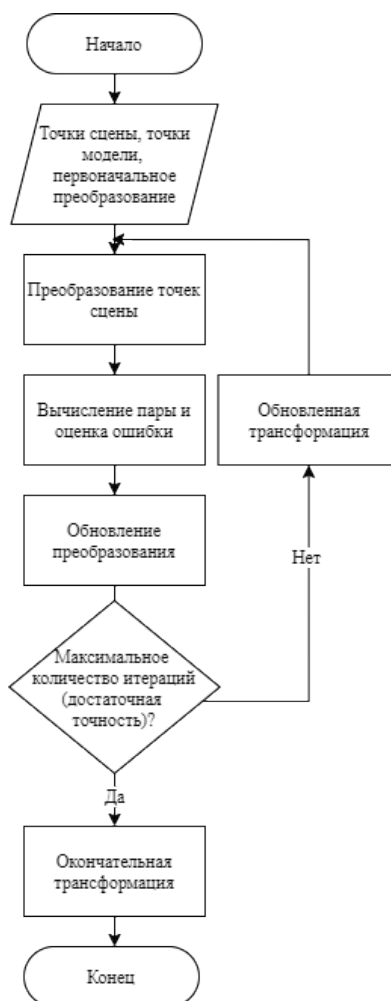


Рисунок 3– Структура метода ICP

2.1. Описание алгоритма ICP

Реализация алгоритма ICP состоит в математическом и дальнейшем программном анализе входных данных и полученной совмещенной трехмерной модели [5].

Для алгоритма регистрации используются следующие представления геометрических данных:

наборы точек;

- 1) наборы линейных сегментов (полилинии);
- 2) неявные кривые: $\vec{g}(x, y, z) = 0$;
- 3) параметрические кривые: $(x(u), y(u), z(u))$;
- 4) наборы треугольников (граненые поверхности);
- 5) неявные поверхности: $g(x, y, z) = 0$;
- 6) параметрические поверхности: $(x(u, v), y(u, v), z(u, v))$.

В описании алгоритма фигура «данных» P перемещается (регистрируется, позиционируется) так, чтобы наилучшим образом соответствовать фигуре «модели» X . Данные и форма модели могут быть представлены в любой из допустимых форм. В данном случае форма данных должна быть разложена на набор точек. Если форма данных не находится в форме набора точек, то действовать надо следующим образом: точки, которые будут использоваться для наборов

треугольников и линий, будут являться вершинами и конечными точками соответственно; если форма данных имеет форму поверхности или кривую, то используются вершины и конечные точки приближения «треугольник / линия». Количество точек в форме данных будет обозначено, как N_p .

Пусть N_j – количество точек, отрезков или треугольников, входящих в форму модели. Будет обозначена метрика расстояния d между отдельной точкой данных p и формой модели X (3).

$$d(\vec{p}, X) = \min_{\vec{x} \in X} \|\vec{x} - \vec{p}\| \quad (3)$$

Ближайшая точка в X , которая дает минимальное расстояние \vec{y} , обозначается $d(\vec{p}, \vec{y}) = d(\vec{p}, X)$, где $\vec{y} \in X$. Вычисление ближайшей точки – это $O(N_x)$, наихудший случай с ожидаемыми затратами $\log(N_x)$. Когда вычисление ближайшей точки (от \vec{p} до X) выполняется для каждой точки в P , этот процесс является наихудшим случаем $O(N_p N_x)$. Пусть Y обозначает результирующий набор ближайших точек, и пусть C – оператор ближайшей точки:

$$Y = C(P, X) \quad (4)$$

Учитывая результирующий соответствующий набор точек Y , регистрация методом наименьших квадратов вычисляется по формуле:

$$(\vec{q}, d) = Q(P, Y) \quad (5)$$

Далее положения набора точек формы данных обновляются с помощью $P = \vec{q}(P)$.

2.2. Математическая формулировка алгоритма ICP

Задан набор точек P с N_p точками $\{\vec{p}_l\}$ из формы данных и формы модели X (с поддерживающими геометрическими примитивами: точками, линиями или треугольниками).

Итерация инициализируется установкой $P_0 = P$, $\vec{q}_0 = [1, 0, 0, 0, 0, 0, 0]^T$ и $k=0$. Векторы регистрации определяются относительно исходного набора данных P_0 таким образом, чтобы окончательная регистрация представляла собой полное преобразование. Шаги 1, 2, 3 и 4 применяются до сходимости в пределах допуска τ . Вычислительная стоимость каждой операции указана в скобках [7].

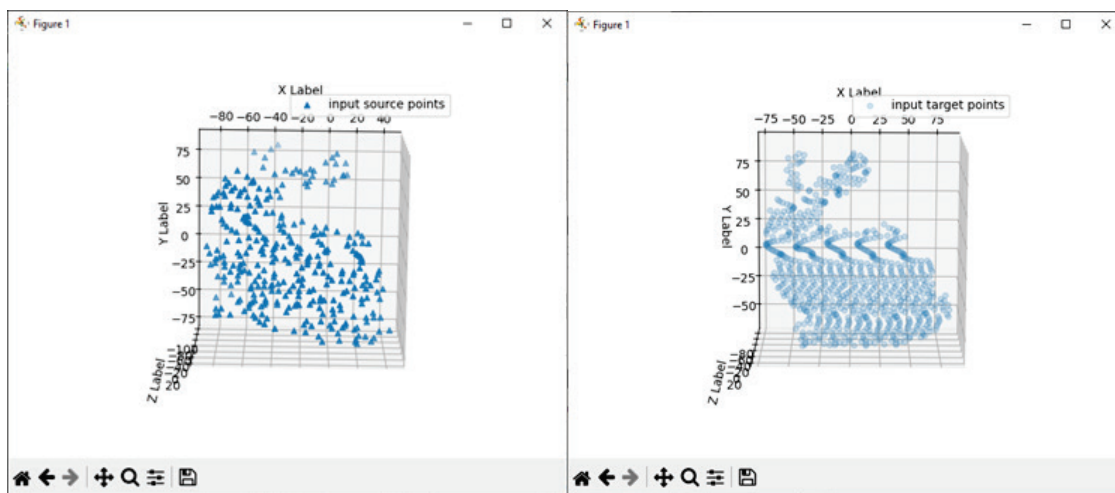
1. Вычисление ближайшей точки: $Y_k = C(P_k, X)$ (вычислительная стоимость: $O(N_p N_x)$ в худшем случае, $O(N_p \log N_x)$ (среднее значение)).
2. Вычисление регистрации: $(\vec{q}_k, d_k) = Q(P_0, Y_0)$ (стоимость: $O(N_p)$).
3. Применение регистрации: $P_{k+1} = \vec{q}_k(P_0)$ (стоимость: $O(N_p)$).
4. Завершение итерации, когда изменение среднеквадратичной ошибки будет меньше порога $\tau > 0$, задающего желаемую точность регистрации: $d_k - d_{k+1} < \tau$.

Если требуется безразмерный порог, то можно заменить τ на $\tau \sqrt{\text{tr}(\sum x)}$, где квадратный корень из следа ковариации формы модели указывает приблизительный размер формы модели.

3. Программная реализация алгоритма ICP

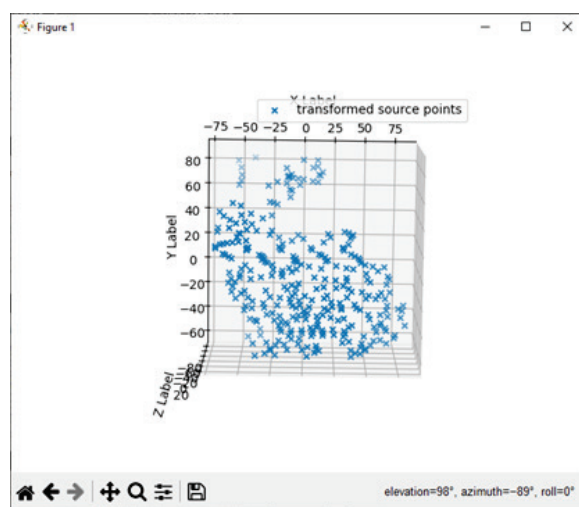
Программная реализация данного алгоритма выполнена на языке Python [8]. Вывод облаков точек представлен на рисунке 4.

Полученное изображение совмещенных облаков точек представлено на рисунке 5.



а

б



в

Рисунок 4 – Облака точек: а – входное облако точек источника; б – входное целевое облако точек; в – трансформированное облако точек источника

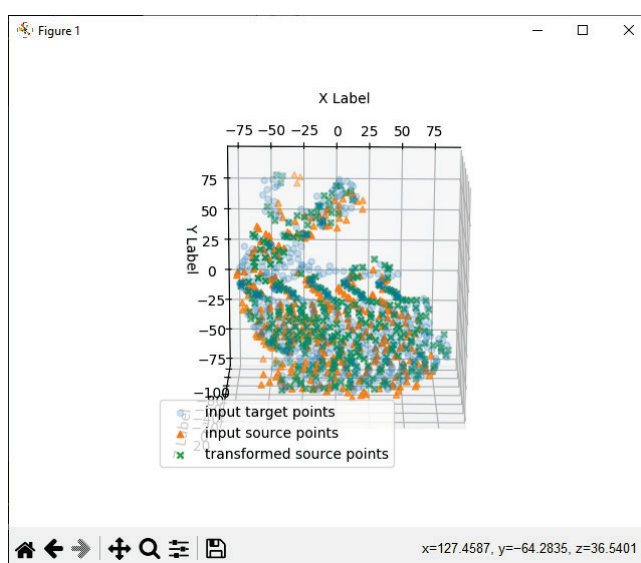


Рисунок 5– Полученное изображение совмещенных облаков точек

3.1. Проблемы применения алгоритма ICP

Основная проблема данного метода – совмещение облаков, обладающих лишь частичным перекрытием. Примером может являться съемка объектов с различных ракурсов. При этом на двух соседних по времени регистрации кадрах и, соответственно, облаках точек, как правило, присутствует часть точек, не имеющих соответствующих на другом изображении. Применение алгоритма ICP в данном случае дает квазиоптимальное решение, однако, с точки зрения визуальной оценки, результат может быть достаточно далек от требуемого и не соответствовать потребностям решаемой задачи.

Стандартный вариант алгоритма ICP дает плохие результаты при совмещении изображений, содержащих однородные поверхности (стол, пол, плоские участки земной поверхности без выраженных особенностей и т.д.). На таких поверхностях трудно находить ключевые точки и устанавливать между ними соответствия. В ряде работ [9] для установления соответствий предлагается использование детектора ключевых точек FAST и алгоритма оптического потока в схеме со стереопарой позволяет находить ключевые точки на соседних кадрах плоских изображений и устанавливать между ними соответствие в двумерном пространстве. Установленные соответствия между точками на плоских изображениях позволяют найти соответствия между ключевыми точками точек и в трехмерном пространстве.

В качестве способа устранения данного недостатка в последующих работах предполагается использование алгоритма установления однозначного соответствия между отдельными точками, грубое совмещение облаков по ним, а затем – досовмещение с применением модифицированной версии ICP.

Для формирования множества $G = \left\{ M_j, M'_j \right\}_{j=1}^k$ соответствующих точек разработан

алгоритм, основанный на использовании сигнатур спин-изображений [10, 11]. Общая схема предлагаемого метода такова. В каждой точке M_{i_0} одного и второго облаков точек методом наименьших квадратов находятся оценки \hat{a}, \hat{b} коэффициентов a, b уравнения плоскости $z = ax + by$, проходящей через данную точку. Оптимальные оценки \hat{a}, \hat{b} находятся в результате минимизации уклонения искомой плоскости от некоторого множества точек $\{M_i\}_{i=1}^k$ из окрестности заданного радиуса r точки M_{i_0} . Указанные оценки \hat{a}, \hat{b} необходимы для формирования в каждой точке вектора нормали $\mathbf{n} = (\hat{a}, \hat{b}, -1)$. Все векторы нормалей затем нормируются.

Каждой точке M_{i_0} из одного и второго облаков точек ставится в соответствие k -мерный вектор $\left\{ (\gamma_j, \alpha_j, \beta_j) \right\}_{j=1}^k$, элементами которого являются тройки чисел $(\gamma_j, \alpha_j, \beta_j)$ – дескрипторы точки M_{i_0} . Здесь: γ_j – угол между вектором нормали $\mathbf{n}_{i_0}^0$ текущей точки M_{i_0} и вектором \mathbf{n}_j^0 $j \in \{1, 2, \dots, k\}$ точки M_j из окрестности точки M_{i_0} ; β_j – алгебраическая проекция на вектор $\mathbf{n}_{i_0}^0$ вектора $\overline{M_{i_0}M_j}$, то есть $\beta_j = \text{Pr}_{\mathbf{n}_{i_0}^0} \overline{M_{i_0}M_j} = (\overline{M_{i_0}M_j}, \mathbf{n}_{i_0}^0)$; $\alpha_j = \sqrt{|\overline{M_{i_0}M_j}|^2 - \beta_j^2}$.

Сравнение дескрипторов точек из одного и второго облаков точек производится по следующему алгоритму. Пусть $\left\{ (\gamma_j^{(1)}, \alpha_j^{(1)}, \beta_j^{(1)}) \right\}_{j=1}^r$ – множество дескрипторов фиксированной точки $M_{i_0}^{(1)}$ из первого облака, а $\left\{ (\gamma_j^{(2)}, \alpha_j^{(2)}, \beta_j^{(2)}) \right\}_{j=1}^r$ – множество дескрипторов некоторой произвольной точки $M_i^{(2)}$ из второго облака, проверяемой на соответствие точке $M_{i_0}^{(1)}$. Для каждой точки $M_i^{(2)}$ из второго облака сначала проверяется условие близости по каждому

параметру, а именно в цикле по j от 1 до r проверяется выполнение условий: $|\gamma_j^{(1)} - \gamma_j^{(2)}| < \varepsilon_1$, $|\alpha_j^{(1)} - \alpha_j^{(2)}| < \varepsilon_2$, $|\beta_j^{(1)} - \beta_j^{(2)}| < \varepsilon_3$. Здесь $\varepsilon_1, \varepsilon_2, \varepsilon_3$ – заранее заданные числа. Если хотя бы для одного значения индекса j хотя бы одно из трех условий не выполнено, то осуществляется переход к новой точке второго облака. В противном случае находятся суммы

$$\sum_{\gamma} (M_i^{(2)}) = \sum_{j=1}^r |\gamma_{i_0j}^{(1)} - \gamma_{ij}^{(2)}|, \quad \sum_{\alpha} (M_i^{(2)}) = \sum_{j=1}^r |\alpha_{i_0j}^{(1)} - \alpha_{ij}^{(2)}|, \quad \sum_{\beta} (M_i^{(2)}) = \sum_{j=1}^r |\beta_{i_0j}^{(1)} - \beta_{ij}^{(2)}|, \quad i \in I. \quad (6)$$

Для каждой суммы в составе (6) находятся точки $M_s^{(2)}$, $s = 1, 2, 3$, в каждой из которых достигается минимум соответствующей суммы из (6)

$$M_{i_1}^{(2)} = \arg \min_{M_i^{(2)}} \sum_{\gamma} (M_i^{(2)}), \quad M_{i_2}^{(2)} = \arg \min_{M_i^{(2)}} \sum_{\alpha} (M_i^{(2)}), \quad M_{i_3}^{(2)} = \arg \min_{M_i^{(2)}} \sum_{\beta} (M_i^{(2)}) \quad (7)$$

Если $i_1 = i_2 = i_3 = i^*$, то точка $M_{i^*}^{(2)}$ из второго облака идентифицируется как соответствующая точке $M_{i_0}^{(1)}$, иначе – несоответствующей.

На рисунке 6 приведены исходные модельные облака точек, представляющие собой две седловые поверхности, разнесенные в пространстве на заранее заданный вектор сдвига.

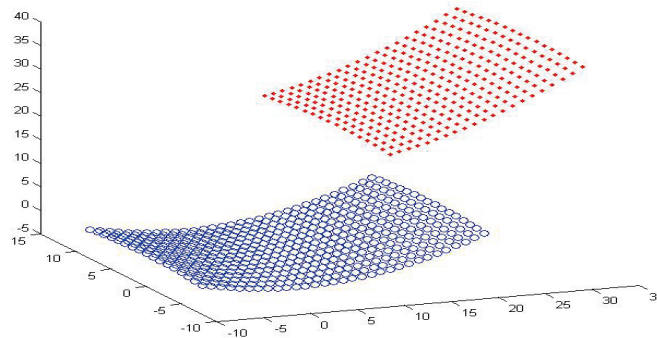


Рисунок 6– Исходные облака точек

На рисунке 7 показаны соответствия, найденные между точками исходных поверхностей.

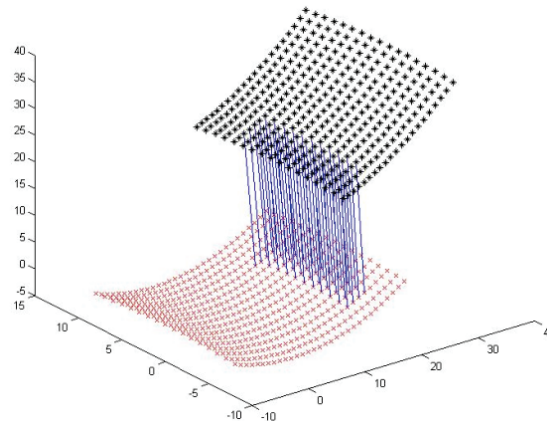


Рисунок 7– Найденные соответствия

Установленные таким образом соответствия являются основой для вычисления коэффициентов трехмерного проективного преобразования [9] и совмещения облаков точек на его основе. После этого для доуточнения полученных результатов, выполняется запуск алгоритма ИСР.

4. Экспериментальные исследования

В ходе экспериментов над модельным облаком точек «Stanford Bunny point cloud» было установлено, что средняя ошибка совмещения по всему облаку точек уменьшается на 3,2 – 4,4 %, а время выполнения меньше классического алгоритма ICP на 6-8%. Зависимость сокращения средней ошибки от степени перекрытия совмещаемых облаков при одинаковых входных искажениях показана на рисунке 8.



Рисунок 8 – Зависимость сокращения средней ошибки от степени перекрытия совмещаемых облаков

5. Заключение

Рассмотренный в статье способ совмещения облаков точек на основе алгоритма ICP является достаточно надежным и дающим устойчивые результаты вне зависимости от качества совмещаемых облаков. Однако, применение предварительного совмещения на основе трехмерных дескрипторов способно значительно повысить точность получаемого совмещения в случае лишь частичного перекрытия совмещаемых облаков.

6. Список источников

- [1] Point-set registration [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Point-set_registration (дата обращения: 08.12.2022).
- [2] Iterative closest point [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Iterative_closest_point (дата обращения: 01.12.2022).
- [3] Kabsch W. A solution for the best rotation to relate two sets of vectors, Acta Crystallographic a Section A, vol. 32, no. 5, 1976. – Pp. 922-923.
- [4] Vicente Morell-Gimenez, Marcelo Saval-Calvo, Jorge Azorin-Lopez, Jose Garcia-Rodriguez, Miguel Cazorla, Sergio Orts-Escolano and Andres Fuster-Guillo A Comparative Study of Registration Methods for RGB-D Video of Static Scenes// Sensors. 2014, 14, 8547-8576. DOI: 10.3390/s140508547 (дата обращения: 09.12.2022).
- [5] L. Maier-Hein, T. R. dos Santos, A. M. Franz, H.-P. Meinzer. Iterative Closest Point Algorithm in the Presence of Anisotropic Noise// German Cancer Research Center, Div. of Medical and Biological Informatics. [Электронный ресурс]. URL: https://ceur-ws.org/Vol-574/bvm2010_47.pdf (дата обращения: 12.12.2022).
- [6] Vladislav Golyanik, Bertram Taetz and Didier Stricker. Joint pre-alignment and robust rigid point set registration// University of Kaiserslautern, Germany. German Research Center for Artificial Intelligence (DFKI), Germany. 2016. DOI: 10.1109/ICIP.2016.7533212.

- [7] Paul J. Besl, Neil D. McKay. A Method for Registration of 3-D Shapes// *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, IEEE Computer Society, 1992.
- [8] Iterative Closest Point (ICP) Algorithm in Python. [Электронный ресурс]. URL: https://github.com/iitaakash/icp_python (дата обращения: 21.04.2023).
- [9] Новиков А.И. Алгоритмы автоматической идентификации объектов на разнородных изображениях и совмещения изображений / А.И. Новиков, А.И. Ефимов, Д.А. Колчаев // Сборник трудов IV международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2018) - Самара: Новая техника, 2018. - С.650-663.
- [10] Johnson, A. Surface Registration by Matching Oriented Points// A. Johnson, M. Hebert/ *Proc. 3 DIM*. 1997. - Pp. 145-152.
- [11] Черников И.С. Трехмерная реконструкция и распознавание по наборам дальнометрических данных / И.С. Черников, А.А. Крыловецкий// *Компьютерное моделирование 2009: труды международной научно-технической конференции*. – Санкт-Петербург, 2009. - С. 284-287.