

УДК 519.7

АЛГОРИТМ РЕГИСТРАЦИИ НЕКОНГРУЭНТНЫХ ОБЛАКОВ ТОЧЕК

© А. Ю. Маковецкий, С. М. Воронин, В. И. Кобер, А. В. Воронин

*Челябинский государственный университет,
454001, г. Челябинск, ул. Братьев Кашириных, 129
E-mail: artemmac@csu.ru*

Алгоритмы регистрации облаков точек в трёхмерном пространстве вычисляют ортогональное преобразование, которое максимизирует согласованное перекрытие двух облаков точек. Наиболее распространённым методом регистрации с использованием исключительно геометрических характеристик является итерационный алгоритм ближайших точек (ICP). Недостаток классических вариантов ICP — зависимость их от начального расположения облаков точек. Алгоритмы грубой регистрации используются для нахождения подходящей начальной регистрации двух облаков. Предлагается новый алгоритм выделения общих частей и грубой регистрации облаков точек.

Ключевые слова: облака точек, трёхмерное пространство, грубая регистрация, дескриптор, ортогональное преобразование.

DOI: 10.15372/AUT20220503

Введение. Регистрация облаков точек в трёхмерном пространстве — распространённый метод обработки данных, который используется в компьютерном зрении и реконструкции виртуального мира. Задачей регистрации облаков точек является поиск ортогонального преобразования, помещающего два облака точек в общую систему координат. Итерационный алгоритм ближайших точек (ICP, Iterative Closest Points) [1–11] является наиболее известным методом выравнивания с использованием геометрических характеристик облаков точек. Недостатком классических вариантов ICP, таких как точка в точку [1] и точка в плоскость [2], является зависимость от начального положения облаков точек. Если поворот, который может выровнять два облака точек, велик, алгоритм ICP часто попадает в локальный минимум.

Грубая регистрация облаков точек используется для поиска подходящего начального выравнивания двух облаков. В частности, известны методы грубой регистрации на основе признаков. Важным элементом таких методов является дескриптор точки. Часто используются трёхмерные дескрипторы FPFH (Fast Point Feature Histograms) [12, 13] и SHOT (Signature of Histograms of Orientations) [14]. Алгоритм SHOT формирует некоторый дескриптор — вектор из 352 компонент для точки в облаке и позволяет сравнивать дескрипторы точек в пространстве \mathbb{R}^{352} вместо \mathbb{R}^3 . Вычисление дескриптора SHOT основано на использовании локальной системы координат для точки в облаке.

Выбор направления базисного вектора системы координат важен, поскольку стандартные алгоритмы разложения матрицы на основе собственных векторов и собственных значений формируют собственные векторы с непредсказуемыми направлениями. В [14] описан алгоритм выбора направления базисных векторов в системе координат.

Сложность регистрации зависит от типов облаков точек. Синтетические 3D-модели широко используются для проведения вычислительных экспериментов. Более сложный случай — регистрация 3D-моделей с шумом. В этом случае обычно используется аддитивный гауссовский или импульсный шум.

Следующим этапом является задача регистрации неконгруэнтных облаков точек, полученных из одной и той же исходной 3D-модели (усечённых облаков точек). Стандартный

алгоритм ИСР обычно не может найти правильное решение в этом случае. Зашумлённые усечённые облака точек являются наиболее сложным типом данных для регистрации с использованием синтетической модели. Этот тип наиболее близок к регистрации реальных 3D-кадров.

В данной работе описан алгоритм выделения общих частей неконгруэнтных облаков точек и их грубого выравнивания с использованием алгоритма SHOT для соответствия между двумя облаками точек, которое позволяет рассматривать их окрестности. Окрестность определяет локальный векторный базис. Два локальных векторных базиса позволяют вычислить ортогональное преобразование.

Поскольку алгоритмы собственного разложения матриц предоставляют собственные векторы с непредсказуемыми направлениями, необходимо согласовать направления векторов двух базисов. В представленной работе используется алгоритм «предсказания направления». Множество соответствующих окрестностей определяет множество локальных ортогональных преобразований.

Предполагается, что распределение матриц преобразования имеет высокую плотность в окрестности истинной матрицы преобразования. Здесь используем алгоритм кластеризации DBSCAN [15] для определения кластера с самой высокой плотностью. Матрицы, не принадлежащие выбранному кластеру, исключаются из рассматриваемого набора. Таким образом, получается редуцированный набор соответствующих точек в двух облаках.

Предлагаемый алгоритм определяет общие части неконгруэнтных облаков точек. После этого усредняем выбранные матрицы и проецируем результат на группу $SO(3)$. Результирующая матрица ортогонального преобразования является грубой матрицей регистрации. Результаты эксперимента иллюстрируют качество предлагаемого метода.

Задача данной работы — создание нового эффективного алгоритма регистрации частично перекрывающихся облаков точек в трёхмерном пространстве, который позволит также оценивать общие части облаков точек.

Алгоритм грубой регистрации. Пусть $P = \{p^1, \dots, p^{sp}\}$ — первое облако точек, а $Q = \{q^1, \dots, q^{sq}\}$ — второе облако точек в \mathbb{R}^3 . Обозначим через p^k точку из облака P , $k = 1, \dots, sp$. Рассмотрим радиус r окрестности точки p^k в P и подмножество $N_r(p^k)$ точек, принадлежащих этой окрестности. Ковариационная матрица Σ_k определяется следующим образом [14]:

$$\Sigma_k = \sum_{p^i \in N_r(p^k)} (r - d(p^i))(p^i - c_k)(p^i - c_k)^\top / \sum_{p^i \in N_r(p^k)} (r - d(p^i)),$$

где c_k — центр масс $N_r(p^k)$, $d(p^i) = |p^i - c_k|$.

Рассмотрим собственное разложение матрицы Σ_k :

$$\Sigma_k = \mathfrak{R}_k \begin{pmatrix} \lambda_{k1} & 0 & 0 \\ 0 & \lambda_{k2} & 0 \\ 0 & 0 & \lambda_{k3} \end{pmatrix} \mathfrak{R}_k^\top,$$

где $\lambda_{k1}, \lambda_{k2}, \lambda_{k3}$ — собственные значения Σ_k в порядке возрастания; \mathfrak{R}_k — ортогональная матрица соответствующих собственных векторов; \top — транспонирование. Отметим, что алгоритмы собственного разложения вычисляют собственные векторы с непредсказуемыми направлениями. В [14] описан алгоритм выбора направлений собственных векторов. Обозначим через $M(k)$ подмножество точек в окрестности $N_r(p^k)$ такое, что

$$M(k) = \left\{ i: |m - i| \leq k, \quad m = \underset{j}{\operatorname{argmedian}} d(p^j) \right\}. \quad (1)$$

Рассмотрим следующие подмножества:

$$\begin{aligned}
 S_x^+ &= \{ i: d(p^i) \leq r \text{ и } \langle (p^i - c_k), x^+ \rangle \geq 0 \}; \\
 S_x^- &= \{ i: d(p^i) \leq r \text{ и } \langle (p^i - c_k), x^- \rangle < 0 \}; \\
 \tilde{S}_x^+ &= \{ i: d(p^i) \in M(k) \text{ и } \langle (p^i - c_k), x^+ \rangle \geq 0 \}; \\
 \tilde{S}_x^- &= \{ i: d(p^i) \in M(k) \text{ и } \langle (p^i - c_k), x^- \rangle \geq 0 \},
 \end{aligned} \tag{2}$$

где x^+ — третий собственный вектор с начальным направлением, x^- — третий собственный вектор с обратным направлением,

$$x = \begin{cases} x^+, & |S_x^+| > |S_x^-|; \\ x^-, & |S_x^+| < |S_x^-|; \\ x^+, & |S_x^+| = |S_x^-| \text{ и } |\tilde{S}_x^+| > |\tilde{S}_x^-|; \\ x^-, & |S_x^+| = |S_x^-| \text{ и } |\tilde{S}_x^+| < |\tilde{S}_x^-|. \end{cases} \tag{3}$$

Этот же метод используется для первого собственного вектора z , вектор y получается как $z \times x$. Переориентированный локальный векторный базис используется для вычисления дескриптора SHOT точки p^k . В частности, вектор z считается нормальным в точке p^k .

Рассмотрим трёхмерный шар с центром в точке p^k , содержащий все точки окрестности p^k . Разделим шар на 32 подмножества (32 — это число пар, полученное из двух делений по углу места, восьми делений по азимуту и двух делений по радиусу). Обозначим через n_q единичную нормаль к точке q и вычислим $\cos(\alpha_q) = \langle n_q, z \rangle$. В сочетании с тем фактом, что правильное количество бинов для внутренних гистограмм (для значений косинуса) равно 11, получаем общую длину гистограммы 352, т. е. алгоритм SHOT генерирует для каждой точки облака точек P дескриптор, который есть вектор в \mathbb{R}^{352} .

Далее опишем предлагаемый новый алгоритм. Ищем ближайшие точки в облаке Q для каждой точки в P , используя дескрипторы k,d-дерева. После этого шага устанавливается фиксированное соответствие между облаками P и Q . Пусть p^i, q^i — соответствующие друг другу точки, рассмотрим k -окрестности точек p^i и q^i . Вычисляем ковариационные матрицы для этих двух окрестностей и выполняем собственные разложения. Результирующие матрицы собственных векторов обозначаются как \mathfrak{R}_{p^i} и \mathfrak{R}_{q^i} . Рассмотрим векторы главных осей x_{p^i} и x_{q^i} . Целью алгоритма предиктора направления является получение согласованных направлений для векторов x_{p^i} и x_{q^i} . Отметим, что описанный алгоритм и алгоритм (1)–(3) различны.

Предположим, что направления собственных векторов в облаке Q фиксированы. Необходимо выбрать сориентированные направления соответствующих векторов в P и Q . Будем считать x_{q^i} и c_q вектором и центром масс окрестности в Q соответственно. Отметим, что ориентированная ось x локального блока определяется вектором x_{q^i} и точкой c_q .

Рассмотрим ортогональные проекции всех точек облака Q на ось x . Обозначим через q_{ent} и q_{ext} первую и последнюю проекции на ось x . Поделим равномерно отрезок $[q_{\text{ent}}; q_{\text{ext}}]$ точками

$$\{q_0 = q_{\text{ent}}; q_1; \dots; q_m = q_{\text{ext}}\},$$

где m — количество подсегментов. Рассмотрим каждый подотрезок $[q_{j-1}; q_j]$ и обозначим через q'_i проекцию точки q_i на Q , если она принадлежит этому подотрезку. Обозначим через h_j центры подотрезков $[q_{j-1}; q_j]$, $j = 1, \dots, m$. Связываем значение d_j с подсегментом следующим образом:

$$d_j = \sum_{q'_i \in [q_{j-1}; q_j]} (|h_j - q_j| - |h_j - q'_i|) |q_j - q'_i|^2 / \sum_{q'_i \in [q_{j-1}; q_j]} (|h_j - q_j| - |h_j - q'_i|).$$

Рассмотрим вектор $ds_x^q = (d_1, \dots, d_m)$. Вычислим аналогичные дескрипторы $ds_x^p(+)$ и $ds_x^p(-)$. Вектор $ds_x^p(-)$ является обратной версией вектора $ds_x^p(+)$. Сравним вектор ds_x^q с $ds_x^p(+)$ и с $ds_x^p(-)$, используя норму L_2 . Наименьшая норма определяет направление третьего собственного вектора в локальном базисе в P .

Точно так же мы вычисляем дескрипторы ds_z^q в Q и ds_z^p в P и устанавливаем ориентацию базисного вектора z_{p^i} . Вектор y_{p^i} является векторным произведением $z_{p^i} \times x_{p^i}$.

После использования алгоритма предсказателя направления получаем (переориентированные) локальный базис $(x_{p^i}, y_{p^i}, z_{p^i})$ в P и локальный базис $(x_{q^i}, y_{q^i}, z_{q^i})$ в Q . Этот локальный базис позволяет вычислить локальное ортогональное преобразование, отображающее базис $(x_{p^i}, y_{p^i}, z_{p^i})$ в базис $(x_{q^i}, y_{q^i}, z_{q^i})$ по формулам

$$R_i = \mathfrak{R}_{q^i} \mathfrak{R}_{p^i}^\top, \quad T_i = c_{q^i} - R_i c_{p^i},$$

где \mathfrak{R}_{p^i} — матрица размера 3×3 , содержащая векторы-столбцы $(x_{p^i}, y_{p^i}, z_{p^i})$; $\mathfrak{R}_{q^i} = (x_{q^i}, y_{q^i}, z_{q^i})$; c_{p^i} и c_{q^i} — центры масс окрестностей в Q и P соответственно.

Итак, у нас есть n локальных ортогональных преобразований $(R_1, T_1), \dots, (R_n, T_n)$ (n — количество соответствий после SHOT). Предположим, что область в \mathbb{R}^6 (параметризирующая преобразования) с наибольшей плотностью хорошо аппроксимирует истинное ортогональное преобразование между облаками точек P и Q . Чтобы кластеризовать набор $\{(R_1, T_1), \dots, (R_n, T_n)\}$, используем алгоритм DBSCAN [15]. Трансформации из наибольшего кластера $\{(R_{cl_1}, T_{cl_1}), \dots, (R_{cl_t}, T_{cl_t})\}$, где t — число элементов в этом кластере, используются для оценки истинного преобразования:

$$R_{\text{avg}} = \frac{1}{t} \sum_{i=1}^t R_{cl_i}, \quad T_{\text{avg}} = \frac{1}{t} \sum_{i=1}^t T_{cl_i}.$$

Точки в облаках P и Q , соответствующие выбранному кластеру, являются оценкой общих частей P и Q .

Поскольку R_{avg} является матрицей аффинного преобразования, то проецируем R_{avg} на группу $SO(3)$ [6]. Рассмотрим сингулярное разложение UDV^\top матрицы R_{avg} . Проекция R группы R_{avg} на группу $SO(3)$ есть матрица USV^\top , где S задаётся выражением

$$S = \begin{cases} I, & \text{если } \det(U) \det(V) = 1; \\ \text{diag}(1, 1, \dots, 11, -1), & \text{если } \det(U) \det(V) = -1. \end{cases}$$

Вычислительные эксперименты. Покажем качество предложенного алгоритма грубой регистрации и выделения общих частей. Используем следующие четыре облака точек: Bunny и Armadillo, Airplane из базы данных ModelNet40 [16], Human face из базы данных Bosphorus [17]. Облака Bunny, Armadillo и Airplane состоят из 1024 точек, облако Human face — из 1724 точек. Все точки лежат в центрированной единичной сфере.

Применим жёсткое геометрическое преобразование к облаку точек P , определяемому ортогональной матрицей R_{true} и вектором переноса T_{true} . Облако точек Q получается из облака точек P как

$$Q = R_{\text{true}}P + T_{\text{true}}.$$

Информация о матрице R_{true} и векторе переноса T_{true} содержится в матрице M_{true} в однородных координатах.

Статистические эксперименты организованы следующим образом. Зафиксируем некоторое значение угла поворота. Рассмотрим случайный вектор направления. Вектор определяет линию, содержащую центр системы координат. Предположим, что линия проходит через ось вращения. В качестве компонент вектора переноса используем случайные величины на отрезке $[0; 1]$.

Обозначим матрицу истинного геометрического преобразования через M_{true} , которая применяется к активному облаку P . Предлагаемый алгоритм применяется к облакам P и Q . Считаем, что предложенный алгоритм сходится к истинным данным, если восстановленная матрица M_{est} аппроксимирует матрицу M_{true} с заданной точностью. Для статистически правильных результатов используем 100 испытаний для каждого фиксированного угла поворота. Угол поворота изменяется от 0 до 180 градусов с шагом 20 градусов.

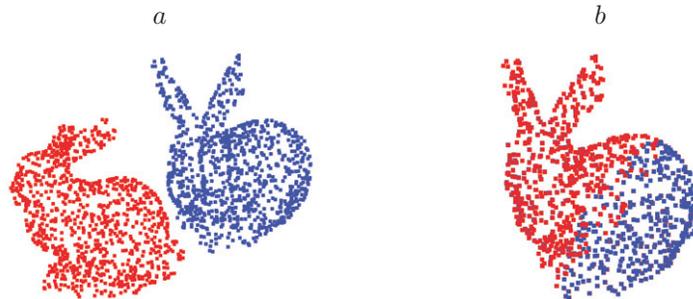


Рис. 1. Регистрация облаков точек: a — облако точек P (красное), облако точек Q (синее); b — результат регистрации

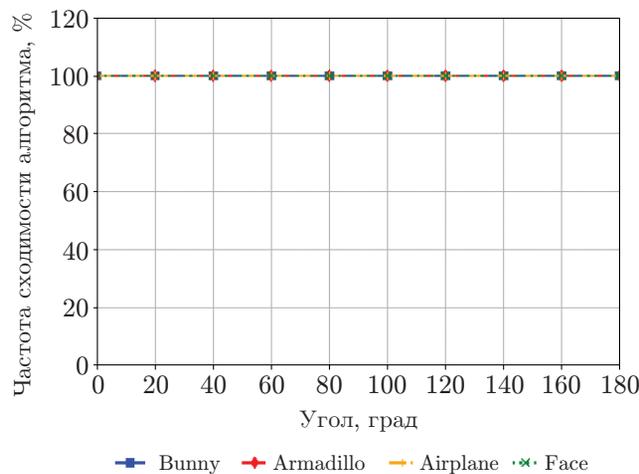


Рис. 2. Частота сходимости алгоритма для четырёх типов облаков точек

Синтетические 3D-модели. На рис. 1, *a* показаны облака P (красное) и Q (синее). Облако Q получается из P преобразованием M_{true} . На рис. 1, *b* показаны облака $M_{\text{est}}P$ (красное) и Q (синее), где преобразование M_{est} — результат регистрации.

В статистических экспериментах для конгруэнтных облаков точек использовался порог $\|M_{\text{est}} - M_{\text{true}}\| < 0,2$ в качестве критерия сходимости к истинному преобразованию.

На рис. 2 показана частота сходимости алгоритма для Bunny (синяя линия), Armadillo (красная линия), Airplane (оранжевая линия), Human face (зелёная линия).

Зашумлённые синтетические 3D-модели. Использовался аддитивный белый гауссов шум. Добавляем к каждой координате точки гауссову случайную величину с нулевым средним значением и стандартным отклонением $s = 0,02$. Облака точек P и Q зашумлены независимо друг от друга.

На рис. 3, *a* даны облака P (красное) и Q (синее). Облако Q получается из P преобразованием M_{true} . На рис. 3, *b* показаны облака $M_{\text{est}}P$ (красное) и Q (синее), где преобразование M_{est} — результат регистрации.

На рис. 4 изображены общие части облаков P (красное) и Q (синее), серые точки не относятся к предполагаемым общим частям.

В статистических экспериментах для конгруэнтных облаков точек использовался порог $\|M_{\text{est}} - M_{\text{true}}\| < 0,6$ в качестве критерия сходимости к истинному преобразованию.

На рис. 5 показана частота сходимости алгоритма для Bunny (синяя линия), Armadillo (красная линия), Airplane (оранжевая линия), Human face (зелёная линия).

Усечённые 3D-модели. На рис. 6, *a* показаны облака P (красное) и Q (синее). Облако Q получается из P преобразованием M_{true} . На рис. 6, *b* показаны облака $M_{\text{est}}P$ (красное) и Q (синее), где преобразование M_{est} — результат регистрации. На рис. 7 даны общие части облаков P (красное) и Q (синее), серые точки не относятся к предполагаемым общим частям.

В статистических экспериментах для конгруэнтных облаков точек использовался порог $\|M_{\text{est}} - M_{\text{true}}\| < 0,6$ в качестве критерия сходимости к истинному преобразованию.

На рис. 8 продемонстрирована частота сходимости алгоритма для Bunny (синяя линия), Armadillo (красная линия), Airplane (оранжевая линия), Human face (зелёная линия).

Зашумлённые усечённые 3D-модели. Использовался аддитивный белый гауссов шум. Добавляем к каждой координате точки гауссову случайную величину с нулевым средним значением и стандартным отклонением $s = 0,02$. Облака точек P и Q зашумлены независимо друг от друга.

На рис. 9, *a* представлены облака P (красное) и Q (синее). Облако Q получается из P преобразованием M_{true} . На рис. 9, *b* показаны облака $M_{\text{est}}P$ (красное) и Q (синее), где преобразование M_{est} — результат регистрации.

На рис. 10 приведены общие части облаков P (красное) и Q (синее), серые точки не относятся к предполагаемым общим частям.

В статистических экспериментах для конгруэнтных облаков точек использовался порог $\|M_{\text{est}} - M_{\text{true}}\| < 0,6$ в качестве критерия сходимости к истинному преобразованию.

На рис. 11 показана частота сходимости алгоритма для Bunny (синяя линия), Armadillo (красная линия), Airplane (оранжевая линия), Human face (зелёная линия).

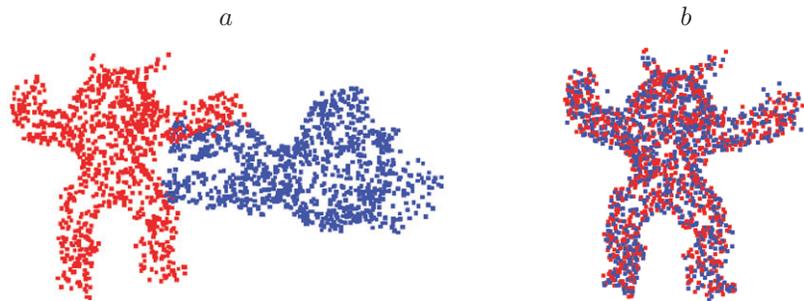


Рис. 3. Регистрация облаков точек: a — облако точек P (красное), облако точек Q (синее); b — результат регистрации

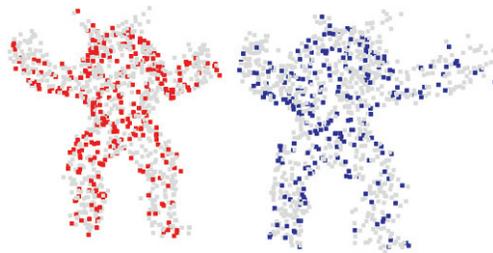


Рис. 4. Вычисленные общие части облака точек P (красного), облака точек Q (синего)

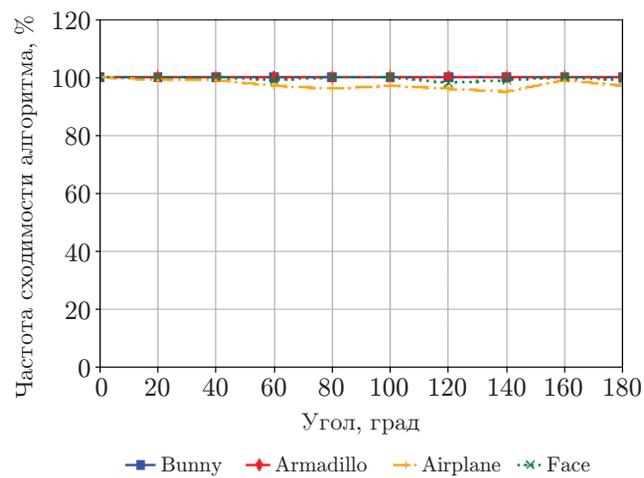


Рис. 5. Частота сходимости алгоритма для четырёх типов облаков точек

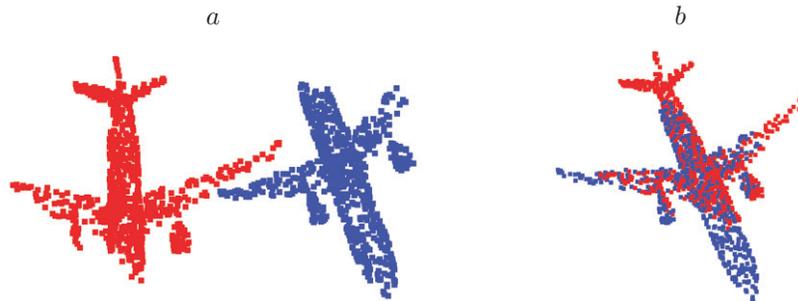


Рис. 6. Регистрация облаков точек: a — облако точек P (красное), облако точек Q (синее); b — результат регистрации

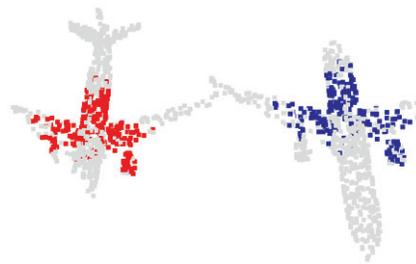


Рис. 7. Вычисленные общие части облака точек P (красного), облака точек Q (синего)

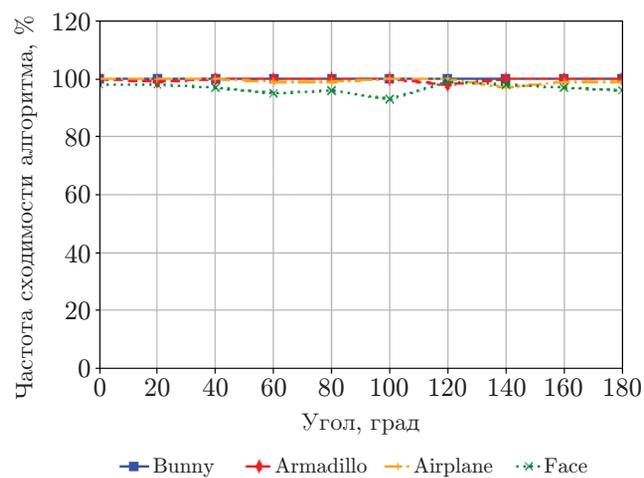


Рис. 8. Частота сходимости алгоритма для четырёх типов облаков точек

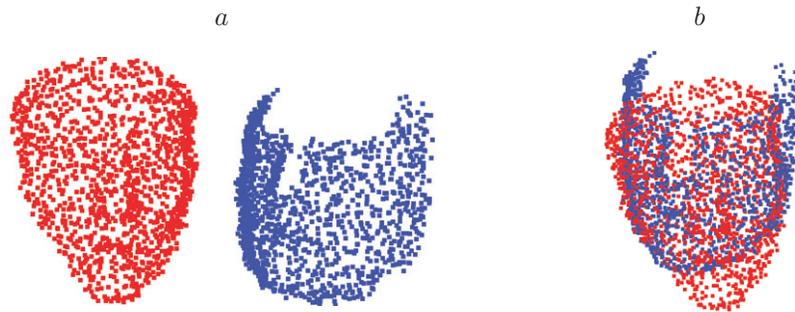


Рис. 9. Регистрация облаков точек: a — облако точек P (красное), облако точек Q (синее); b — результат регистрации

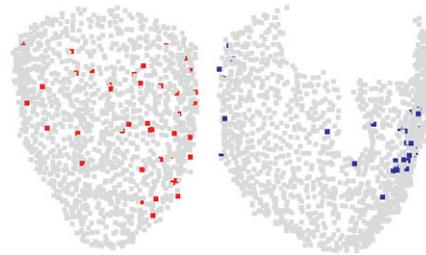


Рис. 10. Вычисленные общие части облака точек P (красного), облака точек Q (синего)

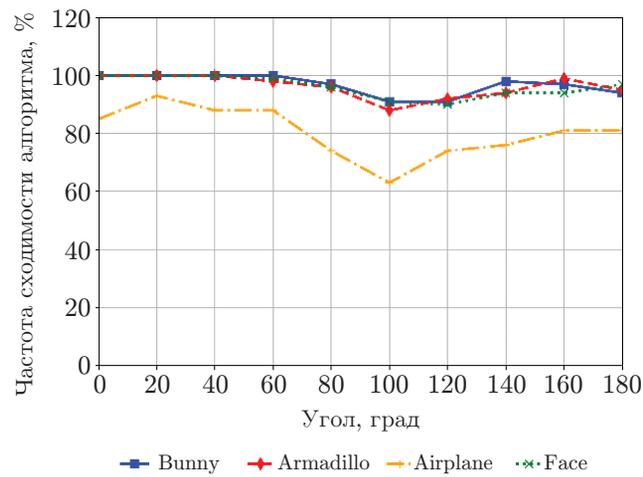


Рис. 11. Частота сходимости алгоритма для четырёх типов облаков точек

Заключение. В представленной работе описан новый алгоритм грубой ортогональной регистрации и выделения общих частей неконгруэнтных облаков точек. Алгоритм ищет соответствие между облаками точек и находит локальные ортогональные преобразования. Кроме того, в этом алгоритме используется новый метод определения направления собственных векторов. Точность работы предложенного алгоритма оценивается на различных классах облаков точек: конгруэнтных облаков точек и различных неконгруэнтных облаков точек. Результаты экспериментов показывают зависимость качества регистрации от точности определения общих частей.

Финансирование. Работа выполнена при поддержке Российского научного фонда (грант № 21-11-00095).

СПИСОК ЛИТЕРАТУРЫ

1. **Besl P., McKay N.** A method for registration of 3-D shapes // *IEEE Trans. Pattern Analysis and Machine Intell.* 1992. **2**. P. 239–256.
2. **Chen Y., Medioni G.** Object modeling by registration of multiple range images // *Image and Vis. Comput.* 1992. **10**. P. 145–155.
3. **Du S., Zheng N., Meng G., Yuan Z.** Affine registration of point sets using ICP and ICA // *IEEE Signal Process. Lett.* 2008. **15**. P. 689–692.
4. **Horn B.** Closed-form solution of absolute orientation using unit quaternions // *JOSA A.* 1987. **4**. P. 629–642.
5. **Horn B., Hilden H., Negahdaripour S.** Closed-form solution of absolute orientation using orthonormal matrices // *JOSA A.* 1988. **5**, Iss. 7. P. 1127–1135.
6. **Umeyama S.** Least-squares estimation of transformation parameters between two point patterns // *IEEE-TPAMI.* 1991. **4**, Iss. 13. P. 376–380.
7. **Makovetskii A., Voronin S., Kober V., Voronin A.** An efficient algorithm for non-rigid object registration // *Comp. Opt.* 2020. **44**, Iss. 1. P. 67–73.
8. **Ruchay A., Dorofeev K., Kalschikov V.** A novel switching bilateral filtering algorithm for depth map // *Comp. Opt.* 2019. **6**, Iss. 43. P. 1001–1007.
9. **Makovetskii A., Voronin S., Kober V., Voronin A.** A non-iterative method for approximation of the exact solution to the point-to-plane variational problem for orthogonal transformations // *Math. Methods in the Appl. Sci.* 2018. **41**. P. 9218–9230.
10. **Makovetskii A., Voronin S., Kober V., Voronin A.** A regularized point cloud registration approach for orthogonal transformations // *Journ. Global Optimization.* 2022. **83**, Iss. 2. P. 497–519.
11. **Makovetskii A., Voronin S., Kober V., Voronin A.** Point cloud registration based on multiparameter functional // *Mathematics.* 2021. **9**, N 20. 2589.
12. **Rusu R. B., Blodow N., Beetz M.** Fast point feature histograms (FPFH) for 3D registration // *Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA 2009).* Kobe, Japan, 12–17 May, 2009. P. 3212–3217.
13. **Mellado N., Aiger D., Mitra N. J.** SUPER 4PCS fast global pointcloud registration via smart indexing // *Eurograph. Symposium on Geometry Processing.* 2014. **5**, Iss.33. P. 205–215.
14. **Salti S., Tombari F., Di Stefano L.** SHOT: Unique signatures of histograms for surface and texture description // *Comput. Vis. Image Underst.* 2014. **125**. P. 251–264.
15. **Ester M., Kriegel H.-P., Sander J., Xu X.** A density-based algorithm for discovering clusters in large spatial databases with noise // *Proc. of the Second International Conf. on Knowledge Discovery and Data Mining (KDD 1996).* Portland, USA, 2–4 Aug., 1996. P. 226–231.

16. **Wu Z., Song S., Khosla A. et al.** 3D Shapenets: A deep representation for volumetric shapes // Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015). Boston, USA, 07–12 June, 2015. P. 1912–1920.
17. **Bosphorus 3D Face Database.** URL: <http://bosphorus.ee.boun.edu.tr/default.aspx> (дата обращения: 15.07.2022).

Поступила в редакцию 15.07.2022

После доработки 29.07.2022

Принята к публикации 15.08.2022
