

doi: 10.32620/oikit.2021.92.11

УДК 004.048

М.Д. Мирненко, Д.М. Крицький,  
О.К. Погудіна, О.С. Крицька

## Порівняльний аналіз методів пошуку найближчих точок на зображеннях об'єктів технічних систем

*Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ»*

**Предметом** вивчення є процес відображення побудови хмар точок об'єктів технічних систем за допомогою алгоритму найближчих точок. **Метою** є мінімізація критерію суміщення, при перетворенні множини точок хмари  $Y$  в множину точок хмари  $X$  у програмному продукті, що застосовує алгоритм ітеративних найближчих точок (iterative closest point – ICP). **Завдання:** провести аналіз властивостей вхідних зображень, які містять хмари точок; провести огляд алгоритмів виявлення та зіставлення ключових точок; реалізувати алгоритм порівняння хмар за допомогою алгоритму ICP; розглянути приклад роботи алгоритму оцінки наближених значень елементів взаємного орієнтування; реалізувати програмне забезпечення, що дозволяє порівняти файли, які містять хмари точок, та зробити висновок щодо можливості їх зіставлення. Використовуваними **методами** є: пошук точок за допомогою алгоритму ітеративних найближчих точок, алгоритм оцінки наближених значень елементів взаємного орієнтування, метод теорії алгоритмів щодо аналізу структур файлів stl (standard template library – стандартна бібліотека шаблонів) формату. Отримані такі **результати**. Обґрунтовано вибір алгоритму ICP для задачі реконструкції об'єкту технічних систем; розглянуто основні особливості роботи алгоритму ICP; наведено алгоритм зіставлення ключових точок, а також оптимізації, що дозволяє зменшити критерій суміщення при реконструкції тривимірних об'єктів технічних систем. **Висновки.** В ході дослідження було виявлено, що ітеративний алгоритм найближчих точок є більш детальним й точним при обчисленні міри близькості складність прорахування цим алгоритмом збільшується в рази. Тоді як алгоритм оцінки наближених значень елементів взаємного орієнтування не потребує інформації про наближену орієнтацію хмар точок, що спрощує роботу і зменшує час моделювання. Виявлено, що не всі файли можливі для порівняння. Тому реалізовано програмне забезпечення, яке дає висновок, щодо можливості зіставлення точок у пропонувананих двох файлах, які містять хмари точок у структурі stl формату. **Ключові слова:** пошук найближчих хмар точок, ітеративний алгоритм найближчих точок, алгоритм оцінки наближених значень елементів взаємного орієнтування.

### Вступ

На даний час використання додатків по обробці хмари точок є популярним напрямком. Велика кількість програм включають використання технології тривимірного сканування для оброблення заготовок на верстатах у машинобудівній промисловості. В даний час можна спостерігати прогрес технологій сканування та обробки в області виготовлення деталей на верстатах з ЧПУ. В інженерних додатках широко використовуються оптичні 3D-сканери. Результатом цього принципу сканування є хмара точок, яку можна описати як набір точок даних у якійсь системі координат, зазвичай визначається  $x$ ,  $y$ ,  $z$  координатами. При переміщенні пристрою сканування або зміни положення сканованого об'єкту, отримана хмара точок частково накладається. Основна мета – синхронізувати всі часткові скани разом, щоб сформувати єдиний об'єкт. Цей процес називається точковою реєстрацією. Процес реєстрації можна розділити на дві основні частини: грубу та точну відповідність. У грубій

відповідній частині ми знаходимо відповідні точки з детекторами та (або) дескрипторами. У цій статті розглядається точна відповідність вирівнювання двох хмар точок. В статті представляється алгоритм найближчої точки ітерації (ICP), який є одним із найпоширеніших алгоритмів, що використовуються на практиці. А також порівнюється з не менш популярним методом – алгоритмом оцінки наближених значень елементів взаємного орієнтування. Наведені алгоритми, а також їх результати, представлені у розділі нижче.

### 1 Ітеративний алгоритм найближчих точок

Алгоритм найближчих точок використовується для суміщення знімків одного і того ж об'єкта, отриманих з різних ракурсів, але мають спільні ділянки – області перекриття. При цьому передбачається, що в областях перекриття знаходяться пари найближчих точок, відстань між якими менше заданого порогу. Якщо в список таких пар потрапляють «помилкові» пари, які не належать області перекриття, це негативно позначається на коректності роботи алгоритму. Є дві хмари точок  $X$  і  $Y$ , які описують тривимірний об'єкт, отриманий з різних точок зору за допомогою 3D-сканера. Для перекладу будь-якої точки з хмари  $X$  в систему координат хмари  $Y$  використовується ортогональне геометричне перетворення [5]:

$$y_i = Rx_i + T, \quad (1.1)$$

де  $T$  – вектор переносу;  $R$  – матриця повороту;  $x$  – довільна точка з хмари точок  $X$ ;  $y$  – довільна точка з хмари точок  $Y$ .

Для обчислення відстані між точкою  $x^1 \in X$  і хмарою точок  $Y$  в першому варіанті алгоритму використовується метрика «точка-точка» (point-to-point distance) [5]:

Метрика (1.2) використовує дискретні моделі поверхонь.

$$E(R, T) = \frac{1}{N} \sum_{i=0}^N \|x_i - Ry_i - T\|, \quad N = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} W_{ij}. \quad (1.2)$$

В [3] пропонується метрика «точка-площина» (point-to-plane distance), використання якої передбачає, що в кожній точці  $X$  і  $Y$  відомі нормалі до поверхонь, заданих хмарами точок  $X$  і  $Y$  відповідно. Така інформація може бути задана спочатку, або обчислена через усереднення нормалей по інцидентним трикутникам при використанні триангуляційної кусочно-лінійної апроксимації поверхні. У цьому випадку буде мінімізуватися сума квадратів відстаней від  $x^1$  до площини  $P_i^2$ , перпендикулярної  $Y$  в точці  $y^1$  по всім парам найближчих точок  $(x^1, y^1)$ :

$$E = \sum_{i=1}^N H^2(x^1, P_i^2) \rightarrow \min, \quad (1.3)$$

де  $H(x, P)$  – евклідова відстань від точки  $x$  до площини  $P$ .

Загальна ітеративна схема алгоритму включає наступні етапи:

- 1) пошук пар найближчих точок  $(x^1, y^1)$ ,  $i = \overline{1, N}$  для поточного становища  $X$  і  $Y$ ;
- 2) пошук перетворення (параметрів зрушень і поворотів) хмари точок  $X$ , що зменшує помилку  $E$  (1.1) або (1.3) за допомогою методу найменших квадратів. Якщо зміна помилки  $E$  менше певного порогового значення, то робота алгоритму закінчується;

3) застосування знайденого на попередньому етапі перетворення до хмари точок  $X$ . Перехід на етап 1.

Етапи 1-3 повторюються до тих пір, поки зменшення помилки перевищує деяке порогове значення. Рішенням задачі є кінцеве положення хмари точок  $X$ .

Вплив «помилкових» пар найближчих точок при використанні метрики «точка-площина» менше, ніж при використанні (1.2). Кількість ітерацій алгоритму при використанні метрики «точка-точка» більше, так як в ній використовуються дискретні моделі поверхонь. Недоліком метрики «точка-площина» є те, що вона сильно залежить від початкового взаємного положення  $X$  і  $Y$  [4].

Основною перевагою алгоритму найближчих точок є його проста реалізація, до недоліків можна віднести сильну залежність від початкового наближення об'єктів, обчислювальну складність, пов'язану з пошуком всіх пар найближчих точок  $\{(x^1 y^1)\}^N$ .

Нехай  $N_1, N_2$  – кількість точок у вихідних хмарах точок  $X$  та  $Y$  відповідно, тоді при простій реалізації оцінка трудомісткості такого пошуку буде  $O(N_1 N_2)$  – квадратичною при  $N_1 \approx N_2$ . За допомогою використання більш складних структур даних – наприклад,  $k$ -d дерева [2] – пошук може бути здійснений за час  $O(N_1 \log N_2)$ . Таким чином, загальна кількість необхідних операцій для пошуку пар найближчих точок при  $m$  ітераціях становить  $O(m N_1 \log N_2)$ .

Велика кількість робіт [5] присвячена різним поліпшенням алгоритму найближчих точок, що полягає в: модифікації методів вибору області перекриття і пар найближчих точок (наприклад, за допомогою введення жорстких обмежень на клас рухів, що переводять одну точку в іншу, з використанням теорії випадкових величин,  $k$ -d дерев, генетичних алгоритмів); призначення ваг парам найближчих точок; модифікації формули для відстані між точками; модифікація формули для мінімізації помилки; модифікації методу мінімізації помилки.

Недоліком поліпшень алгоритму найближчих точок є їх складність, налаштування на конкретні експериментальні дані, що зменшує стійкість алгоритмів.

В [3] розглядалася задача порівняння поверхонь одного і того ж об'єкта, заданих на різних множинах точок, при цьому відстань від точки на одній поверхні до іншої поверхні обчислювалася уздовж нормалей до сплайнів другої поверхні. Міра порівняння, яка заснована на обчисленні різниці відстаней уздовж нормалей до поверхонь, цікава тим, що не вимагає приведення порівнюваних функцій до загальної сітці. Однак складність такого підходу квадратична.

Такі міри, як, наприклад, середня відстань між висотами, які можуть бути безпосередньо обчислені при завданні обох поверхонь у вузлах загальної сітки, не допускають узагальнення на випадок їх завдання в вузлах різних сіток без етапу приведення сіток до регулярних.

В [2] міра для порівняння двох поверхонь  $X$  і  $Y$ , заданих в вузлах сіток  $g_1$  і  $g_2$  відповідно, вводилася як максимальна відстань між парами найближчих точок різних сіток:

$$\rho(X, Y) = \max_{g_1[i] \in g_1} \min_{g_2[j] \in g_2} d(g_1[i], g_2[j]), \quad (1.4)$$

де  $d(g_1[i], g_2[j])$  – евклідова відстань між двома точками. Така міра може бути узагальнена на випадок різних нерегулярних сіток, але її обчислення матиме

квадратичну складність і, крім того, її фізичний зміст не очевидний і прийнятний в малій кількості додатків.

У [1] запропонований алгоритм зіставлення поверхонь, заснований на мінімізації міри близькості між ними. При цьому порівнювані поверхні  $X$  і  $Y$  розглядаються як об'єкти лінійного простору, а міра близькості  $p(S_1, S_2)$  являє собою норму  $\|X - Y\|$  в цьому просторі. Нехай поверхня  $X$  представлена триангуляційною кусочно-лінійною моделлю з  $N$  трикутників,  $c_i$  – центроїд  $i$ -ого трикутника,  $n_i$  – вектор нормалі до  $i$ -ого трикутника, довжина якого дорівнює площі цього трикутника. Норму поверхні  $X$  вводиться як

$$X = \sum_{i=1}^N \sum_{j=1}^N (n_i, n_j) e^{-|c_i - c_j|^2 / \sigma^2}. \quad (1.5)$$

Нехай вихідні поверхні  $S_1, S_2$  задані триангуляційними кусочно-лінійними моделями з  $N_1, N_2$  трикутників відповідно;  $c_i^1(c_i^2)$  – центроїд  $i$ -ого трикутника поверхні  $S_1(S_2)$ ,  $n_i^1(n_i^2)$  – вектор нормалі до  $i$ -ого трикутника поверхні  $S_1(S_2)$ , довжина якого дорівнює площі цього трикутника. Міра близькості, що пропонується в [4], вводиться наступним чином:

$$\begin{aligned} \rho(X, Y) &= \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} (n_i^1, n_j^1) e^{-|c_i^1 - c_j^1|^2 / \sigma^2} + \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} (n_i^2, n_j^2) e^{-|c_i^2 - c_j^2|^2 / \sigma^2} \\ &- 2 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (n_i^1, n_j^2) e^{-|c_i^1 - c_j^2|^2 / \sigma^2} = \|S_1 - S_2\| \end{aligned} \quad (1.6)$$

Далі значення міри (1.6) мінімізується методом найшвидшого спуску. Недоліком такого підходу є квадратична складність обчислення міри близькості.

## 2 Алгоритм оцінки наближених значень елементів взаємного орієнтування

Цей алгоритм заснований на використанні орієнтаційних гістограм, який дозволяє вирішити задачу без використання початкових наближень. Експериментальні результати показують ефективність запропонованого методу на різних прикладах [6]. Алгоритм оцінки наближених значень елементів взаємного орієнтування складається з двох етапів:

- 1) виконання оцінки кутової орієнтації  $R$ ;
- 2) оцінка вектору зсуву  $T$ .

В алгоритмі розглядаються всі точки, у яких нормаль до поверхні спрямована вгору. Також виключаються точки з невизначеним становищем нормалі.

Оцінка кутової орієнтації складається з наступних кроків:

- 1) побудова двох орієнтаційних гістограм  $H_1$  і  $H_2$  для хмар точок  $P_1$  і  $P_2$  відповідно;

2) знаходження максимуму кореляційної функції  $C(H_1, H_2\_rotated)$  для всіх можливих значень матриці повороту  $R$ .

Значення матриці повороту, відповідне максимуму кореляційної функції, і є шуканою кутовою орієнтацією.

Оцінка вектору зсуву виглядає наступним чином: спочатку хмара точок  $P_2$  повертається з використанням знайденої раніше матриці повороту. Потім будуватиметься бінарне вексельне уявлення для хмар точок за допомогою процедури тривимірної дискретизації. Елемент вексельного уявлення дорівнює 1, якщо

всередині векселя знаходиться хоча б одна тривимірна точка, і дорівнює 0 в іншому випадку. Оцінка зсуву між двома системами координат хмар точок вважається як максимум кореляційної функції  $G$ , який розраховується за допомогою перетворення Фур'є [9]. Приклад роботи алгоритму оцінки наближених значень елементів взаємного орієнтування зображено на рисунку 1 [6].

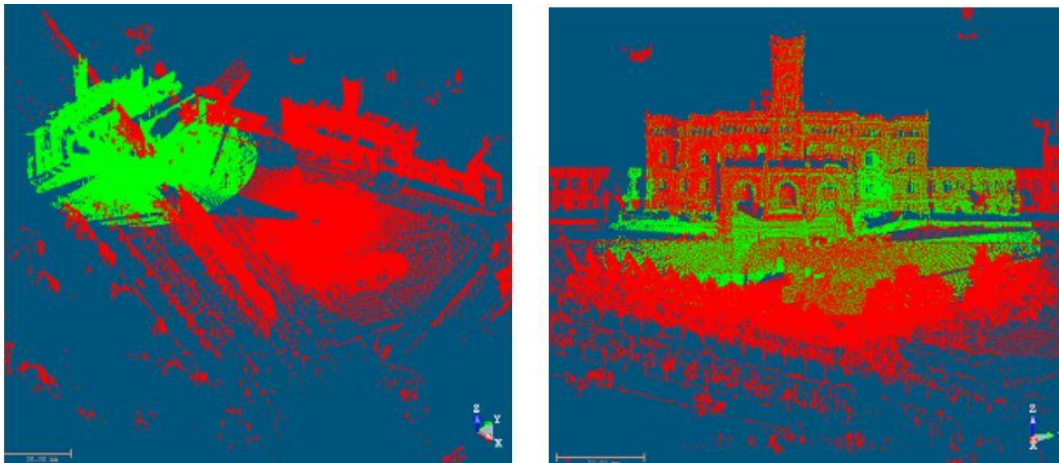


Рисунок 1 – Хмара точок до і після реєстрації. Початковий кут повороту навколо осі  $Z$  складає  $\sim 72$  градуси. Довжина початкового вектору зсуву становить  $\sim 70\%$  від розміру моделі

### Висновки

В ході дослідження було виявлено, що ітеративний алгоритм найближчих точок є більш детальним й точним при моделюванні об'єктів. В той же час цей метод потребує дуже точних значень і при обчисленні міри близькості складність прорахування цим алгоритмом збільшується в рази. Тоді як алгоритм оцінки наближених значень елементів взаємного орієнтування не потребує інформації про наближену орієнтацію хмар точок, що спрощує роботу і зменшує час моделювання.

Розглянуті алгоритми дуже затребувані в промислових галузях. Одним з прикладів застосування обох алгоритмів є моделювання трубопроводів. Для цього виділяється масив точок певного трубопроводу і вказується початкова точка, програма автоматично вписує в цю хмару точок циліндри. Головна перевага такого методу – висока швидкість моделювання. Також часте застосування алгоритмів зустрічається при моделюванні металоконструкцій. Перед побудовою по хмарі точок точно визначаються всі габарити, створюються напрямні площині і площині, що обмежують довжину об'єкта [2].

Для використання пропонованих алгоритмів необхідно, щоб хмари точок були наближеними одна до одної, тому було розроблено програму `Stl_Parser.`, що дозволяє перевіряти файли, які містять множину точок. Наприклад `stl`-файли. Порівняно `.stl`-файл, отриманий після 3D-сканування, й `.stl`-файл, створений у середовищі моделювання `SOLIDWORKS`. У результаті роботи було виявлено, що файли мають різне програмне походження та суттєво відрізняються один від одного за обсягом й кількістю трикутників. Порівняння їх геометрії є неможливим (рис. 2).

```
First file:  
STL HEADER = solid Butyl_Solid  
# triangles = 4838  
  
Second file:  
STL HEADER = STLEXP Object01  
# triangles = 204045  
Model D:/Butyl_Solid.stl is much bigger than model C:/Users/Lenovo/Desktop/Butyl.stl. Impossible to compare.
```

Рисунок 2 – Результат роботи програми Stl\_Parser

Протестована програма на .stl-файлах, створених у середовищі моделювання SOLIDWORKS. Результат представлений на рисунку 3.

```
Please, enter full path to the first file: D:/Part_Full.stl  
Please, enter full path to the second file: D:/Part1.stl  
  
First file:  
STL HEADER = solid Part_Full  
# triangles = 12  
  
Second file:  
STL HEADER = solid Part1  
# triangles = 16  
Models may be comparable.
```

Рисунок 3 – Результат порівняння .stl-файлів, створених у середовищі

Було виявлено, що оскільки файли мають спільне програмне походження та несуттєво відрізняються один від одного за обсягом й кількістю трикутників, то порівняння їх геометрії є можливим.

### Перелік посилань

1. Митцева А. Разработка программной системы для совмещения данных трехмерного сканирования. // Высшая школа электроники и компьютерных наук. 2018. – С. 53.
2. Li, S. Tree point clouds registration using an improved ICP algorithm based on kd-tree / Li S., Wang J., Liang Z., Su L., 2016. – P. 4.
3. Wang, F. A survey of iterative closest point algorithm. / Wang F., Zhao Z., 2017. – P. 34.
4. Старков, А.Д Реализация алгоритма iterative closest point. // Дальневосточный федеральный университет, 2018. – С. 60.
5. Diez, Y. A Qualitative Review on 3D Coarse Registration Methods / Diez Y., Route F., Llado X., Salvi J: ACM Computing Surveys, 2015. – P. 36
6. Велижев А. Автоматическая регистрация облаков точек на основе ориентационных гистограмм. – Режим доступа: [graphics.cs.msu.ru/ru/science/research/3dpoint/alignment](http://graphics.cs.msu.ru/ru/science/research/3dpoint/alignment).
7. Pohudina O. et al. Possibilities of Position Determination //Integrated Computer Technologies in Mechanical Engineering. – Springer, Cham, 2020. – С. 523-537.

### References

1. Myttseva A. Razrabotka prohrammnoy systemy dlya sovmeshchenyya dannykh trekhmernogo skanyrovaniya [Development of a software system for combining 3D scanning data]. // Graduate School of Electronics and Computer Science. 2018. – P. 53.

2. Li, S. Tree point clouds registration using an improved ICP algorithm based on kd-tree / Li S., Wang J., Liang Z., Su L., 2016. – P. 4.
3. Wang, F. A survey of iterative closest point algorithm. / Wang F., Zhao Z., 2017. – P. 34.
4. Starkov, A.D Realyzatsyya alhorytma iterative closest point [Implementation of the iterative closest point algorithm]. // Far Eastern Federal University, 2018. – P. 60.
5. Diez, Y. A Qualitative Review on 3D Coarse Registration Methods / Diez Y., Route F., Llado X., Salvi J: ACM Computing Surveys, 2015. – P. 36
6. Velyzhev A. Avtomatycheskaya rehystratsyya oblakov toчек na osnove oryentatsyonnykh hystohramm [Automatic registration of point clouds based on orientation histograms]. – Access mode: graphics.cs.msu.ru/ru/science/research/3dpoint/alignment.
7. Pohudina O. et al. Possibilities of Position Determination //Integrated Computer Technologies in Mechanical Engineering. – Springer, Cham, 2020. – С. 523-537.

Надійшла до редакції 20.05.2021, розглянута на редколегії 20.05.2021

## **Comparative analysis of methods for finding the nearest points in the images of objects of technical systems**

The subject of the study is the process of mapping the construction of point clouds of technical systems using the algorithm of the nearest points. The goal is to minimize the alignment criterion by converting a set of cloud points  $Y$  into a set of cloud points  $X$  in a software product that uses an iterative closest point (ICP) algorithm. Objectives: to analyze the properties of input images that contain point clouds; to review the algorithms for identifying and comparing key points; implement a cloud comparison algorithm using the ISR algorithm; consider an example of the algorithm for estimating the approximate values of the elements of mutual orientation; implement software that allows you to compare files that contain point clouds and draw conclusions about the possibility of comparing them. The methods used are: search for points using the algorithm of iterative nearest points, the algorithm for estimating the approximate values of the elements of mutual orientation, the method of algorithm theory for the analysis of file structures STL (standard template library - format template library) format. The following results were obtained. The choice of the ICP algorithm for the task of reconstruction of the object of technical systems is substantiated; the main features of the ISR algorithm are considered; the algorithm of comparison of key points, and also optimization that allows reducing criterion of combination at the reconstruction of three-dimensional objects of technical systems results. Conclusions. The study found that the iterative near-point algorithm is more detailed and accurate when modeling objects. At the same time, this method requires very accurate values and when calculating the degree of proximity, the complexity of calculation by this algorithm increases many times. Whereas the algorithm for estimating the approximate values of the elements of mutual orientation does not require information about the approximate orientation of the point clouds, which simplifies the work and reduces the simulation time. It was found that not all files are comparable. Therefore, the software is implemented, which gives an opinion on the

possibility of comparing points in the proposed two files, which contain clouds of points in the structure of the STL format.

**Keywords:** search for the nearest point clouds, iterative algorithm of the nearest points, algorithm for estimating the approximate values of the elements of mutual orientation.

#### **Відомості про авторів:**

**Мирненко Максим Дмитрович**, студент кафедри інформаційні технології проектування Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут» <https://orcid.org/0000-0002-5777-9583>

**Крицький Дмитро Миколайович**, кандидат технічних наук, завідувач кафедри інформаційні технології проектування Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут» <https://orcid.org/0000-0003-4919-0194>

**Погудіна Ольга Костянтинівна**, кандидат технічних наук, доцент кафедри інформаційні технології проектування Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут» <https://orcid.org/0000-0001-5689-2552>

**Крицька Ольга Сергіївна**, аспірант кафедри інформаційні технології проектування Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут» <https://orcid.org/0000-0002-3949-1164>

#### **About the Authors:**

**Myrnenko Maksym Dmytrovych**, student of the Department of Information Technology Design of the National Aerospace University named after. M. Zhukovsky "Kharkiv Aviation Institute"

**Krytskiy Dmytro Mykolayovych**, Assistant at the Department of Information Technology of Design at National Aerospace University Kharkiv Aviation Institute

**Pohudina Olha Kostyantynivna**, Assistant at the Department of Information Technology of Design at National Aerospace University Kharkiv Aviation Institute

**Krytska Olha Sergievna**, PhD student of the Department of Information Technology Design of the National Aerospace University named after. M. Zhukovsky "Kharkiv Aviation Institute"