

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КІЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ**

«До захисту допущено»

Завідувач кафедри

М. В. Грайворонський
(підпис)
(ініціали, прізвище)
“ ” 2015

р.

Дипломна робота

освітньо-кваліфікаційного рівня “бакалавр”

з напряму підготовки 6.040301 «Прикладна математика»
на тему «Ідентифікація випадкових відображень точкових множин в скінчен-
новимірних просторах»

Виконала студентка 4 курсу групи ФІ-41

Лавягіна Ольга Олексіївна

Керівник кандидат ф.-м. наук Рябов Георгій Валентинович

Рецензент

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авто-
рів без відповідних посилань.

Студент _____

ЗМІСТ

Вступ	3
1 Попередні роботи присвячені співставленню точкових множин	5
1.1 Перші описи алгоритму	5
1.2 Сучасні роботи	5
1.3 Постановка задачі	6
1.4 Модифікації	7
2 Розв'язання задачі методом найменших квадратів	8
2.1 Оцінка невідомих параметрів задачі методом найменших квадратів .	8
2.2 Статистичні властивості оцінки найменших квадратів	16
3 Розв'язання задачі ітеративним алгоритмом найближчих точок.	18
3.1 Ітеративний алгоритм найближчих точок.	18
3.2 Збіжність алгоритму	29
3.3 Розподіл оцінок.	29
3.3.1 Теорема про незалежність елементів матриці X	29
3.3.2 Розподіл матриць U та V в сингулярному розкладі матриці X . .	34
3.3.3 Розподіл елементів матриці повороту	35
3.4 Простий приклад незадовільної роботи алгоритму	37
3.5 Стенфордський кролик	39
Висновки	45
Перелік посилань	46
Додаток А	48

ВСТУП

Значною мірою робота завдячує Андрію Анатолійовичу Дороговцеву — професору, доктору фізико-математичних наук, завідувачу відділу теорії випадкових процесів Інституту математики НАН України.

Актуальність роботи. Оцінка положення камери по хмарам точок (або точковим множинам) лежить в основі сканування об'єктів за допомогою 3D сканера, одночасній локалізації і картографування. Для розв'язання цих задач використовується ітеративний алгоритм найближчих точок і його модифікації. У зв'язку з розвитком та компактизацією обчислюальної техніки з'явилася можливість реалізовувати алгоритм на маленьких комп'ютерах (наприклад, бортові комп'ютери дронів). Оскільки потрібно, щоб такі пристрой працювали надійно, треба ретельно вивчити властивості алгоритму, що використовується.

Об'єкт дослідження — методи оцінки параметрів камери.

Предмет дослідження — алгоритм співставлення точкових множин.

Мета дослідження. Аналіз алгоритму співставлення двох точкових множин та отриманих за його допомогою оцінок невідомих параметрів.

Завдання наступні:

- 1) застосувати метод найменших квадратів для розв'язання задачі;
- 2) ознайомитися з ітеративним алгоритмом найближчих точок, що використовується для співставлення двох точкових множин;
- 3) перевірити однозначність оцінки, яка є результатом роботи алгоритму;
- 4) дослідити алгоритм на збіжність;
- 5) розробити програмну реалізацію алгоритму.

Практичне значення одержаних результатів.

Ітеративний алгоритм найближчих точок можна використовувати для відновлення двовимірних або тривимірних поверхонь, отриманих за допомогою 3D сканера. Було з'ясовано, що алгоритм визначений однозначно, тобто за одинакових вхідних даних він дає один і той самий результат. Це додає зру-

чності для подальшого дослідження, а також спрощує реалізацію алгоритму, бо не виникає необхідності вибору оптимального розв'язку в умовах його небільшості.

Публікації.

XVI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики».

1 ПОПЕРЕДНІ РОБОТИ ПРИСВЯЧЕНІ СПІВСТАВЛЕННЮ ТОЧКОВИХ МНОЖИН

В першому розділі розглянуто коротку історію досліджень, що пов'язані зі співсталенням точкових множин за допомогою ітеративного алгоритму найближчих точок. Розбір попередніх робіт дає змогу чітко поставити задачу, що розв'язується в другому та третьому розділах дипломної роботи.

1.1 Перші описи алгоритму

У лютому 1992 року Пол Бесл та Ніл Маккей опублікували статтю з описом ітеративного алгоритму найближчих точок для співставлення точкових множин [1]. Алгоритм полягає в ітеративній мінімізації середньоквадратичної відстані між тривимірними множинами. В роботі доведено, що алгоритм монотонно збігається до локального мінімуму цієї метрики.

У квітні 1992 року Ян Чен та Жерар Медіоні опубліковали статтю, в якій описали застосування ітеративного алгоритму найближчих точок для отримання повної моделі фізичного об'єкта [2]. Результати роботи представлені на прикладі реконструкції гіпсового зубу та бюсту Моцарта (рис. 1.1). На перших двох рисунках зображені дві вхідні множини, на третьому — результат роботи алгоритму.

1.2 Сучасні роботи

На момент написання дипломної роботи одними з новітніх робіт, де був використаний ітеративний алгоритм найближчих точок, є перевірка правильності положення пацієнта на томотерапії при лікуванні онкологічних захворювань [3] та виявлення геометричних деформацій стін, стелі та доріг підземної

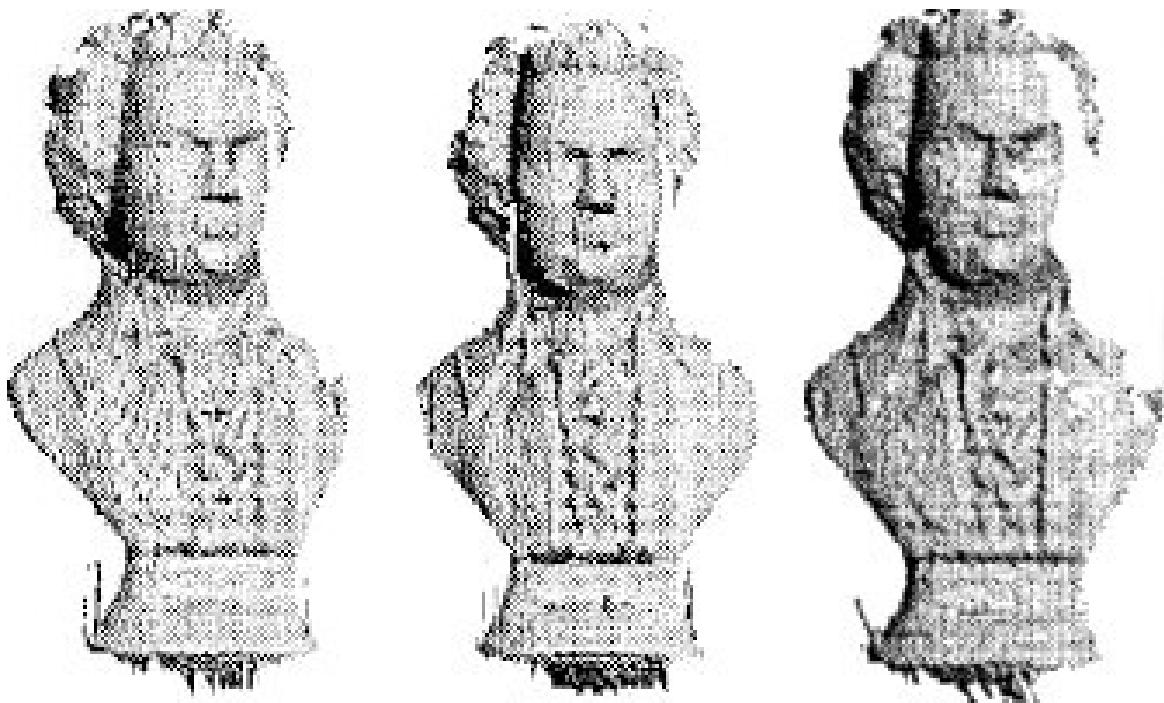


Рисунок 1.1 — Бюст Моцарта

парковки, за якими ведеться спостереження [4]. В останній статті розв'язується задача одночасної локалізації і картографування всередині будівлі, де неможливо використовувати GPS.

1.3 Постановка задачі

Запишемо постановку задачі, яка досліджується у згадуваних статтях. Є дві множини: вихідна $S \subset \mathbb{R}^3$ та цільова $T \subset \mathbb{R}^3$. Точки вихідної множини $s \in S$ повернули за допомогою матриці

$$\begin{cases} R \in \mathbb{R}^{3 \times 3}, \\ R^T = R^{-1}, \\ \det R = 1 \end{cases}$$

та зсунули за допомогою вектора $\mathbf{b} \in \mathbb{R}^3$. Також в процесі сканування з'явився адитивний гаусів шум з незалежними компонентами на невідомою дисперсією

$$\mathbf{k}_s = R \cdot \mathbf{s} + \mathbf{b} + \boldsymbol{\xi}_s, \quad \boldsymbol{\xi}_s \sim N(\mathbf{0}, \sigma^2 \cdot I), \quad (1.1)$$

де $k : S \rightarrow T$ – розмітка, тобто сюр'ективне відображення, яке співставляє кожну точку вихідної множини з точкою з цільової множини.

Задача полягає в такому виборі матриці R та вектора \mathbf{b} , за яких евклідова відстань між \mathbf{k}_s та $R \cdot \mathbf{s} + \mathbf{b}$ для всіх $\mathbf{s} \in S$ була б найменшою.

1.4 Модифікації

В деяких роботах [5, 6] при оптимізації враховують не тільки точки двох множин множин, але й нормальні до них, а також відстані між деякими точками множать на ваги.

Висновки до розділу 1

Проведено огляд задач, при розв'язанні яких використовується ітеративний алгоритм найближчих точок. Поставлена задача, розв'язання якої наведено в наступних розділах дипломної роботи.

2 РОЗВ'ЯЗАННЯ ЗАДАЧІ МЕТОДОМ НАЙМЕНШИХ КВАДРАТІВ

Другий розділ присвячено розв'язку задачі методом найменших квадратів. Досліджуються властивості оцінок найменших квадратів та з'ясовується придатність розв'язку для поставленої задачі.

2.1 Оцінка невідомих параметрів задачі методом найменших квадратів

В тому випадку, коли S і T скінченні, можна скористатися звичайним методом найменших квадратів [7]. Виконаємо оцінку невідомих параметрів у виразі (1.1). Для цього вимагаємо, щоб сума квадратів похибок досягала мінімуму. Іншими словами, знайдемо мінімум величини

$$E(k, R, \mathbf{b}) = \sum_{\mathbf{s} \in S} \|\xi_{\mathbf{s}}\|^2 = \sum_{\mathbf{s} \in S} \|k_{\mathbf{s}} - R \cdot \mathbf{s} - \mathbf{b}\|^2 \rightarrow \min_{k, R, \mathbf{b}}. \quad (2.1)$$

Сума квадратів відхилень між векторами — це те ж саме, що і сума квадратів відхилень між проекціями по кожній координаті

$$E(k, R, \mathbf{b}) = E_x(k, R, \mathbf{b}) + E_y(k, R, \mathbf{b}) + E_z(k, R, \mathbf{b}) \rightarrow \min_{k, R, \mathbf{b}}.$$

Знайдемо, чому дорівнює добуток матриці R та вектора \mathbf{s}

$$R \cdot \mathbf{s} = \begin{bmatrix} r_{xx} & r_{xy} & r_{xz} \\ r_{yx} & r_{yy} & r_{yz} \\ r_{zx} & r_{zy} & r_{zz} \end{bmatrix} \cdot \begin{bmatrix} s_x \\ s_y \\ s_z \end{bmatrix} = \begin{bmatrix} r_{xx} \cdot s_x + r_{xy} \cdot s_y + r_{xz} \cdot s_z \\ r_{yx} \cdot s_x + r_{yy} \cdot s_y + r_{yz} \cdot s_z \\ r_{zx} \cdot s_x + r_{zy} \cdot s_y + r_{zz} \cdot s_z \end{bmatrix} = \begin{bmatrix} \mathbf{r}_x \cdot \mathbf{s} \\ \mathbf{r}_y \cdot \mathbf{s} \\ \mathbf{r}_z \cdot \mathbf{s} \end{bmatrix}.$$

Розпишему суму квадратів відхилень через проекції

$$\begin{aligned}
 E(k, R, \mathbf{b}) &= \sum_{\mathbf{s} \in S} (\mathbf{r}_x \cdot \mathbf{s} + \mathbf{r}_y \cdot \mathbf{s} + \mathbf{r}_z \cdot \mathbf{s} + b_x + b_y + b_z - k_{s_x} - k_{s_y} - k_{s_z})^2 = \\
 &= \sum_{\mathbf{s} \in S} [(\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x}) + (\mathbf{r}_y \cdot \mathbf{s} + b_y - k_{s_y}) + (\mathbf{r}_z \cdot \mathbf{s} + b_z - k_{s_z})]^2 = \\
 &= \sum_{\mathbf{s} \in S} (\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x})^2 + \sum_{\mathbf{s} \in S} (\mathbf{r}_y \cdot \mathbf{s} + b_y - k_{s_y})^2 + \sum_{\mathbf{s} \in S} (\mathbf{r}_z \cdot \mathbf{s} + b_z - k_{s_z})^2.
 \end{aligned}$$

Множини параметрів, які входять в кожну з трьох сум, різні, тому можемо мінімізувати суми квадратів проекцій відхилень на кожну координату окремо

$$\begin{cases} E_x = \sum_{\mathbf{s} \in S} (\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x})^2 \rightarrow \min_{\mathbf{r}_x, b_x}, \\ E_y = \sum_{\mathbf{s} \in S} (\mathbf{r}_y \cdot \mathbf{s} + b_y - k_{s_y})^2 \rightarrow \min_{\mathbf{r}_y, b_y}, \\ E_z = \sum_{\mathbf{s} \in S} (\mathbf{r}_z \cdot \mathbf{s} + b_z - k_{s_z})^2 \rightarrow \min_{\mathbf{r}_z, b_z}. \end{cases}$$

Маємо лінійні функції точок вихідної множини, які підносяться до квадрату. Це випуклі функції. Отже, можна взяти часткові похідні по \mathbf{r}_i та b_i для всіх $i \in \{x, y, z\}$ та прирівняти їх до нуля. Отримаємо по чотири рівняння дляожної координати. Без втрати загальності запишемо ці рівняння для E_x . Таким чином, екстремальне значення E_x отримаємо в результаті розв'язку системи лінійних алгебраїчних рівнянь

$$\begin{cases} \frac{\partial E_x}{\partial b_x} = \sum_{\mathbf{s} \in S} 2 (\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x}) = 0, \\ \frac{\partial E_x}{\partial r_{xx}} = \sum_{\mathbf{s} \in S} 2 (\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x}) \cdot s_x = 0, \\ \frac{\partial E_x}{\partial r_{xy}} = \sum_{\mathbf{s} \in S} 2 (\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x}) \cdot s_y = 0, \\ \frac{\partial E_x}{\partial r_{xz}} = \sum_{\mathbf{s} \in S} 2 (\mathbf{r}_x \cdot \mathbf{s} + b_x - k_{s_x}) \cdot s_z = 0. \end{cases}$$

Розв'язуємо перше рівняння відносно b_x

$$\sum_{\mathbf{s} \in S} b_x = \sum_{\mathbf{s} \in S} (k_{s_x} - \mathbf{r}_x \cdot \mathbf{s}).$$

Спростимо останній вираз

$$|S| \cdot b_x + \mathbf{r}_x \sum_{\mathbf{s} \in S} \mathbf{s} = \sum_{\mathbf{s} \in S} k_{s_x}.$$

Розпишемо скалярний добуток

$$|S| \cdot b_x + \sum_{\mathbf{s} \in S} r_{xx} \cdot s_x + \sum_{\mathbf{s} \in S} r_{xy} \cdot s_y + \sum_{\mathbf{s} \in S} r_{xz} \cdot s_z = \sum_{\mathbf{s} \in S} k_{s_x}.$$

Розв'язуємо інші рівняння відносно r_{xi} для $i \in \{x, y, z\}$. Без втрати загальності знаходимо розв'язок тольки для r_{xx}

$$\mathbf{r}_x \sum_{\mathbf{s} \in S} \mathbf{s} \cdot s_x = \sum_{\mathbf{s} \in S} (k_{s_x} - b_x) \cdot s_x.$$

Розпишемо скалярний добуток

$$\sum_{\mathbf{s} \in S} r_{xx} \cdot s_x^2 + \sum_{\mathbf{s} \in S} r_{xy} \cdot s_x \cdot s_y + \sum_{\mathbf{s} \in S} r_{xz} \cdot s_x \cdot s_z + \sum_{\mathbf{s} \in S} b_x \cdot s_x = \sum_{\mathbf{s} \in S} k_{s_x} \cdot s_x.$$

Маємо систему рівнянь

$$\begin{cases} |S| \cdot b_x + \sum_{s \in S} r_{xx} \cdot s_x + \sum_{s \in S} r_{xy} \cdot s_y + \sum_{s \in S} r_{xz} \cdot s_z = \sum_{s \in S} k_{s_x}, \\ \sum_{s \in S} r_{xx} \cdot s_x^2 + \sum_{s \in S} r_{xy} \cdot s_x \cdot s_y + \sum_{s \in S} r_{xz} \cdot s_x \cdot s_z + \sum_{s \in S} b_x \cdot s_x = \sum_{s \in S} k_{s_x} \cdot s_x, \\ \sum_{s \in S} r_{xx} \cdot s_x \cdot s_y + \sum_{s \in S} r_{xy} \cdot s_y^2 + \sum_{s \in S} r_{xz} \cdot s_y \cdot s_z + \sum_{s \in S} b_x \cdot s_y = \sum_{s \in S} k_{s_x} \cdot s_y, \\ \sum_{s \in S} r_{xx} \cdot s_x \cdot s_z + \sum_{s \in S} r_{xy} \cdot s_y \cdot s_z + \sum_{s \in S} r_{xz} \cdot s_z^2 + \sum_{s \in S} b_x \cdot s_z = \sum_{s \in S} k_{s_x} \cdot s_z. \end{cases}$$

Запишемо її в матричному вигляді

$$\begin{bmatrix} |S| & \sum_{s \in S} s_x & \sum_{s \in S} s_y & \sum_{s \in S} s_z \\ \sum_{s \in S} s_x & \sum_{s \in S} s_x^2 & \sum_{s \in S} s_x \cdot s_y & \sum_{s \in S} s_x \cdot s_z \\ \sum_{s \in S} s_y & \sum_{s \in S} s_x \cdot s_y & \sum_{s \in S} s_y^2 & \sum_{s \in S} s_y \cdot s_z \\ \sum_{s \in S} s_z & \sum_{s \in S} s_x \cdot s_z & \sum_{s \in S} s_y \cdot s_z & \sum_{s \in S} s_z^2 \end{bmatrix} \begin{bmatrix} b_x \\ r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} = \begin{bmatrix} \sum_{s \in S} k_{s_x} \\ \sum_{s \in S} k_{s_x} \cdot s_x \\ \sum_{s \in S} k_{s_x} \cdot s_y \\ \sum_{s \in S} k_{s_x} \cdot s_z \end{bmatrix}.$$

Введемо позначення

$$\sum_{s \in S} s_i = S_i, \quad i \in \{x, y, z\},$$

$$\sum_{s \in S} s_i s_j = S_{ij}, \quad i, j \in \{x, y, z\},$$

$$\sum_{s \in S} k_{s_x} = K,$$

$$\sum_{s \in S} k_{s_x} \cdot s_i = k_i, \quad i \in \{x, y, z\}.$$

Рівняння прийняло наступний вигляд

$$\begin{bmatrix} |S| & S_x & S_y & S_z \\ S_x & S_{xx} & S_{xy} & S_{xz} \\ S_y & S_{xy} & S_{yy} & S_{yz} \\ S_z & S_{xz} & S_{yz} & S_{zz} \end{bmatrix} \begin{bmatrix} b_x \\ r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} = \begin{bmatrix} k \\ k_x \\ k_y \\ k_z \end{bmatrix}.$$

Використовуємо метод Крамера для розв'язання системи лінійних рівнянь.

Визначник Δ

$$\begin{aligned} \Delta = & |S| \cdot S_{xx} \cdot S_{yy} \cdot S_{zz} - \sum_{i \in \{x,y,z\}} L_i + 2 \cdot |S| \cdot S_{xy} \cdot S_{xz} \cdot S_{yz} - \\ & - \sum_{i,j,k \in \{x,y,z\}} L_{ijk} + 2 \sum_{i,j \in \{x,y,z\}} L_{ij}, \end{aligned}$$

де введені позначення при $i, j, k \in \{x, y, z\}$, $i \neq j \neq k$

$$L_i = S_{jk} \cdot (|S| \cdot S_{ii} - S_i^2),$$

$$L_{ij} = S_i \cdot S_j \cdot (S_{ij} \cdot S_k - S_{ik} \cdot S_{jk}),$$

$$L_{ijk} = S_i^2 \cdot S_{jj} \cdot S_{kk}.$$

Визначник Δ_b

$$\begin{aligned} \Delta_b = & k \cdot S_{xx} \cdot S_{yy} \cdot S_{zz} - \sum_{i \in \{x,y,z\}} L_i^b + 2 \cdot S_{xy} \cdot S_{xz} \cdot S_{yz} - \sum_{i,j,k \in \{x,y,z\}} L_{ijk}^b + \\ & + \sum_{i,j \in \{x,y,z\}} L_{ij}^b + \sum_{i,j \in \{x,y,z\}} (L_{ij}^b)', \end{aligned}$$

де введені позначення

$$\begin{aligned} L_{ij}^b &= S_{ij}^2 \cdot S_k \cdot k_k, \\ L_{ijk}^b &= S_i \cdot S_{jj} \cdot S_{kk}, \\ (L_{ij}^b)' &= (S_i \cdot k_j + S_k \cdot k_i) \cdot (S_{ij} \cdot S_{kk} - S_{jk} \cdot S_{ik}) \end{aligned}$$

при $i, j, k \in \{x, y, z\}$, $i \neq j \neq k$. Визначник Δ_{xx}

$$\begin{aligned} \Delta_{xx} = & -K \cdot S_x \cdot S_{yy} \cdot S_{zz} + K \cdot S_x \cdot S_{yz}^2 + K \cdot S_y \cdot S_{xy} \cdot S_{zz} - K \cdot S_y \cdot S_{xz} \cdot S_{yz} - \\ & -K \cdot S_z \cdot S_{xy} \cdot S_{yz} + K \cdot S_z \cdot S_{xz} \cdot S_{yy} + k_x \cdot |S| \cdot S_{yy} \cdot S_{zz} - k_x \cdot |S| \cdot S_{yz}^2 - \\ & -k_x \cdot S_y^2 \cdot S_{zz} + 2 \cdot k_x \cdot S_y \cdot S_z \cdot S_{yz} - k_x \cdot S_z^2 \cdot S_{yy} - k_y \cdot |S| \cdot S_{yz} \cdot S_{zz} + \\ & +k_y \cdot |S| \cdot S_{xz} \cdot S_{yz} + k_y \cdot S_x \cdot S_y \cdot S_{zz} - k_y \cdot S_x \cdot S_y \cdot S_{yz} - k_y \cdot S_y \cdot S_z \cdot S_{xz} + \\ & +k_y \cdot S_z^2 \cdot S_{xy} + k_z \cdot |S| \cdot S_{xy} \cdot S_{yz} - k_z \cdot |S| \cdot S_{xz} \cdot S_{yy} - k_z \cdot S_x \cdot S_y \cdot S_{yz} + \\ & +k_z \cdot S_x \cdot S_z \cdot S_{yy} + k_z \cdot S_y^2 \cdot S_{xz} - k_z \cdot S_y \cdot S_z \cdot S_{xy}. \end{aligned}$$

Визначник Δ_{xy}

$$\begin{aligned} \Delta_{xy} = & K \cdot S_x \cdot S_{xy} \cdot S_{zz} - K \cdot S_x \cdot S_{xz} \cdot S_{yz} - K \cdot S_y \cdot S_{xx} \cdot S_{zz} + K \cdot S_y \cdot S_{xz}^2 + \\ & + K \cdot S_z \cdot S_{xx} \cdot S_{yz} - K \cdot S_z \cdot S_{xy} \cdot S_{xz} - k_x \cdot |S| \cdot S_{xy} \cdot S_{zz} + k_x \cdot |S| \cdot S_{xz} \cdot S_{yz} + \\ & + k_x \cdot S_x \cdot S_y \cdot S_{zz} - k_x \cdot S_x \cdot S_z \cdot S_{yz} - k_x \cdot S_y \cdot S_z \cdot S_{xz} + k_x \cdot S_z^2 \cdot S_{xy} + \\ & + k_y \cdot |S| \cdot S_{xx} \cdot S_{zz} - k_y \cdot |S| \cdot S_{xz}^2 - k_y \cdot S_x^2 \cdot S_{zz} + 2 \cdot k_y \cdot S_x \cdot S_z \cdot S_{xz} - \\ & - k_y \cdot S_x \cdot S_z \cdot S_{xz} - k_y \cdot S_z^2 \cdot S_{xx} - k_z \cdot |S| \cdot S_{xx} \cdot S_{yz} + k_z \cdot |S| \cdot S_{xy} \cdot S_{xz} + \\ & + k_z \cdot S_x^2 \cdot S_{yz} - k_z \cdot S_x \cdot S_y \cdot S_{xz} - k_z \cdot S_x \cdot S_z \cdot S_{xy} + k_z \cdot S_y \cdot S_z \cdot S_{xx}. \end{aligned}$$

Визначник Δ_{xz}

$$\begin{aligned}\Delta_{xz} = & -K \cdot S_x \cdot S_{xy} \cdot S_{yz} + K \cdot S_x \cdot S_{xz} \cdot S_{yy} + K \cdot S_y \cdot S_{xx} \cdot S_{yz} - \\ & -K \cdot S_y \cdot S_{xy} \cdot S_{xz} - K \cdot S_z \cdot S_{xx} \cdot S_{yy} + K \cdot S_z \cdot S_{xy}^2 + S_x \cdot |S| \cdot S_{xy} \cdot S_{yz} - \\ & -k_x \cdot |S| \cdot S_{xz} \cdot S_{yy} - k_x \cdot S_x \cdot S_y \cdot S_{yz} + k_x \cdot S_x \cdot S_y \cdot S_{yy} + k_x \cdot S_y^2 \cdot S_{xz} - \\ & -k_x \cdot S_y \cdot S_z \cdot S_{xy} - k_y \cdot |S| \cdot S_{xx} \cdot S_{yz} + k_y \cdot |S| \cdot S_{xy} \cdot S_{xz} + k_y \cdot S_x^2 \cdot S_{yz} - \\ & -k_y \cdot S_x \cdot S_y \cdot S_{xz} - k_y \cdot S_x \cdot S_z \cdot S_{xy} + k_y \cdot S_y \cdot S_z \cdot S_{xx} + k_z \cdot |S| \cdot S_{xx} \cdot S_{yy} - \\ & -k_z \cdot |S| \cdot S_{xy}^2 - k_z \cdot S_x^2 \cdot S_{yy} + 2 \cdot k_z \cdot S_x \cdot S_y \cdot S_{xy} - k_z \cdot S_y^2 \cdot S_{xx}.\end{aligned}$$

Відомо, що розв'язком є наступні вирази [8]

$$b_x = \frac{\Delta_b}{\Delta}, r_{xx} = \frac{\Delta_{xx}}{\Delta}, r_{xy} = \frac{\Delta_{xy}}{\Delta}, r_{xz} = \frac{\Delta_{xz}}{\Delta}.$$

Інші проекції знаходимо аналогічним чином, прирівнюючи часткові похідні від E_x та E_y до нуля.

Нехай $|S| = n$. Якщо ввести конструкційну матрицю

$$A = \begin{bmatrix} 1 & s_{1x} & s_{1y} & s_{1z} \\ 1 & s_{2x} & s_{2y} & s_{2z} \\ \dots & \dots & \dots & \dots \\ 1 & s_{nx} & s_{ny} & s_{nz} \end{bmatrix}$$

та позначити невідомий вектор через

$$\boldsymbol{\theta}^T = \left[b_x, r_{xx}, r_{xy}, r_{xz} \right]^T,$$

то розв'язок можна записати у більш компактному вигляді. Перепишемо функцію, яку мінімізуємо, через введені позначення

$$E_x = (\mathbf{k}_x - A \cdot \boldsymbol{\theta})^T \cdot (\mathbf{k}_x - A \cdot \boldsymbol{\theta}) = \mathbf{k}_x^T \cdot \mathbf{k}_x - 2\boldsymbol{\theta}^T \cdot A^T \cdot \mathbf{k}_x + \boldsymbol{\theta}^T \cdot A^T \cdot A \cdot \boldsymbol{\theta}, \quad (2.2)$$

де $\mathbf{k}_x = (k_{s_1x}, k_{s_2x}, \dots, k_{s_nx})^T$. Якщо продиференціювати останній вираз по кожному θ_i , то отримаємо систему рівнянь у матричному вигляді

$$-2A^T \cdot \mathbf{k}_x + 2A^T \cdot A \cdot \boldsymbol{\theta} = 0,$$

або

$$(A^T \cdot A) \cdot \boldsymbol{\theta} = A^T \cdot \mathbf{k}_x. \quad (2.3)$$

Звідси маємо оцінку

$$\hat{\boldsymbol{\theta}} = (A^T \cdot A)^{-1} \cdot A^T \cdot \mathbf{k}_x. \quad (2.4)$$

Таким чином,

$$\min_{\boldsymbol{\theta}} E_x(\boldsymbol{\theta}) = E_x(\hat{\boldsymbol{\theta}}).$$

Якщо $A^T \cdot A$ є матрицею з ненульовим визначником, то оцінка найменших квадратів $\hat{\boldsymbol{\theta}}$ єдина. Щоб довести це, припустимо, що $\boldsymbol{\theta}^*$ — довільне фіксоване значення $\boldsymbol{\theta}$. Тоді з (2.2) маємо

$$\begin{aligned} E_x(\hat{\boldsymbol{\theta}}) &= [\mathbf{k}_x - A \cdot \boldsymbol{\theta}^* + A(\boldsymbol{\theta}^* - \boldsymbol{\theta})]^T \cdot [\mathbf{k}_x - A \cdot \boldsymbol{\theta}^* + A(\boldsymbol{\theta}^* - \boldsymbol{\theta})] = \\ &= E_x(\boldsymbol{\theta}^*) + 2(\boldsymbol{\theta}^* - \boldsymbol{\theta})^T \cdot (A^T \cdot \mathbf{k}_x - A^T \cdot A \cdot \boldsymbol{\theta}^*) + (\boldsymbol{\theta}^* - \boldsymbol{\theta})^T \cdot A^T \cdot A \cdot (\boldsymbol{\theta}^* - \boldsymbol{\theta}). \end{aligned}$$

Якщо $\boldsymbol{\theta}^* = \hat{\boldsymbol{\theta}}$, то $E_x(\boldsymbol{\theta}) = E_x(\hat{\boldsymbol{\theta}}) + (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \cdot A^T \cdot A \cdot (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \geq E_x(\hat{\boldsymbol{\theta}})$, оскільки матриця $A^T \cdot A$ невід'ємно визначена. Таким чином, мінімум $E_x(\boldsymbol{\theta})$ дорівнює $E_x(\hat{\boldsymbol{\theta}})$, він досягається при $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$.

Для невиродженої матриці $A^T \cdot A$ рівняння (2.3) однозначно розв'язується, отже, в цьому випадку оцінка найменших квадратів єдина та має вигляд (2.4).

2.2 Статистичні властивості оцінки найменших квадратів

Оцінка $\hat{\theta}$ є незміщеною оцінкою параметра θ . З постановки методу відомо, що $M\mathbf{k}_x = A \cdot \theta$. Отже, $M\hat{\theta} = (A^T \cdot A)^{-1} \cdot A^T \cdot M\mathbf{k}_x = (A^T \cdot A) \cdot A^T \cdot A \cdot \theta$, тобто $M\hat{\theta} = \theta$.

Серед класу оцінок θ^* величини θ , які

1) є незміщеними оцінками,

2) представляють собою лінійні комбінації вихідних даних \mathbf{k}_x ,

за допомогою критерія найменших квадратів можна знайти таку оцінку невідомого параметра θ , що $D\hat{\theta}_j \leq D\theta_j^*$ для будь-якого j . Іншими словами, $\hat{\theta}$ є найбільш точною (або оптимальною) оцінкою θ з усіх можливих, що належать даному класу.

Щоб довести це, насамперед треба знайти вираз для $D\theta_j$. З постановки задачі відомо, що $D\mathbf{k}_x = \sigma^2 \cdot I$. З (2.4) отримуємо

$$D\hat{\theta} = (A^T \cdot A)^{-1} \cdot A^T \cdot D\mathbf{k}_x \cdot A \cdot (A^T \cdot A)^{-1} = \sigma^2 \cdot (A^T \cdot A)^{-1}.$$

Нехай θ^* — якась інша оцінка θ , яка є лінійною комбінацією вихідних даних, тобто $\theta^* = U \cdot \mathbf{k}_x$. Вимагаємо, щоб вона була незміщеною:

$$M\theta^* = U \cdot M\mathbf{k}_x = U \cdot A \cdot \theta = \theta.$$

Остання рівність виконується для всіх θ , отже, можемо стверджувати, що

$$U \cdot A = I.$$

Знайдемо коваріаційну матрицю $D\theta^* = U \cdot Dk_x \cdot U^T = \sigma^2 \cdot U \cdot U^T$. Порівняємо $U \cdot U^T$ з $(A^T \cdot A)^{-1}$. Нехай $K = (A^T \cdot A)^{-1}$. Тоді можемо записати

$$U \cdot U^T = K + (U - K \cdot A^T) \cdot (U - K \cdot A^T)^T.$$

Ця рівність справедлива завдяки тому, що $U \cdot A = I$. Таким чином,

$$D\theta^* = D\hat{\theta} + \sigma^2 (U - K \cdot A^T) \cdot (U - K \cdot A^T)^T.$$

Для $\hat{\theta}$ маємо $U = K \cdot A^T$, тому другий доданок обертається в нуль. Для будь-якої іншої оцінки цей доданок невід'ємний, отже, кожен діагональний елемент матриці $D\theta^*$ не менше відповідного діагонального елемента матриці $D\hat{\theta}$.

Висновки до розділу 2

Пред'явлено розв'язок поставленої задачі методом найменших квадратів. Хоча отримані оцінки мають гарні статистичні властивості, було з'ясовано, що даний метод не дає оптимального розв'язку в даному випадку, адже не вразовує нелінійні обмеження, що накладені на шукані параметри: матриця повороту R має бути ортогональною, а її визначник повинен дорівнювати одиниці. Також за допомогою цього методу не можна оцінити розмітку k , бо це дискретна функція.

3 РОЗВ'ЯЗАННЯ ЗАДАЧІ ІТЕРАТИВНИМ АЛГОРИТМОМ НАЙБЛИЖЧИХ ТОЧОК

У третьому розділі наведено розв'язання поставленої задачі за допомогою ітеративного алгоритму найближчих точок. Досліджено збіжність алгоритму. Перевірено однозначність оцінок, які дає алгоритм, та знайдено їх розподіл. Представлено практичні результати застосування ітеративного алгоритму найближчих точок із зазначенням вхідних даних.

3.1 Ітеративний алгоритм найближчих точок

Ітеративний алгоритм найближчих точок (Iterative Closest Points, ICP) [9] складається з двох операцій, що чергуються. Ініціюється алгоритм одиничною матрицею повороту $R = I$ та нульовим вектором зсуву $b = 0$. Перша ітерація полягає в пошуку такої розмітки $k : S \rightarrow T$, щоб

$$\sum_{s \in S} \|R \cdot s + b - k_s\|^2 \rightarrow \min_k,$$

де R і b фіксовані, k — сюр'єктивне відображення із S в T . Це те ж саме, що шукати такий набір $\{k_s \mid s \in S\}$, щоб

$$\sum_{s \in S} \|Rs + b - k_s\|^2 \rightarrow \min_{k_s}.$$

Запишемо суму явно (nehай множина S має n точок)

$$\|R \cdot s_1 + b - k_{s_1}\|^2 + \dots + \|R \cdot s_n + b - k_{s_n}\|^2 \rightarrow \min_{k_{s_1}, k_{s_2}, \dots, k_{s_n} \in T}.$$

Параметри, що входять у кожний доданок, різні, тож мінімізація всієї суми еквівалентна мінімізації кожного доданку окремо

$$\left\{ \begin{array}{l} \|R \cdot \mathbf{s}_1 + \mathbf{b} - \mathbf{k}_{s_1}\|^2 \rightarrow \min_{\mathbf{k}_{s_1} \in T}, \\ \|R \cdot \mathbf{s}_2 + \mathbf{b} - \mathbf{k}_{s_2}\|^2 \rightarrow \min_{\mathbf{k}_{s_2} \in T}, \\ \vdots \\ \|R \cdot \mathbf{s}_n + \mathbf{b} - \mathbf{k}_{s_n}\|^2 \rightarrow \min_{\mathbf{k}_{s_n} \in T}. \end{array} \right.$$

Таким чином, для кожної точки $\mathbf{s} \in S$ знаходимо точку $\mathbf{t}^* \in T$ таку, щоб відстані між парами $R \cdot \mathbf{s} + \mathbf{b}$ та \mathbf{t}^* для всіх $\mathbf{s} \in S$ були найменшими

$$\|R \cdot \mathbf{s} + \mathbf{b} - \mathbf{t}^*\|^2 = \min_{\mathbf{t} \in T} \|R\mathbf{s} + \mathbf{b} - \mathbf{t}\|^2.$$

Якщо для якоїсь точки \mathbf{s} знайдено декілька таких точок \mathbf{t}^* , то вибираємо будь-яку.

На наступній ітерації виконується пошук повороту R та зсуву \mathbf{b} за поточною розміткою k

$$\sum_{\mathbf{s} \in S} \|R \cdot \mathbf{s} + \mathbf{b} - \mathbf{k}_s\|^2 \rightarrow \min_{R, \mathbf{b}}. \quad (3.1)$$

При цьому матриця $R \in SO(3)$, тобто ортогональна матриця розмірності 3×3 з визначником $+1$. Тому метод найменших квадратів без обмежень не підходить.

Обчислимо зсув \mathbf{b} . Нехай R — відома. Мінімізуємо

$$E(\mathbf{b}) = \sum_{\mathbf{s} \in S} \|R \cdot \mathbf{s} + \mathbf{b} - \mathbf{k}_s\|^2.$$

Можемо знайти оптимальний зсув, взявши градієнт від E по \mathbf{b} та прирівнявши

його до нуля

$$0 = \nabla E(\mathbf{b}) = \sum_{\mathbf{s} \in S} 2(R \cdot \mathbf{s} + \mathbf{b} - \mathbf{k}_s) = 2\mathbf{b} \cdot |S| + 2R \sum_{\mathbf{s} \in S} \mathbf{s} - 2 \sum_{\mathbf{s} \in S} \mathbf{k}_s. \quad (3.2)$$

Позначимо

$$\bar{\mathbf{s}} = \frac{\sum_{\mathbf{s} \in S} \mathbf{s}}{|S|}, \quad \bar{\mathbf{k}}_s = \frac{\sum_{\mathbf{s} \in S} \mathbf{k}_s}{|S|}.$$

Перепишемо рівняння (3.2) в термінах введених позначень

$$\mathbf{b} = \bar{\mathbf{k}}_s - R \cdot \bar{\mathbf{s}}. \quad (3.3)$$

Знайшли оптимальний вектор \mathbf{b} , який однозначно обчислюється за матрицею повороту R . Підставимо його у вираз (3.1)

$$\begin{aligned} \sum_{\mathbf{s} \in S} \|R \cdot \mathbf{s} + \mathbf{b} - \mathbf{k}_s\|^2 &= \sum_{\mathbf{s} \in S} \left\| R \cdot \mathbf{s} + (\bar{\mathbf{k}}_s - R \cdot \bar{\mathbf{s}}) - \mathbf{k}_s \right\|^2 = \\ &= \sum_{\mathbf{s} \in S} \left\| R \cdot (\mathbf{s} - \bar{\mathbf{s}}) - (\mathbf{k}_s - \bar{\mathbf{k}}_s) \right\|^2. \end{aligned}$$

Таким чином, в задачі мінімізації позбулися невідомого вектора \mathbf{b} , тому залишилось знайти лише матрицю R . Нехай $\tilde{\mathbf{s}} = \mathbf{s} - \bar{\mathbf{s}}$, $\tilde{\mathbf{k}}_s = \mathbf{k}_s - \bar{\mathbf{k}}_s$, тоді

$$R = \arg \min_{R \in SO(3)} \sum_{\mathbf{s} \in S} \left\| R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s \right\|^2. \quad (3.4)$$

Спростимо вираз, який мінімізується в (3.4), використавши ортогональність

матриці R : $R^T = R^{-1}$

$$\begin{aligned} \|R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s\|^2 &= (R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s)^T \cdot (R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s) = \\ &= (\tilde{\mathbf{s}}^T \cdot R^T - \tilde{\mathbf{k}}_s^T) \cdot (R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s) = \\ &= \tilde{\mathbf{s}}^T \cdot R^T \cdot R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s^T \cdot \tilde{\mathbf{s}}^T \cdot R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{s}}^T \cdot R^T \cdot \tilde{\mathbf{k}}_s + \tilde{\mathbf{k}}_s^T \cdot \tilde{\mathbf{k}}_s = \\ &= \tilde{\mathbf{s}}^T \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{s}}^T \cdot R^T \cdot \tilde{\mathbf{k}}_s + \tilde{\mathbf{k}}_s^T \cdot \tilde{\mathbf{k}}_s. \end{aligned}$$

Помітимо, що $\tilde{\mathbf{s}}^T \cdot R^T \cdot \tilde{\mathbf{k}}_s$ – це скаляр: $\tilde{\mathbf{s}}^T$ має розмірність 1×3 , R^T має розмірність 3×3 , а $\tilde{\mathbf{k}}_s$ має розмірність 3×1 . Для будь-якого скаляру $a = a^T$, тому

$$\tilde{\mathbf{s}}^T \cdot R^T \cdot \tilde{\mathbf{k}}_s = (\tilde{\mathbf{s}}^T \cdot R^T \cdot \tilde{\mathbf{k}}_s)^T = \tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}}.$$

Маємо

$$\|R \cdot \tilde{\mathbf{s}} - \tilde{\mathbf{k}}_s\|^2 = \tilde{\mathbf{s}}^T \cdot \tilde{\mathbf{s}} - 2\tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}} + \tilde{\mathbf{k}}_s^T \cdot \tilde{\mathbf{k}}_s.$$

Підставимо отриманий вираз у (3.4), відкинувши суми $\tilde{\mathbf{s}}^T \cdot \tilde{\mathbf{s}}$ та $\tilde{\mathbf{k}}_s^T \cdot \tilde{\mathbf{k}}_s$ по всім $s \in S$, бо ці вирази не залежать від R та не впливають на мінімізацію

$$\begin{aligned} R &= \arg \min_{R \in SO(3)} \sum_{s \in S} (\tilde{\mathbf{s}}^T \cdot \tilde{\mathbf{s}} - 2\tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}} + \tilde{\mathbf{k}}_s^T \cdot \tilde{\mathbf{k}}_s) = \\ &= \arg \min_{R \in SO(3)} \left(\sum_{s \in S} \tilde{\mathbf{s}}^T \cdot \tilde{\mathbf{s}} - 2 \sum_{s \in S} \tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}} + \sum_{s \in S} \tilde{\mathbf{k}}_s \cdot \tilde{\mathbf{k}}_s \right) = \\ &= \arg \min_{R \in SO(3)} \left(-2 \sum_{s \in S} \tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}} \right). \end{aligned}$$

Те саме справедливо для константи, на яку множиться сума. Також слід зазнати, що максимізація виразу еквівалентна його мінімізації за знаком «мінус»

$$R = \arg \max_{R \in SO(3)} \sum_{s \in S} \tilde{\mathbf{k}}_s^T \cdot R \cdot \tilde{\mathbf{s}}.$$

Помітимо, що

$$\sum_{\mathbf{s} \in S} \tilde{\mathbf{k}}_{\mathbf{s}}^T \cdot R \cdot \tilde{\mathbf{s}} = \text{tr} \left(\tilde{K}^T \cdot R \cdot \tilde{S} \right),$$

де \tilde{K} і \tilde{S} — це матриці розмірності $3 \times n$ зі стовпцями $\tilde{\mathbf{s}}$ та $\tilde{\mathbf{k}}_{\mathbf{s}}$ відповідно

$$\tilde{K}^T \cdot R \cdot \tilde{S} = \begin{bmatrix} \tilde{\mathbf{k}}_{\mathbf{s}_1}^T \\ \tilde{\mathbf{k}}_{\mathbf{s}_2}^T \\ \vdots \\ \tilde{\mathbf{k}}_{\mathbf{s}_n}^T \end{bmatrix} \cdot R \cdot \begin{bmatrix} \tilde{\mathbf{s}}_1 & \tilde{\mathbf{s}}_2 & \dots & \tilde{\mathbf{s}}_n \end{bmatrix}.$$

Слід квадратної матриці дорівнює сумі її діагональних елементів. Шукаємо таку матрицю R , яка буде максимізувати вираз $\text{tr} \left(\tilde{K}^T \cdot R \cdot \tilde{S} \right)$. Слід матриці має властивість [10]

$$\text{tr} (A \cdot B) = \text{tr} (B \cdot A) \quad (3.5)$$

для будь-яких матриць A та B сумісних розмірностей.

Доведемо цю властивість. Нехай матриця A має розмірність $n \times m$, а матриця $B - m \times n$. Тоді матриця $C = A \cdot B$ — матриця розмірності $n \times n$, що складається з елементів

$$c_{ij} = \sum_{r=1}^m a_{ir} \cdot b_{rj}.$$

Аналогічно, матриця $D = B \cdot A$ має розмірність $m \times m$ і складається з елементів

$$d_{ij} = \sum_{r=1}^n b_{ir} \cdot a_{rj}.$$

Діагональні елементи матриць C та D мають вигляд

$$\begin{cases} c_{ii} = \sum_{r=1}^m a_{ir} \cdot b_{ri}, \\ d_{ii} = \sum_{r=1}^n b_{ir} \cdot a_{ri}. \end{cases}$$

Запишемо слід для добутків

$$\begin{aligned} \operatorname{tr}(A \cdot B) &= \operatorname{tr}(C) = \sum_{i=1}^n c_{ii} = \sum_{i=1}^n \sum_{r=1}^m a_{ir} \cdot b_{ri}, \\ \operatorname{tr}(B \cdot A) &= \operatorname{tr}(D) = \sum_{i=1}^m d_{ii} = \sum_{i=1}^m \sum_{r=1}^n b_{ir} \cdot a_{ri}. \end{aligned}$$

Дві отримані суми однакові. Це означає, що $\operatorname{tr}(A \cdot B) = \operatorname{tr}(B \cdot A)$.

Таким чином,

$$\operatorname{tr}\left(\tilde{K}^T \cdot R \cdot \tilde{S}\right) = \operatorname{tr}\left(\tilde{K}^T \cdot [R \cdot \tilde{S}]\right) = \operatorname{tr}\left(R \cdot \tilde{S} \cdot \tilde{K}^T\right).$$

Позначимо

$$\begin{aligned} X &= \tilde{S} \cdot \tilde{K}^T = \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i}^T = \sum_{i=1}^n \tilde{s}_i \cdot \left(\mathbf{k}_{s_i} - \bar{\mathbf{k}}_s\right)^T = \\ &= \begin{bmatrix} \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i x}^T & \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i y}^T & \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i z}^T \\ \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i y}^T & \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i z}^T & \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i x}^T \\ \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i z}^T & \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i x}^T & \sum_{i=1}^n \tilde{s}_i \cdot \tilde{k}_{s_i y}^T \end{bmatrix}. \end{aligned}$$

Візьмемо сингулярний розклад [11] матриці X

$$X = U \cdot \Sigma \cdot V^T, \quad (3.6)$$

де $U \in \mathbb{R}^{3 \times 3}$ та $V \in \mathbb{R}^{3 \times 3}$ – ортогональні матриці, а $\Sigma \in \mathbb{R}^{3 \times 3}$ – діагональна матриця з невід'ємними елементами, причому $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. Ранг матриці Σ менше трьох, якщо [12]

$$\text{rank } X = \text{rank} (\tilde{S} \cdot \tilde{K}^T) \leq \min \{ \text{rank } \tilde{S}, \text{rank } \tilde{K}^T \} < 3.$$

Таке можливо, якщо $\det X = \det (\tilde{S} \cdot \tilde{K}^T) = 0$ [13].

Розкриємо цей визначник

$$\begin{aligned} \det X &= \\ &= \sum_{i=1}^n \tilde{s}_{ix} \cdot \tilde{k}_{s_i x}^T \cdot \left(\sum_{i=1}^n \tilde{s}_{iy} \cdot \tilde{k}_{s_i y}^T \cdot \sum_{i=1}^n \tilde{s}_{iz} \cdot \tilde{k}_{s_i z}^T - \sum_{i=1}^n \tilde{s}_{iz} \cdot \tilde{k}_{s_i y}^T \cdot \sum_{i=1}^n \tilde{s}_{iy} \cdot \tilde{k}_{s_i z}^T \right) - \\ &\quad - \sum_{i=1}^n \tilde{s}_{iy} \cdot \tilde{k}_{s_i x}^T \cdot \left(\sum_{i=1}^n \tilde{s}_{ix} \cdot \tilde{k}_{s_i y}^T \cdot \sum_{i=1}^n \tilde{s}_{iz} \cdot \tilde{k}_{s_i z}^T - \sum_{i=1}^n \tilde{s}_{iz} \cdot \tilde{k}_{s_i y}^T \cdot \sum_{i=1}^n \tilde{s}_{ix} \cdot \tilde{k}_{s_i z}^T \right) + \\ &\quad + \sum_{i=1}^n \tilde{s}_{iz} \cdot \tilde{k}_{s_i x}^T \cdot \left(\sum_{i=1}^n \tilde{s}_{ix} \cdot \tilde{k}_{s_i y}^T \cdot \sum_{i=1}^n \tilde{s}_{iy} \cdot \tilde{k}_{s_i z}^T - \sum_{i=1}^n \tilde{s}_{ix} \cdot \tilde{k}_{s_i y}^T \cdot \sum_{i=1}^n \tilde{s}_{iy} \cdot \tilde{k}_{s_i z}^T \right) = \\ &= P \left(\tilde{k}_{s_1 x}^T, \tilde{k}_{s_1 y}^T, \tilde{k}_{s_1 z}^T, \dots, \tilde{k}_{s_n x}^T, \tilde{k}_{s_n y}^T, \tilde{k}_{s_n z}^T \right), \end{aligned}$$

де P – многочлен. Покажемо, що ймовірність того, що визначник майже напевно не дорівнює нулю, дорівнює нулю. Випишемо окремо одну змінну мно-

гочлену P

$$\begin{aligned}
 \tilde{\mathbf{k}_{s_1}}^T_x &= \left[R \cdot \mathbf{s}_1 + \mathbf{b} + \boldsymbol{\xi}_{s_1} - \frac{1}{|S|} \sum_{\mathbf{s} \in S} (R \cdot \mathbf{s} + \mathbf{b} + \boldsymbol{\xi}_{\mathbf{s}}) \right]_x^T = \\
 &= \left[R \cdot \mathbf{s}_1 + \mathbf{b} - \frac{1}{|S|} \sum_{\mathbf{s} \in S} (R \cdot \mathbf{s} + \mathbf{b}) \right]_x^T + \left[\boldsymbol{\xi}_{s_1} - \frac{1}{|S|} \sum_{\mathbf{s} \in S} \boldsymbol{\xi}_{\mathbf{s}} \right]_x^T \sim \quad (3.7) \\
 &\sim N \left(\left[R \cdot \mathbf{s}_1 + \mathbf{b} - \frac{1}{|S|} \sum_{\mathbf{s} \in S} (R \cdot \mathbf{s} + \mathbf{b}) \right]_x^T, \frac{|S|-1}{|S|} \cdot \sigma^2 \cdot I \right).
 \end{aligned}$$

Це гаусова випадкова величина, яка, в свою чергу, має неперервний розподіл. Тоді

$$\begin{aligned}
 P(\det X = 0) &= MP \left(\det X = 0 \mid \tilde{\mathbf{k}_{s_1}}_y^T, \tilde{\mathbf{k}_{s_1}}_z^T, \dots, \tilde{\mathbf{k}_{s_n}}_x^T, \tilde{\mathbf{k}_{s_n}}_y^T, \tilde{\mathbf{k}_{s_n}}_z^T \right) = \\
 &= M \mathbb{1} \left\{ F_{\tilde{\mathbf{k}_{s_1}}_y^T, \tilde{\mathbf{k}_{s_1}}_z^T, \dots, \tilde{\mathbf{k}_{s_n}}_x^T, \tilde{\mathbf{k}_{s_n}}_y^T, \tilde{\mathbf{k}_{s_n}}_z^T} (\tilde{\mathbf{k}_{s_1}}_x) = 0 \right\},
 \end{aligned}$$

де $F_{a,b,c,\dots,d}(\cdot)$ — многочлен. Фіксуємо елементи, які стоять в його індексі. Тоді

$$P \left\{ F_{a,b,c,\dots,d} (\tilde{\mathbf{k}_{s_1}}_x) = 0 \right\} = 0.$$

Так як $\tilde{\mathbf{k}_{s_1}}_x$ — випадкова величина з неперервним розподілом, то вона приймає фіксоване значення з нульовою ймовірністю. Таким чином,

$$P(\det X = 0) = M0 = 0,$$

і матриця Σ буде мати ранг 3 з ймовірністю 1, тобто нерівність $\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0$ виконується майже завжди при поточній постановці задачі ідентифікації випадкових точкових відображень. Це означає, що сингулярний розклад матриці X буде єдиним з точністю до перестановки знаків в матрицях U та V .

Підставимо сингулярний розклад в слід

$$\operatorname{tr} \left(R \cdot \tilde{S} \cdot \tilde{K}^T \right) = \operatorname{tr} (R \cdot X) = \operatorname{tr} (R \cdot U \cdot \Sigma \cdot V^T) = \operatorname{tr} (\Sigma \cdot V^T \cdot R \cdot U).$$

Матриці V , R и U — ортогональні, а тому матриця

$$M = V^T \cdot R \cdot U$$

теж ортогональна. Це означає, що $\mathbf{m}_i \cdot \mathbf{m}_i^T = 1$ для кожного рядка \mathbf{m}_i матриці M . Отже, модулі всіх елементів m_{ij} матриці M не перевищують одиниці

$$1 = \mathbf{m}_i \cdot \mathbf{m}_i^T = \sum_{j=1}^3 m_{ij}^2 \Rightarrow m_{ij}^2 \leq 1 \Rightarrow |m_{ij}| \leq 1.$$

За означенням сингулярного розкладу Σ — діагональна матриця з додатними елементами

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 > 0.$$

Тому

$$\operatorname{tr} (\Sigma \cdot M) = \operatorname{tr} \left(\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \cdot \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \right) = \sum_{i=1}^3 \sigma_i \cdot m_{ii} \leq \sum_{i=1}^3 \sigma_i.$$

Отже, слід максимізується при $m_{ii} = 1$, $\forall i \in \{1, 2, 3\}$. Так як M — ортогональна матриця, то вона має бути одиничною

$$I = M = V^T \cdot R \cdot U \Rightarrow V = R \cdot U \Rightarrow R = V \cdot U^T.$$

Помітимо, що зараз R — це ортогональна матриця, але при цьому можливі дві ситуації:

- 1) коли $\det R = \det(V \cdot U^T) = 1$, тобто матриця R діє як поворот;
- 2) коли $\det R = \det(V \cdot U^T) = -1$, тобто матриця R діє як поворот і дзеркальне відображення.

Приклади дзеркальних відображень наведені на рисунку 3.1.

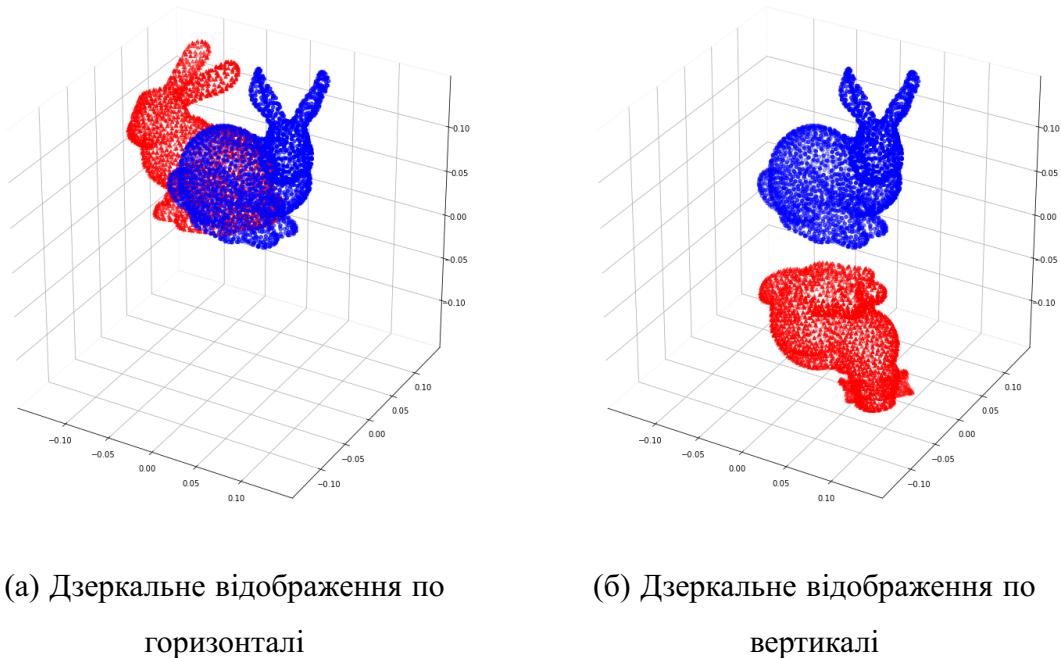


Рисунок 3.1 — Приклади дзеркальних відображень

Припустимо, що $\det(V \cdot U^T) = -1$. Це еквівалентно тому, що

$$\det M = -1.$$

Шукаємо таку матрицю M , яка максимізує вираз

$$\operatorname{tr}(\Sigma \cdot M) = \sigma_1 \cdot m_{11} + \sigma_2 \cdot m_{22} + \sigma_3 \cdot m_{33}.$$

Розглядаємо вектор $(m_{11}, m_{22}, m_{33}) \in \mathbb{R}^3$. Це множина всіх діагоналей ортогональної матриці порядка 3. Альфред Хорн довів [14], що вектор (d_1, \dots, d_n) —

це діагональ матриці повороту порядка n тоді й тільки тоді, коли він лежить у випуклій оболонці точок $(\pm 1, \dots, \pm 1)$, де парне число значень (в тому числі жодне) дорівнює -1 . Для нашого випадку ця теорема приймає такий вигляд: M — матриця повороту тоді й тільки тоді, коли її діагональ (m_{11}, m_{22}, m_{33}) лежить у випуклій оболонці точок $(\pm 1, \pm 1, \pm 1)$, де парна кількість координат (в тому числі жодна) дорівнює -1 . Матриця M — матриця повороту та дзеркального відображення, тому для неї оптимальна діагональ має вигляд $(1, 1, -1)$, коли непарна кількість значень дорівнює -1 , відповідно,

$$\operatorname{tr}(\Sigma \cdot M) = \sigma_1 + \sigma_2 - \sigma_3.$$

Це значення більше будь-якого іншого вектора з $(\pm 1, \pm 1, \pm 1)$ за виключенням $(1, 1, 1)$, бо σ_3 — це найменше сингулярне значення.

Таким чином, якщо $\det(V \cdot U^T) = -1$, то

$$M = V^T \cdot R \cdot U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \Rightarrow R = V \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \cdot U^T.$$

Таким чином, шукана матриця повороту R має вигляд [15]

$$R = V \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(V \cdot U^T) \end{bmatrix} \cdot U^T, \quad (3.8)$$

а оптимальний вектор зсуву обчислюється за формулою (3.3).

Отже, ітеративний алгоритм найближчих точок полягає в почерговому

виконанні двох кроків:

- 1) пошук найкращої розмітки k при фіксованих R і b ;
- 2) пошук матриці повороту R і вектора зсуву b при фіксованій розмітці k ,
- 3) кроки повторюються один за одним, поки не буде досягнений мінімум в (2.1).

3.2 Збіжність алгоритму

Покажемо, що ітеративний алгоритм найближчих точок завжди збігається. Припустимо, що алгоритм нікуди не збігається. Тоді існує така скінченна множина S , матриця повороту R , вектор зсуву b та шум ξ_s , що алгоритм буде виконувати нескінченно багато ітерацій і ніколи не зупиниться.

Нехай алгоритм перебрав усі можливі розмітки, тобто виконав $|T|^{|S|}$ ітерацій, і не зупинився. В цьому випадку на наступному кроці буде вибрана розмітка, яка вже вибиралась на одному з попередніх кроків. Тоді вираз (2.1) або збільшиться (що неможливо за побудовою алгоритму), або не зміниться, і алгоритм закінчить свою роботу. Це означає, що алгоритм завжди збігається.

3.3 Розподіл оцінок

3.3.1 Теорема про незалежність елементів матриці X

Теорема. Асимптотично матриця $X = \tilde{S} \cdot \tilde{K}^T = U \cdot \Sigma \cdot V^T$ має нормальній розподіл, причому елементи матриці незалежні.

Те, що елементи матриці X мають нормальній розподіл, випливає з (3.7). Доведемо незалежність. Для цього знайдемо кореляцію двох довільних

елементів матриці X

$$\text{cor}(x_{ij}, x_{kr}) = \frac{\text{cov}(x_{ij}, x_{kr})}{\sqrt{Dx_{ij} \cdot Dx_{kr}}}, \quad (3.9)$$

де i, k – індекси рядків матриці X , а j, r – індекси стовпців матриці X . Знайдемо окремо вираз у чисельнику

$$\text{cov}(x_{ij}, x_{kr}) = \text{cov} \left[\sum_{t=1}^n \tilde{s}_{ti} \cdot \left(\xi_{s_t i}^T - \frac{1}{n} \sum_{p=1}^n \xi_{s_p j}^T \right), \sum_{q=1}^n \tilde{s}_{qk} \cdot \left(\xi_{s_q r}^T - \frac{1}{n} \sum_{g=1}^n \xi_{s_g r}^T \right) \right].$$

Винесемо суми та константи

$$\text{cov}(x_{ij}, x_{kr}) = \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \cdot \text{cov} \left(\xi_{s_t j}^T - \frac{1}{n} \sum_{p=1}^n \xi_{s_p j}^T, \xi_{s_q r}^T - \frac{1}{n} \sum_{g=1}^n \xi_{s_g r}^T \right).$$

Розкриємо дужки

$$\begin{aligned} \text{cov}(x_{ij}, x_{kr}) &= \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \cdot \text{cov} \left(\xi_{s_t j}^T, \xi_{s_q r}^T \right) + \\ &+ \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \cdot \text{cov} \left(\xi_{s_t j}^T, -\frac{1}{n} \sum_{g=1}^n \xi_{s_g r}^T \right) + \\ &+ \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \cdot \text{cov} \left(-\frac{1}{n} \sum_{p=1}^n \xi_{s_p j}^T, \xi_{s_1 r}^T \right) + \\ &+ \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \cdot \text{cov} \left(-\frac{1}{n} \sum_{p=1}^n \xi_{s_p j}^T, -\frac{1}{n} \sum_{g=1}^n \xi_{s_g r}^T \right) \equiv A + B + C + D, \end{aligned}$$

де через A, B, C, D позначені доданки. Обчислимо кожний з них окремо. По-

чнemo з першого доданку

$$A = \begin{cases} \sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk} \cdot \sigma^2 = \sigma^2 \sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk}, & j = r, \\ 0, & j \neq r. \end{cases}$$

Другий доданок

$$B = -\frac{1}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \sum_{g=1}^n cov \left(\xi_{s_t j}^T, \xi_{s_g r}^T \right) = \begin{cases} -\frac{\sigma^2}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk}, & j = r, \\ 0, & j \neq r. \end{cases}$$

Третій доданок

$$C = -\frac{1}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \sum_{p=1}^n cov \left(\xi_{s_p j}^T, \xi_{s_q r}^T \right) = \begin{cases} -\frac{\sigma^2}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk}, & j = r, \\ 0, & j \neq r. \end{cases}$$

Четвертий доданок

$$D = \frac{1}{n^2} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \sum_{p=1}^n \sum_{g=1}^n cov \left(\xi_{s_p j}^T, \xi_{s_g r}^T \right) = \\ = \begin{cases} \frac{\sigma^2}{n^2} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \cdot n = \frac{\sigma^2}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk}, & j = r, \\ 0, & j \neq r. \end{cases}$$

Бачимо, що другий і третій доданки рівні, а четвертий співпадає з ними з

протилежним знаком. Отже, отримуємо

$$\text{cov}(x_{ij}, x_{kr}) = A + B = \begin{cases} \sigma^2 \sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk} - \frac{\sigma^2}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk}, & j = r, \\ 0, & j \neq r. \end{cases}$$

Знайдемо дисперсію

$$\begin{aligned} Dx_{ij} &= D \left[\sum_{t=1}^n \tilde{s}_{ti} \cdot \left(\xi_{s_t j}^T - \frac{1}{n} \sum_{p=1}^n \xi_{s_p j}^T \right) \right] = \\ &= \sum_{t=1}^n \tilde{s}_{ti}^2 \cdot \left[D\xi_{s_t j}^T + D \left(-\frac{1}{n} \sum_{p=1}^n \xi_{s_p j} \right)^T \right] + 2\text{cov} \left(\xi_{s_t j}^T, -\frac{1}{n} \sum_{p=1}^n \xi_{s_p j}^T \right) = \\ &= \sum_{t=1}^n \tilde{s}_{ti}^2 \cdot \left(\sigma^2 + \frac{\sigma^2}{n} - \frac{2\sigma^2}{n} \right) = \sigma^2 \sum_{t=1}^n \tilde{s}_{ti}^2 \cdot \left(1 - \frac{1}{n} \right). \end{aligned}$$

Аналогічно,

$$Dx_{kr} = \sigma^2 \sum_{t=1}^n \tilde{s}_{tk} \cdot \left(1 - \frac{1}{n} \right).$$

Підставимо отримані вирази в формулу для кореляції (3.9)

$$\begin{aligned} \text{cor}(x_{ij}, x_{kr}) &= \frac{\sigma^2 \left(\sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk} - \frac{1}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \right)}{\left[\sigma^2 \sum_{t=1}^n \tilde{s}_{ti}^2 \cdot \left(1 - \frac{1}{n} \right) \cdot \sigma^2 \sum_{q=1}^n \tilde{s}_{qk}^2 \cdot \left(1 - \frac{1}{n} \right) \right]^{\frac{1}{2}}} \leq \\ &\leq \frac{\left| \sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk} - \frac{1}{n} \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \right|}{\frac{n-1}{n} \cdot \left(\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2 \right)^{\frac{1}{2}}} \leq \frac{\left| \sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk} \right| + \frac{1}{n} \cdot \left| \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \right|}{\frac{n-1}{n} \cdot \left(\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2 \right)^{\frac{1}{2}}}. \end{aligned} \quad (3.10)$$

Поділимо почленно чисельник на знаменник та розглянемо окремо перший

доданок. Використовуємо нерівність Коші-Буняковського [16]

$$\frac{\left| \sum_{t=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{tk} \right|}{\frac{n-1}{n} \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2}} \leq \frac{\sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2 + \sum_{t=1}^n \tilde{s}_{tk}^2}}{\frac{n-1}{n} \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2}} \leq \frac{\sqrt{\sum_{t=1}^n (\tilde{s}_{ti} + \tilde{s}_{tk})^2}}{\frac{n-1}{n} \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2}}.$$

Згідно з нерівністю Мінковського [16] цей вираз не перевищує

$$\frac{\sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2} + \sqrt{\sum_{t=1}^n \tilde{s}_{tk}^2}}{\frac{n-1}{n} \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2}} = \frac{1}{(1 - \frac{1}{n}) \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{tk}^2}} + \frac{1}{(1 - \frac{1}{n}) \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2}}.$$

У вихідній задачі треба знайти функцію перетворення множини точок \tilde{s} . Її прообраз – точкова множина \tilde{S} . Тому чим більше точок міститься в прообразі функції перетворення, тим точніше можна відновити цю функцію. Тобто зростом n сума квадратів координат точок множини \tilde{S} буде тільки збільшуватись. Отже, вираз (3.10) прямує до нуля при $n \rightarrow \infty$.

Розглянемо другий доданок у виразі (3.10)

$$\frac{\left| \sum_{t=1}^n \sum_{q=1}^n \tilde{s}_{ti} \cdot \tilde{s}_{qk} \right|}{(n-1) \cdot \sqrt{\sum_{t=1}^n \tilde{s}_{ti}^2 \sum_{q=1}^n \tilde{s}_{qk}^2}} \leq \frac{1}{n-1} \cdot \frac{\frac{1}{n} \sum_{t=1}^n |\tilde{s}_{ti}| \cdot \frac{1}{n} \sum_{q=1}^n |\tilde{s}_{qk}| \cdot n^2}{\sqrt{\frac{1}{n} \sum_{t=1}^n \tilde{s}_{ti}^2 \cdot \frac{1}{n} \sum_{q=1}^n \tilde{s}_{qk}^2 \cdot n^4}}.$$

Використаємо нерівність між середнім квадратичним та середнім арифмети-

чним. Бачимо, що наступний вираз не менше попереднього

$$\frac{1}{n-1} \cdot \frac{\frac{1}{n} \sum_{t=1}^n |\tilde{s}_{ti}| \cdot \frac{1}{n} \sum_{q=1}^n |\tilde{s}_{qk}|}{\frac{1}{n} \sum_{t=1}^n |\tilde{s}_{ti}| \cdot \frac{1}{n} \sum_{q=1}^n |\tilde{s}_{qk}| \cdot n^2} = \frac{1}{n^2 \cdot (n-1)} \rightarrow 0, n \rightarrow \infty.$$

Доведення теореми завершено. Отже, в асимптотиці елементи матриці X незалежні.

3.3.2 Розподіл матриць U та V в сингулярному розкладі матриці X

Запишемо розподіл ймовірностей елементів матриці X

$$\begin{aligned} \prod_{i=1}^9 \frac{1}{\sqrt{2\pi}} \cdot e^{x_i^2} dx_i &= \frac{1}{(\sqrt{2\pi})^9} \cdot e^{-\frac{1}{2} \sum_{i=1}^9 x_i^2} dx_1 \cdot \dots \cdot dx_9 = \\ &= \frac{1}{(\sqrt{2\pi})^9} \cdot e^{-\frac{1}{2} \cdot \text{tr}(X \cdot X^T)} dX. \end{aligned}$$

Зробимо заміну $X = U \cdot \Sigma \cdot V^T$ та скористаємося властивістю сліду (3.5)

$$\begin{aligned} \frac{1}{(\sqrt{2\pi})^9} \cdot \exp \left\{ -\frac{1}{2} \cdot \text{tr} (U \cdot \Sigma \cdot V^T \cdot V \cdot \Sigma^T \cdot U^T) \right\} dX &= \\ &= \frac{1}{(\sqrt{2\pi})^9} \cdot e^{-\frac{1}{2} \cdot \text{tr} (\Sigma^T \cdot \Sigma)} dX. \end{aligned}$$

Диференціали dX та $dU d\Sigma dV$ пов'язані співвідношенням [17]

$$dX = \prod_{1 \leq i < j \leq 3} (\sigma_i^2 - \sigma_j^2) d\Sigma dU dV.$$

Таким чином, розподіл ймовірностей елементів матриці X має вигляд

$$\frac{1}{(\sqrt{2\pi})^9} \cdot e^{-\frac{1}{2} \cdot \text{tr}(\Sigma^T \cdot \Sigma)} \prod_{1 \leq i < j \leq 3} (\sigma_i - \sigma_j) d\Sigma dU dV.$$

Останній вираз — це добуток трьох розподілів ймовірностей [17]:

$$e^{-\frac{1}{2} \cdot \text{tr}(\Sigma^T \cdot \Sigma)} \prod_{1 \leq i < j \leq 3} (\sigma_i - \sigma_j) d\Sigma$$

для матриці Σ , dU для матриці U та dV для матриці V з точністю до констант. Отже, елементи матриць U та V незалежні та мають рівномірний розподіл.

3.3.3 Розподіл елементів матриці повороту

Шукаємо розподіл елементів матриці $R = V \cdot U^T$, де

$$r_{ij} = \sum_{s=1}^3 v_{is} \cdot u_{sj}^T = \sum_{s=1}^3 r_s.$$

Щільність розподілу випадкових величин v_{is} та u_{sj}^T , які мають рівномірний розподіл на відрізку $[-1, 1]$

$$p_{v_{is}}(x) = p_{u_{sj}^T}(x) = \frac{1}{2} \cdot \mathbb{1}\{x \in [-1, 1]\}.$$

При цьому модулі випадкових величин v_{is} та u_{sj}^T мають стандартний рівномірний розподіл

$$p_{|v_{is}|}(x) = p_{|u_{sj}^T|}(x) = \mathbb{1}\{x \in [0, 1]\}.$$

Знайдемо розподіл величини $|r_s| = |v_{is}| \cdot |u_{sj}^T|$. Візьмемо логарифм від правої і лівої частин останньої рівності

$$\ln |r_s| = \ln |v_{is}| + \ln |u_{sj}^T|.$$

Так як випадкові величини v_{is} і u_{sj}^T — незалежні, то $\ln |v_{is}|$ і $\ln |u_{sj}^T|$ — теж незалежні. Виразимо функції розподілу логарифмів модулів випадкових величин

$$F_{\ln|v_{is}|}(x) = P(\ln |v_{is}| \leq x) = P(|v_{is}| \leq e^x) = F_{|v_{is}|}(e^x).$$

Тоді щільність розподілу

$$p_{\ln|v_{is}|}(x) = [F_{\ln|v_{is}|}(x)]' = e^x \cdot p_{|v_{is}|}(e^x) = e^x \cdot \mathbb{1}\{e^x \in [0, 1]\} = e^x \cdot \mathbb{1}\{x \leq 0\}.$$

Аналогічно,

$$p_{\ln|u_{sj}^T|}(x) = e^x \cdot \mathbb{1}\{x \leq 0\}.$$

Отримуємо щільність розподілу суми незалежних випадкових величин

$$\begin{aligned} p_{\ln|r_s|}(x) &= \int_{-\infty}^{+\infty} p_{\ln|v_{is}|}(x-y) \cdot p_{\ln|u_{sj}^T|}(y) dy = \\ &= \int_{-\infty}^{+\infty} e^{x-y} \cdot \mathbb{1}\{x-y \leq 0\} \cdot e^y \cdot \mathbb{1}\{y \leq 0\} dy = e^x \int_{\min(x,0)}^0 dy = -e^x \cdot \min(x, 0). \end{aligned}$$

Перейдемо від розподілу логарифму випадкової величини $\ln |r_s|$ до розподілу самої випадкової величини $|r_s|$

$$p_{|r_s|}(x) = \frac{1}{x} \cdot p_{\ln|r_s|}(\ln x) = -\frac{1}{x} \cdot e^{\ln x} \cdot \min(\ln x, 0) = -\ln x, \quad 0 \leq x \leq 1.$$

Замінюємо x на $|x|$ в останньому виразі та ділимо його на 2, щоб отримати щільність розподілу випадкової величини r_s . Маємо

$$p_{r_s}(x) = -\frac{1}{2} \cdot \ln|x| \cdot \mathbb{1}\{x \in [-1, 1]\}.$$

Знайдемо математичне сподівання цієї випадкової величини

$$Mr_s = \int_{-\infty}^{+\infty} x \cdot p(x) dx = -\frac{1}{2} \int_{-1}^1 x \cdot \ln|x| dx = -\frac{1}{2} \cdot \ln|x| \cdot \frac{x^2}{2} \Big|_{-1}^1 + \frac{1}{2} \int_{-1}^1 \frac{x^2}{2} \cdot \frac{1}{x} dx = 0.$$

Тоді математичне сподівання елементу знайденої матриці повороту R (3.8) дорівнює $Mr_{ij} = M(r_1 + r_2 + r_3) = 0$. Результат є очікуваним, бо математичне сподівання гаусового шуму (1.1) дорівнює нулю. Тому при малих значеннях шуму поворот точок множини вихідної множини S теж буде малим.

3.4 Простий приклад незадовільної роботи алгоритму

На рисунку 3.2 зображено приклад двох множин, для яких алгоритм не дає очікуваного результату.

Множини представляють собою два ідентичних прямокутник трикутника з катетами, які дорівнюють одиниці, що відрізняються кутом повороту $\theta = \pi$ та зсувом

$$\mathbf{b} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix} \approx \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}.$$

Множину S зображене пунктиром. Це трикутник з вершинами $(0, 0), (0, 1)$ і $(1, 0)$. Множину T , яка складається з точок $\mathbf{t} = R \cdot \mathbf{s} + \mathbf{b}$, зображене суцільною

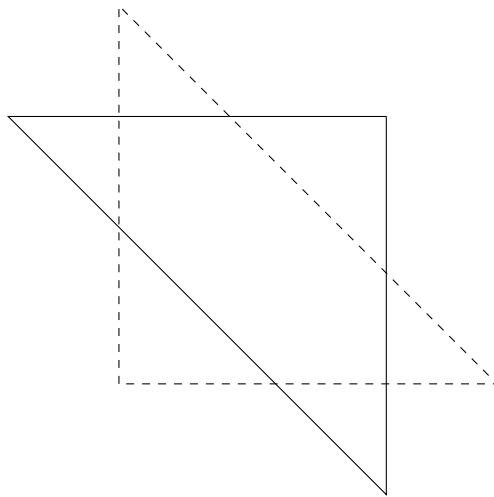


Рисунок 3.2 – Два трикутника, при співставленні яких ICP не дає очікуваного результату

лінією, де s – точки множини S , а R – матриця повороту

$$R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Таким чином, множини S і T мають вигляд

$$S = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, T = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} - 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} - 1 & \frac{\sqrt{2}}{2} \end{bmatrix}.$$

До цих множин був застосований ітеративний алгоритм найближчих точок. Алгоритм зупинився, виконавши дві ітерації, бо розмітка, знайдена на другій ітерації, співпала з розміткою, знайденою на першій ітерації. Отримані взаємні розміщення множин зображені на рисунку 3.3.

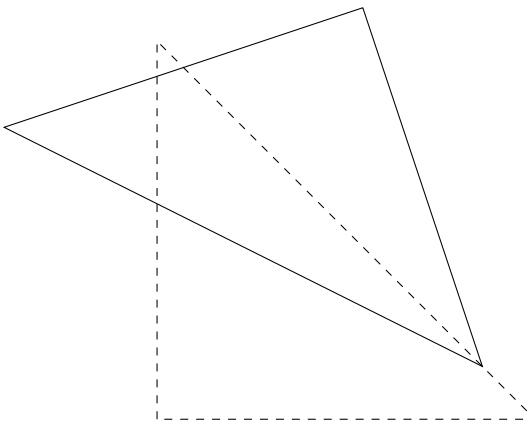


Рисунок 3.3 – Результат ICP для трикутників

Оцінка матриці повороту та вектора зсуву

$$\hat{R} = \begin{bmatrix} 0.9487 & 0.3162 \\ -0.3162 & 0.9487 \end{bmatrix}, \quad \hat{\mathbf{b}} = \begin{bmatrix} 0.0969 \\ 0.1939 \end{bmatrix}.$$

Згідно алгоритму, кут повороту між множинами становить приблизно 18 градусів. Отримані значення параметрів сильно відрізняються від істинних, що видно і на рисунку 3.3. Отже, алгоритм не зміг правильно оцінити кут повороту та зсув множини T відносно множини S .

Правильним результатом були б такі поворот і зсув, коли всі відповідні точки двох трикутників мали б однакові положення, тобто коли трикутники повністю наклалися б один на один. Через те, що найближчі точки не є відповідними, глобальний мінімум не досягається.

3.5 Стенфордський кролик

Для цього прикладу була використана тривимірна модель [18], що складається з 2503 точок, створена Грегом Тюрком та Марком Левоєм в Стенфордському університеті в 1994 році для тестування алгоритмів комп’ютерної графіки, включаючи зменшення кількості полігонів, стиснення даних та згла-

джування поверхонь. Точки вихідної множини повернули відносно осі z на 40 градусів, що становить $\theta = 0.6981$ радіани, та зсунули на 0.1 в бік, протилежний спостерігачеві. При цьому розміри моделі по цій осі становлять приблизно 0.08. Матриця повороту та вектор зсуву мають вигляд

$$R = \begin{bmatrix} 0.766 & -0.6428 & 0 \\ 0.6428 & 0.766 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix}.$$

До точок вихідної множини також доданий гаусів шум з нульовим математичним сподіванням та дисперсією 0.001. Вихідна та отримана множини, тобто вхідні дані, що подаються алгоритму на обробку, зображені на рисунку 3.4. Синім кольором зображена вихідна множина, а червоним — множина, отримана після перетворення вихідної. Назвемо її цільовою.

Ці множини були співставлені за допомогою ітеративного алгоритму найближчих точок. На рисунку 3.5 зображені ітерації роботи алгоритму на моделі кролика зі Стенфордського університету. Обрані ітерації, кратні ступеню двійки, тому що так краще за все помітні зміни.

Видно, що за декілька перших ітерацій алгоритм знайшов правильное розташування вихідної множини відносно цільової, а впродовж наступних ітерацій зміни в положенні вихідної множин, а отже, і зміни в оцінках параметрів, були незначними. Остаточний результат співставлення зображений на рисунку 3.6.

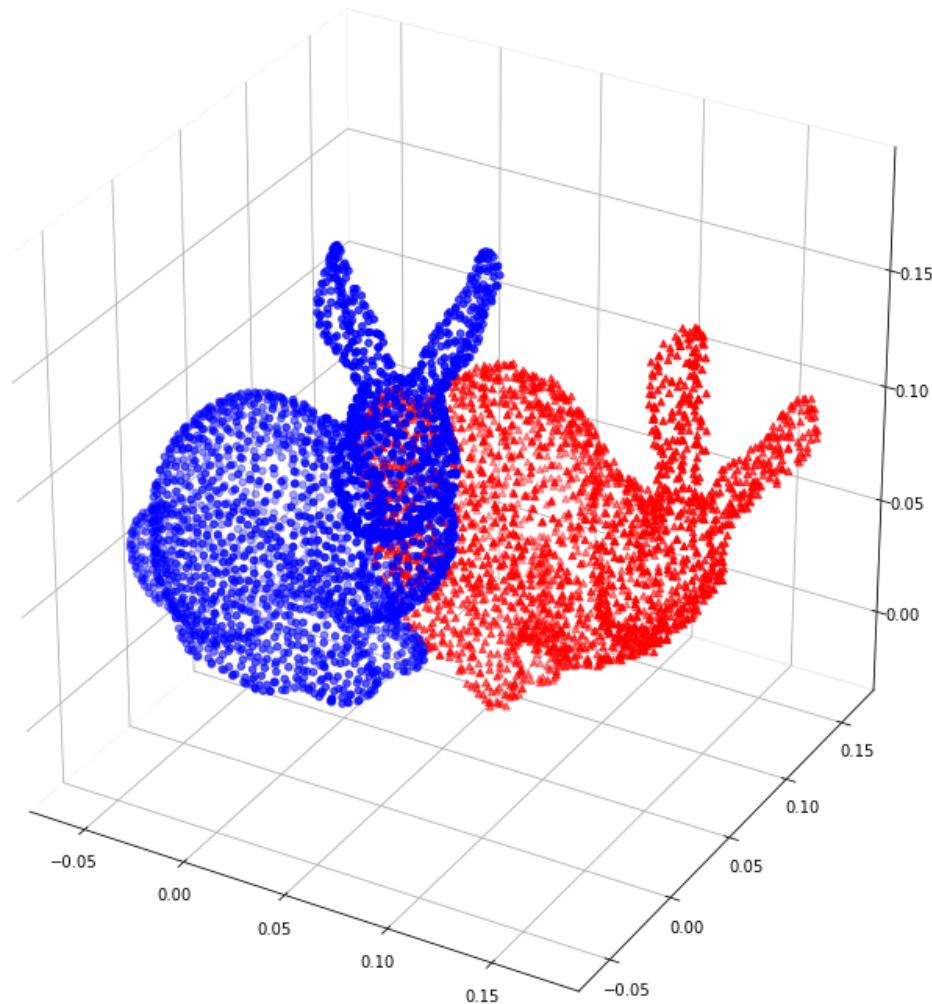


Рисунок 3.4 – Вхідні дані

Оцінка матриці повороту та вектора зсуву має вигляд

$$\hat{R} = \begin{bmatrix} 0.7659 & -0.643 & 0 \\ 0.643 & 0.7659 & 0.0008 \\ -0.0006 & -0.0006 & 1 \end{bmatrix}, \quad \hat{\mathbf{b}} = \begin{bmatrix} -0.0001 \\ 0 \\ 1 \end{bmatrix}.$$

Знайдемо оцінку кута повороту $\hat{\theta} = \arcsin(0.643) = 0.6984$ радіани, що становить

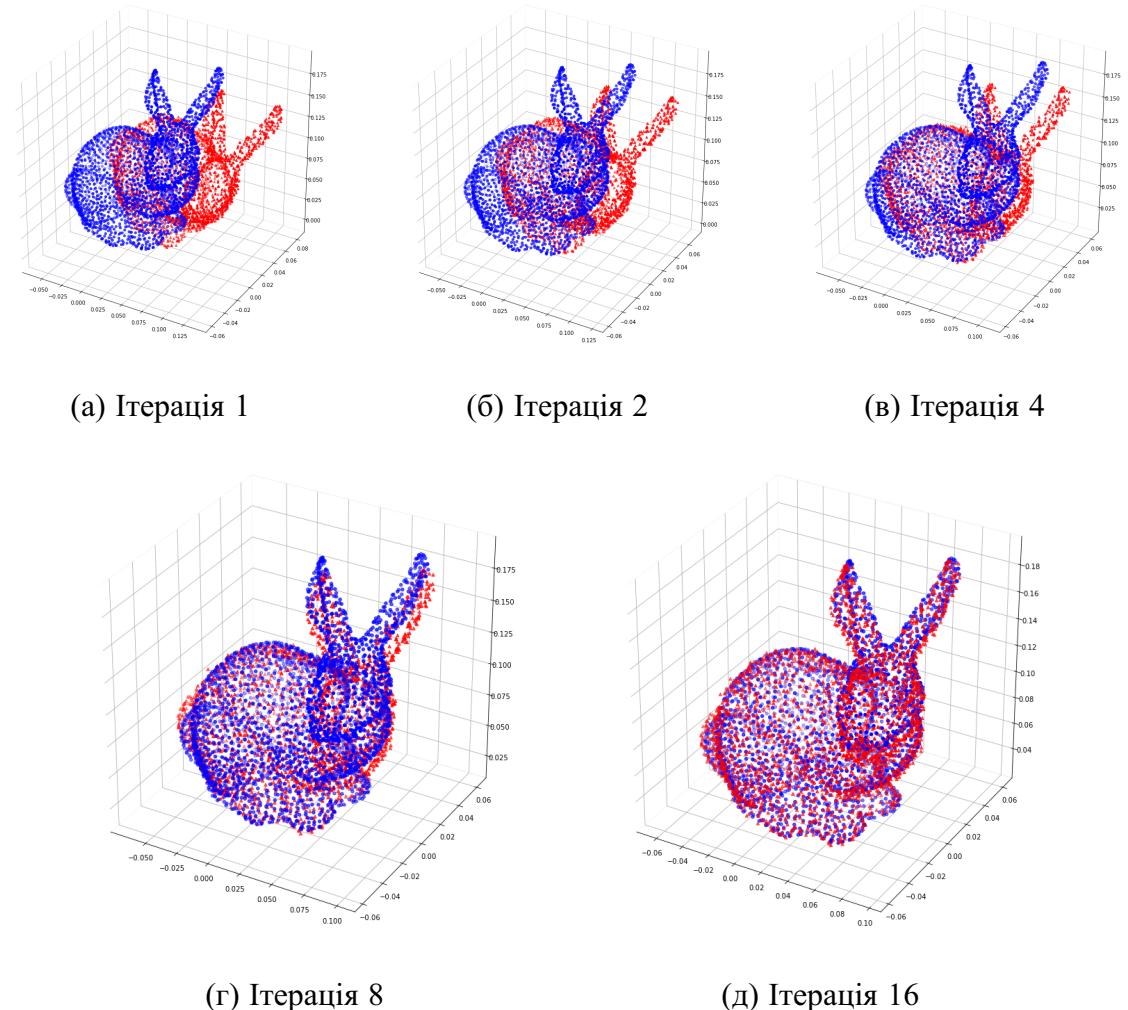


Рисунок 3.5 — Результати проміжних ітерацій

вить 40.0126 градусів. Порахуємо абсолютну похибку оцінки в радіанах

$$\hat{\theta} - \theta = 0.6984 - 0.6981 = 0.0003.$$

Порахуємо відносну похибку

$$\Delta_\theta = \frac{\Delta}{\theta} \cdot 100\% = \frac{0.0003}{0.6981} \cdot 100\% = 0.043\%.$$

Відносна похибка дуже мала, отже, алгоритм відпрацював дійсно добре. Такого результату вдалося досягти за 524 мілісекунди на комп'ютері з процесором

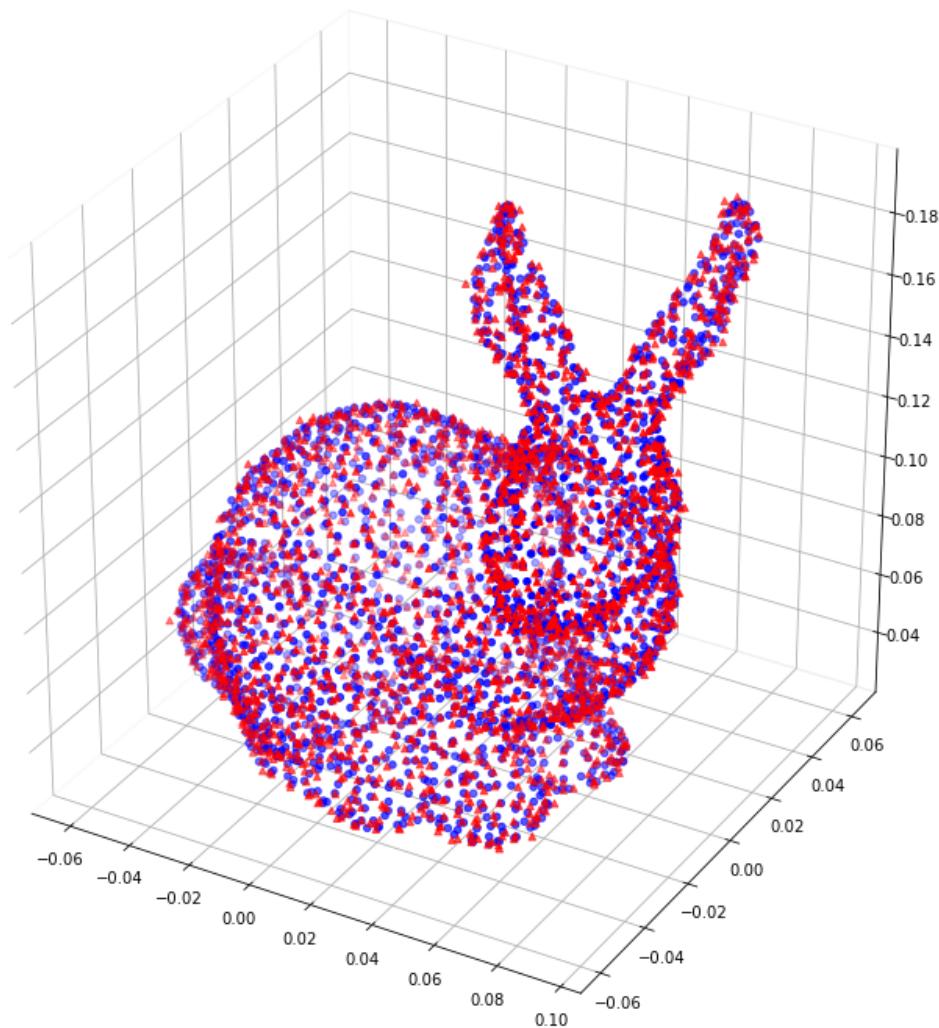


Рисунок 3.6 – Результат роботи алгоритму

Intel Core i5-3317U, виконавши 28 ітерацій алгоритму. Програмне забезпечення будо написано на Python 2.7 спеціально для цієї роботи.

Висновки до розділу 3

Проведено теоретичний опис ітеративного алгоритму найближчих точок для розв'язання задачі, поставленої в першому розділі. Алгоритм базується на обчисленні сингулярного розкладу певної матриці, тому що цей підхід дає змогу отримати матрицю повороту. Виявилося, що в даному випадку сингу-

лярний розклад єдиний з точністю по перестановки знаків. Отже, невідомі параметри оцінюються однозначно. Знайдено розподіл оцінки матриці повороту та доведено, що алгоритм завжди зупиняється за скінченну кількість ітерацій.

Для деяких вхідних даних алгоритм дає дуже добре результати, однак, був знайдений простий приклад, для якого алгоритм не працює задовільно. Таким чином, необхідні подальші дослідження, щоб виявити ще слабкі та сильні сторони алгоритму.

ВИСНОВКИ

В результаті виконання роботи було досліджено задачу співставлення двох точкових множин та ітеративний алгоритм найближчих точок, що її розв'язує. Для розв'язання поставленої задачі було використано метод найменших квадратів, сингулярне розкладення та відомості з теорії ймовірностей. З'ясовано, що звичайний метод найменших квадратів не дає оптимального розв'язку в даному випадку, адже на шукані параметри накладені нелінійні обмеження.

Доведено, що ітеративний алгоритм найближчих точок дає однозначну оцінку шуканих параметрів. Знайдено розподіл оцінки матриці повороту та доведено, що алгоритм завжди зупиняється за скінченну кількість ітерацій.

Було реалізовано демонстративне програмне забезпечення, що за допомогою ітеративного алгоритму найближчих точок точно оцінює шукані параметри.

На даний момент поставлена задача правильно розв'язується тільки повним перебором, який неможливо практично застосовувати через час його роботи для множин, що складаються з великої кількості точок.

ПЕРЕЛІК ПОСИЛАНЬ

- 1 Besl, Paul J. A method for registration of 3-D shapes / Paul J. Besl, Neil D. McKay // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 1992. — February. — Vol. 14. — Pp. 239–256.
- 2 Chen, Yang. Object modelling by registration of multiple range images / Yang Chen, Gérard Medioni // *Image and Vision Computing*. — 1992. — April. — Vol. 10. — Pp. 145–155.
- 3 Assessment of Iterative Closest Point Registration Accuracy for Different Phantom Surfaces Captured by an Optical 3D Sensor in Radiotherapy / Gerald Krell, Nazila Saeid Nezhad, Mathias Walke et al. // *Computational and Mathematical Methods in Medicine*. — 2017. — January. — Vol. 13.
- 4 A Modified Iterative Closest Point Algorithm for 3D Point Cloud Registration / Roberto Marani, Vito Reno, Massimiliano Nitti et al. // *Computer-Aided Civil and Infrastructure Engineering*. — 2016. — Pp. 515–534.
- 5 Robust Single-View Geometry and Motion Reconstruction / Hao Li, Bart Adams, Leonidas J. Guibas, Mark Pauly // *ACM TRANSACTIONS ON GRAPHICS*. — 2009. — Vol. 28. — P. 175.
- 6 Li, Hao. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans / Hao Li, Robert W. Sumner, Mark Pauly // *Computer Graphics Forum (Proc. SGP'08)*. — 2008. — July. — Vol. 27, no. 5.
- 7 Hudson, D.J. Статистика для физиков: лекции по теории вероятностей и элементарной статистике / D.J. Hudson, В.Ф. Грушин, Е.М. Лейкин. — Мир, 1967.
- 8 Воеводин, В.В. Линейная алгебра / В.В. Воеводин. — Издательство "Нauка Главная редакция физико-математической литературы, 1974.

- 9 Zhang, Zhengyou. Iterative point matching for registration of free-form curves and surfaces / Zhengyou Zhang // *International journal of computer vision*. — 1994. — October. — Vol. 13. — Pp. 119–152.
- 10 Fukunaga, K. Introduction to Statistical Pattern Recognition / K. Fukunaga // Computer science and scientific computing. — Elsevier Science, 2013.
- 11 Golub, Gene H. Matrix Computations / Gene H. Golub, Charles F. Van Loan. — Third edition. — The Johns Hopkins University Press, 1996.
- 12 Винберг, Э. Курс алгебры / Э. Винберг. — ЛитРес, 2017.
- 13 Фаддеев, Д.К. Лекции по алгебре / Д.К. Фаддеев. — "Наука,"Глав. ред. физико-математической лит-ры, 1984.
- 14 Horn, Alfred. Doubly Stochastic Matrices and the Diagonal of a Rotation Matrix / Alfred Horn // *American Journal of Mathematics*. — 1954. — July. — Vol. 76. — Pp. 620–630.
- 15 Sorkine-Hornung, Olga. Least-Squares Rigid Motion Using SVD / Olga Sorkine-Hornung, Michael Rabinovich. — 2017. — January. — Technical note.
- 16 Дороговцев, А. Я. Математический анализ. Краткий курс в современном изложении / А. Я. Дороговцев. — Киев: Факт, 2004. — 560 с.
- 17 Mathai, A.M. Jacobians of Matrix Transformations and Functions of Matrix Argument / A.M. Mathai. — World Scientific Pub., 1997.
- 18 Laboratory, Stanford University Computer Graphics. The Stanford Models. — <http://graphics.stanford.edu/data/3Dscanrep/>. — 1994. — [Online; accessed 22-May-2018].

ДОДАТОК А

Лістинг коду програми з реалізацією ітеративного алгоритму найближчих точок та відображенням результатів

```

1  from numpy import genfromtxt, mean, linalg, allclose, diag, matrix, array, cos, sin, pi
2  from numpy.random import normal, random, uniform
3  from matplotlib import pyplot
4  from mpl_toolkits.mplot3d import Axes3D
5  from scipy.spatial import cKDTree
6  from scipy.stats import special_ortho_group
7
8  # Set cardinality
9  n = 100
10 # Generate set T
11 target = random((n, 3))
12 target = array(target)
13 # Generate rotation matrix
14 R = special_ortho_group.rvs(3)
15 R = matrix(R)
16 # Generate shift vector
17 b = random((3, 1))
18 # Generate random noise
19 xi = normal(0, .001, target.shape)
20 # Transform set T into set S
21 source = array((R.dot(data.T)).T + b.T + xi)
22
23 def visualize(source, result):
24     """Plotting sets"""
25     fig = pyplot.figure(figsize=(10,10))
26     ax = Axes3D(fig)
27     ax.xaxis.pane.fill = False
28     ax.yaxis.pane.fill = False
29     ax.zaxis.pane.fill = False
30     ax.scatter(-source[:,0], -source[:,2], source[:,1], c='b', s=20, marker='o')
31     ax.scatter(-result[:,0], -result[:,2], result[:,1], c='r', s=20, marker='^')
32     pyplot.show()
33
34 tree = cKDTree(target)
35 def find_labeling(target, source):
36     """Finding labeling"""
37     return target[tree.query(source)[1]]
38
39 def find_transformation(nearest_neighbours, source):
40     """Finding transformation"""
41     centroid_target = mean(nearest_neighbours, axis=0)
42     centroid_source = mean(source, axis=0)
43     H = ((source - centroid_source).T).dot(nearest_neighbours - centroid_target)
44     U, S, V = linalg.svd(H)
45     R = ((V.T).dot(diag([1, 1, linalg.det((V.T).dot(U.T))])).dot(U.T))

```

```
46     t = centroid_target - R.dot(centroid_source.T).T
47     return R.dot(source.T).T + t
48
49 def icp(target, source, max_iterations=1000):
50     """ICP algorithm"""
51     labelings = []
52     transformations = []
53     labelings.append(find_labeling(target, source))
54     transformations.append(find_transformation(labelings[0], source))
55     i = 1
56     print "Iteration", i
57     visualize(data, transformations[-1])
58     while (len(labelings) < 2 or not allclose(labelings[-1], labelings[-2])) and i < max_iterations:
59         i += 1
60         print "Iteration", i
61         labelings.append(find_labeling(target, transformations[-1]))
62         transformations.append(find_transformation(labelings[-1], source))
63         visualize(data, transformations[-1])
64     return transformations
65
66 # Running the algorithm
67 result = icp(data, source)[-1]
68
69 # Visualizing the result
70 visualize(data, result)
```