МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В. Н. КАРАЗІНА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП'ЮТЕРНОЇ ФІЗИКИ ТА ЕНЕРГЕТИКИ
Кафедра комп'ютерної фізики

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи
за освітньо-кваліфікаційним рівнем магістр

«Система для пошуку розбіжностей між 3D-моделлю і хмарою точок»

Виконав: студент 2 курсу, групи НФ-64
Ребрик В.О.
спеціальність:
105 Прикладна фізика та наноматеріали
освітня програма: комп'ютерна фізика
Керівник: к.т.н., доцент Лісін Д.О.

Рецензент:

Харків – 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут комп'ютерної фізики та енергетики
Кафедра комп'ютерної фізики
Рівень вищої освіти (освітньо-кваліфікаційний рівень) магістр
Спеціальність 105 Прикладна фізика та наноматеріали

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Костянтин НЕМЧЕНКО
підпис

"____"_____2020 року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ (ПРОЕКТ)

Ребрик Владислав Олегович

1.Тема роботи «Система для пошуку розбіжностей між 3D-моделлю і хмарою точок» керівник роботи: Лісін Денис Олександрович, кандидат технічних наук.

затверджені наказом по університету від "10" квітня 2020 року №0210-05/513

2. Строк подання студентом роботи 12.05.2020

3. Перелік питань, які потрібно розробити:

1) Провести аналіз різних підходів для виявлення та оцінки 3D-моделей CAD у хмарі точок ;

2) Обрати та адаптувати алгоритм реєстрації 3D-моделей CAD;

3) Ознайомитися із різними стратегіями вибірки точок, розроблених для проблем комп'ютерного зору із використанням підходу PPF (point pair feature).

4) Розробити програмне забезпечення для виявлення та оцінки 3D-моделей CAD у хмарі точок;

5) Провести тестування розробленого програмного забезпечення.

4. План роботи

№ з/п	Назви етапів роботи
1	Вибір теми, узгодження, а також затвердження її на кафедрі
2	Робота над літературою та іншими джерелами
3	Затвердження теми на кафедрі
4	Написання вступу до роботи та першого розділу
5	Написання другого розділу
6	Написання третього розділу
7	Завершення написання дипломної роботи (висновки та література)
8	Подача науковому керівнику
9	Відгук, рецензія, підготовка до захисту
10	Попередній захист
11	Захист

5.Дата видачі завдання

Студент _____ Ребрик Владислав
підпис

Керівник роботи _____ Лісін Денис Олександрович
підпис

РЕФЕРАТ

Ребрик В.О. Система для пошуку розбіжностей між 3D-моделлю і хмарою точок.

Дипломна робота представлена на 44 сторінках; складається з вступу, 3 розділів та висновків; містить 15 рисунків, список використаної літератури з 37 найменувань.

Мета даної роботи – розробка і програмна реалізація методу перестроювання сітки, застосовного до будь-якої сітки, незалежно від того, чи є трикутники великими, малими, гострими або подовженими. Для цього використовується ефективний алгоритм для визначення глобального розподілу результуючих вершин і нормалей за допомогою відбору проб Пуассона та нормального просторового відбору, щоб задовольнити вимоги відповідності Point pair feature. Для того, щоб щоб збалансувати регулярність та випадковість використовується обмежена релаксація Ллойда.

Об’єктом даної роботи є тривимірні хмари точок.

Предмет роботи – реєстрація 3D-моделей CAD.

Виявлення та оцінка 3D-моделей CAD в хмарі точок мають повсюдне застосування в робототехніці і промислових додатках. Типові моделі CAD розроблені у параметричній формі, де поверхня представлена сукупністю математичних конструкцій. Ці конструкції збираються разом для складання повних моделей. Хоча це стандартний режим роботи у виробництві, інші сфери, такі як комп'ютерна графіка, обробляють і дискретизують ці моделі в 3D-сітках для ефективного використання ресурсів.

КЛЮЧОВІ СЛОВА: ТРИВИМІРНА ХМАРА ТОЧОК, РЕЄСТРАЦІЯ 3D МОДЕЛЕЙ CAD, POINT PAIR FEATURE, ОБМЕЖЕНА РЕЛАКСАЦІЯ ЛЛОЙДА, ВІДБІР ПРОБ ПУАССОНА, 3D МОДЕЛЬ CAD.

РЕФЕРАТ

Ребрик В.О. Система для поиска расхождений между 3D-моделью и облаком точек.

Дипломная работа представлена на 44 страницах; состоит из введения, 3 глав и выводов; содержит 15 рисунков, список использованной литературы из 37 наименований.

Цель данной работы - разработка и программная реализация метода перестроения сетки, применимого к любой сетке, независимо от того, являются ли треугольники большими, маленькими, острыми или удлиненными. Для этого используется эффективный алгоритм для определения глобального распределения результирующих вершин и нормалей с помощью отбора проб Пуассона и нормального пространственного отбора для того, чтобы удовлетворить требованиям Point pair feature. Для того, чтобы чтобы сбалансировать регулярность и случайность используется ограниченная релаксация Ллойда.

Объектом данной работы является трехмерные облака точек.

Предмет работы - регистрация 3D-моделей CAD.

Выявление и оценка 3D-моделей CAD в облаке точек имеют повсеместное применение в робототехнике и промышленных приложениях. Типичные модели CAD разработаны как параметрические формы, где поверхность представлена совокупностью математических конструкций. Эти конструкции собираются вместе для составления полных моделей. Хотя это стандартный режим работы в производстве, другие сферы, такие как компьютерная графика, обрабатывают и дискретизируют эти модели в 3D-сетках для эффективного использования ресурсов.

КЛЮЧЕВЫЕ СЛОВА: ТРЕХМЕРНОЕ ОБЛАКО ТОЧЕК, РЕГИСТРАЦИЯ 3D МОДЕЛЕЙ CAD, POINT PAIR FEATURE, ОГРАНИЧЕННАЯ РЕЛАКСАЦИЯ ЛЛОЙДА, ОТБОР ПРОБ ПУАССОНА, 3D МОДЕЛЬ CAD.

ABSTRACT

Rebrik V.O. A system for finding discrepancies between the 3D model and the point cloud.

The thesis is presented on 44 pages; consists of introduction, 3 chapters and conclusions; contains 15 figures, a list of references from 37 titles.

The goal of this work is to develop and programmatically implement a grid rebuild method applicable to any mesh, regardless of whether the triangles are large, small, sharp or elongated. For this, an efficient algorithm is used to determine the global distribution of the resulting vertices and normals using Poisson sampling and normal space sampling in order to satisfy the requirements of the Point pair feature. In order to balance regularity and randomness, Lloyd's limited relaxation is used.

The object of this work is three-dimensional clouds of points.

The subject of the work is the registration of 3D CAD models.

The identification and evaluation of 3D CAD models in a point cloud is ubiquitous in robotics and industrial applications. Typical CAD models are designed as parametric forms, where the surface is represented by a combination of mathematical constructions. These designs come together to make complete models. Although this is the standard mode of operation in production, other areas, such as computer graphics, process and sample these models in 3D grids for efficient use of resources.

KEYWORDS: 3D POINT CLOUD, REGISTRATION OF 3D CAD MODELS, POINT PAIR FEATURE, LIMITED LOYDE RELAXATION, POISSON SAMPLING, 3D CAD MODEL.

ЗМІСТ

ВСТУП 8
1 АЛГОРИТМИ РЕЄСТРАЦІЇ ХМАР ТОЧОК 11
1.1. Алгоритм Principal Component Analysis (PCA) 15
1.2. Алгоритм Singular Value Decomposition (SVD) 16
1.3. Алгоритм ітеративної найближчої точки (ICP) 17
2 АЛГОРИТМ ВІДБІРУ ВИБІРКИ 24
2.1. Виявлення об’єктів із використанням підходу PPF (Point Pair Feature) 24
3 ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ. 34
3.1. Спектральний аналіз 34
3.2. Реальний набір даних та параметрів 35
3.3. Порівняння із відомими методами та якісні оцінки. 35
3.4. Обчислювавльний час 39
ВИСНОВКИ 40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 41

ВСТУП

Виявлення та оцінка пози камери із шістю ступенями свободи для 3D-моделей CAD має повсюдне застосування в робототехніці та промислових застосуваннях. Типові моделі CAD розроблені як параметричні форми, де поверхня представлена сукупністю математичних конструкцій. Ці конструкції збираються разом для складання повних моделей. Хоча це стандартний режим роботи у виробництві, інші сфери, такі як комп’ютерна графіка, обробляють і дискретизують ці моделі в 3D-сітках для ефективного використання ресурсів. Ця дискретизація називається створенням сітки і зазвичай вводить неоднорідні, анізотропні та подовжені трикутники, а також гострі краї та вершини, скупчені навколо багатих на особливості областей. Сучасні методи сприйняття об'єкта / сцени трактують вхідну модель як добре обумовлену хмару точок. Хоча дослідження видобутку ключових точок чисельні, ці методи працюють на досить добре розподіленій вершинній множині, що охоплює всю поверхню об'єкта. Це робить вищезазначену форму дискретизації не дуже зручною для комп'ютерного зору.

З появою недорогих приладів глибинного зондування, робототехніки, комп'ютерного зору та навколишнього інтелекту, технології перейшли від двовимірної візуалізації та лазерного виявлення зображень та діапазону (LIDAR) до реконструкції реального середовища на основі даних хмарних даних 3D. З одного боку, є структуровані датчики на основі світла, такі як датчик Microsoft Kinect та Asus Xtion, які генерують структуровану хмару точок на регулярній сітці, а з іншого - існує багато датчиків на основі технології Time-Of-Flight, наприклад, камера Softkinetic Depthsense, яка дає неструктуровану хмару точок. Ці хмари точок можуть бути використані безпосередньо для виявлення та розпізнавання об'єктів у навколишньому середовищі, де використовується технологія навколишнього інтелекту, або можуть бути інтегровані для повного відновлення 3D-карти оточення камери. Однак в останньому випадку хмари точок, отримані в різні періоди часу,

потрібно вирівняти - процес, який часто називають реєстрацією. Алгоритми реєстрації здатні оцінити рух робота, обчисливши перетворення, яке оптимально відображає дві точкові хмари, кожна з яких піддається шуму камери.

Метою даної роботи є розробка і програмна реалізація методу перестроювання сітки, застосовного до будь-якої сітки, незалежно від того, чи є трикутники великими, малими, гострими або подовженими. Для цього використовується ефективний алгоритм для визначення глобального розподілу результуючих вершин і нормалей за допомогою відбору проб Пуассона та нормального просторового відбору, щоб задовольнити вимоги відповідності Point pair feature (PPF). Для того, щоб щоб збалансувати регулярність та випадковість використовується обмежена релаксація Ллойда.

Завдання, які були вирішені для досягнення мети роботи:

- Створення технічного завдання;
- Проведення аналізу різних підходів для виявлення та оцінки хмари точок у 3D-моделі CAD;
- Ознайомлення із різними стратегіями вибірки точок, розроблених для проблем комп’ютерного зору із використанням підходу PPF;
- Вибір та адаптування алгоритму реєстрації 3D-моделей CAD та хмари точок;
- Розробка програмного забезпечення для виявлення та оцінки хмари точок у 3D-моделі CAD;
- Тестування програми.

Об’єктом даної роботи є тривимірні хмари точок та сітчаті моделі CAD.

Предмет роботи – реєстрація 3D-моделей CAD.

Актуальність: Виявлення та оцінка 3D-моделей CAD в хмарі точок мають повсюдне застосування в робототехніці і промислових додатках. Типові моделі CAD розроблені як параметричні форми, де поверхня представлена сукупністю математичних конструкцій. Ці конструкції

збираються разом для складання повних моделей. Хоча це є стандартним режимом роботи у виробництві, інші сфери, такі як комп'ютерна графіка, обробляють і дискретизують ці моделі в 3D-сітках для ефективного використання ресурсів. Проблема створення 3D-точкових зображень, сумісних із зором, з нерегулярних, неузгоджених геометрій сітки, на яких необхідно вирішити задачу виявлення та реєстрації об'єктів, є дуже актуальною в наш час.

1 АЛГОРИТМИ РЕЄСТРАЦІЇ ХМАР ТОЧОК

Алгоритми реєстрації використовуються в різних областях знань і програмах, таких як 3D-сканування об'єктів, 3D-картографія, 3D-локалізація та візуальна одометрія або виявлення людського тіла. Більшість цих найсучасніших додатків використовують або простий алгоритм розкладання сингулярного значення (SVD) [1], або алгоритм реєстрації Principal Component Analysis (PCA), або використовують більш вдосконалену ітераційну схему, засновану на алгоритмі ітерації найближчої точки (ICP) [2]. Останнім часом було запропоновано багато варіантів оригінального підходу ICP, найважливішими з яких є нелінійна ICP [3] та узагальнена ICP [4].

Вибір алгоритму, як правило, залежить від декількох важливих характеристик, таких як точність, обчислювальна складність та швидкість збіжності, кожна з яких залежить від напрямку застосування. Більше того, характеристики більшості алгоритмів реєстрації сильно залежать від використовуваних даних, а отже, і від самого середовища. Як результат, складно порівняти дані цих алгоритмів незалежно. Тому в цьому розділі будуть обговорені математичні основи, які є загальними для найбільш широко використовуваних алгоритмів 3D-реєстрації та буде порівняна їх надійність та точність у реальних ситуаціях.

Важливими областями застосування як жорстких, так і нежорстких методологій реєстрації є робототехніка, охорона здоров'я, астрофотографія та інше. У цих застосуваннях спільною метою є визначення положення чи позиції об'єкта стосовно даної точки зору. Тоді як жорсткі перетворення визначені 6 ступенями свободи, нежорсткі перетворення дозволяють збільшити кількість ступенів свободи для того, щоб впоратися з нелінійним або частковим розтягуванням або зменшенням об'єкта [5].

З моменту впровадження недорогих датчиків глибини, таких як камера Microsoft Kinect, у робототехнічному досягненні було досягнуто значного

прогресу в напрямку одночасної локалізації та картографування (SLAM) [6], [7], [8], [9]. Реконструйована тривимірна карта зайнятості сітки представлена набором точкових хмар, які вирівнюються за допомогою реєстрації і можуть використовуватися для таких прийомів, як уникнення перешкод, вивчення карти та управління автономним транспортним засобом [10], [11], [12] . Крім того, інформація про глибину часто поєднується з традиційною камерою RGB [13], [14] для того, щоб значно полегшити реальні проблеми, такі як виявлення об'єктів у зашумлених сценах, відстеження об'єктів та розпізнавання об'єктів [15]. Основна мета робототехнічних додатків - розробити надійний і точний алгоритм, який можна виконати майже в режимі реального часу. Для досягнення цієї мети в даний час багато досліджень зосереджуються на використанні графічних процесорів (GPU) та багатоядерній обробці, що дозволяє виконувати багато обчислювальних завдань протягом одного часового інтервалу на декількох ядрах обробки [16], [17].

Типові застосування нежорстких алгоритмів реєстрації можна знайти в галузі охорони здоров’я, де модель м'якого тіла часто потрібно точно узгоджувати з набором 3D-вимірювань. Застосуваннями є виявлення ракових тканин, виявлення дірок, розпізнавання артефактів тощо [5], [18]. Аналогічно, нежорсткі перетворення використовуються для отримання багатомодального зображення сцени, поєднуючи магнітно-резонансну томографію (МРТ), комп’ютерну томографію (КТ) та позитронно-емісійну томографію обсягів ПЕТ в єдину 3D-модель [5].

До жорсткої реєстрації можна підійти, визначивши функцію вартості, яка представляє поточну помилку, яка вказує на те, наскільки добре перекриваються дві хмарні точки. Ця функція вартості потім мінімізується за допомогою загальних методів оптимізації. Якщо відстань між відповідними точками у кожному хмарі 3D-точок потрібно зменшити, це може бути

спрощено до лінійної задачі мінімізації квадратичних значень, шляхом представлення кожної точки за допомогою однорідних координат.

Однорідне перетворення в трьох вимірах визначається матрицею проективних перетворень 4x4 [19]. Ця матриця використовується для проектування кожної точки декартового простору щодо конкретної точки зору. Оскільки ми використовуємо (рухомі) жорсткі ортонормальні референтні кадри, ми можемо обмежити свої міркування жорсткими перетвореннями. Далі, нехай $\tilde{v}_1 = (x_1, y_1, z_1, 1)^T$ є стандартними однорідними координатами точки в ортонормальній базі, визначеній першою точкою огляду, і нехай $\tilde{v}_2 = (x_2, y_2, z_2, 1)^T$ є стандартними однорідними координатами однієї і тієї ж точки в ортонормальній базі, визначеній другою точкою огляду. Тоді можна виразити $\tilde{v}_2$ відносно основної точки огляду як $T\tilde{v}_1 = \tilde{v}_2$, де T є евклідовою матрицею перетворення, визначеною (1.1).

$$T = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.1}$$

Матриця перетворення (1.1) складається з матриці обертання (1.2) розміром 3x3,

$$R = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ r_{2,1} & r_{2,2} & r_{2,3} \\ r_{3,1} & r_{3,2} & r_{3,3} \end{pmatrix} \tag{1.2}$$

та вектора-стовбця $\vec{t} = (\vec{t}_1, \vec{t}_2, \vec{t}_3)^T$, який представляє трансляцію. Оскільки дев'ять елементів матриці обертання можна генерувати за трьома параметрами (наприклад, кутами Ейлера), можно зробити висновок, що жорстке перетворення має шість ступенів свободи.

Жорстке перетворення визначається лише шістьма ступенями свободи, тоді як доступно багато зашумлених спостережень, тобто точкових координат. Тому кількість параметрів будь-якої функції вартості для цієї задачі значно менша, ніж кількість рівнянь, внаслідок чого виникає погано обумовлена

задача, яка не має точного рішення. Добре відомою методикою отримання прийнятного рішення в такому випадку є мінімізація квадрату залишкової помилки. Такий підхід називається оптимізацією найменших квадратів і часто використовується для вирішення проблем наближення та регресії.

Хоча лінійну задачу з найменшими квадратами можна вирішити аналітично, це часто не стосується нелінійних задач оптимізації найменших квадратів. У цьому випадку може бути використаний ітеративний підхід шляхом ітеративного дослідження простору пошуку всіх можливих рішень у напрямку градієнтного вектора функції вартості. Це проілюстровано на рисунку 1, де функція вартості $f(d)$ алгоритму реєстрації ICP мінімізується ітеративно. Функція вартості в цьому випадку представляє суму квадратів евклідових відстаней, визначених обертанням і трансляцією, між усіма відповідними точками двох позиціями камери для двох хмар точок.

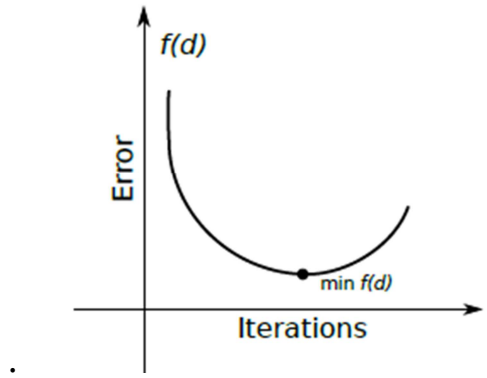


Рис. 1. Підход ICP з найменшими квадратами.

Як жорсткі, так і нежорсткі алгоритми реєстрації можуть бути додатково класифіковані на алгоритми парної реєстрації та багатореєстраційні методи. Алгоритми парної реєстрації обчислюють жорстке перетворення між двома наступними хмарними точками, тоді як процес реєстрації в багатореєстраційних методах враховує кілька хмарних точок, щоб виправити накопичений дрейф, який вводиться методами парної реєстрації.

1.1. Алгоритм Principal Component Analysis (PCA)

PCA часто використовується в методах класифікації та стиснення для проектування даних на новій ортонормальній основі у напрямку найбільшої дисперсії [20]. Напрямок найбільшої дисперсії відповідає найбільшому власному вектору матриці коваріації даних, тоді як величина цієї дисперсії визначається відповідним власним значенням.

Отже, якщо матриця коваріації двох точкових хмар відрізняється від матриці ідентичності, грубу реєстрацію можна отримати, просто вирівнюючи власні вектори їх матриць коваріації. Таке вирівнювання виконується наступним чином.

По-перше, дві точкові хмари відцентровані. Центрування точкової хмари просто відповідає відніманню центроїдних координат від кожної з точкових координат. Центроїд точкової хмари відповідає середній координаті і, таким чином, отримується діленням суми всіх точкових координат на кількість точок у точковому хмарі.

Оскільки реєстрація на основі PCA просто вирівнює напрямки, в яких хмари точок найбільше змінюються, другий крок складається з обчислення матриці коваріації кожної хмари точок. Коваріаційна матриця - це ортогональна матриця 3x3, діагональні значення якої представляють собою відхилення, тоді як значення поза діагональних елементів являють собою коваріації.

По-третє, обчислюються власні вектори обох коваріаційних матриць. Найбільший власний вектор - це вектор у напрямку найбільшої дисперсії хмари 3D-точок, а отже, він представляє орієнтацію точкової хмари. Далі нехай A - матриця коваріації, $\vec{v}$ - власний вектор цієї матриці, і нехай λ буде відповідним власним значенням. Задача про власне значення задачі декомпозиції визначається як:

$$A\tilde{x} = \lambda\tilde{x} \tag{1.3}$$

Із подальшим зведенням до

$$\tilde{x}(A - \lambda I) = 0. \quad (1.4)$$

Очевидно, що (1.4) має ненульовий розв'язок якщо $A - \lambda I$ є сингулярним та, відповідно, якщо його визначник дорівнює нулю:

$$\det(A - \lambda I) = 0. \quad (1.5)$$

Власні значення можна просто отримати шляхом розв'язування (5), тоді як відповідні власні вектори отримуються шляхом підстановки власних значень у (3).

Як тільки власні вектори будуть відомі для кожної хмари точок, реєстрація досягається вирівнюванням цих векторів. Далі нехай матриця T_t^y являє собою перетворення, яке б вирівняло найбільший власний вектор t хмари цільової точки з віссю y. Нехай матриця T_y^s являє собою перетворення, яке вирівняло б найбільший власний вектор s хмари вихідної точки з віссю y. Тоді остаточну матрицю перетворення T_t^s, яка вирівнює вхідну хмару точок з цільовою хмарою точок, можна легко отримати, як це показано на Рис. 2.

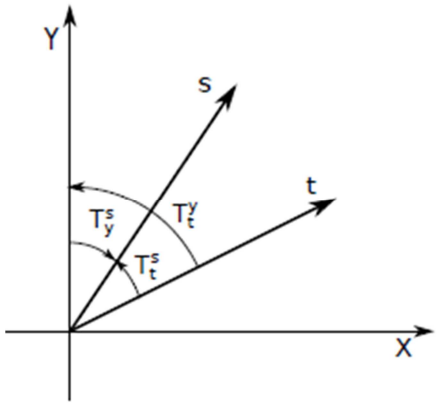


Рис. 2. Вирівнювання вхідної та цільової хмар точок у PCA.

Нарешті, центроїд цільових даних додається до кожної з перетворених координат для трансляції вирівняної хмари точок, таким чином, що її центр відповідає центру хмари цільової точки.

1.2. Алгоритм Singular Value Decomposition (SVD)

Реєстрація на основі PCA просто вирівнює напрямки найбільшої дисперсії кожної хмари точок, і, отже, не мінімізує евклідову відстань між відповідними точками наборів даних. Отже, такий підхід дуже чутливий до шуму, і добре працює лише в тому випадку, якщо кожна хмара точок приблизно нормально розподілена.

Однак, якщо наявні точкові відповідності між двома точковими хмарами, більш надійним підходом було б безпосередньо мінімізувати суму евклідових відстаней між цими точками. Це відповідає лінійній задачі з найменшими квадратами, яку можна вирішити грубо за допомогою методу SVD [1].

Виходячи з точкових відповідностей, матрицю перехресної кореляції M між двома центральними точковими хмарами можна обчислити, після чого декомпозицію власного значення отримують наступним чином:

$$M = USV^T. \tag{1.6}$$

Оптимальне рішення задачі з найменшими квадратами визначається матрицею обертання R як:

$$R_t^s = UV^T, \tag{1.7}$$

А трансляцію з цільової хмари точок в вхідну хмару точок визначається:

$$\tilde{t} = \tilde{c}_s - R_t^s \tilde{c}_t. \tag{1.8}$$

1.3. Алгоритм ітеративної найближчої точки (ICP)

Тоді як алгоритм SVD безпосередньо вирішує задачу найменших квадратів, тим самим передбачаючи ідеальні дані, Беслі та Мак. Кей [2] ввели метод, який ітераційно ігнорує сторонні шуми, щоб покращити попередню оцінку параметрів обертання та трансляції. Їх метод називається "ICP" і його концептуально проілюстровано на Рис. 3.

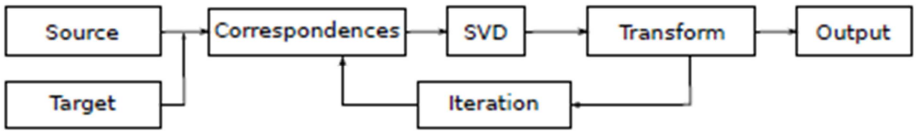


Рис. 3. Загальна схема ICP

Вхід алгоритму ICP складається з хмари точок джерела та цільової хмари точок. Відповідність точок між цими хмарами точок визначається на основі підходу найближчого сусіда або більш детальної схеми з використанням геометричних особливостей або кольорової інформації. SVD, як пояснено в попередньому розділі, використовується для отримання початкової оцінки матриці афінного перетворення, яка вирівнює обидві хмари точок. Після трансформації весь цей процес повторюється шляхом видалення залишків та переосмислення точкових відповідностей.

Два широко використовувані варіанти ICP - це алгоритми ICP «точка-точка» та ICP «точка-поверхня». Ці підходи відрізняються лише своїм визначенням точкових відповідностей.

1) ICP «точка-точка»: алгоритм був спочатку описаний в [2] і просто отримує відповідність точок шляхом пошуку найближчої сусідньої цільової точки $\vec{q}_i$ до точки $\vec{p}_j$ у вхідній хмарі точок. Відповідність найближчого сусіда визначається з точки зору евклідової метрики відстані:

$$i = arg\ \min_j \left\| \vec{p}_j - \vec{q}_i \right\|^2, \tag{1.9}$$

де $j \in [0,1,\dots,N]$, а N представляє собою кількість точок у цільовій хмарі точок.

Подібно до підходу SVD, параметри обертання R та трансляції $\vec{t}$ оцінюються за допомогою мінімізації квадратів відстаней між цими відповідними парами:

$$\hat{R}, \hat{\vec{t}} = arg\ \min_{R,\vec{t}} \sum_{i=1}^{N} \left\| \left(R\vec{p}_i + \vec{t}\right) - \vec{q}_i \right\|^2. \tag{1.10}$$

Потім ICP ітеративно вирішує (1.9) та (1.10), щоб покращити оцінки попередніх ітерацій. Це проілюстровано на Рис. 4, де поверхня s вирівняна до поверхні t після n Ітерацій ICP.

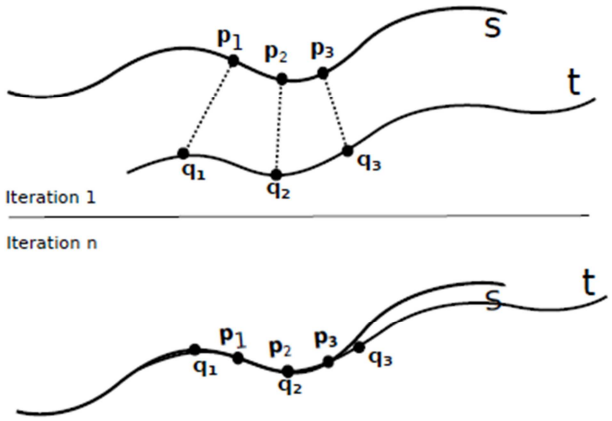


Рис. 4. Вирівнювання ICP, що основується на підході «точка-точка».

2) ICP "точка-поверхня": Завдяки спрощеному визначенню відповідності точок, алгоритм ICP "точка-точка", запропонований [24], є дуже чутливим до шуму. Замість того, щоб безпосередньо знайти найближчого сусіда до точки-джерела $\vec{p}_j$ у цільовій хмарі точок, можна взяти до уваги локальне сусідство відповідного кандидата $\vec{q}_i$, щоб зменшити чутливість алгоритму до шуму.

Алгоритм ICP "точка-поверхня" передбачає, що локальне сусідство точки в хмарі точок є копланарним. Потім ця локальна поверхня може бути визначена за вектором нормалі $\vec{n}$, який отримують як найменший власний вектор матриці коваріації точок, які оточують відповідного кандидата $\vec{q}_i$.

Замість прямої мінімізації евклідової відстані між відповідними точками ми можемо мінімізувати скалярну проекцію цієї відстані на площинну поверхню, визначену вектором нормалі $\vec{n}$:

$$\hat{R}, \hat{\vec{t}} = arg\ \min_{R,\vec{t}} \sum_{i=1}^{N} \left\| \left((R\vec{p}_i + \vec{t}) - \vec{q}_i \right) \vec{n}_i \right\| \quad . \tag{1.11}$$

Простіше це проілюстровано на Рис.5.

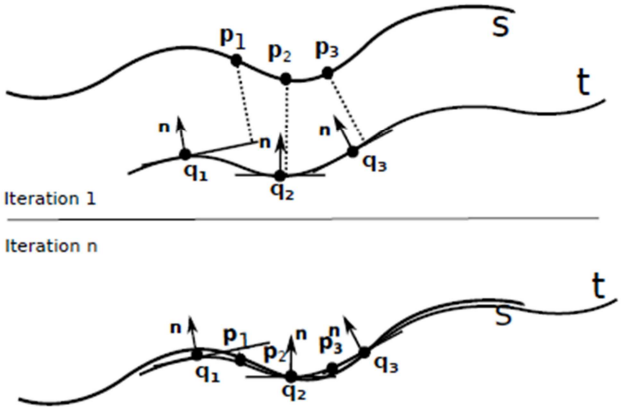


Рис. 5. Вирівнювання ICP, що основується на підході «точка-поверхня».

3) ICP нелінійний: і підхід ICP «точка-точка» і «точка-поверхня» визначав диференційовану, опуклу функцію витрат у квадраті, в результаті чого виникає проста проблема лінійної оптимізації лінійних найменших квадратів, відома як L2-оптимізація, яку можна вирішити чисельно за допомогою SVD. Однак L2-оптимізація, як відомо, дуже чутлива до шуму, тому що похибки зводяться у квадрат. Підхід, що вирішує цю проблему, відомий як L1-оптимізація, де сума абсолютних значень похибок мінімізується замість квадрата. Однак функція витрат L1 не диференційована за походженням, що ускладнює отримання оптимального рішення.

Як компроміс між оптимізацією L1 та L2 може використовуватися так звана функція витрат Хубера, як показано в (7). Функція втрат Хубера є квадратичною для малих значень і, таким чином, поводиться як проблема L2 у цих випадках. Однак для великих значень функція витрат стає лінійною і, отже, поводиться як функція витрат L1. На Рис.6 показано різницю між функцією Губера-Лосса (зелена крива), синя крива показує квадратичну функцію L2. Більше того, функція витрат Хубера є плавною та диференційованою, що дозволяє використовувати традиційні методи оптимізації обчислень для ефективного обходу простору пошуку.

$$e(n) = \begin{cases} \frac{n^2}{2}, & \text{якщо } |n| \le k \\ k|n| - \frac{k^2}{2}, & \text{якщо } |n| \ge k \end{cases}, \qquad (1.12)$$

де k - емпірично визначений поріг, а n - міра відстані.

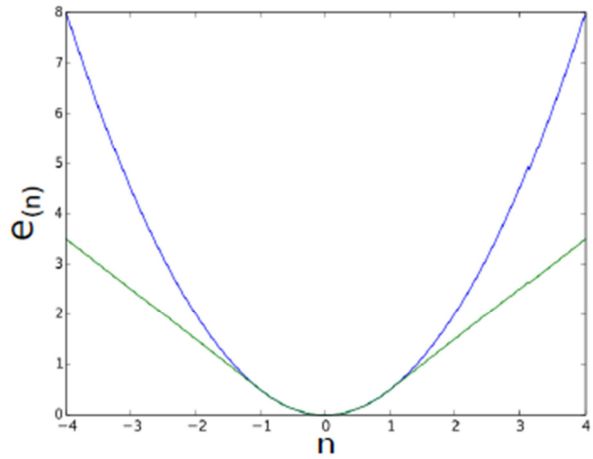


Рис.6. Функція витрат Хубера.

Нелінійний алгоритм ICP використовує функцію витрат Хубера замість наївної квадратичної функції витрат для зменшення впливу шуму:

$$\hat{R}, \hat{\vec{t}} = arg\ \min_{R,\vec{t}} \sum_{i=1}^{N} e^2(n), \tag{1.13}$$

де

$$n = \left\| (R\vec{p} + \vec{t}) - \vec{q} \right\|. \tag{1.14}$$

Для отримання оптимальних оцінок $\hat{R}, \hat{\vec{t}}$ в (13) використовується алгоритм Левенберга-Маркарда (LMA) [3]. Метод LMA - це ітеративна процедура, схожа на добре відомий метод градієнтного спуску та алгоритми Гаусса-Ньютона, яка дозволяє швидко знайти локальний мінімум у нелінійних функціях.

4) Узагальнена ICP: Основним недоліком традиційного алгоритму ICP «точка-точка» є те, що він передбачає, що хмара точок джерела взята з відомої геометричної поверхні, а не отримується за допомогою зашумлених вимірювань. Однак через помилки дискретизації, як правило, неможливо отримати ідеальне співвідношення «точка-точка» навіть після повного зближення алгоритму. Алгоритм ICP «точка-поверхня» послаблює це обмеження, дозволяючи зміщення точок уздовж поверхні, щоб впоратися з різницевими відмінностями. Однак цей підхід все ще передбачає, що хмара

вихідних точок являє собою дискретизований набір зразків відомої геометричної моделі поверхні, оскільки зміщення вздовж поверхні дозволено лише в цільовій хмарі точок.

Для вирішення цього питання Segal та ін. [4] запропонував алгоритм узагальненого ICP (GICP), який виконує порівняння площини до площини. Вони ввели ймовірнісну інтерпретацію процесу мінімізації таким чином, що структурна інформація як з хмари вхідних точки, так і з цільової хмари може легко включатися в алгоритм оптимізації. Більше того, вони показали, що традиційні алгоритми ICP «точка-точка» та «точка-поверхня» - це лише окремі випадки узагальненої структури ICP.

Замість того, щоб припускати, що хмара вхідних точок отримана з відомої геометричної поверхні, Segal та ін. припустили, що і хмара вхідних точок $A = \{\vec{a}_i\}$, і хмара цільових точок $B = \{\vec{b}_i\}$ складаються з випадкових вибірок з невідомої хмари точок $\hat{A} = \{\vec{\hat{a}}_i\}$ та $\hat{B} = \{\vec{\hat{b}}_i\}$. Для невідомих точкових хмар $\hat{A}$ і $\hat{B}$ існують досконалі відповідності, тоді як це не стосується спостережуваних хмар точок A і B, оскільки кожна точка $\vec{a}_i$ і $\vec{b}_i$ вважається вибіркою з нормального розподілу, такого що $\vec{a}_i \sim \mathrm{N}(\vec{a}_i, C_i^a)$ і $\vec{b}_i \sim \mathrm{N}(\vec{b}_i, C_i^b)$. Матриці коваріації C_i^a і та C_i^b невідомі. Якщо обидві хмари точок складалися б детермінованих зразків з відомих геометричних моделей, то обидві матриці коваріації були б нульовими, так що тоді $A = \hat{A}$ і $B = \hat{B}$.

Далі нехай T - матриця афінного перетворення, яка визначає відображення від $\hat{A}$ до $\hat{B}$ таким, що $\vec{b}_i = T\vec{a}_i$. Якщо T було б відомо, можливо було б застосувати це перетворення до спостережуваної вхідної хмари точок A і визначити помилку, яку слід мінімізувати як $d_i^T = \vec{b}_i - T\vec{a}_i$. Тому що і $\vec{a}_i$, і $\vec{b}_i$ мають бути виведеними з незалежних нормальних розподілів, d_i^T, що є лінійною комбінацією $\vec{a}_i$, і $\vec{b}_i$, також виводиться з нормального розподілу:

$$d_i^T \sim N\left(\vec{\hat{b}}_i - T\hat{a}_i, C_i^B + TC_i^A T^T\right) \qquad (1.15)$$

$$= N\left(0, C_i^B + TC_i^A T^T\right) \quad (1.16)$$

Оптимальна матриця перетворення $\hat{T}$ – тоді є перетворенням, яке мінімізує логарифм функції правдоподібності спостережуваних помилок d_i:

$$\hat{T} = arg\min_T \sum_i \lg\left(p\left(d_i^T\right)\right)$$

$$= arg\min_T \sum_i d_i^{TT}(C_i^B + TC_i^A T^T)^{-1} d_i^T \quad (1.17)$$

Segal та ін. показали, що ICP «точка-точка» та «точка-площина» - це конкретні випадки (1.17), що відрізняються лише вибором коваріаційних матриць C_i^A і та C_i^B; Якщо вважається, що хмара вихідних точок отримана з відомої геометричної поверхні, $C_i^A = 0$. Крім того, якщо точкам у цільовій хмарі точок дозволено три ступені свободи, то $C_i^B = I$; У цьому випадку (17) зводиться до:

$$\hat{T} = arg\min_T \sum_i d_i^{TT} d_i^T = arg\min_T \sum_i \left\| d_i^T \right\|^2, \quad (1.18)$$

що справді є саме проблемою оптимізації, яка вирішується традиційним алгоритмом ICP «точка-точка». Аналогічно, C_i^A та C_i^B можуть бути обрані таким чином, що отримання максимальної оцінки правдоподібності відповідає мінімізації відстаней від точки до точки або від точки до площини між обома хмарами точок.

2 АЛГОРИТМ ВІДБІРУ ВИБІРКИ

Виявлення та оцінка 3D-моделей CAD у хмарі точок мають повсюдне застосування в робототехніці та промислових додатках. Типові моделі CAD розроблені як параметричні форми, де поверхня представлена сукупністю математичних конструкцій. Ці конструкції збираються разом для складання повних моделей. Хоча це стандартний режим роботи у виробництві, інші сфери, такі як комп’ютерна графіка, обробляють і дискретизують ці моделі в 3D-сітках для ефективного використання ресурсів. Ця дискретизація називається генерацією сітки [21] і зазвичай вводить неоднорідні, анізотропні та подовжені трикутники, а також гострі краї та вершини, скупчені навколо багатих на особливості областей.

Сучасні методи сприйняття об'єкта / сцени трактують вхідну модель як добре обумовлену хмару точок [2], [3], [4], [5]. Хоча дослідження видобутку ключових точок численні [6], [7], [5], ці методи вже працюють на досить добре розподіленій вершинній множині, що охоплює всю поверхню об'єкта. Це робить вищезазначену форму дискретизації дуже недоброю для комп'ютерного зору.

У цій роботі вирішується особлива проблема створення 3D-точкових зображень, сумісних із зором, з нерегулярних, неузгоджених геометрій сітки, на яких необхідно вирішити задачу виявлення та реєстрації об'єктів. Розвинуто алгоритм точкового перекомпонування, шляхом додавання до нього геометричної структури хешування Дроста та ін. [2] та Бірдала та Іліц [3].

2.1. Виявлення об’єктів із використанням підходу PPF (Point Pair Feature)

Із заданою вхідною сіткою, модель поверхні описується як база даних з точкових пар $F = \{f_{ij}\}$, така що:

$$f_{ij} = \left(\|d\|_2, \angle(n_i, d), \angle\left(n_j, d\right), \angle\left(n_i, n_j\right)\right), \quad (2.1)$$

де d - вектор різниці, n_i і n_j - поверхневі нормали в точках m_iі m_j. $\|.\|$ - евклідова відстань між точковими парами. i посилається на опорну точку, а j позначає будь-яку іншу парну точку. База даних підтримується як перевернутий файл **H**, де самоподібні PPF 3D-моделі кластеруються, щоб належати однієї і тієї ж корзині. У корзинах зберігаються індекси опорних точок, а також локальне посилання точкової пари. Цей опис є напівглобальним, в тому сенсі, що опорна точка описується набором інших точок, з якими вона сполучена. Створення ключа бази даних із заданої функції здійснюється квантуванням:

$$\hat{f}_{ij} = \left(\tau\|d\|_2, \alpha\angle(n_i, d), \alpha\angle\left(n_j, d\right), \alpha\angle\left(n_i, n_j\right)\right), \quad (2.2)$$

де τ і α - відносні коефіцієнти квантування відстаней і кутів відповідно. З рівняння (2.2) видно, що для заповнення хеш-таблиці якомога рівномірно та для збільшення розрізняющої сили необхідна рівномірність вибірки відстаней і кутів. Це мотивує до розробки індивідуального алгоритму перекомпонування точок.

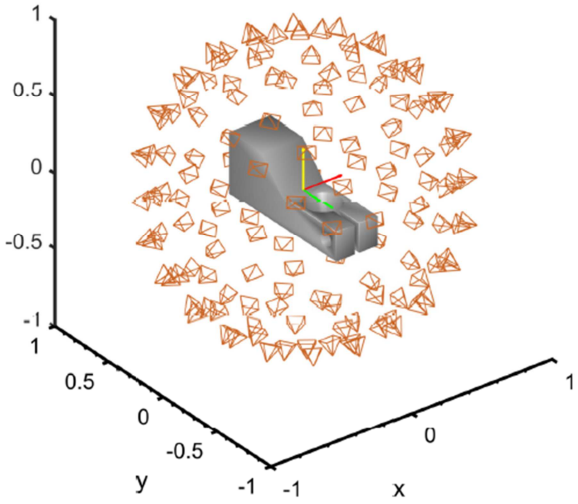


Рис.7. Генерація камер

Із наданою об'єктною моделлю, ми генеруємо набір синтетичних камер на кулі, яка окружає об'єкт, як на рис. 7, і випускає промені для кожного

огляду, від центру камери до об’єкту. Кожен промінь перетинає сітку і створює точку і нормаль на перетині. Потім ми збираємо всі ці зразки і обрізаємо їх за допомогою ефективної сітки вокселів із обмеженнями Пуассона. Нарешті, з релаксацією Лліода відбір проб набуває збалансованої регулярності. Вся процедура узагальнена у наступному алгоритмі:

Require: Mesh $(\{\mathbf{T}_i\}, \{\mathbf{M}_i\})$, Relative sample threshold τ, Weight threshold τ_w and # Max Samples N_m
Ensure: Sampled point cloud $\mathbf{D}$ with normals $\mathbf{N_D}$
  Normalize and Align $\mathbf{M}$ to canonical frame as in Section IIIa
  Generate camera poses : $\mathbf{V} = \{V_1 ... V_N\}$
  $(\mathbf{S}, \mathbf{N}) \leftarrow []$
  for $V_i \in \mathbf{V}$ **do** ▷ Sample pool generation
    Find best $\mathbf{K}$ via Eq. (2)
    Shoot rays: $\mathbf{r}_i(\lambda) \leftarrow \mathbf{o} + \lambda \mathbf{d}_i$
    $\{(\mathbf{v}^i_{int}, \mathbf{n}^i_{int})\} \leftarrow \{\mathbf{r}_i\} \cap (\mathbf{T}, \mathbf{M})$
    Compute $\mathbf{w}(\phi)$ via Eq. (3)
    Exclude vertices with $\mathbf{w}(\phi) < \tau_w$
    $(\mathbf{S}, \mathbf{N}) = (\mathbf{S}, \mathbf{N}) \cup (\{\mathbf{v}^i_{int}\}, \{\mathbf{n}^i_{int}\})$
  end for
  Randomize$(\mathbf{S}, \mathbf{N})$ as explained in Section IIIg
  Compute CDF from Section IIIi.
  $R_d \leftarrow diameter(\mathbf{S})$
  $(\mathbf{D}, \mathbf{N}_D) \leftarrow []$
  $cnt \leftarrow 0$
  for $cnt < N_m$ **do** ▷ Prune and Merge
    $i \leftarrow find(CDF, random(0, N))$ ▷ See section IIIi
    $(\mathbf{s}, \mathbf{n}) \leftarrow (\mathbf{S}_i, \mathbf{N}_i)$
    $d_{min} = \min_{(\mathbf{t} \in \mathbf{D})} |\mathbf{s} - \mathbf{t}|$ ▷ See Section IIIh.
    if $(d_{min} > \tau R_d)$ **then**
      $\mathbf{D} \leftarrow \mathbf{D} \bigcup \mathbf{s}$
      $\mathbf{N}_D \leftarrow \mathbf{N}_D \bigcup \mathbf{n}$
    end if
    $cnt \leftarrow cnt + 1$
  end for
  Apply Lloyd relaxation on $\{\mathbf{D}, \mathbf{N}_D, \mathbf{T}\}$

Формально дано сітчасту модель $C = (M; T)$, з вершинами $M = \{m_1 .. m_{N_m}\} \in \mathbb{R}^{N_m x3}$ та трикутниками $T = \{t_1 .. t_{n_t}\} \in \mathbb{Z}^{N_t x3}$, ми прагнемо генерувати точкову хмару $P \in \mathbb{R}^{N_p x3}$ та нормалі $N \in \mathbb{R}^{N_p x3}$ такі, що вибіркові точки підпорядковуються рівномірному розподілу як в тривимірному, так і в нормальному просторі.

а) *Попередня обробка*: CAD-моделі, створені на виробництві, можуть не мати дизайнерських обмежень щодо орієнтації чи позиціонування, і вони можуть лежати в будь-якому просторі. Щоб алгоритм працював незалежно від позиціонування моделі, першим кроком є вирівнювання моделі до канонічної

системи відліку та шкали. Для цього спочатку обчислюємо матрицю коваріації вершин C і v, нормалізовані власні вектори C: $Cv = \lambda v$. Ми фіксуємо проміжну систему відліку як $(v0;\ v1;\ v0 \times v1)$. Обертання моделі вирівнює цей новий координатний кадр до базового кадру C_0. Більше того, орієнтований обмежувальний куб обчислюється як куб, орієнтований по осям в трансформованому кадрі. Навіть незважаючи на те, що цей підхід не обов'язково видає обмежувальне поле з мінімізацією обсягу, він є досить точним для наших цілей. Нарешті, сітка перекидається на кулю діаметром $d = \sqrt{2}$. Звідси модель M буде посилатися на цю нормалізовану сітку.

б) *Налаштування з декількома точками зору*: генеруємо набір позицій камер $V = \{V_1 \dots V_N\}$, рівномірно розподілених на кулі S, отриманих шляхом поділу граней 12-вершинного ікосаедра на однаково розташовані вершини N_S, як на рис. 7. Позиція складається з матриці обертання R і вектора трансляції ξ: $Q_i = [R_i|\xi_i]$, та $o_i = -R_i^T \xi_i$ є координатами камери. Більше того, для кожної позиції ми підтримуємо набір внутрішніх матриць камери $K = \{K_i\}$, відповідно до моделі «стеноп». Нехай $f = (f_x;\ f_y)$ - фокусна відстань у пікселях, а $c = (c_x;\ c_y)$ - головна точка. Встановлюємо $c_x = w/2, c_y = h/2$, де $(w;\ h)$ - потрібна роздільна здатність камери.

Під час синтезування позиції камери Q_iіз сфери S нам гарантується перегляд усієї проекції моделі, але ми не гарантуємо використання повної роздільної здатності, якщо не налаштовано f. Щоб максимально використовувати область перегляду, спочатку встановлюємо f на розслаблене початкове значення $f = (f_0;\ f_0)$ і проектуємо модель. Враховуючи цю проекцію, ми обчислюємо щільно облягаючий 2D обмежувальний квадрат і масштаб f відповідно:

$$f^* = \min\left(\frac{w}{b_w}, \frac{h}{b_h}\right) f_0, \quad c^* = 2c - c_b, \tag{2.3}$$

де (b_w, b_h) - розміри, а c_b - центр 2D обмежувального квадрату. Таким чином, область проекції максимально збільшується, зберігаючи співвідношення

сторін. Оскільки проектований силует об'єкта різний у всіх видах, f і c відрізняються для кожного виду, результатом чого є набір $K = \{K_i\}$.

в) *Ефективний пошук перетина променя та трикутника*: На цьому наступному етапі зразки, що проектуються на камеру, зворотньо проектуються та перетинаються із самою 3D-сіткою.

Нехай $r = \{o, d\}$визначає промінь з початком в o і нормалізованим вектором напрямку d. Будь-яка точка цього променя потім параметризується як $r(\lambda) = o + \lambda d$. Потім запишемо ребра трикутника, що перетинається, як $E = \{e_0, e_1, e_0\}$ і виразимо трикутник точкою та двома ребрами $t = (v; e_0; e_1)$. Точку перетину v_{int} можна описати, використовуючи 2D барицентричні коефіцієнти $(u; v)$ як $v_{int} = p + ue_0 + ve_1$. Використовуючи відомий алгоритм Мёллера-Тромбора [28], отримуються λ, u і v. Інформація про нормаль у v_{int} отримується як нормаль для грані: $n_{int} = \frac{(p_1-p)\times(p_2-p)}{\|(p_1-p)\times(p_2-p)\|}$.

Треба зауважити, що, оскільки ми використовуємо номери нормалей, нам не доведеться проводити нормальні обчислення для кожної точки вибірки. Натомість ми могли б попередньо обчислити всі нормалі трикутника і зменшити нормальні обчислення до одиничного пошуку. Якщо зберігання викликає занепокоєння, завжди можна проіндексувати нормалі під час виконання в хеш-таблицю та забезпечити єдине обчислення на грані. Результатом цього є вибірковий пул X, де модель CAD охоплюється завищеною кількістю точок.

г) *Зважування семплів*. Через відмінності точок зору та числову точність знайдених перетинів променів з трикутниками не кожен обраний зразок має однакову якість. Завдяки променям, для кожної точки m_j та пересіченого трикутника t_i ми можемо зважити семпли. Спочатку перевертаємо нормалі $\{n_i\}$ у напрямку камери, отримуємо $\{n_i^v\}$. Вага для зразка i обернено пропорційна куту між нормаллю і випущеним променем і визначається як:

$$\omega(\varphi) = \frac{1}{1+\exp(-\lambda(\varphi-\mu_\varphi))}, \quad \varphi = 1 - \left|n_i^v d_j\right|. \tag{2.4}$$

Використовуємо $\lambda = 10$ і $\mu_\varphi = 0.5$. Максимальна вага $\omega^{max} \approx 1$ досягається, коли вектори n_i^v і d_j паралельні ($\varphi = 1$), при цьому мінімум $\omega^{min} \to 0$ отримується, коли кут наближається до 90^0 ($\varphi = 0$). Треба зауважити , що через проекцію камери після $\pm 90^0$ об'єкт не видно або його видно з іншого боку. Як правило, ми відхиляємо семпли, якщо вага виявляється дуже низьким $\omega_i < \tau_\omega$. Це дозволяє нам вибирати семпли, які розглядаються фронтально-паралельно. Ця процедура може працювати з відкритими сітками, оскільки нормалі до сітки надають інформацію про спрямованість.

д) *Реалізація на GPU*: Для систем з підтримкою графіки перетин трикутника і променів, а також зважування можуть бути реалізовані шляхом надання координат перетину на 3-канальне (RGB) зображення. Це можна обчислити в Шейдері. Крім того, ідентифікатор трикутника для кожного пікселя (семпл точки) може зберігатися в альфа-каналі. Це зменшує передачу всієї інформації від GPU до CPU до однієї текстури RGBA.

е) *Об'єднання семплів, що базується на позиції камери*: Враховуючи всі перетинання променів, нам залишається набір 3D-точок $\{p_i\} \in X$ на кожну позицію камери, які слід злити в повну тривимірну хмару точок $\{p_i\} \in P$. Деякі з цих точок p_iможуть бути дублікатами у різних позиціях камер , або навіть якщо дублікатів немає, вони можуть бути розташовані дуже близько (через помилки квантування). Більше того, ще не досягнуто задовільного розподілу точок по поверхні об'єкта. Мета полягає в тому, щоб зачистити цей великий вибірковий пул X з дотриманням певних обмежень. Треба зауважити, що на цьому етапі також можна застосувати бажані характеристики вибірки 3D. Оскільки типові алгоритми виявлення об'єктів [23] покладаються на рівномірну вибірку, а також на локальні характеристики поверхні (наприклад, нормалі), ми адаптуємо дві стратегії: Ми використовуємо алгоритм відбора проб Пуассона для обмежень, що базуються на відстані, і вибірку з

нормального простором для створення рівномірного розподілу локальних характеристик поверхні.

ж) *відбір проб Пуассона (PDS):* Останнім часом PDS виявляється особливо надійним для відповідності PPF [23], через його генерування рівних відстаней та хороших спектральних властивостей. У нашому випадку це також допоможе зменшити упередженість, спричинену дискретною вибіркою на різних поглядах. PDS намагається отримати однакові випадкові точки на основі критерію мінімальної відстані між зразками. Формально він намагається задовольнити наступним двом умовам:

$$\forall p_i \in P_i, \forall \Omega \in D_{i-1}, P(P_i \in \Omega) = \frac{Area(\Omega)}{Area(D_{i-1})} \tag{2.5}$$

$$\forall (i, j : i \neq j), \left\| p_i - p_j \right\| > 2r \tag{2.6}$$

де D - область вибіркового пулу, а D_{i-1} позначає доступну (ще не вибіркову) область. Перша умова (2.5 - проба Пуассона) визначає, що рівномірно розподілений зразок p_iпотрапляє в субдомен Ω з вірогідністю, пропорційною площі Ω, за умови, що $D_{i-1} \cap \Omega = \emptyset$. Завдяки обчислювальній складності обчислення регіону вибірки ми ослабляємо перше обмеження. Ми пов'язуємо кожен зразок p_i із сферою, з центром у p_i, радіусом r. r однаковий для всіх семплів. Потім ймовірність $P(P_i \in \Omega)$ обчислюється як $P(P_i \in \Omega) = 1 - \sum_{j=1}^{i} \alpha(j) 4/3\pi r^3$, де $\alpha(j)$ впливає на процес пуассона в інтервалі (0; 1], впливаючи на відсоток можливого простору для вибірки для точки i. Замість використання $\{\alpha(j)\}$ як в процесі Пуассона, ми просто замінюємо його очікуваним значенням $\bar{\alpha}$, приходячи до $P(P_i \in \Omega) = 1 - \sum_{j=1}^{i} 4/3K\bar{\alpha}\pi r^3$. Тим самим нам дозволяється отримувати зразки з пулу послідовно з однаковою ймовірністю, тобто малюнок розподіляється рівномірно.

Задоволення другої умови (рівняння 2.6 - відбір дисків) є більш простішим, але потрібно зробити його ефективним для великої кількості семплів. Основна ідея полягає в тому, щоб відбирати семпли з пулу X ітераційно і перевіряти наявність існуючих сем плів P_{i-1} на предмет

порушення рівняння 2.6. Якщо рівняння 2.6 задоволено, семпл прийнято. Наївна реалізація передбачає пошук усіх на цей час відібраних семплів або позначення всіх сусідів вибірки як відхилених. Обидва обчислювально вимогливі. Можна піти іншим шляхом і побудувати 3D-воксельну сітку G над існуючими семплами. Потім послідовно брати зразок p_i з пулу і вставляти його в G. Якщо семпл задовольняє рівнянню 2.6, він зберігається, інакше відхиляється. Бічна довжина сітки налаштована таким чином, що куля пошуку залишається в межах 9 вокселів, і запит можна виконати за час $O(1)$, що дозволяє завершити весь процес вибірки за $O(N)$. Навіть незважаючи на те, що ця процедура є жадібною - оскільки це залежить від першої вибірки, вона виявляє хороший розподіл на практиці.

з) Стисле представлення вокселів: для великих радіусів (меншої кількості семплів) пошук у щільних воксельних сітках буде швидким. Однак невеликі радіуси, де відбираються більш близькі точки, є проблематичними через надто великі потреби у пам’яті. З цієї причини пропонується скористатися більш рідкою сіткою вокселів. Наша структура дерева нагадує структуру дерева B +. За конструкцією дерево врівноважене по висоті, неглибоке і широке. Це зменшує кількість операцій з переходу по дереву від кореня до листяного вузла. Створена структура має 1 кореневий вузол, 2 внутрішні шари та шар листя, як показано на Рис. 8. Інформація про точку зберігається лише на шарі листа, а внутрішні шари підтримують бітову маску для кодування активних потомків на основі їх просторових координат. Щоб збільшити швидкість обходу, вся структура обмежується степенями двійки: перший шар має розмір 8 одиниць, другий шар 4 та вузли листків 4. Також реалізований кеш, який містить останні відвідувані внутрішні та листкові вузли. Роздільна здатність сітки автоматично регулюється на 2r. Таким чином, маючи точку, можна відповісти на питання, «чи слід відбирати цю точку?» за константний час, і таким чином зменшити розмір вибірки для дуже великих хмар точок, таких як на рис. 9.

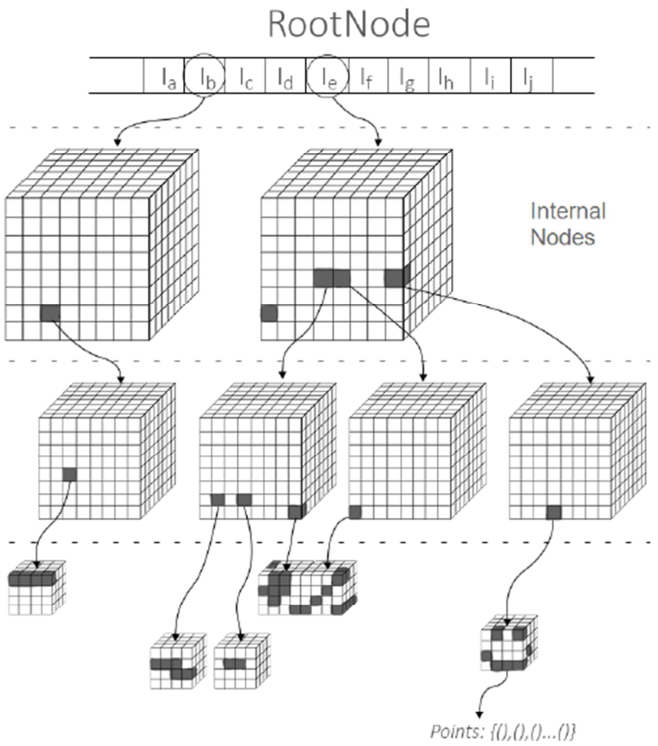


Рис. 8. Сжате представлення вокселів

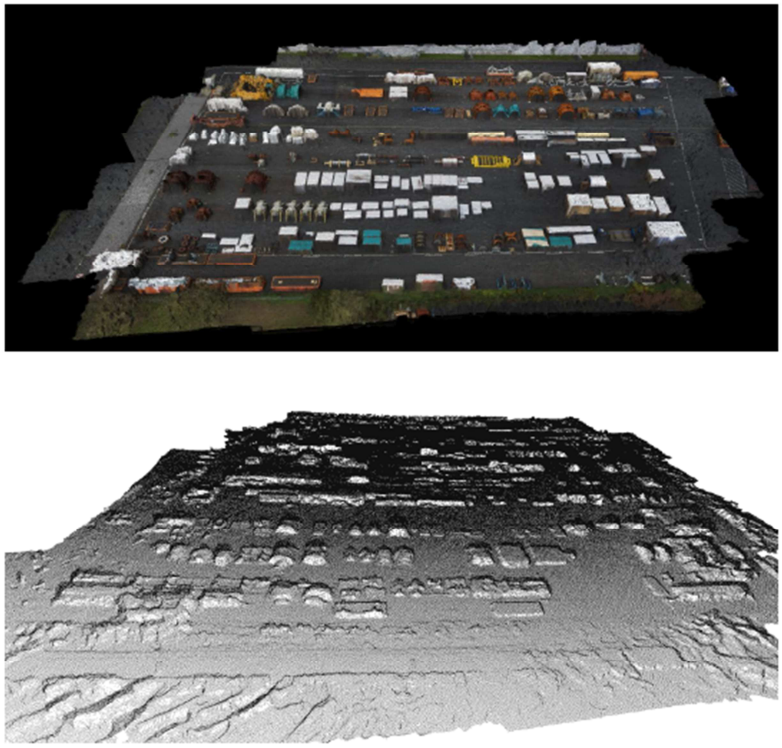


Рис.9. Дуже велика сітка та її вибірка із використанням стислого представлення вокселів.

j) *Релаксація Ллойда*: Через показану рандомізованість та жадібність алгоритму вибірки вибіркові точки не є регулярними. Щоб отримати хороший баланс між властивістю синього шуму і регулярністю, ми завершуємо вибірку, застосовуючи кілька ітерацій Ллойда, в яких центри точок зміщуються до центру діаграми Вороного поступово. Хоча це легко застосувати на синтезованих зразках (зразки не на певній поверхні), гарантуючи, що центри осередків Вороного залишаються на поверхні, важко. Для цього використовуються ітерації обмеженого центроїду Вороного (R-CVD), які ефективно застосовуються у [29]. R-CVD працює шляхом перетину діаграми Вороного з поверхнею. Схема Ллойда формулюється як варіаційна мінімізація енергії і для швидкої збіжності проводиться квазі-ньютонівське наближення. Оскільки на попередніх кроках приблизно задовольняється мінімальна обмежена відстань, ця схема має гарну ініціалізацію, і для задовільних результатів достатньо лише декількох ітерацій.

3 ЕКСПЕРИМЕНТАЛЬНІ РЕЗУЛЬТАТИ.

3.1. Спектральний аналіз

а) Спектральний аналіз: Оскільки алгоритм відбору вибірки демонструє характеристики синього шуму, використовується аналіз частотної області для оцінки якості. Лагае і Дутре [30] стандартизують цей аналіз як дослідження спектру потужності. По-перше, генеруються 50 різних синтетичних 2D сіток, як показано на рис. 10, і вони відбираються за допомогою нашого методу. Для кожного відбору проб обчислюється періодограма. Ці періодограми усереднюються для оцінки спектру потужності $P(f)$. Ці 2D-спектри представлені на рис. 10 з логарифмічною тональною картою. $P(f)$ виявляє типові властивості синього шуму: Центральний пік оточений кільцем з низькою енергією, за яким слідує різкий перехідний регіон, низькочастотний обріз та більш плоска область високої частоти. Як результат, міжпробні відстані відповідають певному закону, причому високі частоти зустрічаються частіше. Треба зауважити, що вибірка зберігає ці спектральні властивості.

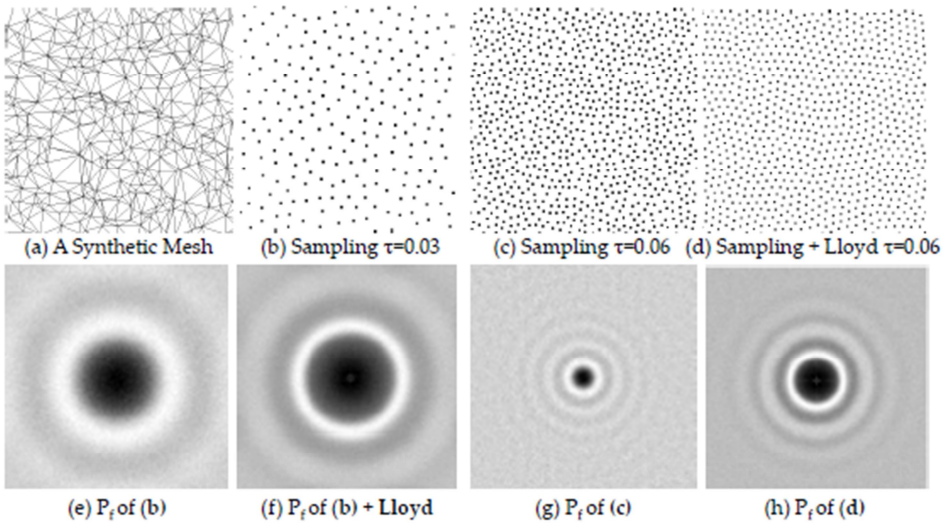


Рис. 10. Спектральний аналіз.

3.2. Реальний набір даних та параметрів

Оскільки використовуємий метод найкраще справляється з промисловими моделями САПР в сітчастих формах, ми використовуємо набір даних Toshiba, запропонований в [31]. Цей набір даних складається з реальних 3D-моделей з каркасною схемою та сканованих хмар точок. Моделі мають внутрішні структури, подовжені трикутники та гострі краї, що робить їх важкими для роботи. Модифікуємо цей набір даних, щоб включати нормалі для 20 сцен, кожна сцена містить кілька об'єктів на різних рівнях оклюзії. Набір даних підсумований на Рис.11. Для всіх експериментів у цій роботі використовується набір фіксованих параметрів: $\tau = 0.03$; роздільна здатність зображення - 640x480, і ми використовуємо 2 підрозділи сфери, в результаті чого камери переглядають об'єкт з 162 позицій.

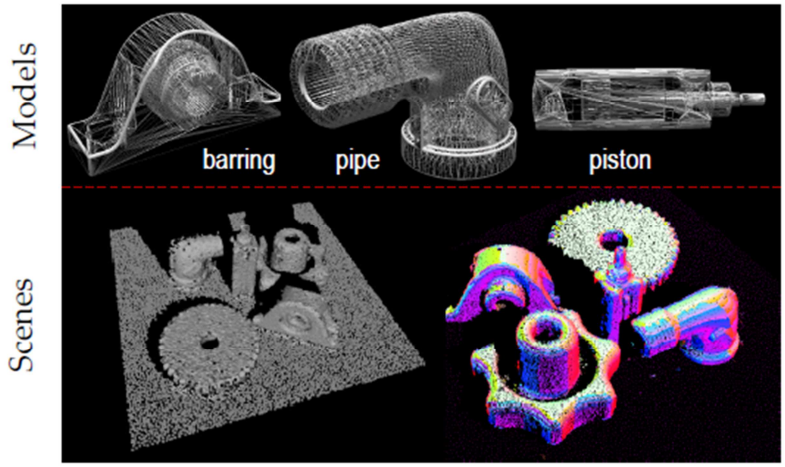


Рис.11. Набір даних Toshiba.

3.3. Порівняння із відомими методами та якісні оцінки.

Алгоритм відбору вибірки було порівняно із 4 іншими методами, які здатні перетворювати сітки на точкові вибірки. Нашою базовою лінією є вибірка Монте-Карло (рівномірна випадкова вибірка) на гранях. Більш якісним методом є метод стратифікованої вибірки або трикутника [32], в якому кожен трикутник отримує вибірку залежно від роздільної здатності основної сітки вокселів. Популярним та успішним підходом є відбір проб

Пуассона [33], яка також використовується в роботі [23]. Нарешті, метод відбору вибірки порівнювався з рівномірним перекомплектуванням сітки, яке складається з побудови рівномірного об'ємного означеного поля відстаней та застосування методу рухомих кубів для отримання рівномірно розподілених семплів. Це також дуже схоже на метод, запропонований у [34]. Треба зауважити, що велике сімейство 3d-методів вилучення ключових точок працює безпосередньо на вершинах [35], [36] і не може бути застосоване до набору даних Toshiba.

Візуальна якість методу відбору вибірки тестувалась на об'єкті «Порстон» з набору даних Toshiba, а також на двох додаткових промислових CAD-моделях бензинового двигуна та великого трансформатора. Всі моделі мають складне розташування трикутників, приховані грані та структурні з'єднання. Вершини нерівномірно розподілені та згруповані у певних областях. На рис. 12 представлені результати вибірки 1000 точок за алгоритмом відбору вибірки та попередніми методами. Помітно, що метод відбору вибірки краще зберігає глобальну форму, має рівномірний розподіл вершин і може позбутися внутрішніх структур. Більше того, завдяки синім шумовим характеристикам артефакти дискретизації зникають. Це призводить до підвищення якості сприйняття сформованих точкових наборів. Найближчий до цього результату - відбір проб Пуассона.

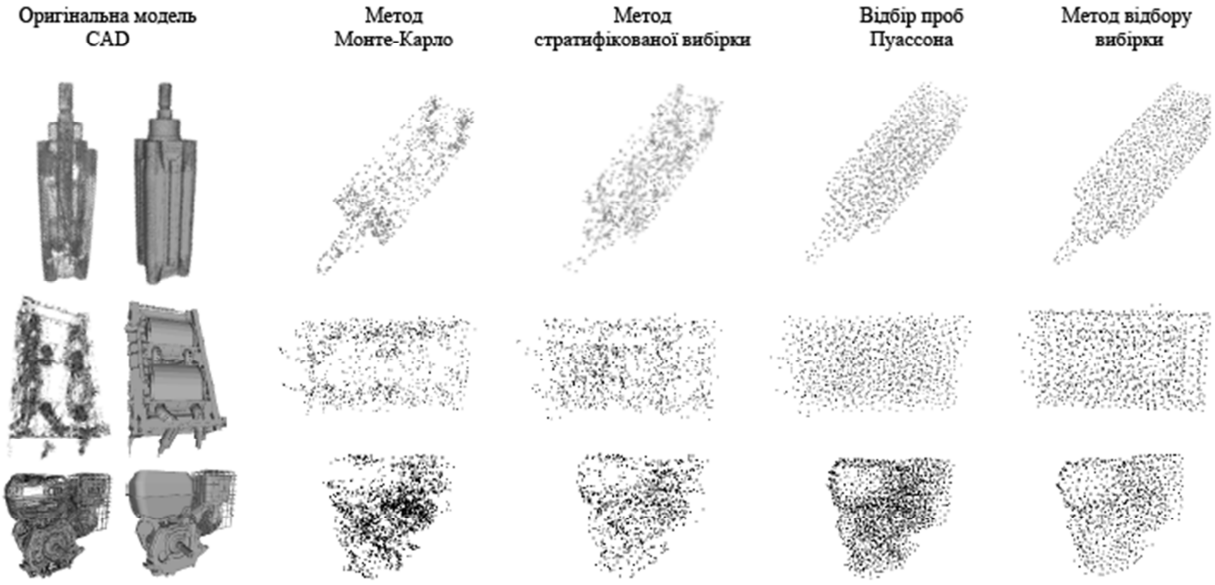


Рис. 12. Порівняння роботи різних методів

Також були візуалізовані норми, оцінені за допомогою різних методик на рис. 13. Тут слід шукати набір напрямків, які найбільше нагадують модель CAD. Оскільки ми проводимо вибірку безпосередньо на гранях, якість побудови нормалей також є найбільш близькою до оригінальної моделі CAD серед оцінюваних алгоритмів.

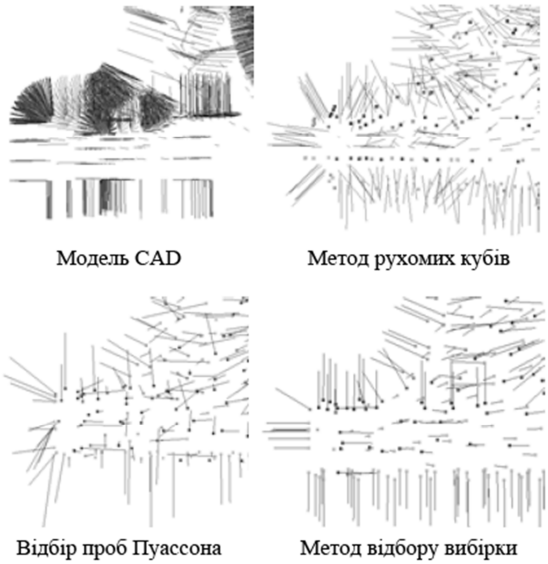


Рис.13. Відібрані нормалі

Тепер оцінимо ефективність методу у випробуванні виявлення 3D-об’єктів, що є головною причиною його використання. Ми використовуємо

метод PPF [23], [22] на наборі даних Toshiba [31]. Оцінки проводяться заміною існуючої вибірки [22] іншою та порівнянням результатів визначення пози. Висновки щодо різної кількості зразків та алгоритмів вибірки наведені на рис. 14. Надано графіки для обертових (ε_R) та поступальних (ε_t) компонентів помилок, які відповідно вимірюються як:

$$\varepsilon_R(R_t, R_g) = \frac{1}{\pi} arc\ c?s(\frac{trace(R_t^{-1}R_g)-1}{2}) \quad (3.1)$$

та $\varepsilon_t(t_t, t_g) = \|t_t - t_g\|$. Визначення, є правильним, якщо воно може бути уточнене до коректної пози шляхом послідовного застосування ICP (методу найближчої ітеративної точки). Залежно від пози, у якій об'єкт знаходиться на сцені, точність ICP може відрізнятися, однакові налаштування були використані у всіх методах. Кількість коректних виявлень виявлення на кожну вибірку показано в самому лівому стовпчику на рис. 14. З цього рисунка послідовно видно, що вибір рівномірно розташованих точок на видимій поверхні при рівномірному розподілі нормалей допомагає алгоритму відбору вибірки виявити більше об'єктів і зробити меншу помилку в позі при виявленні об'єкта. Це означає, що цьому алгоритму вдається відібрати більше точок, що підходять для завдань, які мають більшу ймовірність бачення та мають краще покриття усієї поверхні - як для [22], поза залежить від точки, яка виявляється на поверхні.

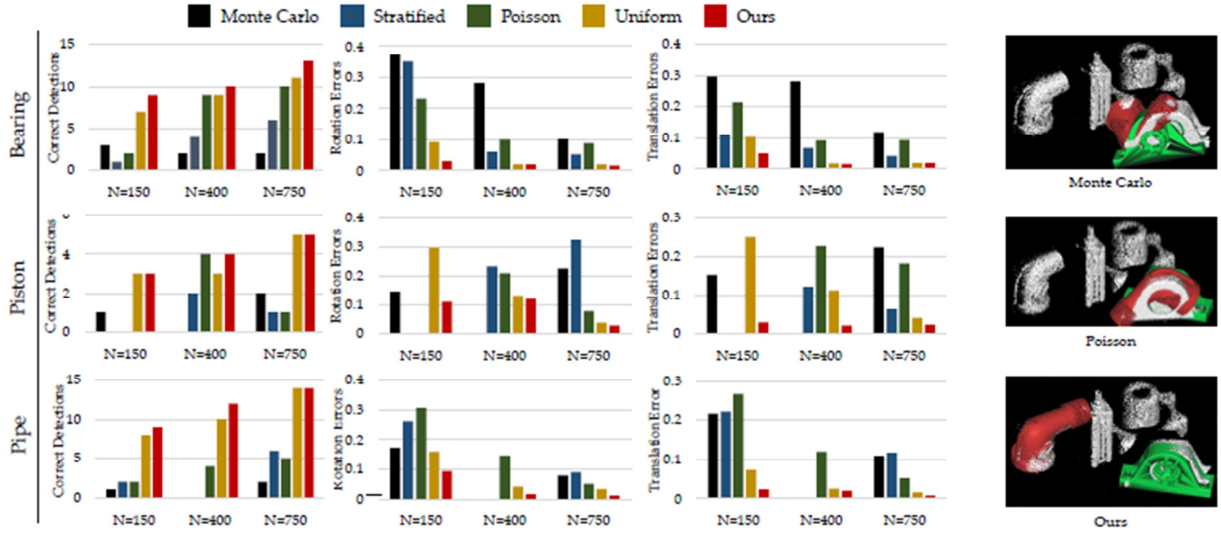


Рис. 14. Порівняння ефективності методів у виявленні 3D-об’єктів

3.4. Обчислювальний час

Алгоритм було реалізовано на процесорі Intel i5 2.3GhZ. Для випускання променів було використано вільно доступну бібліотеку Intel Embree [37], ретельно оптимізовану для платформи Intel. Середній час виконання на використаному наборі даних наведені на рис.15. Найважливіший параметр часу виконання - це роздільна здатність синтетичних поглядів, коли ми передаємо промінь для кожного пікселя виду. Кількість граней моделі CAD має зменшений ефект завдяки просторовому індексуванню.

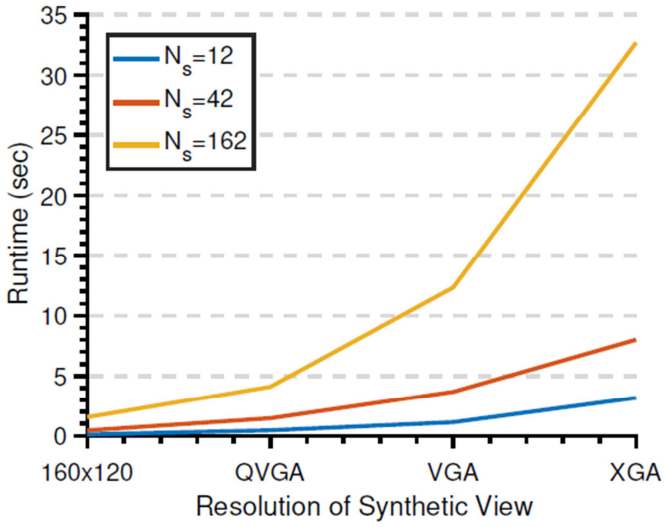


Рис. 15. Час виконання реалізації аллгоритму

ВИСНОВКИ

Мета та задачі, поставлені в цій роботі, були виконані. А саме:

- Було створено технічне завдання;
- Проведено аналіз різних підходів для вибірки точок у застосуваннях комп’ютерного зору;
- Обрано алгоритм вибірки;
- Реалізовано обраний алгоритм у середовищі MATLAB;
- Програму було протестовано.

Було запропоновано дуже ефективну, практичну та просту у впровадженні стратегію вибірки точок, розроблену для проблем комп’ютерного зору із використанням підходу PPF (point pair feature). Алгоритм дозволяє безпосередньо обчислити рандомізований, рівномірно розташований відбір точок видимої частини поверхні. Обчислення нормалей та якість точок добре інтегровані у схему і завжди присутні у результуючей виборці. Підхід було якісно та кількісно оцінено на реальному наборі даних та продемонстровано приріст у задачах пошуку відповідності та виявлення пози. Крім того, метод має потенціал розширення для застосування до різних проблем, таких як реконструкція та реєстрація в багатьох видах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] S. Marden and J. Guivant, “Improving the Performance of ICP for Real-Time Applications using an Approximate Nearest Neighbour Search”, 2012, pp. 3–5.

[2] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,”P. S. Schenker, Ed., Apr. 1992, pp. 586–606.

[3] S. Fantoni, U. Castellani, and A. Fusiello, “Accurate and automatic alignment of range surfaces,” in Proceedings - 2nd Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2012. IEEE, Oct. 2012, pp. 73–80.

[4] A. V. Segal, D. Haehnel, and S. Thrun, “Generalized-ICP,” in Proceedings of Robotics: Science and Systems, Seattle, 2009, p. 8.

[5] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, “Nonrigid registration using free-form deformations: application to breast MR images.” IEEE Transactions on Medical Imaging, vol. 18, no. 8, Aug. 1999, pp. 712–21. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10534053

[6] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, “An evaluation of the RGB-D SLAM system,” Robotics and Automation (ICRA), 2012 IEEE International Conference, vol. 3, no. c, May 2012, pp. 1691–1696. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs all.jsp?arnumber=6225199

[7] J. Aulinas, Y. Petillot, J. Salvi, and X. Llad´o, “The SLAM problem: a survey.” CCIA, 2008, pp. 363—-371.

[8] P. F. I. N. D. E. Carrera, “MADRID RGB-D SLAM Author : Jorge Garcıa Bueno,” no. October, 2011.

[9] K. Berger, S. Meister, R. Nair, and D. Kondermann, “A state of the art report on kinect sensor setups in computer vision,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8200 LNCS, 2013, pp. 257–272.

[10] C. Kerl, J. Sturm, and D. Cremers, “Dense visual SLAM for RGBD cameras,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, Nov. 2013, pp. 2100–2106.
[11] J. Sprickerhof and A. N¨uchter, “An Explicit Loop Closing Technique for 6D SLAM.” ECMR, 2009, pp. 1–6. [Online]. Available: http://plum.eecs.jacobs-university.de/download/ecmr2009.pdf
[12] A. S. Huang and A. Bachrach, “Visual odometry and mapping for autonomous flight using an RGB-D camera,” International Symposium on Robotics Research (ISRR), 2011, pp. 1–16.
[13] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” in 2011 10th IEEE International Symposium on Mixed and Augmented Reality. IEEE, Oct. 2011, pp. 127–136.
[14] M. Ruhnke, L. Bo, D. Fox, and W. Burgard, “Compact RGBD Surface Models Based on Sparse Coding.” AAAI, 2013.
[15] S. Savarese, “3D generic object categorization, localization and pose estimation,” in 2007 IEEE 11th International Conference on Computer Vision. IEEE, 2007, pp. 1–8.
[16] R. Shams, P. Sadeghi, R. Kennedy, and R. Hartley, “A survey of medical image registration on multicore and the GPU,” pp. 50–60, 2010. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs all.jsp?arnumber=5438962
[17] D. Lee, H. Kim, and H. Myung, “Image feature-based real-time RGBD 3D SLAM with GPU acceleration,” Journal of Institute of Control, Robotics and Systems, vol. 19, no. Urai, 2013, pp. 457–461.
[18] W. R. Crum, “Non-rigid image registration: theory and practice,” British Journal of Radiology, vol. 77, no. suppl 2, Dec. 2004, pp. S140–S153.
[19] J. Kay, “Introduction to Homogeneous Transformations & Robot Kinematics,” Rowan University Computer Science Department, no. January, 2005, pp. 1–25.

[20] B. Draper, W. Yambor, and J. Beveridge, “Analyzing pca-based face recognition algorithms: Eigenvector selection and distance measures,” Empirical Evaluation Methods in Computer Vision, Singapore, 2002, pp. 1–14.
[21] J. Schoberl, “Netgen an advancing front 2d/3d-mesh generator based on abstract rules,” Computing and Visualization in Science, 1997.
[22] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3d object recognition,” in Computer Vision and Pattern Recognition, 2010.
[23] T. Birdal and S. Ilic, “Point pair features based object detection and pose estimation revisited,” in 3D Vision. IEEE, 2015, pp. 527–535.
[24] Y. Zhong, “Intrinsic shape signatures: A shape descriptor for 3d object recognition,” in International Conference on Computer Vision Workshops. IEEE, 2009, pp. 689–696.
[25] J. Serafin, E. Olson, and G. Grisetti, “Fast and robust 3d feature extraction from sparse point clouds,” in Intelligent Robots and Systems, International Conference on. IEEE, 2016.
[26] F. Tombari, S. Salti, and L. Di Stefano, “Unique signatures of histograms for local surface description,” in European conference on computer vision. Springer, 2010, pp. 356–369.
[27] S. M. Prakhya, B. Liu, and W. Lin, “Detecting keypoint sets on 3d point clouds via histogram of normal orientations,” Pattern Recognition Letters, vol. 83, 2016.
[28] T. M¨oller and B. Trumbore, “Fast, minimum storage ray/triangle intersection,” in ACM SIGGRAPH Courses. ACM, 2005, p. 7.
[29] D.-M. Yan, B. L´evy, Y. Liu, F. Sun, and W. Wang, “Isotropic remeshing with fast and exact computation of restricted voronoi diagram,” in Computer graphics forum, vol. 28, 2009.
[30] A. Lagae and P. Dutr´e, “A comparison of methods for generating poisson disk distributions,” in Computer Graphics Forum, 2008.

[31] M.-T. Pham, O. J. Woodford, F. Perbet, A. Maki, B. Stenger, and R. Cipolla, “A new distance for scale-invariant 3D shape recognition and registration,” in ICCV, 2011.

[32] M. Alexa, S. Rusinkiewicz, D. Nehab, and P. Shilane, “Stratified point sampling of 3d models,” in Eurographics, 2004.

[33] M. Corsini, P. Cignoni, and R. Scopigno, “Efficient and flexible sampling with blue noise properties of triangular meshes,” Visualization and Computer Graphics, vol. 18, pp. 914–924, 2012.

[34] A. S. Mian, M. Bennamoun, and R. Owens, “Three-dimensional model-based object recognition and segmentation in cluttered scenes,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, Oct. 2006.

[35] Y. Zhong, “Intrinsic shape signatures: A shape descriptor for 3d object recognition,” in International Conference on Computer Vision Workshops. IEEE, 2009, pp. 689–696.

[36] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, and J. Wan, “3d object recognition in cluttered scenes with local surface features: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 11, 2014.

[37] S. Woop, L. Feng, I. Wald, and C. Benthin, “Embree ray tracing kernels for cpus and the xeon phi architecture,” in ACM SIGGRAPH 2013 Talks. ACM, 2013, p. 44.